

## PENGUNAAN MATLAB DALAM PENYELESAIAN SISTEM PERSAMAAN LINEAR MENGGUNAKAN JARINGAN HOPFIELD LINEAR

**Rosihan Ari Yuana**

Program Studi Pendidikan Matematika  
Universitas Sebelas Maret

### ABSTRAK

Aplikasi jaringan syaraf tiruan untuk menyelesaikan permasalahan komputasi telah banyak digunakan. Salah satunya adalah algoritma jaringan Hopfield Linear untuk menyelesaikan sistem persamaan linear. Algoritma ini memungkinkan mesin (komputer) dapat menyelesaikan sistem persamaan linear dengan serangkaian proses training. Dalam artikel ini digunakan MATLAB untuk mengimplementasikan algoritma tersebut. Output dari script MATLAB yang dibuat meliputi penyelesaian persamaan linear, informasi performa (running time) dan juga error beserta grafiknya. Keyword: sistem persamaan linear, jaringan syaraf buatan, jaringan Hopfield linear, MATLAB

### PENDAHULUAN

Jaringan syaraf tiruan (JST), adalah model matematis atau model komputasi yang terinspirasi oleh struktur dan / atau aspek fungsional dari jaringan saraf biologis. Sebuah jaringan saraf terdiri dari kelompok yang saling berhubungan dari neuron buatan, dan memproses informasi menggunakan pendekatan koneksionis untuk perhitungan. Menurut Kusumadewi (2003), dalam kebanyakan kasus, sebuah JST merupakan sistem adaptif yang berubah strukturnya berdasarkan informasi eksternal atau internal yang mengalir melalui jaringan selama fase pembelajaran. Salah satu model JST yang dapat digunakan adalah Jaringan Hopfield Linear (JHL). JHL dapat digunakan untuk menyelesaikan sistem persamaan linear.

### Jaringan Hopfield Linear (JHL)

Jaringan Hopfield diperkenalkan pada tahun 1982, yang merupakan salah satu tipe jaringan syaraf buatan (Fausett, 1994). Jaringan tersebut dapat digunakan untuk menyelesaikan masalah optimisasi khususnya masalah sistem persamaan linear, apabila fungsi aktivasi yang digunakan

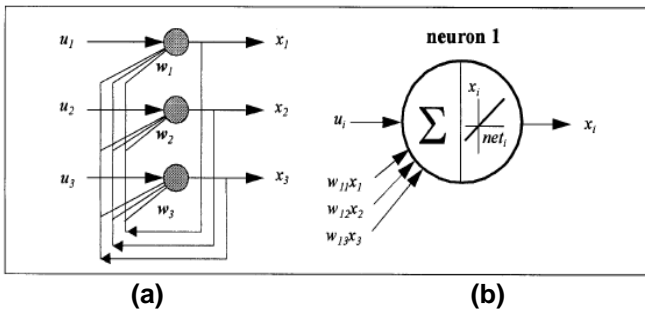
adalah fungsi linear. Lebih lanjut, jaringan ini disebut jaringan Hopfield linear. Demikian seperti yang dijelaskan oleh Lendaris dkk. (1994).

Jaringan Hopfield linear tersusun dari beberapa neuron yang membentuk satu lapisan (*single layer*). Topologi jaringan Hopfield linear dengan 3 neuron digambarkan pada Gambar 1. Secara umum, antar neuron dihubungkan dengan sinapsis berbobot  $\mathbf{W}$  yang berbentuk matriks. Output dari jaringan tersebut merupakan vektor  $\mathbf{x}$ , serta input eksternal adalah vektor  $\mathbf{u}$ .

Proses iterasi yang terjadi pada jaringan Hopfield linear dapat dituliskan sebagai

$$\mathbf{x}_{k+1} = \mathbf{W}\mathbf{x}_k + \mathbf{u}, \quad (1)$$

dengan  $k = 0, 1, 2, 3, \dots$ . Jaringan ini akan konvergen apabila *spectral radius* dari matriks  $\mathbf{W}$  lebih kecil dari 1 atau  $\rho(\mathbf{W}) < 1$ . Spectral radius suatu matriks merupakan nilai eigen yang terbesar dari matriks tersebut.



**Gambar 1.** (a) Topologi jaringan Hopfield dengan 3 neuron . (b) Fungsi aktivasi linear pada masing-masing neuron.

### Penyelesaian Sistem Persamaan Linear dengan JHL

Diperhatikan sistem persamaan linear

$$\mathbf{Ax} = \mathbf{b}, \quad (2)$$

dengan  $\mathbf{A}$  adalah matriks bujur sangkar berdimensi  $n \times n$  dan mempunyai invers. Selanjutnya diinginkan menyelesaikan persamaan (2) dengan menggunakan jaringan Hopfield linear. Untuk menyelesaikan persamaan (2) dengan jaringan Hopfield linear, jumlah neuron yang diperlukan adalah sejumlah  $n$ . Dalam hal ini bobot yang menghubungkan antar neuron dapat dinyatakan sebagai matriks  $\mathbf{W}$  berdimensi  $n \times n$ , sedangkan vektor  $\mathbf{x}$  yang merupakan output dari jaringan berdimensi dan vektor input eksternal  $\mathbf{u}$  berdimensi  $n$ . Selanjutnya terlebih dahulu akan dikaitkan antara persamaan (1) dengan (2). Persamaan (2) dapat dituliskan sebagai

$$-\mathbf{Ax} + \mathbf{b} = \mathbf{0}. \quad (3)$$

Apabila persamaan (3) diselesaikan dengan proses iterasi, maka proses iterasi tersebut adalah

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha (-\mathbf{Ax}_k + \mathbf{b}), \quad (4)$$

dengan  $\alpha$  adalah tetapan untuk konvergensi dan  $k = 0, 1, 2, \dots$ . Persamaan (4) disusun berdasarkan asumsi bahwa untuk  $k \rightarrow \infty$ ,

berakibat nilai  $\mathbf{x}_{k+1} = \mathbf{x}_k$  karena nilai  $\mathbf{x}$  telah konvergen yang berarti bahwa penyelesaian (3) diperoleh, sehingga  $\mathbf{Ax}_k + \mathbf{b} = \mathbf{0}$ . Selanjutnya persamaan (4) dapat dituliskan sebagai

$$\mathbf{x}_{k+1} = (\mathbf{I} - \alpha \mathbf{A}) \mathbf{x}_k + \alpha \mathbf{b}. \quad (5)$$

Apabila persamaan (5) dikaitkan dengan (1), maka (4) merupakan proses iterasi jaringan Hopfield linear dengan bobot  $\mathbf{W} = \mathbf{I} - \alpha \mathbf{A}$  dan  $\mathbf{u} = \alpha \mathbf{b}$ . Di sini, matriks  $\mathbf{A}$  dapat mempunyai sifat:

1. Definit positif (nilai eigen semuanya bernilai positif).

Untuk sifat ini nilai  $\alpha$  yang dipilih adalah  $0 < \alpha < 2/\rho(\mathbf{A})$  untuk memenuhi syarat kekonvergenan  $\rho(\mathbf{W}) = \rho(\mathbf{I} - \alpha \mathbf{A}) < 1$ .

2. Tidak definit positif (nilai eigen tidak semuanya bernilai positif).

Apabila  $\mathbf{A}$  tidak definit positif, maka kriteria pengambilan  $\alpha$  seperti di atas tidak bisa menjamin kekonvergenan. Lendaris dkk (1994) menjelaskan bahwa  $\mathbf{A}$  dapat dibalik (mempunyai invers) jika dan hanya jika  $\mathbf{A}^T \mathbf{A}$  definit positif, dimana  $\mathbf{A}^T$  adalah matriks transpose dari  $\mathbf{A}$ . Di sini persamaan  $\mathbf{A}^{-1} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T$  digunakan. Karena  $\mathbf{A}$  mempunyai invers, maka penyelesaian (2) dapat dituliskan sebagai

$$\mathbf{x} = \mathbf{A}^{-1} \mathbf{b} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b}. \quad (6)$$

Misalkan  $\mathbf{B} = \mathbf{A}^T \mathbf{A}$  dan  $\mathbf{d} = \mathbf{A}^T \mathbf{b}$ , maka persamaan (6) menjadi

$$\mathbf{x} = \mathbf{B}^{-1} \mathbf{d}$$

atau

$$\mathbf{Bx} = \mathbf{d} \quad (7)$$

dimana  $\mathbf{B}$  definit positif. Selanjutnya persamaan (7) dapat diselesaikan dengan jaringan Hopfield linear, dengan proses iterasi

$$\mathbf{x}_{k+1} = (\mathbf{I} - \alpha \mathbf{B}) \mathbf{x}_k + \alpha \mathbf{d}. \quad (8)$$

Pada persamaan (8) di atas, matriks  $\mathbf{W}$  adalah  $\mathbf{W} = (\mathbf{I} - \alpha \mathbf{B})$  dan pengambilan nilai  $\alpha$  adalah  $0 < \alpha < 2/\rho(\mathbf{B})$ .

### Implementasi dengan MATLAB

MATLAB merupakan software untuk tool visualisasi dan sebagai bahasa pemrograman yang mampu menyelesaikan permasalahan dalam bidang teknik, komputasi, dan matematika. MATLAB dibangun dengan basis array, sehingga sangatlah handal apabila digunakan untuk proses komputasi yang melibatkan operasi array atau matriks (Hanselman dan Littefield, 1998).

Dalam penulisan ini, dibuat m-file yang berisi script untuk menyelesaikan sistem persamaan linear menggunakan jaringan Hopfield linear. Script ini dibuat karena dalam MATLAB belum ada function untuk mengkonstruksi jaringan Hopfield linear tersebut (Demuth dan Mark, 1992).

Untuk menyelesaikan sistem persamaan linear  $\mathbf{Ax} = \mathbf{b}$ , parameter yang digunakan dalam script meliputi matriks  $\mathbf{A}$ ,  $\mathbf{b}$ , dan juga  $\mathbf{x}$  untuk nilai awal iterasi. Selain menampilkan hasil penyelesaian sistem persamaan linear yang diberikan, juga ditampilkan error iterasi dan juga plotnya. Running time untuk proses penyelesaian juga ditampilkan.

Script yang dibuat terdiri dari beberapa langkah proses, yaitu:

1. Input  $\mathbf{A}$ ,  $\mathbf{b}$ , dan  $\mathbf{x}$
2. Dicari nilai eigen matriks  $\mathbf{A}$
3. Dicek apakah matriks  $\mathbf{A}$  definit positif atau tidak
4. Jika  $\mathbf{A}$  definit positif, maka
  - 4.1 Hitung spectral radius  $\mathbf{A}$

- 4.2 Input nilai  $\alpha$
- 4.3 Input banyak iterasi/epoch
- 4.4 Lakukan proses seperti pada persamaan (5)
- 4.5 Tampilkan error dan grafik error setiap iterasi/epoch
- 4.6 Tampilkan hasil penyelesaian dan running time
5. Jika  $\mathbf{A}$  tidak definit positif, maka
  - 5.1 Hitung  $\mathbf{B} = \mathbf{A}^T \mathbf{A}$
  - 5.2 Hitung  $\mathbf{d} = \mathbf{A}^T \mathbf{b}$
  - 5.3 Hitung spectral radius matriks  $\mathbf{B}$
  - 5.4 Input nilai  $\alpha$
  - 5.5 Input banyak iterasi/epoch
  - 5.6 Lakukan proses seperti pada persamaan (8)
  - 5.7 Tampilkan error dan grafik error setiap iterasi/epoch (error dihitung menggunakan norm error Euclid)
  - 5.8 Tampilkan hasil penyelesaian dan running time
6. Selesai

### Kasus 1

Diberikan sistem persamaan linear  $\mathbf{Ax} = \mathbf{b}$  dengan

$$\mathbf{A} = \begin{bmatrix} 3 & 5 & 6 & -4 \\ 1 & 2 & -5 & 3 \\ 7 & -1 & -2 & 1 \\ 3 & 5 & 7 & 9 \end{bmatrix} \text{ dan } \mathbf{b} = \begin{bmatrix} 20 \\ -4 \\ 6 \\ 3 \end{bmatrix}.$$

Selanjutnya akan dicari penyelesaian  $\mathbf{x}$  dengan jaringan Hopfield linear. Misalkan nilai awal  $\mathbf{x}$  diberikan sebagai berikut

$$x = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

Berikut ini informasi dan hasil penyelesaian Kasus 1 untuk 300 iterasi/epoch menggunakan MATLAB.

```
Matriks A tidak definit positif
Batas pengambilan alpha adalah 0 < alpha < 0.01100
Nilai alpha yang anda pilih = 0.01
Banyaknya epoch = 300
```

Epoch	Error
100	1.46085e-008
200	8.30815e-016
300	8.30815e-016

```
Epoch : 300
Solusinya adalah :
```

```
x =
 1.42905493361104
 1.16922676386358
 0.73496485290289
-1.36422806560791
```

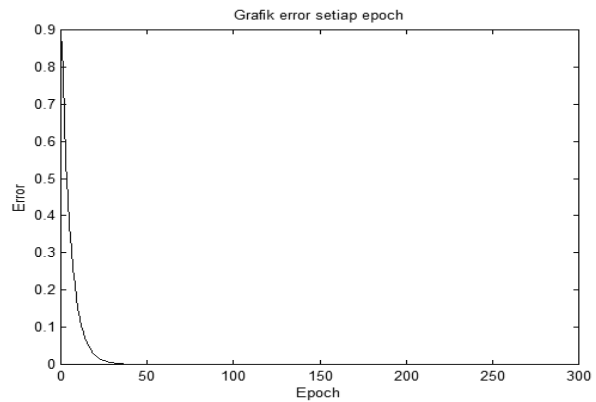
```
Errornya : 8.30815e-016
```

```
time =
 3.84000000000000
```

```
eksak =
 1.42905493361104
 1.16922676386358
 0.73496485290289
-1.36422806560791
```

Nilai  $\alpha$  yang dipilih untuk kasus ini adalah 0.01. Dari hasil di atas tampak bahwa untuk 100 iterasi pertama diperoleh error sebesar  $1.46085e-008$ . Sedangkan untuk iterasi ke 200 sampai 300, errornya cenderung stabil (MATLAB hanya menampilkan error untuk iterasi berkelipatan 100). Running time yang diperlukan MATLAB untuk menyelesaikan Kasus 1 ini adalah 3.84 sekon.

Grafik yang memperlihatkan besar error untuk setiap iterasi digambarkan sebagai Gambar 2. Dari grafik error terlihat bahwa error hampir mendekati nol ketika iterasi ke 40 an.



Gambar 2. Grafik Error Setiap Iterasi untuk Kasus 1

## Kasus 2

Diberikan sistem persamaan linear  $Ax=b$  sebagai berikut

$$A = \begin{bmatrix} 0.0011 & -1.2413 & 0.0134 \\ 0.1234 & 1.2412 & -0.2410 \\ 1.1124 & 2.0189 & 0.0241 \end{bmatrix},$$

$$b = \begin{bmatrix} 0.1023 \\ 1.1923 \\ -0.2341 \end{bmatrix},$$

Misal diberikan nilai awal  $x = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$ , dengan

menggunakan MATLAB diperoleh hasil sebagai berikut:

```
Matriks A tidak definit positif
Batas pengambilan alpha adalah 0 < alpha < 0.24949104
Nilai alpha yang anda pilih = 0.24
Banyaknya epoch = 3000
```

```
-----
Epoch | Error
```

```

-----
100      | 2.62765
200      | 1.25634
300      | 0.600688
400      | 0.287203
500      | 0.137319
600      | 0.0656553
700      | 0.0313914
800      | 0.015009
900      | 0.00717615
1000     | 0.00343109
1100     | 0.00164048
1200     | 0.000784354
1300     | 0.000375019
1400     | 0.000179305
1500     | 8.57301e-005
1600     | 4.09896e-005
1700     | 1.95981e-005
1800     | 9.37033e-006
1900     | 4.48018e-006
2000     | 2.14208e-006
2100     | 1.02418e-006
2200     | 4.89684e-007
2300     | 2.3413e-007
2400     | 1.11943e-007
2500     | 5.35227e-008
2600     | 2.55905e-008
2700     | 1.22354e-008
2800     | 5.85005e-009
2900     | 2.79705e-009
3000     | 1.33734e-009
-----

```

Epoch : 3000  
Solusinya adalah :

x =

```

0.16969367179510
-0.14266374575145
-5.59516199980986

```

Errornya : 1.33734e-009

time =

```
24.72000000000000
```

eksak =

```

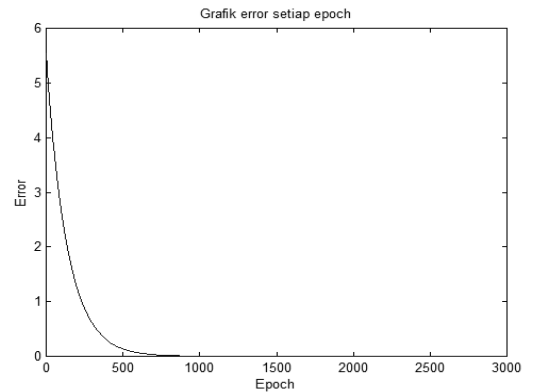
0.16969367206756
-0.14266374589190
-5.59516200111160

```

Nilai  $\alpha$  yang dipilih untuk kasus ini adalah 0.24. Dari hasil di atas tampak bahwa untuk 3000 iterasi diperoleh error sebesar 1.33734e-009. Running time yang diperlukan MATLAB

untuk menyelesaikan Kasus 2 ini adalah 24.72 sekon.

Grafik yang memperlihatkan besar error untuk setiap iterasi digambarkan sebagai Gambar 3.



Gambar 3. Grafik Error Setiap Iterasi untuk Kasus 2

### Kasus 3

Diberikan matriks A dan b untuk sistem persamaan linear  $Ax = b$  sebagai berikut

$$A = \begin{bmatrix} 1 & 2 & -3 \\ 1 & 7 & 1 \\ 0 & 1 & 2 \end{bmatrix} \text{ dan } b = \begin{bmatrix} 5 \\ -5 \\ 2 \end{bmatrix}.$$

Apabila diberikan nilai awal  $x = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$ ,

diperoleh hasil seperti di bawah ini.

Matriks A definit positif  
Batas pengambilan alpha adalah  $0 < \alpha < 0.26989359$   
Nilai alpha yang anda pilih = 0.2  
Banyaknya epoch = 100

```

-----
Epoch   | Error
-----
100     | 0.013932
-----

```

Epoch : 100

Solusinya adalah :

```

x =
    24.31965765797024
   -4.66439688091523
    3.33194614497277

Errornya      : 0.013932

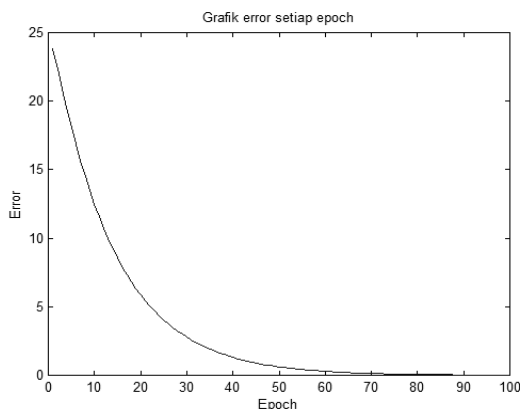
time =
    0.660000000000000

eksak =
    24.33333333333333
   -4.66666666666667
    3.33333333333333

```

Nilai  $\alpha$  yang dipilih untuk kasus ini adalah 0.2. Dari hasil di atas tampak bahwa untuk 100 iterasi diperoleh error sebesar 0.013932. Running time yang diperlukan MATLAB untuk menyelesaikan Kasus 3 ini adalah 0.66 sekon.

Grafik yang memperlihatkan besar error untuk setiap iterasi digambarkan dalam Gambar 4.



**Gambar 4.** Grafik Error Setiap Iterasi untuk Kasus 3

## Kesimpulan

Dari hasil eksperimentasi, disimpulkan bahwa MATLAB dapat digunakan untuk menyelesaikan sistem persamaan linear dengan jaringan Hopfield linear. Adapun algoritma yang

dapat digunakan untuk implementasinya adalah sebagai berikut:

1. Input A, b, dan x
2. Dicari nilai eigen matriks A
3. Dicek apakah matriks A definit positif atau tidak
4. Jika A definit positif, maka
  - 4.1 Hitung spectral radius A
  - 4.2 Input nilai  $\alpha$
  - 4.3 Input banyak iterasi/epoch
  - 4.4 Lakukan proses seperti pada persamaan (5)
  - 4.5 Tampilkan error dan grafik error setiap iterasi/epoch
  - 4.6 Tampilkan hasil penyelesaian dan running time
5. Jika A tidak definit positif, maka
  - 5.1 Hitung  $B = A^T A$
  - 5.2 Hitung  $d = A^T b$
  - 5.3 Hitung spectral radius matriks B
  - 5.4 Input nilai  $\alpha$
  - 5.5 Input banyak iterasi/epoch
  - 5.6 Lakukan proses seperti pada persamaan (8)
  - 5.7 Tampilkan error dan grafik error setiap iterasi/epoch (error dihitung menggunakan norm error Euclid)
  - 5.8 Tampilkan hasil penyelesaian dan running time

6. Selesai

Dari algoritma di atas, hasil output dari program berupa grafik error setiap iterasi, besaran error serta penyelesaian beserta running timenya untuk melihat efisiensi dari komputasi.

## DAFTAR PUSTAKA

- Demuth, H., and Mark, B., 1992, *Neural Network Toolbox, For Use in MATLAB*,  
Mathworks
- Fausett, L., 1994, *Fundamentals of Neural Networks, Architectures, Algorithms, and Applications*,  
Prentice Hall, Englewood Cliffs, New Jersey
- Hanselman, D, and Littlefield, B., 1998, *Mastering MATLAB 5, A Comprehensive Tutorial and  
Reference*, Prentice Hall, New Jersey
- Kusumadewi, S., 2003, *Artificial Intelligence, Teknik dan Aplikasinya*, Graha Ilmu
- Lendaris, G. G., Mathia, K., and Seeks, R.1994, *Linear Hopfield Networks and Constrained  
Optimization*, Submitted for Government Review.