

## PERAMALAN MENGGUNAKAN METODE *BACKPROPAGATION NEURAL NETWORK*

Made Nita Dwi Sawitri<sup>1§</sup>, I Wayan Sumarjaya<sup>2</sup>, Ni Ketut Tari Tastrawati<sup>3</sup>

<sup>1</sup>Program Studi Matematika, Fakultas MIPA – Universitas Udayana [Email: [dwi.sawitri@gmail.com](mailto:dwi.sawitri@gmail.com)]

<sup>2</sup>Program Studi Matematika, Fakultas MIPA – Universitas Udayana [Email: [sumarjaya@unud.ac.id](mailto:sumarjaya@unud.ac.id)]

<sup>3</sup>Program Studi Matematika, Fakultas MIPA – Universitas Udayana [Email: [tastrawati@unud.ac.id](mailto:tastrawati@unud.ac.id)]

<sup>§</sup>Corresponding Author

### ABSTRACT

*The purpose of the study is to forecast the price of rice in the city of Denpasar in 2017 using backpropagation neural network method. Backpropagation neural network is a model of artificial neural network by finding the optimal weight value. Artificial neural networks are information processing systems that have certain performance characteristics similar to that of human neural networks. This analysis uses time series data of rice prices in the city of Denpasar from January 2001 until December 2016. The results of this research, concludes that the lowest rice price is predicted in July 2017 at Rp9791.5 while the highest rice price in April 2017 for Rp9839.4.*

**Keywords:** *Backpropagation Neural Network, Rice Price, Forecasting.*

### 1. PENDAHULUAN

*Backpropagation neural network* adalah salah satu model jaringan saraf tiruan dengan arsitektur *multilayer* yang sering digunakan dengan mencari bobot optimal pada jaringan saraf tiruan. Jaringan saraf tiruan atau *artificial neural network* telah berkembang sejak lima puluh tahun yang lalu. Menurut Fausett (1994) jaringan saraf tiruan merupakan sistem pemrosesan informasi yang memiliki karakteristik kinerja tertentu yang sama dengan jaringan saraf manusia. Pada *backpropagation neural network*, jaringan diberikan sepasang pola yang terdiri atas pola masukan (*input*) dan pola yang diinginkan (*target*).

Berbagai jenis data yang memiliki pola yang tidak teratur cenderung sulit untuk diramalkan seperti pada data produksi produk pertanian dan data harga produk pertanian. Hal ini disebabkan akhir-akhir ini terjadi perubahan iklim dan produksi hasil pertanian semakin sulit untuk diperkirakan sehingga harga produk pertanian sulit pula untuk diramal dan cenderung menyebabkan berfluktuasinya harga produk pertanian. Namun dengan

menggunakan metode *backpropagation neural network*, meramal harga produk pertanian dapat dilakukan karena melalui proses pelatihan jaringan saraf tiruan *backpropagation* dapat mengenali suatu pola di dalam data dan memperkirakan nilai-nilai masa depan.

Provinsi Bali memiliki produktivitas padi sawah yang didukung oleh sistem irigasi dan kelembagaan tani yang cukup baik. Terdapat organisasi kemasyarakatan yang khusus mengatur sistem pengairan sawah yang digunakan dalam cocok tanam padi di Bali yaitu subak. Semakin meningkatnya laju konversi lahan sawah ke penggunaan nonpertanian lain mengakibatkan usaha untuk meningkatkan produksi padi melalui ekstensifikasi tidak memungkinkan. BPS Provinsi Bali (2012) melaporkan bahwa dalam periode 1997 sampai 2011 laju konversi lahan pertanian di Bali rata-rata 436 ha/tahun. Hal ini berakibat pada semakin sempitnya lahan pertanian, terutama sawah, di Bali. Pada tahun 2011 luas lahan sawah di Bali 81.744 ha (14,5%) dan luas lahan pertanian bukan sawah

273.655 ha (48,6%), sementara luas lahan bukan pertanian 208.267 ha (37%). Terkait dengan kondisi harga beras, sepanjang tahun 2015 rata-rata harga beras di Kota Denpasar tercatat sebesar Rp 9847 per kg. Secara umum, penurunan harga beras umumnya terjadi pada saat musim panen raya yaitu pada bulan April-Juni (Badan Pusat Statistik Provinsi Bali, 2015). Namun mengingat akhir-akhir ini terjadi perubahan iklim dan produksi hasil pertanian semakin tidak menentu yang tentunya berdampak terhadap harga produk pertanian yang sulit diperkirakan.

Tujuan dari penelitian ini adalah untuk memperoleh arsitektur *backpropagation neural network* yang terbaik dengan menggunakan data set yang tersedia untuk peramalan harga beras di Kota Denpasar, dan meramalkan harga beras di Kota Denpasar menggunakan arsitektur yang diperoleh.

## 2. METODE PENELITIAN

### Jenis dan Sumber Data

Jenis data yang digunakan dalam penelitian ini adalah data sekunder berupa data harga beras eceran (Rp/kg) periode Januari 2001 sampai Desember 2016 yang diperoleh dari Kantor Badan Pusat Statistik Provinsi Bali.

### Arsitektur Jaringan Saraf Tiruan

Pada jaringan saraf tiruan, metode yang digunakan untuk melakukan peramalan adalah metode *backpropagation neural network* dengan momentum. Arsitektur jaringan saraf tiruan yang digunakan pada penelitian ini terdiri dari satu lapisan *input*, satu lapisan tersembunyi, dan satu lapisan *output*. Pada lapisan *input* jumlah neuron yang ditetapkan adalah 12 neuron mengacu pada jumlah bulan dalam satu tahun (12 bulan), pada lapisan *output* jumlah neuron yang ditetapkan adalah satu neuron mengacu pada waktu peramalan yang dilakukan berkisar satu bulan berikutnya, dan untuk lapisan tersembunyi jumlah neuron yang digunakan adalah 1 sampai 12 neuron. Fungsi aktivasi yang digunakan pada lapisan

tersembunyi adalah fungsi *sigmoid biner* atau fungsi *sigmoid bipolar*. Fungsi aktivasi pada lapisan *output* mengikuti fungsi aktivasi pada lapisan sebelumnya. Arsitektur terbaik dipilih berdasarkan pada nilai *mean square error* (MSE) yang terkecil dari hasil pelatihan dan validasi. Arsitektur dengan performa terbaik selanjutnya akan digunakan untuk melakukan peramalan harga beras periode Januari sampai Desember 2017.

### Pembagian Data

Data harga beras di Kota Denpasar yang berjumlah 192 data selanjutnya dibagi menjadi dua kelompok yaitu: (1) kelompok data periode Januari 2001 sampai Desember 2012. Sehingga pada kelompok satu terdapat sebanyak 144 data, (2) kelompok data periode Januari 2013 sampai Desember 2016. Sehingga pada kelompok dua terdapat sebanyak 48 data. Data tersebut kemudian dinormalisasikan sesuai dengan fungsi aktivasi yang digunakan.

### Bobot Awal

Proses awal adalah menginisialisasi nilai bobot dan bias awal menggunakan inisialisasi Nguyen-Widrow. Algoritme Nguyen-Widrow diawali dengan inisialisasi bobot awal secara acak dengan nilai antara  $-0,5$  dan  $0,5$ . Setelah itu dilakukan beberapa tahap perhitungan. Perhitungan pertama adalah menghitung faktor skala ( $\beta$ ). Nilai dari faktor skala ( $\beta$ ) adalah  $0,7^n \sqrt{p}$  dengan  $n$  adalah jumlah neuron pada lapisan *input* dan  $p$  adalah jumlah neuron pada lapisan tersembunyi. Nilai faktor skala dari lapisan *input* ke lapisan tersembunyi dapat dilihat pada Tabel 1.

Sedangkan untuk nilai vektor bobot dari lapisan tersembunyi ke lapisan *output* dilakukan dengan mengambil nilai acak antara  $-0,5$  sampai  $0,5$ . Perhitungan selanjutnya adalah menghitung besaran vektor bobot menggunakan persamaan

$$\|v_{ij}\| = \sqrt{v_{1j}^2 + v_{2j}^2 + \dots + v_{nj}^2}. \quad (1)$$

Kemudian bobot yang dipakai sebagai inisialisasi diperoleh dengan rumus

$$v_{ij} = \frac{\beta v_{ij}(lama)}{\|v_{ij}\|}. \quad (2)$$

Tabel 1. Nilai Faktor Skala dari Lapisan *Input* ke Lapisan Tersembunyi

No	Jumlah Neuron Pada Lapisan <i>Input</i> ( <i>n</i> )	Jumlah Neuron Pada Lapisan Tersembunyi ( <i>p</i> )	Nilai Faktor Skala ( $\beta$ ) dari Lapisan <i>Input</i> Ke Lapisan Tersembunyi
1	12	1	0,7000
2	12	2	0,7416
3	12	3	0,7671
4	12	4	0,7857
5	12	5	0,8005
6	12	6	0,8127
7	12	7	0,8232
8	12	8	0,8324
9	12	9	0,8407
10	12	10	0,8481
11	12	11	0,8548
12	12	12	0,8611

Sumber: Data diolah, 2018

### Metode Analisis Data

Adapun algoritme pelatihan standar *backpropagation neural network* adalah sebagai berikut:

**Langkah 1:** Definisikan pola masukan dan targetnya,

**Langkah 2:** Inisialisasi bobot awal,

**Langkah 3:** Tentukan maksimum interasi, target *error*, dan *learning rate* yang diinginkan,

**Langkah 4:** Kerjakan langkah-langkah berikut selama (*Epoch* < Maksimum *Epoch*) dan (*error* > target *error*).

#### Tahap I : Fase propagasi maju:

**Langkah 5:** Jumlahkan semua sinyal yang masuk ke lapisan unit. ( $x_i, i = 1,2,3, \dots, n$ ) ( $x_j, j = 1,2,3, \dots, p$ )

$$z_{net_j} = v_{0j} + \sum_i^n x_i v_{ij}. \quad (3)$$

Hitung keluaran semua lapisan unit *j* pada lapisan tersembunyi berdasarkan fungsi aktivasi

$$z_j = f(z_{net_j}). \quad (4)$$

**Langkah 6:** Jumlahkan semua sinyal yang masuk ke keluaran unit *k*, ( $y_k, k = 1,2,3, \dots, m$ )

$$y_{net_k} = w_{ok} + \sum_{j=1}^p z_j w_{jk}. \quad (5)$$

Gunakan fungsi aktivasi untuk menghitung sinyal *output* dan kirimkan sinyal tersebut ke semua unit lapisan atasnya (unit-unit *output*).

$$y_k = f(y_{net_k}). \quad (6)$$

#### Tahap II Fase propagansi mundur:

**Langkah 7:** Hitung faktor kesalahan pada keluaran *layer* (lapisan) ( $y_k, k = 1,2,3, \dots, m$ )

$$\delta_k = (t_k - y_k) f'(y_{net_k}). \quad (7)$$

Kemudian hitung koreksi bobot

$$\Delta w_{jk} = \alpha \delta_k z_j. \quad (8)$$

Hitung juga koreksi bias ;  $\Delta w_{ok} = \alpha \delta_k$ .

**Langkah 8:** Hitung penjumlahan kesalahannya ( $z_j, j = 1,2,3, \dots, p$ )

$$\delta_{net_j} = \sum_{k=1}^m \delta_k w_{jk}. \quad (9)$$

Kalikan nilai ini dengan turunan dari fungsi aktivasinya untuk menghitung informasi *error*

$$\delta_j = \delta_{net_j} f'(z_{net_j}). \quad (10)$$

Kemudian hitung korelasi bobot

$$\Delta v_{ji} = \alpha \delta_j x_i. \quad (11)$$

Hitung juga koreksi bias

$$\Delta v_{0j} = \alpha \delta_j. \quad (12)$$

#### Tahap III Perubahan bobot:

**Langkah 9:** Tiap-tiap unit *output* ( $Y_k, k = 1,2,3, \dots, m$ ) memperbaiki bias dan bobotnya ( $j = 0,1,2,3, \dots, p$ )

$$w_{jk}(baru) = w_{jk}(lama) + \Delta w_{jk}. \quad (13)$$

Tiap-tiap unit pada suatu lapisan tersembunyi ( $z_j, j = 1,2,3, \dots, p$ ) memperbaiki bias dan bobotnya ( $i = 0,1,2, \dots, n$ )

$$v_{ij}(baru) = v_{ij}(lama) + \Delta v_{ij}. \quad (14)$$

**Langkah 10:** Uji kondisi berakhir (akhir iterasi).

## 3. HASIL DAN PEMBAHASAN

### Pelatihan Jaringan

Setelah diperoleh nilai bobot dan bias awal menggunakan algoritme Nguyen-Widrow dilakukan tahap pelatihan jaringan. Tahap pelatihan jaringan menggunakan algoritma *backpropagation* dengan momentum dan *adaptive learning rate*. Pelatihan jaringan dilakukan menggunakan *software* Matlab R2014a.

Pelatihan dilakukan terhadap data-data pelatihan dengan *learning rate* ditentukan sebesar 0,1; nilai MSE target ditentukan sebesar 0,1, untuk nilai momentum ditentukan sebesar 0,9 dan nilai maksimum iterasi ditentukan sebesar 1.000 iterasi. Pada proses pelatihan ini, nilai *learning rate* bersifat *adaptive* yang berubah-ubah selama proses pelatihan. Proses pelatihan jaringan akan berhenti jika iterasi telah mencapai batas maksimum yang ditentukan yaitu 1.000 iterasi, atau pelatihan akan berhenti jika target *error* yang ditentukan telah tercapai.

Tahapan pelatihan jaringan akan dilakukan sebanyak dua kali. Pada setiap pelatihan jumlah neuron pada lapisan tersembunyi divariasikan pada rentang 1 hingga 12 neuron. Tabel 2 menunjukkan arsitektur jaringan yang dicoba pada tahap pelatihan.

Tabel 2. Arsitektur Jaringan pada Tahap Pelatihan

Pelatihan ke-	Jml <i>input</i>	Jml Neuron	Jml <i>Output</i>	Fungsi Aktivasi
1	12	1-12	1	<i>Sigm. Biner</i>
2	12	1-12	1	<i>Sigm. Bipolar</i>

Kedua arsitektur  $12 - n - 1$ ,  $n = 1, 2, \dots, 12$  selanjutnya dilatih dengan menggunakan *software* MATLAB, dengan hasil pelatihan sebagai berikut:

1. Hasil pelatihan dengan arsitektur jaringan  $12 - n - 1$  dengan fungsi aktivasi *sigmoid biner* (0,1) adalah seperti pada Tabel 3.

Pada Tabel 3 terlihat bahwa nilai MSE akhir terkecil dihasilkan oleh arsitektur yang

menggunakan 3 neuron pada lapisan tersembunyi yaitu sebesar 0,000645.

Tabel 3. Hasil pelatihan dengan arsitektur jaringan  $12 - n - 1$  dengan Fungsi Aktivasi *Sigmoid biner* (0,1)

Arsitektur Jaringan	Jumlah Neuron Tersembunyi ( $n$ )	Jumlah <i>Epoch</i> Maksimum	MSE
12-1-1	1	1.000	0,069313
12-2-1	2	1.000	0,032205
12-3-1	3	1.000	0,000645
12-4-1	4	1.000	0,001554
12-5-1	5	1.000	0,000699
12-6-1	6	1.000	0,000756
12-7-1	7	1.000	0,001299
12-8-1	8	1.000	0,001127
12-9-1	9	1.000	0,00078
12-10-1	10	1.000	0,001007
12-11-1	11	1.000	0,000831
12-12-1	12	1.000	0,001086

Sumber : Data diolah, 2018

2. Hasil pelatihan dengan arsitektur jaringan  $12 - n - 1$  dengan fungsi aktivasi *sigmoid bipolar* (-1,1) adalah sebagai berikut

Tabel 4. Hasil pelatihan dengan arsitektur jaringan  $12 - n - 1$  dengan Fungsi Aktivasi *Sigmoid bipolar* (-1,1)

Arsitektur Jaringan	Jumlah Neuron Tersembunyi ( $n$ )	Jumlah <i>Epoch</i> Maksimum	MSE
12-1-1	1	1.000	0,156499
12-2-1	2	1.000	0,15582
12-3-1	3	1.000	0,154846
12-4-1	4	1.000	0,155076
12-5-1	5	1.000	0,155284
12-6-1	6	1.000	0,155104
12-7-1	7	1.000	0,15515
12-8-1	8	1.000	0,155586

12-9-1	9	1.000	0,155031
12-10-1	10	1.000	0,155001
12-11-1	11	1.000	0,154666
12-12-1	12	1.000	0,155025

Sumber : Data diolah, 2018

Pada Tabel 4 terlihat bahwa nilai MSE akhir terkecil dihasilkan oleh arsitektur yang menggunakan 11 neuron pada lapisan tersembunyi yaitu sebesar 0,154666.

### Tahapan Validasi

Arsitektur yang digunakan pada tahap validasi adalah jaringan yang menghasilkan nilai MSE terkecil saat pelatihan. Tahap validasi dilakukan dengan menggunakan data yang tidak ikut dilatih pada masing-masing jaringan yaitu pada periode Januari 2015 sampai Desember 2016. Arsitektur jaringan terbaik dipilih dengan menihat nilai MSE terkecil yang dihasilkan pada tahap validasi.

1. Hasil validasi arsitektur jaringan 12 – 3 – 1 dengan fungsi aktivasi biner (0,1) adalah sebagai berikut:

Tabel 5. Hasil validasi Arsitektur Jaringan 12 – 3 – 1 dengan Fungsi Aktivasi Biner (0,1)

Data Ke-	Target	Output	Error
1	0,7335	0,70638	0,02712
2	0,7504	0,70665	0,04375
3	0,7631	0,70739	0,05571
...	...	...	...
...	...	...	...
36	0,8602	0,72332	0,13688

Sumber : Data diolah, 2018

Perhitungan nilai MSE antara *output* jaringan dan target, yaitu:

$$\begin{aligned}
 MSE &= \frac{\sum_{t=1}^n (t_k - y_k)^2}{n_{pola}} \\
 &= \frac{0,484995351}{36} \\
 &= 0,013472
 \end{aligned}$$

2. Hasil validasi arsitektur jaringan 12 – 11 – 1 dengan fungsi aktivasi bipolar (-1,1) adalah sebagai berikut:

Tabel 6. Hasil validasi Arsitektur Jaringan 12 – 11 – 1 dengan Fungsi Aktivasi Bipolar (-1,1)

Data Ke-	Target	Output	Error
1	0,5253	0,46734	0,05796
2	0,5634	0,46757	0,09583
3	0,5919	0,46867	0,12323
...	...	...	...
...	...	...	...
36	0,8104	0,48766	0,32274

Sumber : Data diolah, 2018

Perhitungan nilai MSE antara *output* jaringan dan target, yaitu:

$$\begin{aligned}
 MSE &= \frac{\sum_{t=1}^n (t_k - y_k)^2}{n_{pola}} \\
 &= \frac{2,6272636777}{36} \\
 &= 0,072979
 \end{aligned}$$

### Hasil Pelatihan dan Validasi Jaringan

Tabel 7. Hasil Pelatihan dan Validasi Jaringan

Pelatihan		Validasi	
Jaringan	MSE	MSE	MSE
Fungsi Aktivasi Sigmoid biner	12-3-1	0,000591	0,013472
Fungsi Aktivasi Sigmoid bipolar	12-11-1	0,141777	0,072979

Berdasarkan Tabel 3 dan Tabel 4 terlihat bahwa nilai MSE pelatihan terkecil pada jaringan yang menggunakan fungsi *sigmoid biner* dihasilkan oleh jaringan 12 – 3 – 1 yaitu sebesar 0,000591. Untuk jaringan yang menggunakan fungsi *sigmoid bipolar*, jaringan dengan arsitektur 12 – 11 – 1 menghasilkan nilai MSE terkecil yaitu sebesar 0,141777. Pada tahap validasi, nilai MSE paling minimum dihasilkan oleh jaringan dengan fungsi *sigmoid*

*bipolar* dengan arsitektur 12 – 3 – 1 yaitu sebesar 0,013472. Sehingga untuk meramalkan harga beras di Kota Denpasar dipilih jaringan yang terdiri dari 12 neuron *input*, 3 neuron tersembunyi, dan 1 neuron *output* dan menggunakan fungsi aktivasi *sigmoid biner*.

Tabel 8. Hasil Ramalan Harga Beras di Kota Denpasar Tahun 2017

Bulan	Hasil Ramalan
Januari	9813,2
Februari	9823,5
Maret	9819,8
April	9839,4
Mei	9803,9
Juni	9798,4
Juli	9791,5
Agustus	9826
September	9838,1
Oktober	9806,3
November	9837,2
Desember	9811,4

Hasil ramalan menunjukkan harga beras terendah diprediksikan terjadi pada bulan Juli yaitu seharga Rp9791,5 sedangkan harga beras tertinggi terjadi pada bulan April yaitu seharga Rp9839,4.

#### 4. KESIMPULAN DAN SARAN

Kesimpulan dalam penelitian ini adalah

1. Berdasarkan hasil pelatihan diperoleh arsitektur jaringan saraf tiruan dengan fungsi aktivasi *sigmoid biner* yang menghasilkan nilai MSE terkecil adalah jaringan yang terdiri dari satu unit *input*, tiga unit neuron lapisan tersembunyi dan satu unit lapisan *output* dengan nilai MSE yang dihasilkan sebesar 0,000645.
2. Berdasarkan hasil pelatihan diperoleh arsitektur jaringan saraf tiruan dengan fungsi aktivasi *sigmoid bipolar* yang menghasilkan nilai MSE terkecil adalah jaringan yang terdiri dari satu unit *input*, 11 unit neuron lapisan tersembunyi dan satu unit lapisan *output* dengan masing-masing nilai MSE yang dihasilkan sebesar 0,154666.

3. Berdasarkan hasil validasi diperoleh arsitektur jaringan terbaik untuk meramalkan harga beras di periode berikutnya adalah jaringan dengan fungsi aktivasi *sigmoid biner* yang terdiri dari satu unit *input*, tiga unit neuron lapisan tersembunyi dan satu unit lapisan *output* dengan nilai MSE sebesar 0,013472.
4. Hasil ramalan menunjukkan harga beras di Kota Denpasar pada periode 2017 akan mengalami fluktuasi tiap bulannya, dimana harga beras terendah diprediksikan terjadi pada bulan Juli yaitu sebesar Rp9791,5 sedangkan harga beras tertinggi terjadi pada bulan April yaitu sebesar Rp9839,4.

Arsitektur jaringan yang digunakan dalam penelitian ini belum dapat dikatakan sebagai hasil terbaik mengingat masih banyak kemungkinan kombinasi yang digunakan, baik itu pola *input* data, jumlah neuron pada lapisan tersembunyi, jumlah maksimum *epoch* dan nilai MSE target. Hal ini disebabkan karena pemilihan parameter-parameter tersebut harus dilakukan dengan cara *trial and error*. Selain itu, perlu adanya penelitian lebih lanjut tentang peramalan harga beras di Kota Denpasar dengan penambahan variabel lain yang kemungkinan berpengaruh terhadap harga beras sehingga untuk penelitian selanjutnya dapat menghasilkan peramalan yang lebih akurat.

#### DAFTAR PUSTAKA

- Badan Pusat Statistik Provinsi Bali. (2015). *Harga Konsumen Beberapa Barang Dan Jasa Di Ibukota Kabupaten Provinsi Bali 2015*. Bali: BPS Bali.
- Fausett, L. (1994). *Fundamentals of Neural Network (Architectures, Algorithms, and Applications)*. Upper Saddle River, New-Jersey: Prentice-Hall.
- Fithri, Y. A., Suryani, E., & Vinarti, R. A. (2012). *Analisis Fluktuasi dan Prediksi Harga Beras. Jurnal Teknik POMITS* Vol.1 No.1, 1-6.
- Herjanto, E. (2009). *Sains Manajemen*. Jakarta: Grasindo.

- Hu, Y. H., Hwang, & Jenq-Neng. (2002). *Handbook of Neural Network Signal Processing*. New York: CRC Press.
- Mardianto, S., & Suryana, A. (2001). *Bunga Rampai Ekonomi Beras*. Jakarta: Lembaga Penyelidikan Ekonomi dan Masyarakat, Fakultas Ekonomi, Universitas Indonesia.
- Pangastuti, P. (2014). *Metode Jaringan Saraf Tiruan Backpropagation untuk mengukur tingkat korelasi prestasi Mahasiswa (Studi Kasus pada Universitas Dian Nuswantoro Semarang)*. Semarang: Universitas Dian Nuswantoro.
- Santosa, P. B., & Hamdani, M. (2007). *Statistika Deskriptif dalam Bidang Ekonomi dan Niaga*. Jakarta: Erlangga.
- Siang, J. J. (2004). *Jaringan Saraf Tiruan dan Pemrogramannya Menggunakan Matlab*. Yogyakarta: Andi Offset.