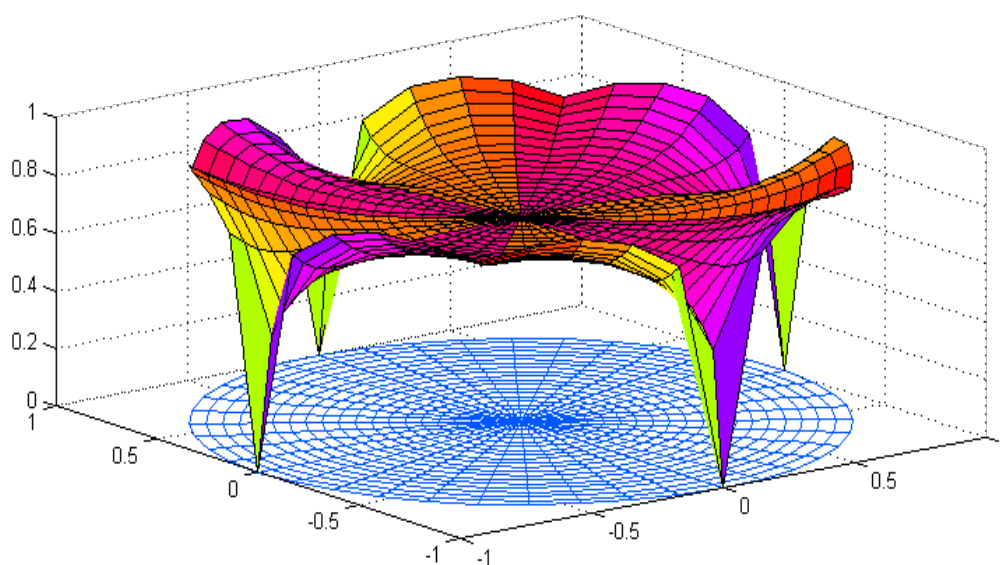


# **B**arekeng

**jurnal ilmu matematika dan terapan**

ISSN 1978-7227





**jurnal ilmu matematika dan terapan**

ISSN 1978-7227

**Volume 10 Nomor 2 | Desember 2016**

**PENANGGUNG JAWAB**

Ketua Jurusan Matematika  
FMIPA - Universitas Pattimura

**KETUA DEWAN REDAKSI**

M. I. Tilukay, S.Si., M.Si

**PENYUNTING AHLI**

Prof. Drs. Subanar, Ph.D (UGM Yogyakarta)  
Prof. Dr. Edi Tri Baskoro (ITB Bandung)  
Prof. Dr. Ir. Siswadi, M.Sc (IPB Bogor)  
Prof. Dr. Basuki Widodo, M.Sc (ITS Surabaya)  
Prof. Dr. Thomas Pentury, M.Si (Unpatti Ambon)  
Prof. Dr. T. G. Ratumanan, M.Pd. (Unpatti Ambon)

**PENYUNTING PELAKSANA**

B.P. Tomasouw, S.Si, M.Si  
D. L. Rahakbauw, S.Si, M.Si  
V. Y. Ilwaru, S.Si, M.Si

**PENERBIT (PUBLISHER)**

Jurusan Matematika FMIPA  
Universitas Pattimura Ambon

**SEKRETARIAT DAN ALAMAT EDITOR (EDITORIAL ADDRESS)**

Jurusan Matematika  
Fakultas Matematika dan Ilmu Pengetahuan Alam  
Universitas Pattimura  
Alamat:  
Kampus FMIPA UNPATTI  
Jl. Ir. M. Putuhena, Poka 97233  
Ambon - Maluku



## PENELITIAN

<b>KARAKTERISASI MATRIKS CENTRO-SIMETRIS</b> <i>The Characterization of Centrosymmetric Matrices</i>	Berny Pebo Tomasouw	69-76
<b>BILANGAN TERHUBUNG TITIK PELANGI UNTUK GRAF LINGKARAN BINTANG (<math>S_m C_n</math>)</b> <i>The Rainbow Vertex Connection Number of Star Cycle Graph (<math>S_m C_n</math>)</i>	Ariestha Widyastuty Bustan	77-81
<b>APLIKASI OPERASI HIMPUNAN DAN MATEMATIKA MORFOLOGI PADA PENGOLAHAN CITRA DIGITAL</b> <i>Application of Set Operation and Mathematics Morphology on Digital Image Processing</i>	V. Y. I. Ilwaru Y. A. Lesnussa E. M. Sahetapy Zeth A. Leleury	83-96
<b>ANALISIS DAN PREDIKSI PENYAKIT JANTUNG KORONER DI KOTA AMBON MENGGUNAKAN JARINGAN SARAF TIRUAN</b> <i>Analysis and Prediction of Coronary Heart Disease in Ambon City Using Neural Network</i>	Dorteus Lodewyik Rahakbauw Ferry Kondo Lembang Yuniasih M. J. Taihuttu	97-105
<b>MODEL REGULASI PADAPROSES BIODEGRADASI POLYETHYLENE TEREPHTHALATE (PET)</b> <i>Regulation Model on Biodegradation Process of Polyethylene Terephthalate (PET)</i>	Taufan Talib	107-115
<b>PERAMALAN JUMLAH KENDARAAN DI DKI JAKARTA DENGAN JARINGAN BACKPROPAGATION</b> <i>Forecasting Number of Vehicles in DKI Jakarta with Backpropagation Network</i>	Alwi Smith Zumrotus Sya'diyah	117-125
<b>PREDIKSI PENGGUNAAN BANDWIDTH MENGGUNAKAN ELMAN RECURRENT NEURAL NETWORK</b> <i>Prediction of Bandwidth Using Elman Recurrent Neural Network</i>	Jefri Radjabaycolle Reza Pulungan	127-135
<b>KINERJA DIAGRAM KONTROL W DAN DIAGRAM KONTROL G</b> <i>Performance of W Control Chart and G Control Chart</i>	M. Stivo Noya Van Delsen Mozart Winston Talakua	137-147



merupakan Jurnal Ilmu Matematika dan Terapannya sebagai suatu wahana informasi ilmiah yang menyajikan artikel (naskah) hasil penelitian meliputi bidang-bidang sebagai berikut: matematika analisis, aljabar, matematika terapan, statistika, pendidikan matematika dan ilmu komputer. Jurnal ini diterbitkan dua kali dalam setahun yaitu pada bulan Maret dan bulan Desember. Artikel atau naskah-naskah di dalam jurnal ini merupakan hasil-hasil penelitian pribadi ataupun kelompok yang belum pernah diterbitkan di jurnal-jurnal atau majalah ilmiah lainnya.

---

Diterbitkan oleh:

Jurusan Matematika

Fakultas Matematika dan Ilmu Pengetahuan Alam

Universitas Pattimura

Ambon

2016

Copyright © Jurusan Matematika FMIPA UNPATTI 2016

## KARAKTERISTIK MATRIKS *CENTRO*-SIMETRIS

**Berny Pebo Tomasouw**

Jurusan Matematika FMIPA Universitas Pattimura  
Jl. Ir. M. Putuhena, Kampus Unpatti, Poka-Ambon, Indonesia  
e-mail: peboberny@gmail.com

---

### Abstrak

Matriks *centro*-simetris merupakan matriks simetris yang memenuhi sifat tertentu. Dalam penelitian ini, akan dibahas bentuk dan beberapa sifat dasar dari matriks *centro*-simetris serta nilai eigen dan vektor eigennya.

*Kata Kunci:* Matriks *centro*-simetris, nilai eigen, vector eigen.

## THE CHARACTERISTICS OF CENTROSYMMETRIC MATRICES

### Abstract

Centrosymmetric is a symmetric matrix which satisfy some certain conditions. In this paper, we will discuss the form and some basic properties of the centrosymmetric matrix also its eigenvalues and eigenvectors.

*Keywords:* Centrosymmetric matrix, eigen value, eigen vector.

---

### 1. Pendahuluan

Konsep matriks *centro*-simetris muncul dalam buku yang ditulis A. C. Aitken [1] dengan judul “*Determinants and Matrices*”. Sedangkan F. Graybill juga membahas konsep matriks yang sama namun dengan nama matriks *cross*-simetris [2]. Walaupun memiliki nama yang mirip dengan matriks simetris namun sifat yang dimiliki oleh matriks *centro*-simetris sangatlah berbeda.

Beberapa penelitian memperlihatkan bahwa matriks *centro*-simetris mempunyai peranan penting dalam masalah analisis numerik, teori informasi, proses Markov, persamaan diferensial, persegi ajaib maupun pengenalan pola [3-8]. Oleh karena itu, dalam penelitian ini akan dibahas karakteristik dari matriks *centro*-simetris beserta nilai eigen dan vektor eigennya.

#### Definisi 1. (Matriks Simetris)

Matriks  $A \in \mathbf{M}_n(\mathbb{R})$  dikatakan simetris jika berlaku

$$A = A^T$$

#### Definisi 2. (Nilai Eigen dan Vektor Eigen)

Diberikan matriks  $A \in \mathbf{M}_n(\mathbb{R})$  dan vektor tak nol  $\mathbf{x} \in \mathbb{R}^n$ . Skalar  $\lambda \in \mathbb{R}$  merupakan nilai eigen dari A jika memenuhi  $A\mathbf{x} = \lambda\mathbf{x}$ . Vektor  $\mathbf{x}$  disebut vektor eigen yang bersesuaian dengan nilai eigen  $\lambda$ .

**Definisi 3. (Matriks Similar)**

Diberikan matriks  $A, B \in \mathbf{M}_n(\mathbb{R})$ . Matriks  $A$  dikatakan similar dengan  $B$  jika terdapat matriks non-singular  $P$  sedemikian sehingga berlaku

$$A = PBP^{-1}.$$

**Lema 1.** Jika matriks  $A$  similar dengan matriks  $B$  maka  $\det(A) = \det(B)$ .

**Akibat 1.** Misalkan matriks  $A$  similar dengan matriks  $B$ . Jika  $\lambda$  adalah nilai eigen dari  $A$  maka  $\lambda$  juga merupakan nilai eigen dari  $B$ .

**Definisi 4. (Matriks Ortogonal)**

Matriks non-singular  $Q \in \mathbf{M}_n(\mathbb{R})$  disebut matriks ortogonal jika memenuhi  $Q^T = Q^{-1}$ .

**Definisi 5. (Matriks contra-identitas)**

Matriks persegi  $J_n$  disebut matriks contra-identitas jika berbentuk

$$J_n = \begin{bmatrix} 0 & 0 & \cdots & 0 & 1 \\ 0 & 0 & \ddots & 1 & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 1 & \ddots & 0 & 0 \\ 1 & 0 & \cdots & 0 & 0 \end{bmatrix}$$

**Lema 2.** Matriks contra-identitas memenuhi sifat-sifat berikut :

- $J_n = J_n^T$ .
- $J_n^{-1} = J_n$ .
- $J_n^k = \begin{cases} I_n, & \text{jika } k \text{ genap} \\ J_n, & \text{jika } k \text{ ganjil} \end{cases}$

**Definisi 6. (Rotasi Matriks)**

Diberikan  $A \in \mathbf{M}_n(\mathbb{R})$ . Rotasi dari matriks  $A$  dinotasikan dengan  $A^R$  dan didefinisikan sebagai

$$A^R = J_n A J_n.$$

**Lema 3.** Rotasi matriks  $A$  memenuhi sifat-sifat berikut :

- $(A^R)^R = A$ .
- $(A^R)^T = (A^T)^R$ .
- $(A^{-1})^R = (A^R)^{-1}$ , jika  $A$  non-singular.

## 2. Hasil dan Pembahasan

### Definisi 7. (Matriks *Centro-Simetris*)

Diberikan matriks  $S$  berorde  $n \times n$ . Matriks  $S$  disebut matriks *centro-simetris* jika memenuhi

$$S^R = S$$

**Contoh 1.** Matriks  $S = \begin{bmatrix} 3 & -2 \\ -2 & 3 \end{bmatrix}$  adalah matriks *centro-simetris* karena

$$\begin{aligned} S^R &= J_2 S J_2 \\ &= \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 3 & -2 \\ -2 & 3 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \\ &= \begin{bmatrix} 3 & -2 \\ -2 & 3 \end{bmatrix} \\ &= S \end{aligned}$$

**Contoh 2.** Matriks  $S = \begin{bmatrix} 2 & 1 & 3 \\ 0 & 5 & 0 \\ 3 & 1 & 2 \end{bmatrix}$  adalah matriks *centro-simetris*.

Contoh 1 di atas memperlihatkan contoh matriks simetris yang juga merupakan matriks *centro-simetris*. Namun hal ini tidak berlaku umum ataupun sebaliknya seperti tampak pada contoh 2 di atas. Lema 4 berikut ini memperlihatkan bentuk umum dari matriks *centro-simetris* berukuran  $2 \times 2$  dan  $3 \times 3$ .

### Lema 4.

- i. Jika  $S = \begin{bmatrix} a & b \\ b & a \end{bmatrix}$  maka  $S$  adalah matriks *centro-simetris*.
- ii. Jika  $S = \begin{bmatrix} a & c & b \\ d & e & d \\ b & c & a \end{bmatrix}$  maka  $S$  adalah matriks *centro-simetris*.

Selanjutnya, bentuk umum matriks *centro-simetris* dengan orde  $n \geq 4$  dibagi ke dalam 2 kasus yakni  $n$  genap dan  $n$  ganjil. Untuk kasus  $n$  genap yakni  $n = 2m$  maka misalkan matriks  $S$  berbentuk matriks blok sebagai berikut

$$S = \begin{bmatrix} A & C \\ B & D \end{bmatrix},$$

dengan  $A, B, C$ , dan  $D$  adalah matriks persegi berorde  $m$ . Matriks  $S$  harus memenuhi  $S^R = S$  sehingga

$$J_n S J_n = S$$

$$\begin{bmatrix} O_m & J_m \\ J_m & O_m \end{bmatrix} \begin{bmatrix} A & C \\ B & D \end{bmatrix} \begin{bmatrix} O_m & J_m \\ J_m & O_m \end{bmatrix} = \begin{bmatrix} A & C \\ B & D \end{bmatrix}$$

$$\begin{bmatrix} J_m D J_m & J_m B J_m \\ J_m C J_m & J_m A J_m \end{bmatrix} = \begin{bmatrix} A & C \\ B & D \end{bmatrix}$$

Dari bentuk terakhir diperoleh  $D = J_m A J_m$  dan  $C = J_m B J_m$ . Hasil ini diperlihatkan dalam Lema berikut.

**Lema 5.** Jika  $S$  adalah matriks *centro*-simetris berorde  $n$  genap yakni  $n = 2m$  maka  $S$  dapat ditulis dalam bentuk matriks blok sebagai berikut

$$S = \begin{bmatrix} A & J_m B J_m \\ B & J_m A J_m \end{bmatrix},$$

dengan  $A$  dan  $B$  adalah matriks persegi berorde  $m$ .

Bentuk matriks  $S$  pada Lema di atas bukanlah bentuk tunggal karena matriks  $S$  dapat juga ditulis dalam bentuk

$$S = \begin{bmatrix} A & B \\ J_m B J_m & J_m A J_m \end{bmatrix} \text{ atau } S = \begin{bmatrix} A & J_m B \\ B J_m & J_m A J_m \end{bmatrix}.$$

**Contoh 3.** Matriks  $S = \begin{bmatrix} 2 & 1 & 2 & -3 \\ 3 & 5 & 3 & 1 \\ 1 & 3 & 5 & 3 \\ -3 & 2 & 1 & 2 \end{bmatrix}$  adalah matriks *centro*-simetris karena  $S$  dapat ditulis dalam bentuk

$$S = \begin{bmatrix} A & J_m B J_m \\ B & J_m A J_m \end{bmatrix} \text{ dengan } A = \begin{bmatrix} 2 & 1 \\ 3 & 5 \end{bmatrix} \text{ dan } B = \begin{bmatrix} 1 & 3 \\ -3 & 2 \end{bmatrix}.$$

Selanjutnya, untuk matriks persegi berorde  $n$  ganjil yakni  $n = 2m + 1$  maka misalkan  $S$  berbentuk matriks blok sebagai berikut

$$S = \begin{bmatrix} A & \mathbf{p} & C \\ \mathbf{q}^T & r & \mathbf{r}^T \\ B & \mathbf{s} & D \end{bmatrix}$$

dengan  $A, B, C, D$  adalah matriks persegi berorde  $m$ , vektor  $\mathbf{p}, \mathbf{q}, \mathbf{r}, \mathbf{s} \in \mathbb{R}^m$  dan  $r \in \mathbb{R}$ . Matriks  $S$  harus memenuhi  $S^R = S$  sehingga

$$J_n S J_n = S$$

$$\begin{bmatrix} O_m & \mathbf{0} & J_m \\ \mathbf{0}^T & 1 & \mathbf{0}^T \\ J_m & \mathbf{0} & O_m \end{bmatrix} \begin{bmatrix} A & \mathbf{p} & C \\ \mathbf{q}^T & r & \mathbf{r}^T \\ B & \mathbf{s} & D \end{bmatrix} \begin{bmatrix} O_m & \mathbf{0} & J_m \\ \mathbf{0}^T & 1 & \mathbf{0}^T \\ J_m & \mathbf{0} & O_m \end{bmatrix} = \begin{bmatrix} A & \mathbf{p} & C \\ \mathbf{q}^T & r & \mathbf{r}^T \\ B & \mathbf{s} & D \end{bmatrix}$$

$$\begin{bmatrix} J_m D J_m & J_m \mathbf{s} & J_m B J_m \\ \mathbf{r}^T J_m & r & \mathbf{q}^T J_m \\ J_m C J_m & J_m \mathbf{p} & J_m A J_m \end{bmatrix} = \begin{bmatrix} A & \mathbf{p} & C \\ \mathbf{q}^T & r & \mathbf{r}^T \\ B & \mathbf{s} & D \end{bmatrix}$$

Dari bentuk terakhir diperoleh persamaan :

- i.  $J_m D J_m = A$  dan  $D = J_m A J_m$ .
- ii.  $J_m \mathbf{s} = \mathbf{p}$  dan  $J_m \mathbf{p} = \mathbf{s}$ .
- iii.  $\mathbf{r}^T J_m = \mathbf{q}^T$  dan  $\mathbf{r}^T = \mathbf{q}^T J_m$ .
- iv.  $J_m C J_m = B$  dan  $C = J_m B J_m$ .

Dari hasil ini dapat dibentuk Lema berikut.



**Lema 6.** Jika  $S$  adalah matriks *centro*-simetris berorde  $n$  ganjil yakni  $n = 2m + 1$  maka  $S$  dapat ditulis dalam bentuk matriks blok sebagai berikut

$$S = \begin{bmatrix} A & \mathbf{p} & J_m B J_m \\ \mathbf{q}^T & r & \mathbf{q}^T J_m \\ B & J_m \mathbf{p} & J_m A J_m \end{bmatrix},$$

dengan  $A, B$  adalah matriks persegi berorde  $m$ , vektor  $\mathbf{p}, \mathbf{q} \in \mathbb{R}^m$  dan  $r \in \mathbb{R}$ .

Selanjutnya, Lema berikut akan membantu dalam mencari matriks yang similar dengan matriks *centro*-simetris.

**Lema 7.**

a. Matriks  $Q_1 = \frac{1}{\sqrt{2}} \begin{bmatrix} I_m & I_m \\ -J_m & J_m \end{bmatrix}$  adalah matriks ortogonal berorde  $n = 2m$ .

b. Matriks  $Q_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} I_m & \mathbf{0} & I_m \\ \mathbf{0}^T & \sqrt{2} & \mathbf{0}^T \\ -J_m & \mathbf{0} & J_m \end{bmatrix}$  adalah matriks ortogonal berorde  $n = 2m + 1$ .

Dengan menggunakan matriks  $Q_1$  dan  $Q_2$  pada Lema ini maka dapat dicari matriks yang similar dengan matriks *centro*-simetris.

Untuk kasus  $n$  genap maka misalkan  $D_1$  adalah matriks yang similar. Matriks *centro*-simetris  $S$  dan  $D_1$  harus memenuhi  $S = Q_1 D_1 Q_1^T$  atau

$$\begin{aligned} D_1 &= Q_1^T S Q_1 \\ &= \left( \frac{1}{\sqrt{2}} \begin{bmatrix} I_m & -J_m \\ I_m & J_m \end{bmatrix} \right) \begin{bmatrix} A & J_m B J_m \\ B & J_m A J_m \end{bmatrix} \left( \frac{1}{\sqrt{2}} \begin{bmatrix} I_m & I_m \\ -J_m & J_m \end{bmatrix} \right) \\ &= \frac{1}{2} \begin{bmatrix} 2(A - J_m B) & O_m \\ O_m & 2(A + J_m B) \end{bmatrix} \\ &= \begin{bmatrix} A - J_m B & O_m \\ O_m & A + J_m B \end{bmatrix} \end{aligned}$$

Sedangkan untuk kasus  $n$  ganjil diperoleh matriks  $D_2$  yang similar yaitu

$$\begin{aligned} D_2 &= Q_2^T S Q_2 \\ &= \left( \frac{1}{\sqrt{2}} \begin{bmatrix} I_m & \mathbf{0} & -J_m \\ \mathbf{0}^T & \sqrt{2} & \mathbf{0}^T \\ I_m & \mathbf{0} & J_m \end{bmatrix} \right) \begin{bmatrix} A & \mathbf{p} & J_m B J_m \\ \mathbf{q}^T & r & \mathbf{q}^T J_m \\ B & J_m \mathbf{p} & J_m A J_m \end{bmatrix} \left( \frac{1}{\sqrt{2}} \begin{bmatrix} I_m & \mathbf{0} & I_m \\ \mathbf{0}^T & \sqrt{2} & \mathbf{0}^T \\ -J_m & \mathbf{0} & J_m \end{bmatrix} \right) \\ &= \frac{1}{2} \begin{bmatrix} 2(A - J_m B) & \mathbf{0} & O_m \\ \mathbf{0}^T & 2r & 2\sqrt{2} \mathbf{q}^T \\ O_m & 2\sqrt{2} \mathbf{p} & 2(A + J_m B) \end{bmatrix} \end{aligned}$$

$$= \begin{bmatrix} A - J_m B & \mathbf{0} & O_m \\ \mathbf{0}^T & r & \sqrt{2} \mathbf{q}^T \\ O_m & \sqrt{2} \mathbf{p} & A + J_m B \end{bmatrix}$$

Dari hasil ini dapat dibentuk Teorema berikut.

**Teorema 1.**

a. Jika  $S = \begin{bmatrix} A & J_m B J_m \\ B & J_m A J_m \end{bmatrix}$  dan  $Q_1 = \frac{1}{\sqrt{2}} \begin{bmatrix} I_m & I_m \\ -J_m & J_m \end{bmatrix}$  maka  $S$  similar dengan matriks diagonal blok

$$D_1 = \begin{bmatrix} A - J_m B & O_m \\ O_m & A + J_m B \end{bmatrix}.$$

b. Jika  $S = \begin{bmatrix} A & \mathbf{p} & J_m B J_m \\ \mathbf{q}^T & r & \mathbf{q}^T J_m \\ B & J_m \mathbf{p} & J_m A J_m \end{bmatrix}$  dan  $Q_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} I_m & \mathbf{0} & I_m \\ \mathbf{0}^T & \sqrt{2} & \mathbf{0}^T \\ -J_m & \mathbf{0} & J_m \end{bmatrix}$  maka  $S$  similar dengan matriks diagonal

$$\text{blok } D_2 = \begin{bmatrix} A - J_m B & \mathbf{0} & O_m \\ \mathbf{0}^T & r & \sqrt{2} \mathbf{q}^T \\ O_m & \sqrt{2} \mathbf{p} & A + J_m B \end{bmatrix}$$

Dari Teorema 1 diperoleh tiga akibat sebagai berikut.

**Akibat 2.** Determinan dari matriks centro-simetris adalah

a. Untuk  $n$  genap

$$\det \left( \begin{bmatrix} A & J_m B J_m \\ B & J_m A J_m \end{bmatrix} \right) = \det(A - J_m B) \det(A + J_m B).$$

b. Untuk  $n$  ganjil

$$\det \left( \begin{bmatrix} A & \mathbf{p} & J_m B J_m \\ \mathbf{q}^T & r & \mathbf{q}^T J_m \\ B & J_m \mathbf{p} & J_m A J_m \end{bmatrix} \right) = \det(A - J_m B) \det \left( \begin{bmatrix} r & \sqrt{2} \mathbf{q}^T \\ \sqrt{2} \mathbf{p} & A + J_m B \end{bmatrix} \right).$$

**Akibat 3.** Jika  $S = \begin{bmatrix} A & J_m B J_m \\ B & J_m A J_m \end{bmatrix}$  serta matriks  $A - J_m B$  dan  $A + J_m B$  tak singular maka

$$S^{-1} = \frac{1}{2} \begin{bmatrix} F^{-1} + G^{-1} & (G^{-1} - F^{-1}) J_m \\ J_m (G^{-1} - F^{-1}) & J_m (F^{-1} + G^{-1}) J_m \end{bmatrix}.$$

dengan  $F = A - J_m B$  dan  $G = A + J_m B$ .

**Akibat 4.** Nilai eigen dari matriks *centro*-simetris adalah

a. Untuk  $n$  genap

Nilai eigen dari  $S = \begin{bmatrix} A & J_m B J_m \\ B & J_m A J_m \end{bmatrix}$  adalah nilai eigen dari matriks  $A - J_m B$  dan nilai eigen dari matriks  $A + J_m B$ .

b. Untuk  $n$  ganjil

Nilai eigen dari  $S = \begin{bmatrix} A & \mathbf{p} & J_m B J_m \\ \mathbf{q}^T & r & \mathbf{q}^T J_m \\ B & J_m \mathbf{p} & J_m A J_m \end{bmatrix}$  adalah nilai eigen dari matriks  $A - J_m B$  dan nilai eigen dari matriks  $\begin{bmatrix} r & \sqrt{2} \mathbf{q}^T \\ \sqrt{2} \mathbf{p} & A + J_m B \end{bmatrix}$ .

Bentuk vektor eigen dari matriks *centro*-simetris diperlihatkan dalam Teorema berikut.

**Teorema 2.** Diberikan matriks *centro*-simetris  $S = \begin{bmatrix} A & J_m B J_m \\ B & J_m A J_m \end{bmatrix}$ .

a. Jika vektor eigen dari  $A - J_m B$  adalah  $\mathbf{y}$  maka vektor eigen dari  $S$  adalah  $\mathbf{v} = \begin{bmatrix} \mathbf{y} \\ -J_m \mathbf{y} \end{bmatrix}$ .

b. Jika vektor eigen dari  $A + J_m B$  adalah  $\mathbf{z}$  maka vektor eigen dari  $S$  adalah  $\mathbf{w} = \begin{bmatrix} \mathbf{z} \\ J_m \mathbf{z} \end{bmatrix}$ .

**Bukti.**

a. Misalkan nilai eigen dari  $A - J_m B$  adalah  $\lambda$  dengan vektor eigen yang bersesuaian adalah  $\mathbf{y}$ . Harus ditunjukkan bahwa berlaku  $S\mathbf{v} = \lambda \mathbf{v}$ .

$$\begin{aligned} S\mathbf{v} &= \begin{bmatrix} A & J_m B J_m \\ B & J_m A J_m \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ -J_m \mathbf{y} \end{bmatrix} \\ &= \begin{bmatrix} A\mathbf{y} - J_m B\mathbf{y} \\ B\mathbf{y} - J_m A\mathbf{y} \end{bmatrix} \\ &= \begin{bmatrix} (A - J_m B)\mathbf{y} \\ -J_m (A - J_m B)\mathbf{y} \end{bmatrix} \\ &= \begin{bmatrix} \lambda \mathbf{y} \\ -J_m \lambda \mathbf{y} \end{bmatrix} \\ &= \lambda \begin{bmatrix} \mathbf{y} \\ -J_m \mathbf{y} \end{bmatrix} \\ &= \lambda \mathbf{v} \end{aligned}$$

Karena  $S\mathbf{v} = \lambda \mathbf{v}$  maka terbukti  $\mathbf{v} = \begin{bmatrix} \mathbf{y} \\ -J_m \mathbf{y} \end{bmatrix}$  adalah vektor eigen dari  $S$ .

b. Misalkan nilai eigen dari  $A + J_m B$  adalah  $s$  dengan vektor eigen yang bersesuaian adalah  $\mathbf{z}$ .

Harus ditunjukkan bahwa berlaku  $S\mathbf{w} = s\mathbf{w}$ .

$$\begin{aligned}
 S\mathbf{v} &= \begin{bmatrix} A & J_m B J_m \\ B & J_m A J_m \end{bmatrix} \begin{bmatrix} \mathbf{z} \\ J_m \mathbf{z} \end{bmatrix} \\
 &= \begin{bmatrix} A\mathbf{z} + J_m B\mathbf{z} \\ B\mathbf{z} + J_m A\mathbf{z} \end{bmatrix} \\
 &= \begin{bmatrix} (A + J_m B)\mathbf{z} \\ J_m (A + J_m B)\mathbf{z} \end{bmatrix} \\
 &= \begin{bmatrix} s\mathbf{z} \\ J_m s\mathbf{z} \end{bmatrix} \\
 &= s \begin{bmatrix} \mathbf{z} \\ J_m \mathbf{z} \end{bmatrix} \\
 &= s\mathbf{w}
 \end{aligned}$$

Karena  $S\mathbf{w} = s\mathbf{w}$  maka terbukti bahwa  $\mathbf{w} = \begin{bmatrix} \mathbf{z} \\ J_m \mathbf{z} \end{bmatrix}$  adalah vektor eigen dari  $S$ .

## Daftar Pustaka

- [1] A.C.Aitken., Determinants and Matrices, Edinburgh. Oliver and Boyd, 1939.
- [2] F. Graybill, Introduction to Matrices with Application in Statistics, Wadsworth. Belmont., 1969.
- [3] A. C. a. P. Butler, "Properties of the Eigen vectors of Persymmetric Matrices with Applications to Communication Theory. IEEE Transactions on Communications.," vol. 24, pp. pp 804-809, 1976.
- [4] J. Delmas, "On Adaptive EVD Asymptotic Distribution of Centro-Symmetric Covariance Matrices". IEEE Transactions on Signal Processing," Vol.47, pp.1402-1406, 1999.
- [5] W. Chen, Y.Yu and X. Wang, "Reducing the Computational Requirement of Differential Quadrature Method". Numerical Methods for Partial Differential Equations," Vol.12, pp.565-577, 1996.
- [6] F. Stenger, "Matrices of Sinc Methods," *Journal of Computational and Applied Mathematics*, Vol. 86, pp.297-310, 1997.
- [7] R. B. Mattingly, "Even Order Regular Magic Squares Are Singular," *The American Mathematical Monthly*, Vol.107, pp. pp.777-782, 2000.
- [8] L. Datta and S. Morgera, "Some Results on Matrix Symmetries and a Pattern Recognition Application," *IEEE Transactions on Signal Processing*, Vol.34, No.4, pp.992-994, 1986.

## BILANGAN TERHUBUNG TITIK PELANGI UNTUK GRAF LINGKARAN BINTANG ( $S_m C_n$ )

Ariestha Widyastuty Bustan

Program Studi Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam  
Universitas Pasifik Morotai, Jl Siswa Darame, Kec. Morotai Selatan,  
Kab. Pulau morotai, Maluku Utara, Indonesia  
e-mail: ariesthawidyastutybustan@gmail.com

---

### Abstrak

Pewarnaan pada graf  $G = (V(G), E(G))$  dikatakan terhubung titik pelangi, jika untuk setiap dua titik yang berbeda  $u$  dan  $v$  di  $V(G)$ , terdapat lintasan  $u - v$  dengan semua titik internalnya memiliki warna yang berbeda. Bilangan terhubung titik pelangi dari  $G$ , dinotasikan dengan  $r_1(G)$ , adalah minimum banyak warna yang dibutuhkan sehingga  $G$  terhubung titik pelangi. Misalkan  $m$  dan  $n$  adalah dua bilangan bulat positif dengan  $m \geq 3$  dan  $n \geq 3$ ,  $S_m$  adalah graf bintang dengan  $m + 1$  titik, dan  $C_n$  adalah graf lingkaran dengan  $n$  titik. Graf lingkaran bintang ( $S_m C_m$ ) adalah graf yang diperoleh dengan menempelkan satu salinan graf  $C_n$  ke masing-masing titik pendant graf  $S_m$ . Pada paper ini ditentukan bilangan terhubung titik pelangi untuk graf lingkaran bintang.

*Kata Kunci:* bilangan terhubung titik pelangi, bintang, lingkaran, pewarnaan pelangi.

## THE RAINBOW VERTEX CONNECTION NUMBER OF STAR CYCLE GRAPHS ( $S_m C_n$ )

### Abstract

A vertex-colored graph  $G = (V(G), E(G))$  is said to be rainbow vertex-connected, if for every two vertices  $u$  and  $v$  in  $V(G)$ , there exist a  $u - v$  path with all internal vertices have distinct colors. The rainbow vertex-connection number of  $G$ , denoted by  $r_1(G)$ , is the smallest number of colors needed to make  $G$  rainbow vertex-connected. Let  $m$  and  $n$  be two integers at least 3,  $S_m$  be a star with  $m + 1$  vertices, and  $C_n$  be a cycle with  $n$  vertices. A star cycle ( $S_m C_n$ ) is a graph obtained by embedding a copy of  $C_n$  to each pendant of  $S_m$ . In this paper, we determine the rainbow vertex connection number of star cycle graphs.

*Keywords:* cycle, rainbow vertex-coloring, rainbow vertex connection-number, star.

---

### 1. Pendahuluan

Misalkan  $G$  adalah graf sederhana, berhingga dan terhubung tak trivial, maka didefinisikan suatu pewarnaan titik dari  $G$  yaitu  $\alpha : V(G) \rightarrow \{1, 2, \dots, n\}$  untuk suatu bilangan bulat  $n$  sedemikian sehingga untuk setiap dua titik yang bertetangga memiliki warna yang berbeda. Suatu lintasan  $P$  pada  $G$  merupakan lintasan titik pelangi jika titik-titik dalam lintasan  $u - v$  untuk setiap  $u, v \in V(G)$  memiliki warna yang berbeda. Pewarnaan titik pada  $G$  dikatakan terhubung titik pelangi, jika setiap dua titik dihubungkan oleh paling sedikit satu lintasan titik pelangi. Bilangan terhubung titik pelangi dari graf terhubung  $G$ , dinotasikan dengan  $r_1(G)$ , adalah warna minimum yang dibutuhkan untuk membuat  $G$  terhubung titik pelangi.

Krivelevich dan Yuster memberikan batasan untuk bilangan terhubung titik pelangi, yaitu

**Teorema 1.1** Misalkan  $n \geq 3$  dan  $G$  graf terhubung dengan  $n$  titik, maka  $(G) - 1 \leq r_1(G) \leq n - 2$ .

**Teorema 1.2** Misalkan  $G$  graf terhubung dengan  $c$  titik pemotong, maka  $r_1(G) \geq c$ .

**2. Hasil dan Pembahasan**

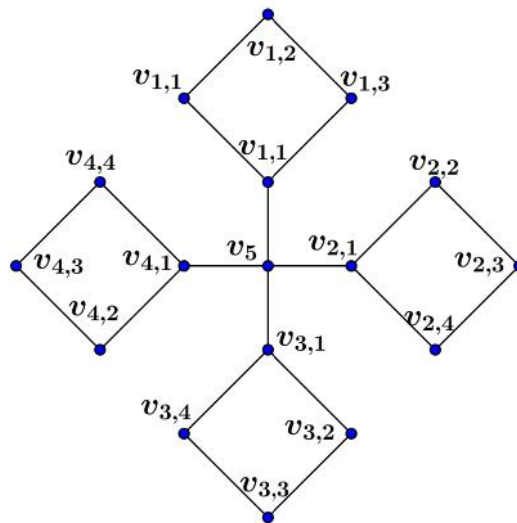
**Definisi 2.1.** Misalkan  $m$  dan  $n$  adalah dua bilangan positif dengan  $m \geq 3$  dan  $n \geq 3$ ,  $S_m$  adalah graf bintang dengan  $m + 1$  titik,  $C_n$  adalah graf lingkaran dengan  $n$  titik. Graf lingkaran bintang yang dinotasikan dengan  $S_m C_n$  adalah graf yang diperoleh dengan menempelkan satu salinan graf  $C_n$  ke masing-masing titik pendants graf  $S_m$ .

Graf  $S_m C_n$  memiliki himpunan titik dan himpunan sisi yang dituliskan secara berturut-turut sebagai berikut:

$$V(S_m C_n) = \{v_{i,j} | i, j \in [1, m]\} \cup \{v_{m+1}\}, d$$

$$E(S_m C_n) = \{v_{m+1}v_{i,1} | i \in [1, m]\} \cup \{v_{i,j}v_{i,j+1} | i \in [1, m], j \in [1, m-1]\}$$

$$\cup \{v_{i,m}v_{i,1} | i \in [1, m]\} \cup \{v_{i,j}v_{i,j+\frac{m}{2}} | i \in [1, m], j \in [1, \frac{m}{2}]\}$$



Gambar 1. Graf  $S_4 C_4$

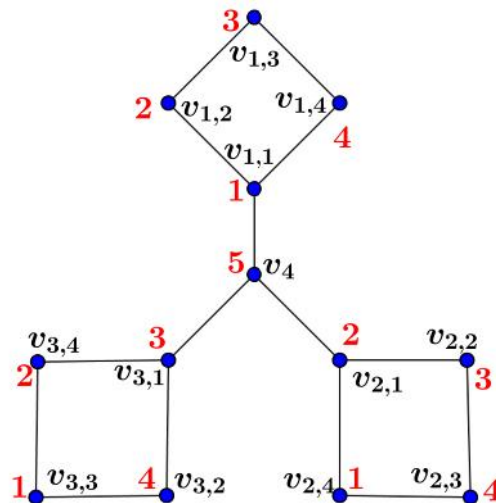
**Teorema 2.2.** Misalkan  $m$  dan  $n$  adalah dua bilangan positif dengan  $3 \leq d \leq n \leq [3,5] \cdot S_m C_n$  adalah graf lingkaran bintang, maka

$$r(S_m C_n) = \begin{cases} d - 1 & u = S_3 C_4 \\ c & u = m, 3n = 3 \\ d & u = m, 4d = n = 4 \\ d + 1 & u = S_3 C_5 \\ & u = S_4 C_5 \end{cases}$$

**Bukti.**

**Kasus 1.**  $r(S_3 C_4) = 5$

Berdasarkan teorema 1.1 diperoleh  $r(S_3 C_4) = d(S_3 C_4) - 1 = 6 - 1 = 5$ . Sehingga diperoleh suatu pewarnaan 5-titik pelangi pada graf  $S_3 C_4$  seperti yang ditunjukkan pada Gambar 2.

Gambar 2. Graf  $S_3C_4$ **Kasus 2.**  $r(S_mC_n) = c$ 

Berdasarkan teorema 1.2 diperoleh  $r(S_mC_n) = c = m + 1$ . Selanjutnya akan didefinisikan suatu pewarnaan  $\alpha: V(S_mC_n) \rightarrow [1, m + 1]$  sebagai berikut:

$$\alpha(v_{m+1}) = m + 1$$

$$\alpha(v_{i,j}) = \begin{cases} i + j - 1, & \text{if } i + j = m + 1, d \in [1, m]; \\ (i + j)m - d(m + 1), & \text{if } l \in [1, m]. \end{cases}$$

**Kasus 3.**  $r(S_3C_5) = d$ 

Berdasarkan teorema 1.1, diperoleh  $r(S_3C_5) = d - 1 = 6 - 1 = 5$ . Andaikan terdapat suatu pewarnaan titik pelangi pada  $(S_3C_5)$  dengan 5 warna. Tanpa mengurangi keumuman warnai titik pemotong dengan warna berbeda dengan aturan sebagai berikut:

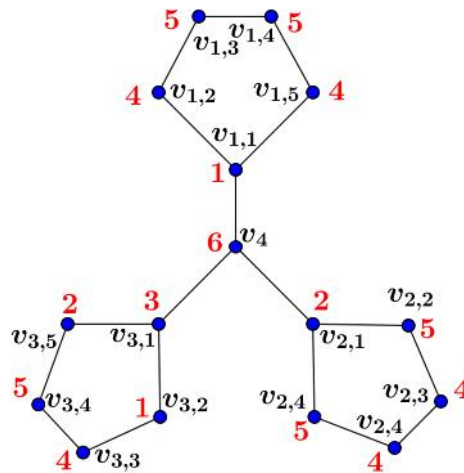
$$\alpha'(v_{m+1}) = m + 1$$

$$\alpha'(v_{i,m+1}) = i, i \in [1, m]$$

Perhatikan titik  $v_{1,4}$  dan titik  $v_{3,4}$ , satu-satunya lintasan yang menghubungkan kedua titik tersebut dengan panjang sama dengan diameter adalah lintasan  $v_{1,4}, v_{1,5}, v_{1,1}, v_4, v_{3,1}, v_{3,5}, v_{3,4}$ . Hal ini mengakibatkan  $v_{1,5}$  atau  $v_{3,5}$  harus diwarnai dengan salah satu warna yang ada pada titik pemotong yaitu warna 2. Misalkan  $v_{1,5}$  diwarnai dengan warna 2, maka tidak terdapat lintasan titik pelangi yang menghubungkan titik  $v_{1,4}$  dengan  $v_{2,4}$ . Misalkan  $v_{3,5}$  diwarnai dengan warna 2 maka tidak terdapat lintasan titik pelangi yang menghubungkan titik  $v_{3,4}$  dengan  $v_{2,4}$ . karena pada graf  $(S_3C_5)$  tidak berlaku pewarnaan 5-titik pelangi, maka diperoleh

$$r(S_3C_5) = 6 = d$$

Selanjutnya akan dibuktikan  $r(S_3C_5) = 6$  dengan mendefinisikan pewarnaan 6-titik pelangi untuk graf  $S_3C_5$  seperti pada Gambar 3.



Gambar 3. Graf  $S_3C_5$

**Kasus 4.**  $r(S_4C_5) = d + 1$

Andaikan terdapat suatu pewarnaan titik pelangi pada  $(S_4C_5)$  dengan  $d = 6$  warna. Tanpa mengurangi keumuman warnai titik pemotong dengan warna berbeda dengan aturan sebagai berikut:

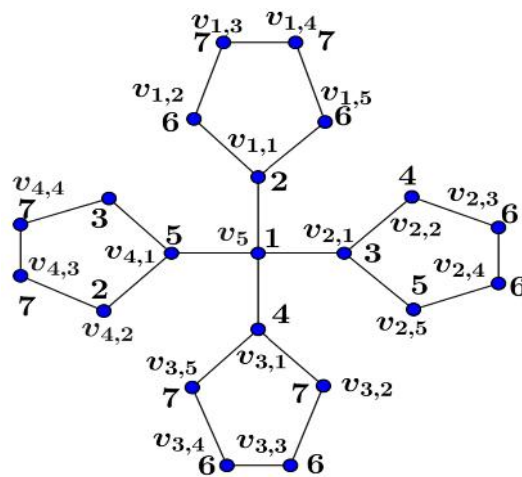
$$\alpha'(v_{m+1}) = m + 1$$

$$\alpha'(v_{i,m+1}) = i, i \in [1, m]$$

Perhatikan titik-titik  $v_{i,2}$  dan  $v_{i,5}$ , pasti ada yang diwarnai dengan warna pada titik pemotong, misal  $v_{i,2}$  yang diwarnai dengan warna yang ada pada titik pemotong dan dimisalkan dengan warna  $k$ , maka titik  $v_{k,2}$  dan titik  $v_{k,5}$  harus diwarnai dengan warna yang sama. Jika warna yang sama tersebut adalah warna pada titik pemotong, maka tidak terdapat pewarnaan titik pelangi untuk lintasan yang melalui titik pemotong tersebut. Sebaliknya jika warna yang sama adalah warna yang tidak digunakan pada titik pemotong, maka perhatikan titik  $v_{i,5}$ , titik tersebut pasti diwarnai dengan warna yang ada pada titik pemotong, misal diwarnai dengan warna  $m - \alpha(v_{i,2})$ . Selanjutnya perhatikan titik-titik  $v_{m,2}$  dan titik  $v_{m,5}$  harus diwarnai dengan warna yang sama, hal ini mengakibatkan tidak terdapat lintasan titik pelangi yang melewati titik potong yang diwarnai dengan warna  $m$ . Karena pada graf  $(S_4C_5)$  tidak berlaku pewarnaan 6-titik pelangi, maka diperoleh

$$r(S_4C_5) = 7 = d + 1$$

Selanjutnya akan dibuktikan  $r(S_4C_5) = 7$  dengan mendefinisikan pewarnaan 7-titik pelangi untuk graf  $S_4C_5$  seperti pada Gambar 4.



Gambar 4. Graf  $S_4C_5$



### 3. Kesimpulan

Berdasarkan hasil dan pembahasan, diperoleh dapat disimpulkan bahwa bilangan terhubung titik pelangi untuk graf lingkaran bintang ( $S_m C_n$ ) adalah sebagai berikut:

Misalkan $m$ dan $n$ adalah dua bilangan positif dengan $3 \leq d \leq n \leq [3,5]$ . $S_m C_n$ adalah graf lingkaran bintang, maka
$r(S_m C_n) = \begin{cases} d - 1 \\ c \\ d + 1 \end{cases}$
$\begin{matrix} u & S_3 C_4 \\ u & m & 3 & n = 3 \\ u & m & 4 & d & n = 4 \\ u & S_3 C_5 \\ u & S_4 C_5 \end{matrix}$

### Daftar Pustaka

- [1] Krivelevich, M. Yuster, R. , The Rainbow Connection of a Graph is (at most) reciprocal to its minimum degree. Graph Theory., 2010..
- [2] X. d. S. Li, Tight Upper Bound of The Rainbow vertex-connection for 2-connected graphs. Discrete Applied Mathematics, 2014, pp. 62-69.



## APLIKASI OPERASI HIMPUNAN DAN MATEMATIKA MORFOLOGI PADA PENGOLAHAN CITRA DIGITAL

V. Y. I. Ilwaru<sup>1</sup>, Y. A. Lesnussa<sup>2</sup>, E. M. Sahetapy<sup>3</sup>, Z. A. Leleury<sup>4</sup>

<sup>1,2,3,4</sup> Jurusan Matematika FMIPA Universitas Pattimura  
Jln. Ir. M. Putuhena, Kampus Unpatti, Poka-Ambon, Kode Pos 97233  
e-mail: <sup>1</sup>yopi\_a\_lesnussa@yahoo.com; <sup>2</sup>zetharthur82@gmail.com

---

### Abstrak

Perkembangan teknologi pengolahan citra dewasa ini berkembang dengan sangat pesat, baik itu jumlah pemakai maupun jenis teknologi yang menggunakan pengolahan citra. Salah satu teknik pengolahan citra yang digunakan adalah deteksi tepi (*edge detection*). Pendeteksian tepi digunakan untuk meningkatkan penampakan garis batas suatu daerah atau obyek di dalam citra. Selain itu deteksi tepi digunakan untuk memperbaiki detail dari citra yang kabur, yang terjadi karena error atau adanya efek dari akuisisi citra. Pada penelitian ini dibahas pengolahan citra digital melibatkan operasi himpunan dan matematika morfologi dengan operasi yang digunakan yaitu Erosi, Dilasi, *Opening*, *Closing*, *Internal Boundary*, *External Boundary*, Morfologi Gradien, dan Filtering. Kemudian menerapkan setiap operasi dimaksud dalam mengolah image sehingga diperoleh hasil citra image dari setiap operasi himpunan dan matematika morfologi.

*Kata Kunci* : Matematika morfologi, operasi himpunan, pengolahan citra digital.

## APPLICATION OF SET OPERATIONS AND MATHEMATIC MORPHOLOGY ON DIGITAL IMAGE PROCESSING

### Abstract

The development of image processing technologies today are growing very rapidly, either the development number of users or the development kind of technology that uses image processing. There are many benefits in image processing. One of the image processing techniques used are edge detection. Edge detection is used to improve the appearance of boundary lines of an area or object in the image. Besides that edge detection is used to improve the detail of a blurred image, which occurs due to an error or the effect of image acquisition. Digital image processing involves the set operation sand the mathematical morphology that has been learned. Operations were used that erosion, dilation, *Opening*, *Closing*, *Internal Boundary*, *External Boundary*, *Morphology Gradient*, and *Filtering*.

*Keywords*: Digital image processing, mathematical morphology, set operations

---

### 1. Pendahuluan

Perkembangan teknologi pengolahan citra dewasa ini berkembang dengan sangat pesat, baik itu perkembangan jumlah pemakai maupun perkembangan jenis teknologi yang menggunakan pengolahan citra, misalnya bidang biomedis, astronomi, penginderaan jauh, dan arkeologi yang umumnya banyak memerlukan teknik peningkatan mutu citra. Aplikasi lain yang kemudian menyusul adalah pengolahan citra digital di bidang robotika, industri, serta arsip citra dan dokumen. Umumnya citra digital berbentuk persegi panjang atau bujur sangkar (pada beberapa sistem pencitraan ada pula yang berbentuk segi enam) yang memiliki lebar dan tinggi tertentu. Ukuran ini biasanya dinyatakan dalam banyaknya titik atau piksel sehingga ukuran citra selalu bernilai bulat. Setiap titik juga memiliki nilai berupa angka digital yang merepresentasikan informasi yang diwakili oleh titik tersebut.

Ada banyak manfaat dalam pengolahan citra antara lain pengolahan citra dalam dunia komunikasi yaitu digunakan untuk memperjelas foto permukaan bumi yang dihasilkan dari satelit cuaca atau memperjelas foto planet-planet yang dihasilkan satelit penyidik, pengolahan citra dalam dunia fotografi digunakan sebagai pengganti kamera filter untuk membuat film hitam putih, efek berkabut, memberikan efek cahaya pada bagian tertentu pada foto, dll. Pengolahan citra dalam dunia kedokteran digunakan untuk memperjelas foto hasil X-ray organ tubuh manusia, hasil CT scan, dan lain-lain. Manfaat lainnya pengolahan citra digital dalam pengenalan pola yaitu pengenalan pola huruf, pola wajah, pola sidik jari, pola iris mata, dan sebagainya, juga pengolahan citra digital digunakan untuk keamanan data dan proteksi hak cipta.

Salah satu teknik pengolahan citra yang digunakan adalah deteksi tepi (*edge detection*). Pendeteksian tepi digunakan untuk meningkatkan penampakan garis batas suatu daerah atau obyek di dalam citra. Selain itu deteksi tepi digunakan untuk memperbaiki detail dari citra yang kabur, yang terjadi karena error atau adanya efek dari akuisisi citra. Citra dari sudut pandang matematis, merupakan fungsi kontinu dari intensitas cahaya pada bidang dua dimensi  $f(x, y)$ , dengan  $x$  dan  $y$  adalah koordinat spasial dan amplitudo  $f$  pada pasangan koordinat  $(x, y)$  yang disebut intensitas atau derajat keabuan citra pada titik tersebut. Jika  $x, y$ , dan  $f$  semuanya berhingga, dan nilainya diskrit, citra tersebut merupakan citra digital. Pengolahan citra digital melibatkan operasi himpunan dan matematika morfologi yang telah dipelajari. Morfologi matematika adalah teori yang sangat penting, dimana operasi tersebut harus didefinisikan oleh himpunan aritmatika. Morfologi matematika terdiri oleh serangkaian morfologi operator aljabar aritmatika. Dalam penelitian ini akan membahas tentang “Penerapan Operasi Himpunan dan Matematika Morfologi Pada Pengolahan Citra Digital”.

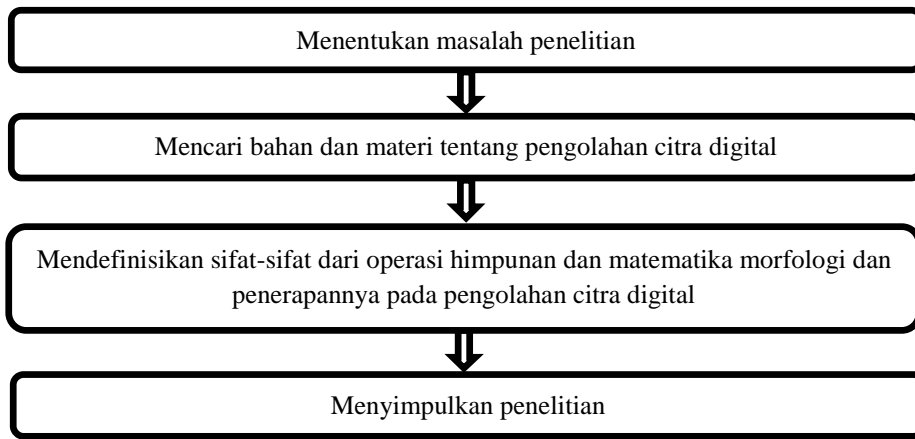
## 2. Tinjauan Pustaka

Citra adalah sinyal diskrit dua dimensi. Secara matematis, sinyal ini dapat direpresentasikan sebagai fungsi dari variabel-variabel dua dimensi. Setiap elemen dari *array* tersebut disebut piksel [1]. Suatu fungsi dua dimensi, dimana harga-harga fungsi tersebut  $f(x, y)$  pada koordinat spasial  $(x, y)$  di bidang  $x$  dan  $y$  mendefinisikan suatu ukuran intensitas cahaya atau kecermelangan titik tersebut [2]. Sedangkan menurut Foley [3], citra adalah *array* dari nilai-nilai dimana sebuah nilai tersebut adalah sekumpulan angka yang mendeskripsikan atribut dari piksel di dalamnya. Ada dua macam citra, yaitu citra diam dan citra bergerak. Citra diam merupakan citra tunggal yang tidak bergerak tetap, biasa disebut citra. Sedangkan citra bergerak merupakan rangkaian citra diam yang ditampilkan secara beruntun sehingga memberikan kesan pada mata sebagai gambar yang bergerak. Setiap citra di dalam rangkaian tersebut disebut *frame*.

Citra digital secara umum didefinisikan sebagai pemrosesan citra dua dimensi dengan komputer. Dalam definisi yang lebih luas, pengolahan citra digital juga mencakup semua data dua dimensi. Citra digital adalah barisan bilangan nyata maupun kompleks yang diwakili oleh bit-bit tertentu. Unit terkecil data digital adalah bit, yaitu angka biner, 0 dan 1. Kumpulan dari data sejumlah 8 bit data adalah sebuah unit data yang disebut *byte*, dengan nilai dari 0 – 225. Selanjutnya, Shih [4] dalam bukunya yang berjudul *Image Processing and Mathematical Morphology* membahas tentang bentuk-bentuk operasi yang melibatkan operasi himpunan dan logika matematika pada pengolahan citra digital meliputi dilasi dan erosi biner, serta operasi *opening* dan *closing* biner.

## 3. Metodologi Penelitian

Tipe penelitian adalah studi pustaka, yaitu mempelajari beberapa literatur yang berhubungan dengan penelitian kemudian mencoba membahas inti permasalahan tersebut dengan menuangkannya secara benar. Bahan atau materi yang digunakan dalam penelitian ini berupa karya ilmiah para matematikawan yang disajikan dalam bentuk pdf, buku cetak dan informasi ilmiah lainnya yang diperoleh dari materi-materi kuliah maupun media elektronik seperti internet.



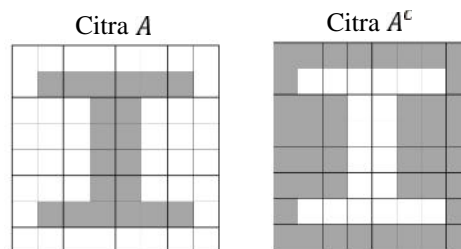
Gambar 1. Prosedur Penelitian

#### 4. Hasil dan Pembahasan

##### 4.1. Operasi Himpunan Pada Citra Biner

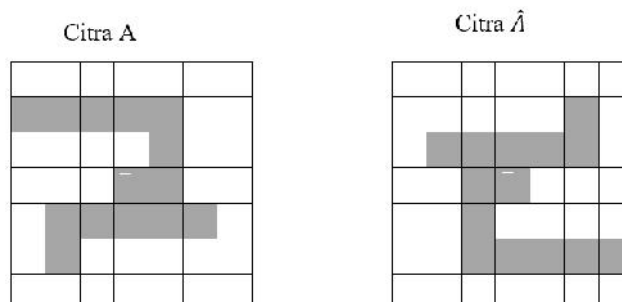
Dalam matematika morfologi, citra biner didefinisikan sebagai himpunan vektor (koordinat) dalam ruang Euclid  $E^N$ : komposisi dari hitam dan putih dalam  $N$ -dimensi.  $E^N$  menunjukkan himpunan semua titik  $p = (x_1, x_2, \dots, x_N)$  di ruang Euclid berdimensi  $N$ . Setiap himpunan  $A$  sesuai dengan citra biner; yaitu dimensi  $N$  dalam warna hitam dan putih, dimana titik  $p$  adalah hitam pada gambar jika dan hanya jika  $p \in A$ ; sebaliknya,  $p$  adalah putih. Sebuah citra biner dalam  $E^2$  adalah siluet, satu himpunan mewakili daerah *foreground* (atau piksel hitam). Sebuah citra biner dalam  $E^3$  adalah solid, satu himpunan mewakili permukaan dan interior benda. Citra biner berkorelasi dengan piksel hitam dan putih untuk sistem koordinat cartesius untuk citra biner.

Misalkan  $A$  menunjukkan suatu himpunan (atau citra biner) di  $E^2$ . Jika himpunan tidak mengandung unsur-unsur, itu disebut himpunan kosong atau himpunan nol, dinotasikan  $\emptyset$ . Misalkan  $a \in A$  menunjukkan elemen  $a = (a_1, a_2)$  di  $A$ .



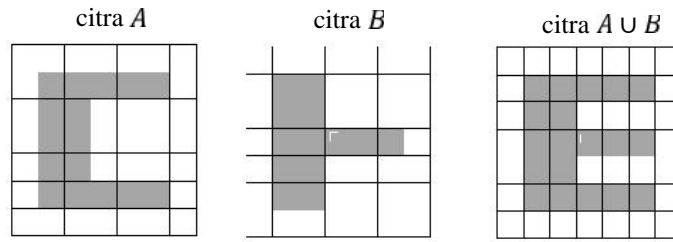
Gambar 2. Komplemen (atau terbalik) dari citra A

$$\bar{A} = (A)^c = \{a | a \notin A\}$$

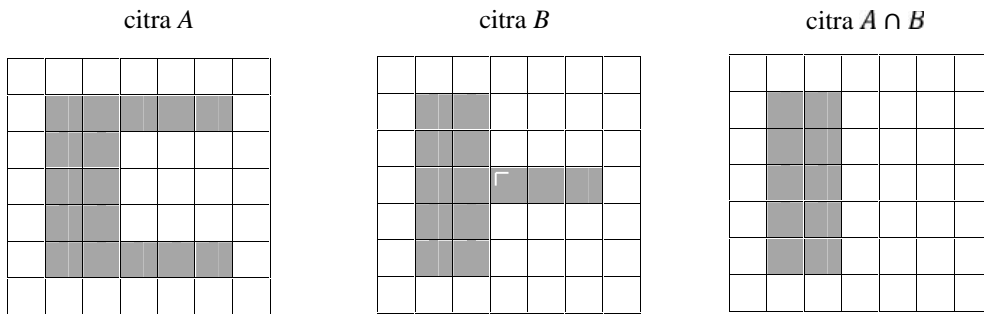


Gambar 3. Refleksi citra A terhadap titik asal, identik dengan rotasi citra A terhadap titik asal sebesar  $180^\circ$

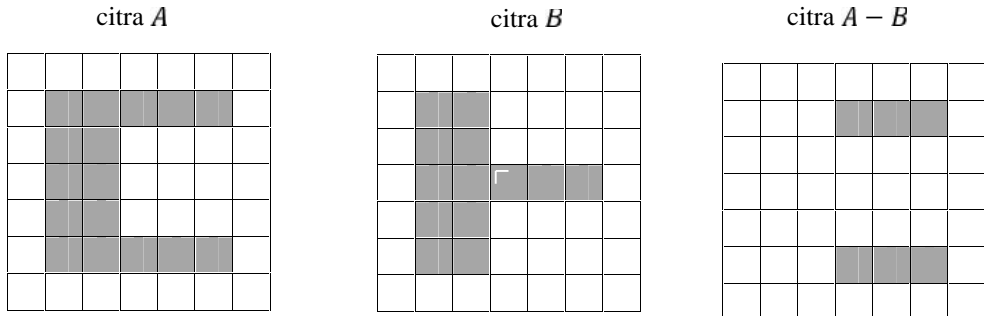
$$\hat{A} = \{b | b = -a, a \in A\}$$



**Gambar 4. Gabungan (union) citra A dan B**  
 $A \cup B = \{c | c \in A \vee c \in B\}$



**Gambar 5. Irisan (intersection) citra A dan B**  
 $A \cap B = \{c | c \in A \wedge c \in B\}$



**Gambar 6. Selisih citra A oleh B**  
 $A - B = A \setminus B = \{c | c \in A \wedge c \notin B\} = A \cap B^c$

#### 4.2. Translasi, Dilasi dan Erosi Biner

**Definisi 1.** Misal  $A \subset E^N$  dan  $b \in E^N$ . Translasi A oleh b, ditulis  $(A)_b$  adalah

$$(A)_b = \{c \in E^N | c = a + b, \quad a \in A\}.$$

**Contoh 1.**  $A = \{(0,2), (1,1), (1,2), (2,0), (2,2), (3,1)\}$ ,  $b = (0,1)$ ,  $(A)_b = \{(0,3), (1,2), (1,3), (2,1), (2,3), (3,2)\}$

	0	1	2	3
0	0	0	1	0
1	0	1	1	0
2	1	0	1	0
3	0	1	0	0

*A*

	0	1
0	0	1

*b*

	0	1	2	3
0	0	0	0	1
1	0	0	1	1
2	0	1	0	1
3	0	0	1	0

$(A)_b$

**Definisi 2.** Misal  $A, B \subset E^N$ . Dilasi biner dari  $A$  oleh  $B$ , ditulis  $A \oplus_b B$  adalah

$$A \oplus_b B = \{c \in E^N \mid c = a + b, a \in A, b \in B\}$$

(Indeks  $b$  menyatakan biner)

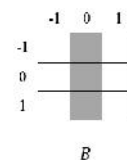
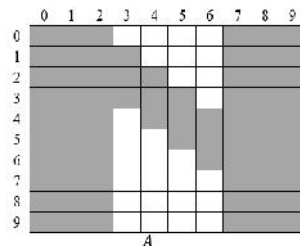
Perhatikan bahwa

$$A \oplus_b B = \bigcup_{b \in B} (A)_b = \bigcup_{a \in A} (B)_a$$

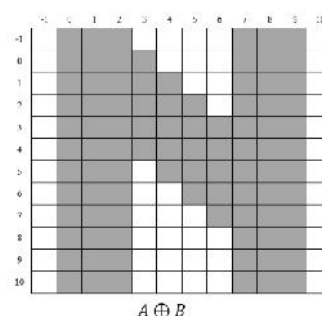
**Contoh 2.**

$A = \{(0,0), (0,1), (0,2), (0,7), (0,8), (0,9), (1,0), (1,1), (1,2), (1,3), (1,7), (1,8), (1,9), (2,0), (2,1), (2,2), (2,3), (2,4), (2,7), (2,8), (2,9), (3,0), (3,1), (3,2), (3,3), (3,4), (3,5), (3,7), (3,8), (3,9), (4,0), (4,1), (4,2), (4,4), (4,5), (4,6), (4,7), (4,8), (4,9), (5,0), (5,1), (5,2), (5,5), (5,6), (5,7), (5,8), (5,9), (6,0), (6,1), (6,2), (6,6), (6,7), (6,8), (6,9), (7,0), (7,1), (7,2), (7,7), (7,8), (7,9), (8,0), (8,1), (8,2), (8,7), (8,8), (8,9), (9,0), (9,1), (9,2), (9,7), (9,8), (9,9)\}$

$B = \{(-1,0), (0,0), (1,0)\}$



$A \oplus_b B = \{(-1,0), (0,0), (1,0), (-1,1), (0,1), (1,1), (-1,2), (0,2), (1,2), (-1,7), (0,7), (1,7), (-1,8), (0,8), (1,8), (-1,9), (0,9), (1,9), (2,0), (2,1), (2,2), (0,3), (1,3), (2,3), (2,8), (2,9), (3,0), (3,1), (3,2), (3,3), (1,4), (2,4), (3,4), (3,7), (3,8), (3,9), (4,0), (4,1), (4,2), (4,3), (4,4), (2,5), (3,5), (4,5), (4,7), (4,8), (4,9), (5,0), (5,1), (5,2), (5,4), (5,5), (3,6), (4,6), (5,6), (5,7), (5,8), (5,9), (6,0), (6,1), (6,2), (6,5), (6,6), (6,7), (6,8), (6,9), (7,0), (7,1), (7,2), (7,6), (7,7), (7,8), (7,9), (8,0), (8,1), (8,2), (8,7), (8,8), (8,9), (9,0), (9,1), (9,2), (9,7), (9,8), (9,9), (10,0), (10,1), (10,2), (10,7), (10,8), (10,9)\}$



**Definisi 3.** Erosi biner dari  $A$  oleh  $B$ , ditulis  $A \ominus_b B$ , adalah

$$A \ominus_b B = \{x \in E^N \mid x + b \in A \quad \forall b \in B\}$$

Definisi di atas ekuivalen dengan  $A \ominus_b B = \bigcap_{b \in B} (A)_{-b}$

Erosi  $A \ominus_b B$  dapat dipandang sebagai tempat kedudukan semua pusat  $c$ , sehingga translasi  $(B)_c$  termuat di dalam himpunan  $A$

$$A \ominus_b B = \{c \in E^N \mid (B)_c \subseteq A\}$$

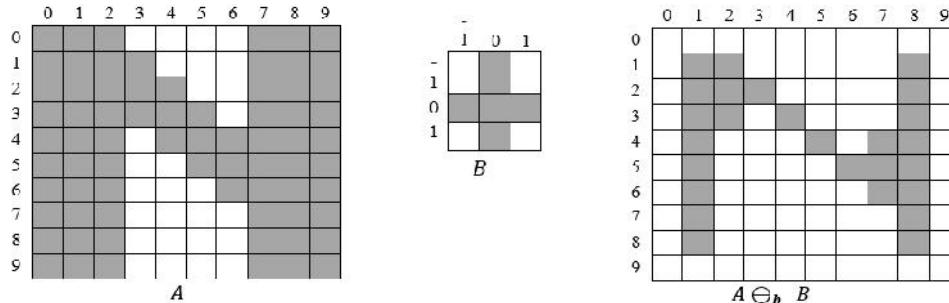
**Contoh 3.**

$A = \{(0,0), (0,1), (0,2), (0,7), (0,8), (0,9), (1,0), (1,1), (1,2), (1,3), (1,7), (1,8), (1,9), (2,0), (2,1), (2,2), (2,4), (2,7), (2,8), (2,9), (3,0), (3,1), (3,2), (3,3), (3,4), (3,5), (3,7), (3,8), (3,9), (4,0), (4,1), (4,2), (4,4), (4,5), (4,6), (4,7), (4,8), (4,9), (5,0), (5,1), (5,2), (5,5), (5,6), (5,7), (5,8), (5,9), (6,0), (6,1), (6,2), (6,6), (6,7), (6,8), (6,9), (7,0), (7,1), (7,2), (7,7), (7,8), (7,9), (8,0), (8,1), (8,2), (8,7), (8,8), (8,9), (9,0), (9,1), (9,2), (9,7), (9,8), (9,9), (10,0), (10,1), (10,2), (10,7), (10,8), (10,9)\}$

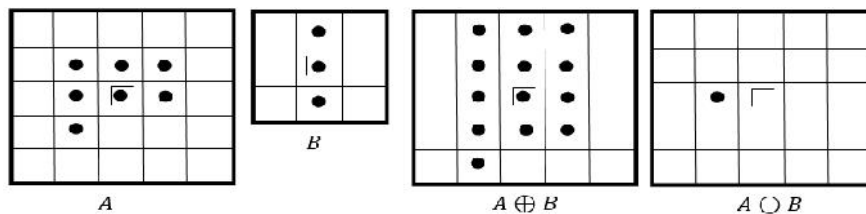
(6,8), (6,9), (7,0), (7,1), (7,2), (7,7), (7,8), (7,9), (8,0), (8,1), (8,2), (8,7), (8,8), (8,9), (9,0), (9,1), (9,2), (9,7), (9,8), (9,9))

$$B = \{(-1,0), (0, -1), (0,0), (0,1), (1,0)\}$$

$$A \ominus_b B = \{(1,1), (1,2), (1,8), (2,1), (2,2), (2,3), (2,8), (3,1), (3,2), (3,4), (3,8), (4,1), (4,5), (4,7), (4,8), (4,9), (5,0), (5,1), (5,6), (5,7), (5,8), (6,1), (6,7), (6,8), (7,1), (7,8), (8,1), (8,8)\}$$



Apabila telah jelas dilasi dan erosi digunakan untuk citra biner A dan elemen pembentuk (*structuring element*) biner B, maka indeks *b* dapat dihilangkan.



Gambar 7. Dilasi Biner dan Erosi Biner

### 4.3. Opening dan Closing

**Definisi 4.** Opening pada suatu citra *A* menggunakan elemen pembentuk *B*, ditulis

*A* ◦ *B*, didefinisikan sebagai

$$A \circ B = (A \ominus B) \oplus B$$

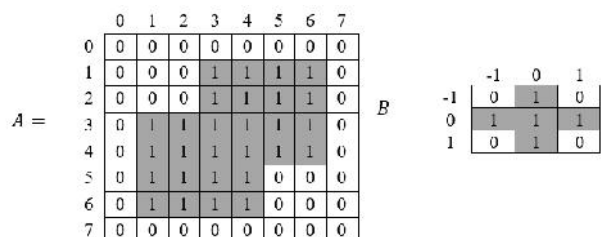
Dengan operasi himpunan, opening dapat dipandang sebagai gabungan semua translasi dari *B* yang termuat di *A*:

$$A \circ B = \bigcup_{B_x \subseteq A} B_x$$

**Contoh 4.**

$A = \{(1,3), (1,4), (1,5), (1,6), (2,3), (2,4), (2,5), (2,6), (3,1), (3,2), (3,3), (3,4), (3,5), (3,6), (4,1), (4,2), (4,3), (4,4), (4,5), (4,6), (5,1), (5,2), (5,3), (5,4), (6,1), (6,2), (6,3), (6,4)\}$

$$B = \{(-1,0), (0, -1), (0,0), (0,1), (1,0)\}$$



Untuk memperoleh opening *A* ◦ *B* maka elemen pembentuk *B* digerakkan di dalam citra *A* kemudian ditandai bagian yang dilalui. Opening dapat juga dipandang sebagai gabungan semua translasi dari *B* yang termuat di *A*

$$A \circ B = (A \ominus B) \oplus B$$



**Contoh 5.**

$A = \{(1,3), (1,4), (1,5), (1,6), (2,3), (2,4), (2,5), (2,6), (3,1), (3,2), (3,3), (3,4), (3,5), (3,6), (4,1), (4,2), (4,3), (4,4), (4,5), (4,6), (5,1), (5,2), (5,3), (5,4), (6,1), (6,2), (6,3), (6,4)\}$

$B = \{(-1,0), (0, -1), (0,0), (0,1), (1,0)\}$

$(A \ominus B) = \{(2,4), (2,5), (3,3), (3,4), (3,5), (4,2), (4,3), (4,4), (5,2), (5,3)\}$

$(A \ominus B) =$

	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0
2	0	0	0	0	1	1	0	0
3	0	0	0	1	1	1	0	0
4	0	0	1	1	1	0	0	0
5	0	0	1	1	0	0	0	0
6	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0

$(A \circ B) = \{(2,4), (2,5), (3,3), (3,4), (3,5), (4,2), (4,3), (4,4), (5,2), (5,3)\} \oplus \{(-1,0), (0, -1), (0,0), (0,1), (1,0)\}$

$A \circ B =$

	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0
2	0	0	0	0	1	1	0	0
3	0	0	0	1	1	1	0	0
4	0	0	1	1	1	0	0	0
5	0	0	1	1	0	0	0	0
6	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0

$\oplus$

	-1	0	1
-1	0	1	0
0	1	1	1
1	0	1	0

$(A \circ B) = \{(1,4), (1,5), (2,3), (2,4), (2,5), (3,2), (3,3), (3,4), (4,2), (4,3), (4,1), (5,1), (5,2), (3,5), (4,4), (5,3), (2,6), (3,6), (4,5), (4,5), (5,4), (6,2), (6,3)\}$

$A \circ B =$

	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	0	0	0	1	1	0	0
2	0	0	0	1	1	1	0	0
3	0	0	1	1	1	1	0	0
4	0	1	1	1	1	1	0	0
5	0	1	1	1	1	0	0	0
6	0	0	1	1	0	0	0	0
7	0	0	0	0	0	0	0	0

**Definisi 5.** *Closing* pada suatu citra  $A$  menggunakan elemen pembentuk  $B$ , ditulis  $A \int B$ , didefinisikan sebagai

$$A \int B = (A \oplus B) \ominus B.$$

Dengan operasi himpunan, *closing* dapat dipandang sebagai komplemen dari gabungan semua translasi  $B$  yang termuat di  $A^c$ :

$$A \circ B = \left( \bigcup_{\hat{B}_y \subseteq A^c} \hat{B}_y \right)^c = \{x: \hat{B}_x^c \cap A = \emptyset\} \hat{B}_x^c$$

**Contoh 6.** Contoh yang digunakan sama dengan contoh di atas. *Closing*  $A \int B$  dapat dipandang sebagai komplemen dari gabungan semua translasi dari  $B$  yang termuat di  $A^c$ .  $A \int B$  adalah bagian yang dilalui ketika titik pusat  $B$  digerakan pada citra  $A$ .

$$A \int B = (A \oplus B) \ominus B$$

$$(A \oplus B) = \{(0,3), (0,4), (0,5), (0,6), (1,3), (1,4), (1,5), (1,6), (2,1), (2,2), (2,3), (2,4), (2,5), (2,6), (3,1), (3,2), (3,3), (3,4), (3,5), (3,6), (4,1), (4,2), (4,3), (4,4), (5,1), (5,2), (5,3), (5,4), (1,2), (3,0), (4,0), (4,5), (5,0), (6,0), (6,1), (6,2), (6,3), (4,6), (6,4), (1,7), (2,7), (3,7), (4,7), (5,5), (6,5), (5,6), (7,1), (7,2), (7,3), (7,4)\}$$

	0	1	2	3	4	5	6	7
0	0	0	0	1	1	1	1	0
1	0	0	1	1	1	1	1	1
2	0	1	1	1	1	1	1	1
3	1	1	1	1	1	1	1	1
4	1	1	1	1	1	1	1	1
5	1	1	1	1	1	1	1	0
6	1	1	1	1	1	1	0	0
7	0	1	1	1	1	0	0	0

$$A \setminus B = \{(0,3), (0,4), (0,5), (0,6), (1,3), (1,4), (1,5), (1,6), (2,1), (2,2), (2,3), (2,4), (2,5), (2,6), (3,1), (3,2), (3,3), (3,4), (3,5), (3,6), (4,1), (4,2), (4,3), (4,4), (5,1), (5,2), (5,3), (5,4), (1,2), (3,0), (4,0), (4,5), (5,0), (6,0), (6,1), (6,2), (6,3), (4,6), (6,4), (1,7), (2,7), (3,7), (4,7), (5,5), (6,5), (5,6), (7,1), (7,2), (7,3), (7,4)\} \ominus \{(-1,0), (0,-1), (0,0), (0,1), (1,0)\}$$

	0	1	2	3	4	5	6	7
0	0	0	0	1	1	1	1	0
1	0	0	1	1	1	1	1	1
2	0	1	1	1	1	1	1	1
3	1	1	1	1	1	1	1	1
4	1	1	1	1	1	1	1	1
5	1	1	1	1	1	1	1	0
6	1	1	1	1	1	1	0	0
7	0	1	1	1	1	0	0	0

	-1	0	1
-1	0	1	0
0	1	1	1
1	0	1	0

$$A \setminus B = \{(1,3), (1,4), (1,5), (1,6), (2,2), (2,3), (2,4), (2,5), (2,6), (3,1), (3,2), (3,3), (3,4), (3,5), (3,6), (4,1), (4,2), (4,3), (4,4), (4,5), (4,6), (5,1), (5,2), (5,3), (5,4), (5,5), (6,1), (6,2), (6,3), (6,4)\}$$

	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	0	0	1	1	1	1	0
2	0	0	1	1	1	1	1	0
3	0	1	1	1	1	1	1	0
4	0	1	1	1	1	1	1	0
5	0	1	1	1	1	1	0	0
6	0	1	1	1	1	0	0	0
7	0	0	0	0	0	0	0	0

#### 4.4. Deteksi Bidang

Salah satu aplikasi operasi erosi adalah untuk deteksi bidang batas suatu objek dalam citra. Jika A suatu citra dan B adalah suatu *structuring element* yang kecil yang terdiri atas titik yang diletakkan secara sistematis terhadap origin, maka dapat didefinisikan bidang batas A dengan beberapa metode sebagai berikut.

- i.  $A - (A \ominus B)$  “*internal boundary*”
- ii.  $(A \oplus B) - A$  “*external boundary*”
- iii.  $(A \oplus B) - (A \ominus B)$  “*morphological gradient*”

**Contoh 7.**

$$A = \{(0,2), (0,3), (1,1), (1,2), (1,3), (1,4), (2,0), (2,1), (2,2), (2,3), (2,4), (2,5), (3,0), (3,1), (3,2), (3,3), (3,4), (3,5), (4,1), (4,2), (4,3), (4,4), (5,2), (5,3)\}$$

$$B = \{(-1, -1), (-1,1), (0,0), (1, -1), (1,1)\}$$

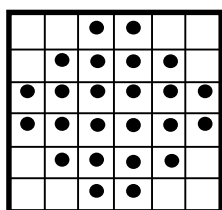
$$A \oplus B = \{(-1,1), (-1,3), (0,2), (1,1), (1,3), (-1,2), (-1,4), (0,3), (1,2), (1,4), (0,0), (2,0), (2,2), (0,1), (2,1), (2,3), (2,4), (0,5), (2,5), (1,-1), (3,-1), (3,1), (1,0), (3,0), (3,2), (1,5), (3,3), (3,4), (2,4), (3,5), (4,-1), (4,1), (4,0), (4,2), (4,3), (4,4), (2,3), (2,5), (4,5), (2,6), (3,5), (4,6), (5,0), (5,2), (5,1), (5,3), (5,4), (5,5), (6,1), (6,3), (6,2), (6,4)\}$$

$$A \ominus B = \{(2,2), (2,3), (3,2), (3,3)\}$$

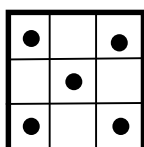
$$(A \oplus B) - A = \{(-1,1), (-1,3), (-1,2), (-1,4), (0,0), (0,1), (0,4), (0,5), (1,-1), (3,-1), (1,0), (1,6), (3,6), (2,-1), (4,-1), (4,0), (4,5), (2,6), (4,6), (5,0), (5,1), (5,4), (5,5), (6,1), (6,3), (6,2), (6,4)\}$$

$$A - (A \ominus B) = \{(0,2), (0,3), (1,1), (1,2), (1,3), (1,4), (2,0), (2,1), (2,4), (2,5), (3,0), (3,1), (3,4), (3,5), (4,1), (4,2), (4,3), (4,4), (5,2), (5,3)\}$$

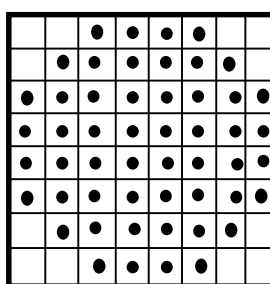
$$(A \oplus B) - (A \ominus B) = \{(-1,1), (-1,3), (0,2), (1,1), (1,3), (-1,2), (-1,4), (0,3), (1,2), (1,4), (0,0), (2,0), (0,1), (2,1), (0,4), (2,4), (0,5), (2,5), (1,-1), (3,-1), (3,1), (1,0), (3,0), (1,5), (1,6), (3,6), (2,-1), (4,-1), (4,1), (4,0), (4,2), (4,3), (4,4), (2,5), (4,5), (2,6), (3,5), (4,6), (5,0), (5,2), (5,1), (5,3), (5,4), (5,5), (6,1), (6,3), (6,2), (6,4)\}$$



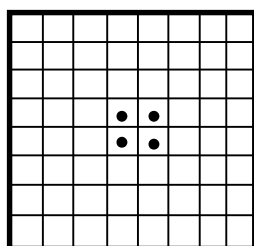
A



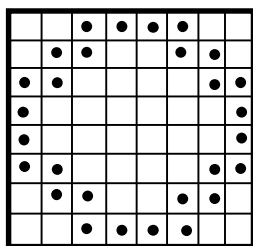
B



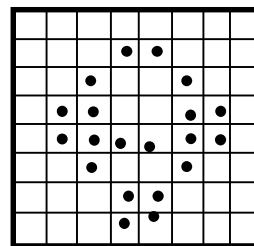
(A ⊕ B)



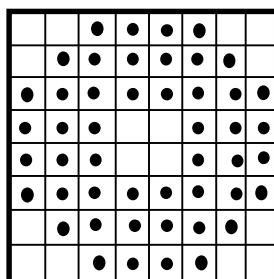
A ⊖ B



(A ⊕ B) - A



A - (A ⊖ B)



(A ⊕ B) - (A ⊖ B)

#### 4.5. Filtering

Misalkan terdapat citra biner  $A$  yang terkena derau implus – piksel yang seharusnya hitam menjadi putih dan sebaliknya. Maka  $(A \ominus B)$  akan menghilangkan piksel hitam tunggal namun akan memperbesar lubang-lubang yang ada. Untuk menghilangkan lubang dapat dilakukan dilasi dua kali secara berurutan.

$$((A \ominus B) \oplus B) \oplus B$$

Dilasi yang pertama akan mengembalikan lubang-lubang ke ukuran semula dan operasi dilasi yang kedua akan menghilangkan lubang-lubang tersebut. Namun hal ini juga akan memperbesar objek pada citra. Untuk mengembalikan objek ke ukuran semula maka dapat dilakukan erosi, sehingga operasi secara keseluruhan dapat dinyatakan sebagai berikut.

$$(((A \ominus B) \oplus B) \oplus B) \ominus B$$

Operasi tersebut juga dapat dinyatakan sebagai berikut

$$(A \circ B) \bar{\cap} B$$

Operasi tersebut disebut *morphological filtering*.

#### Contoh 8.

$A = \{(0,1), (0,3), (0,4), (0,5), (0,6), (1,2), (1,3), (1,4), (1,5), (1,6), (1,7), (1,9), (2,1), (2,2), (2,3), (2,6), (2,7), (2,8), (3,1), (3,2), (3,3), (3,6), (3,7), (3,8), (4,1), (4,2), (4,3), (4,6), (4,7), (4,8), (5,1), (5,2), (5,3), (5,6), (5,7), (5,8), (5,10), (6,1), (6,2), (6,3), (6,6), (6,7), (6,8), (7,1), (7,2), (7,3), (7,6), (7,7), (7,8), (8,0), (8,2), (8,3), (8,4), (8,5), (8,6), (8,7), (9,3), (9,4), (9,5), (9,6), (9,9), (10,1), (10,6), (10,9)\}$

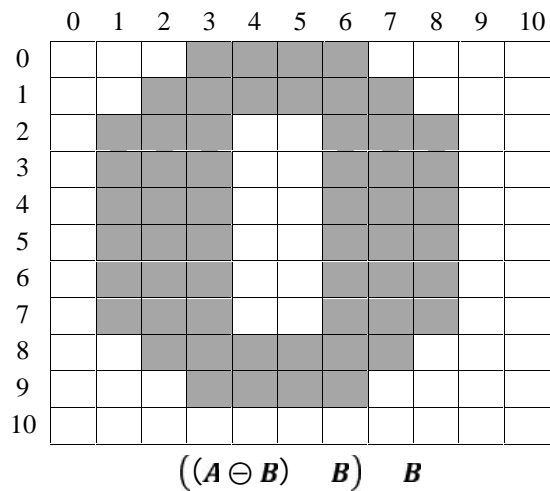
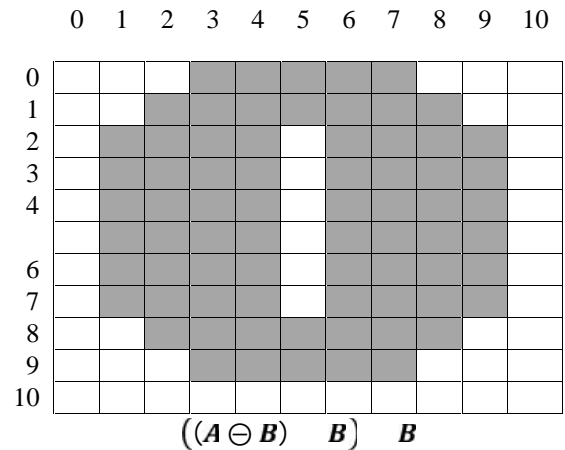
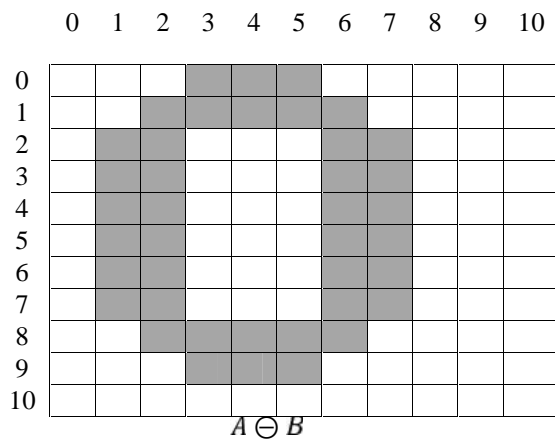
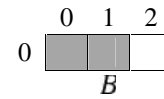
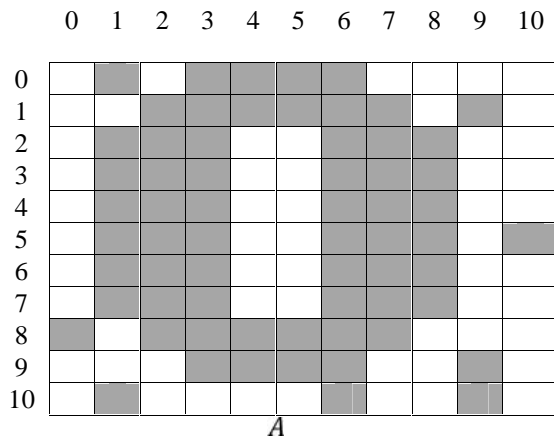
$B = \{(0,0), (0,1)\}$

$A \ominus B = \{(0,3), (0,4), (0,5), (1,2), (1,3), (1,4), (1,5), (1,6), (2,1), (2,2), (2,6), (2,7), (3,1), (3,2), (3,6), (3,7), (4,1), (4,2), (4,6), (4,7), (4,8), (5,1), (5,2), (5,3), (5,6), (5,7), (5,8), (5,10), (6,1), (6,2), (6,3), (6,6), (6,7), (6,8), (7,1), (7,2), (7,6), (7,7), (7,8), (8,0), (8,2), (8,3), (8,4), (8,5), (8,6), (8,7), (9,3), (9,4), (9,5)\}$

$(A \ominus B) \oplus B = \{(0,3), (0,4), (0,5), (0,6), (1,2), (1,3), (1,4), (1,5), (1,6), (1,7), (2,1), (2,2), (2,3), (2,6), (2,7), (2,8), (3,1), (3,2), (3,3), (3,6), (3,7), (3,8), (4,1), (4,2), (4,3), (4,6), (4,7), (4,8), (5,1), (5,2), (5,3), (5,6), (5,7), (5,8), (6,1), (6,2), (6,3), (6,6), (6,7), (6,8), (7,1), (7,2), (7,3), (7,6), (7,7), (7,8), (8,2), (8,3), (8,4), (8,5), (8,6), (8,7), (9,3), (9,4), (9,5), (9,6)\}$

$((A \ominus B) \oplus B) \oplus B = \{(0,3), (0,4), (0,5), (0,6), (0,7), (1,2), (1,3), (1,4), (1,5), (1,6), (1,7), (1,8), (2,1), (2,2), (2,3), (2,4), (2,6), (2,7), (2,8), (2,9), (3,1), (3,2), (3,3), (3,4), (3,6), (3,7), (3,8), (3,9), (4,1), (4,2), (4,3), (4,4), (4,6), (4,7), (4,8), (4,9), (5,1), (5,2), (5,3), (5,4), (5,6), (5,7), (5,8), (5,9), (6,1), (6,2), (6,3), (6,4), (6,6), (6,7), (6,8), (6,9), (7,1), (7,2), (7,3), (7,4), (7,6), (7,7), (7,8), (7,9), (8,2), (8,3), (8,4), (8,5), (8,6), (8,7), (8,8), (9,3), (9,4), (9,5), (9,6), (9,7)\}$

$((((A \ominus B) \oplus B) \oplus B) \ominus B) = \{(0,3), (0,4), (0,5), (0,6), (1,2), (1,3), (1,4), (1,5), (1,6), (1,7), (2,1), (2,2), (2,3), (2,6), (2,7), (2,8), (3,1), (3,2), (3,3), (3,6), (3,7), (3,8), (4,1), (4,2), (4,3), (4,6), (4,7), (4,8), (5,1), (5,2), (5,3), (5,6), (5,7), (5,8), (6,1), (6,2), (6,3), (6,6), (6,7), (6,8), (7,1), (7,2), (7,3), (7,6), (7,7), (7,8), (8,2), (8,3), (8,4), (8,5), (8,6), (8,7), (9,3), (9,4), (9,5), (9,6)\}$



Dari gambar di atas dapat dilihat, pada citra biner A terdapat derau yang seharusnya tidak ada pada citra tersebut. Untuk menghilangkan derau tersebut maka dilakukan erosi dengan menggunakan elemen pembentuk B. Setelah dilakukan erosi ternyata derau yang sebelumnya ada pada citra A hilang namun terjadi pengikisan pada objek. Untuk mengembalikan pada citra aslinya maka dilakukan dilasi dua kali secara berurutan. Namun dengan dilakukan dilasi dua kali terjadi penebalan pada objek, sehingga untuk mengembalikan objek ke ukuran semula maka dilakukan erosi. Setelah dilakukan erosi maka objek kembali ke bentuk aslinya.

#### 4.6. Simulasi Dengan MATLAB

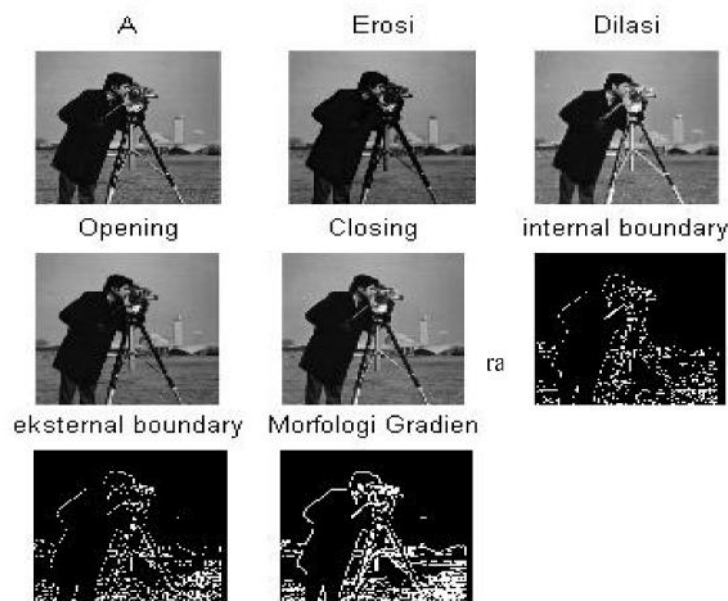
Berikut adalah contoh listing program untuk Erosi, Dilasi, *Opening Closing*, *Exsternal Boundary*, *Internal Boundary*, dan Morfologi Gradien, dengan citra A yaitu “Matematika Morfologi”. Output yang dihasilkan adalah sebagai berikut.



Gambar 8. Output Citra ‘Matematika Morfologi’

Dari output di atas dapat terlihat bahwa hasil erosi merupakan teks yang sama namun lebih tipis. Hasil dilasi lebih mempertebal teks. Hasil opening terlihat menghilangkan objek kecil (dapat dilihat pada huruf ‘e’, ‘r’, dan ‘g’) dan secara umum menghaluskan batas dari teks tanpa mengubah area objek secara signifikan. Efek yang dihasilkan closing sendiri yaitu mengisi lubang kecil pada objek, dan juga secara umum menghaluskan batas teks. Internal Boundary menghasilkan efek dimana citra yang diambil hanyalah batas teks bagian luar dan lebih tipis. Eksternal Boundary mengambil batas teks bagian dalam dan lebih tegas dari internal yang lebih tipis. Morfologi Gradien mengambil batas bagian dalam dan luar teks.

Contoh lainnya untuk Erosi, Dilasi, *Opening Closing*, *Exsternal Boundary*, *Internal Boundary*, dan Morfologi Gradien.



Gambar 9. Output Citra

Dari output di atas dapat terlihat bahwa hasil erosi membuat citra lebih kabur. Hasil dilasi lebih memperjelas citra sehingga citra lebih jelas. Hasil opening terlihat menghilangkan objek dan secara umum

menghaluskan batas citra. *Closing* sendiri mengisi lubang kecil pada objek, dan juga secara umum menghaluskan batas teks. *Internal Boundary* menghasilkan efek dimana citra yang diambil hanyalah bagian luar citra dan lebih tipis. *Eksternal Boundary* mengambil batas citra bagian dalam dan lebih tegas dari internal tegas. *Morfologi Gradien* mengambil batas bagian dalam dan luar dari citra sehingga lebih jelas.

Sedangkan *listing* program untuk *Filtering* citra awalnya menggunakan model warna RGB yang dikonversikan ke model warna citra *grayscale*.



**Gambar 10. Output Citra ‘Eunike’**

Pada output diatas citra asli merupakan citra yang menggunakan model warna RGB yang dikonversikan ke model warna citra *grayscale*. Dari Gambar 10 terlihat bahwa Hasil Filtering 1 yang menggunakan elemen pembentuk matriks ones (3) membuat noda pada citra derau yang sebelumnya sangat banyak terlihat lebih sedikit. Begitu pula untuk Hasil Filtering 2 yang menggunakan elemen pembentuk lain. Namun Hasil Filtering 2 terlihat lebih bersih (nodanya lebih sedikit) dibandingkan Hasil Filtering 1, sehingga dapat diambil kesimpulan bahwa hasil filtering yang baik tergantung pada elemen pembentuk yang digunakan.

## 5. Kesimpulan

Berdasarkan pembahasan pada bagian sebelumnya, maka dapat disimpulkan bahwa:

- 1) Operasi Himpunan dan Matematika Morfologi dapat diterapkan dalam pengolahan citra digital yaitu operasi Erosi, Dilasi, *Opening*, *Closing*, *External Boundary*, *Internal Boundary*, Morfologi Gradien, dan Filtering. Erosi membuat citra menjadi lebih tipis, Dilasi membuat citra menjadi lebih jelas atau mempertebal citra, *Opening* menghilangkan objek-objek kecil juga memperhalus citra, *Closing* menutup celah-celah yang hilang serta memperhalus citra, *Internal Boundary* mengambil batas luar dari sebuah citra, *External Boundary* mengambil batas dalam dari citra, Morfologi Gradien mengambil batas dalam dan luar dari citra, Filtering menghilangkan / mengurangi *noise* yang ada pada citra.
- 2) Elemen pembentuk yang digunakan memiliki pengaruh yang sangat besar pada hasil filtering sebab baik tidaknya hasil filtering tergantung pada elemen pembentuk yang digunakan.

## Daftar Pustaka

1. Fairhurst, M. C. *Visi Komputer Untuk Sistem Robotik*. Jakarta : UI Press, 1994.
2. Wolfray, R. I. *Digital Image Processing*. 2002.
3. Shih, F. Y. *Image Processing and Mathematical Morphology*. s.l. : CRC Press, 2009.
4. Foley, J. D. A. *Computer Graphics: Principles and Practice in C*. Second. s.l. : Addison-Wesley, Addison-Wesley.



## ANALISIS DAN PREDIKSI PENYAKIT JANTUNG KORONER DI KOTA AMBON MENGGUNAKAN JARINGAN SARAF TIRUAN

Dorteus Lodewyik Rahakbauw<sup>1</sup>, Ferry Kondo Lembang<sup>2</sup>,  
Yuniasih M. J. Taihuttu<sup>3</sup>

<sup>1,2</sup> *Staf Dosen Jurusan Matematika FMIPA UNPATTI*

<sup>3</sup> *Staf Dosen Pendidikan Dokter Fakultas Kedokteran UNPATTI*

Jl. Ir. M. Putuhena, Kampus Unpatti, Poka-Ambon  
(Telp: 085776707053 e-mail: lodewyik@gmail.com)

---

### Abstrak

Penyakit jantung koroner disebabkan oleh pembentukan plak yang terdiri atas kolesterol, kalsium dan bahan lain di dalam pembuluh darah yang lama-kelamaan menumpuk di dalam dinding pembuluh darah jantung (arteri koronaria) serta arteri di tempat lain. Di Indonesia 36 juta penduduk menderita penyakit ini dan 80% diantaranya meninggal akibat serangan jantung mendadak, selain itu Indonesia menduduki urutan 5 dunia untuk PJK. Penyakit Jantung Koroner (PJK) merupakan keadaan dimana terjadi penimbunan plak pembuluh darah koroner. Diperlukan diagnosa yang tepat dari pihak medis untuk pemberian pengobatan yang tepat bagi penderita PJK.

Dalam penelitian ini akan diprediksi PJK ini menggunakan metode pengenalan pola dari data catatan rekam medis penderita penyakit jantung koroner yang ada di kota Ambon periode 2014-2015 menggunakan metode Jaringan Syaraf Tiruan (JST) Backpropagation. Berdasarkan data rekam medis penderita penyakit jantung koroner dan orang sehat tersebut dilakukan pelatihan terhadap JST ini, yang nantinya JST ini mampu mengenali polanya. Terdapat 13 faktor risiko yang merupakan node input jaringan dan juga merupakan penyebab timbulnya penyakit jantung koroner, node input tersebut adalah: jenis kelamin, merokok, kelebihan berat badan, pengguna alkohol, pola makan garam yang tinggi, pola makan lemak yang tinggi, olahraga, keturunan, tekanan darah, kolesterol, tekanan gula, tekanan jantung, dan kebiasaan sehari-hari. Dalam penelitian ini data akan langsung diuji dengan 5 pola jaringan: 13-2-1, 13-3-1, 13-4-1, 13-5-1, dan 13-6-1 dimana 13 node input sebagai masukan, unit hidden layer ditentukan secara acak dengan rentang 2 sampai 6 dan 1 unit output dengan fungsi aktivasi sigmoid. Pengujian jaringan saraf tiruan ini dilakukan secara berulang sampai menghasilkan error keluaran yang terkecil pada setiap output data jaringan bersamaan dengan mengubah setiap fungsi aktivasi yang ada pada setiap lapisan pola jaringan saraf tiruan yang dipakai. Setelah proses pelatihan dilakukan, diperoleh konfigurasi jaringan yang terbaik dari 5 tipe yang diuji yaitu tipe dengan 13 input dan 6 hidden layer dengan 1 output (13-6-1) dengan bobot terbaik yang sudah diperoleh dengan kemampuan jaringan mendeteksi seseorang terkena penyakit jantung koroner dengan skala error 0,0009 sampai 0,0727 yang dipandang sangat kecil error yang diperoleh dengan persamaan regresi yang terbentuk adalah  $y = 0,91T + 0,079$  dengan koefisien korelasi sebesar 0,91205.

*Kata Kunci:* Jaringan Saraf Tiruan, *Backpropagation*, Penyakit Jantung Koroner.

## ANALYSIS AND PREDICTION OF CORONARY HEART DISEASE IN AMBON CITY USING NEURAL NETWORK

### Abstract

Coronary heart disease is caused by the formation of plaque consisting of cholesterol, calcium and other substances in the blood vessels that gradually accumulate inside the walls of the coronary arteries (coronary arteries) and arteries elsewhere. In Indonesia 36 million people suffer from this disease and 80% of them died from sudden cardiac arrest, in addition Indonesia is ranked 5th world for CHD. Coronary Heart Disease (CHD) is a condition where there is accumulation of coronary artery plaque. Medical diagnosis is necessary for proper treatment of CHD patients.

In this research will be predicted PJK uses pattern recognition method of medical record records data of patients with coronary heart disease in Ambon city period 2014-2015 using Backpropagation Neural Network (ANN) method. Based on medical record data of patients with coronary heart disease and

healthy people are training on this ANN, which later ANN is able to recognize the pattern. There are 13 risk factors that are the network input nodes and are also the cause of coronary heart disease, the input nodes are: sex, smoking, overweight, alcohol users, high salt diet, high fat diet, exercise, offspring, blood pressure, cholesterol, sugar pressure, heart pressure, and daily habits. In this research the data will be directly tested with 5 network patterns: 13-2-1, 13-3-1, 13-4-1, 13-5-1, and 13-6-1 where 13 input nodes as input, The hidden layer is randomly assigned to the range 2 to 6 and 1 unit of output with the sigmoid activation function. Testing of artificial neural networks is done repeatedly to produce the smallest output error on each network data output simultaneously by changing every activation function that exists on every layer of artificial neural network patterns used. After the training process is done, the best network configuration of the 5 types tested is the type with 13 inputs and 6 hidden layers with 1 output (13-6-1) with the best weights that have been obtained with the ability of the network to detect a person affected by coronary heart disease with Scale error 0.0009 to 0,0727 which is considered very small error obtained with regression equation formed is  $y = 0,91T + 0,079$  with correlation coefficient equal to 0,91205.

*Keywords:* Artificial Neural Network, Backpropagation, Coronary Heart Disease.

## 1. Pendahuluan

Setiap tahunnya lebih dari 36 juta orang meninggal karena Penyakit Tidak Menular (PTM) (63% dari seluruh kematian). Lebih dari 9 juta kematian yang disebabkan oleh penyakit tidak menular terjadi sebelum usia 60 tahun, dan 90% dari kematian “dini” tersebut terjadi di negara berpenghasilan rendah dan menengah. Secara global PTM penyebab kematian nomor satu setiap tahunnya adalah penyakit kardiovaskuler. Penyakit kardiovaskuler adalah penyakit yang disebabkan gangguan fungsi jantung dan pembuluh darah, seperti: Penyakit Jantung Koroner, Penyakit Gagal jantung atau Payah Jantung, Hipertensi dan Stroke (*Kementerian Kesehatan RI*, 2014). Hiperkolesterolemia adalah peninggian kadar kolesterol didalam darah. Tingginya kadar kolesterol darah merupakan problem yang serius karena merupakan salah satu factor resiko utama terjadinya penyakit jantung koroner. Penyakit jantung koroner (PJK) mempunyai angka kematian dan kesakitan yang tinggi. Walaupun penyebab dasar terjadinya penyakit jantung koroner belum diketahui dengan pasti, para ahli telah mengidentifikasi sejumlah faktor yang berhubungan dengan terjadinya penyakit jantung yang disebut sebagai faktor risiko. Faktor risiko penyakit jantung koroner ada yang membaginya dalam faktor risiko primer (independen) dan sekunder [1], yaitu:

- 1) Faktor risiko primer; faktor ini dapat menyebabkan gangguan arteri berupa aterosklerosis tanpa harus dibantu oleh factor lain (independen), termasuk faktor risiko primer, yaitu hiperlidemi, merokok, dan hipertensi.
- 2) Faktor risiko sekunder; Faktor ini baru dapat menimbulkan kelainan arteri bila ditemukan faktor lain secara bersamaan, termasuk faktor risiko sekunder, yaitu diabetes melitus (DM), obesitas, stres, kurang olah raga, alkohol, dan riwayat keluarga.

Mengingat faktor-faktor yang menyebabkan PJK sangat kompleks maka peluang seseorang untuk terkena PJK juga sangat tinggi, sehingga diperlukan suatu solusi untuk menyelesaikan masalah tersebut dalam bentuk pemodelan. Dalam bentuk pemodelan tersebut dapat ditentukan seseorang tergolong penderita PJK sehingga dapat lebih diwaspadai lebih dini. Dari berbagai macam kemungkinan penyebab PJK yang telah disebutkan diatas, faktor umum yang mempengaruhi adalah makanan. Dalam hal ini jaringan Syaraf Tiruan merupakan salah satu bentuk khusus pemrograman komputasi yang dapat digunakan untuk membuat pemodelan penyakit jantung koroner. Algoritma JST yang tepat diterapkan dalam masalah ini adalah backpropagation karena dapat memprediksi seseorang tergolong penderita berdasarkan data rekam medis yang lain. Dengan mengambil 80% data sebagai pelatihan dan 20 % data sebagai data pengujian untuk pengenalan pola penyakit jantung koroner.

## 2. Tinjauan Pustaka

### 2.1. Artificial Intelligence

*Artificial intelligence* atau kecerdasan buatan merupakan sebuah studi tentang bagaimana membuat komputer melakukan hal-hal yang pada saat ini dapat dilakukan lebih baik oleh manusia. Dalam *wikipedia*, 2017 Kecerdasan Buatan atau kecerdasan yang ditambahkan kepada suatu sistem yang bisa diatur dalam konteks ilmiah atau Intelegensi Artifisial didefinisikan sebagai kecerdasan *entitas* ilmiah. Sistem

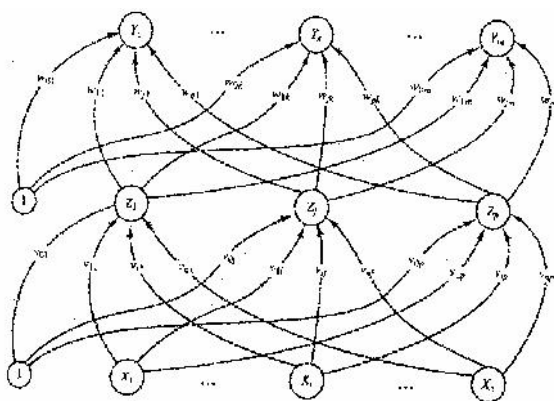
seperti ini umumnya dianggap komputer. Kecerdasan diciptakan dan dimasukkan ke dalam suatu mesin (komputer) agar dapat melakukan pekerjaan seperti yang dapat dilakukan manusia. Beberapa macam bidang yang menggunakan kecerdasan buatan antara lain sistem pakar, permainan komputer (*games*), *logika fuzzy*, jaringan syaraf tiruan dan robotika. Pada awal abad 17, René Descartes mengemukakan bahwa tubuh hewan bukanlah apa-apa melainkan hanya mesin-mesin yang rumit. Blaise Pascal menciptakan mesin penghitung digital mekanis pertama pada 1642. Pada 19, Charles Babbage dan Ada Lovelace bekerja pada mesin penghitung mekanis yang dapat diprogram.

Bertrand Russell dan Alfred North Whitehead menerbitkan *Principia Mathematica*, yang merombak logika formal. Warren McCulloch dan Walter Pitts menerbitkan "Kalkulus Logis Gagasan yang tetap ada dalam Aktivitas " pada 1943 yang meletakkan pondasi untuk jaringan syaraf. Tahun 1950-an adalah periode usaha aktif dalam AI. Program AI pertama yang bekerja ditulis pada 1951 untuk menjalankan mesin Ferranti Mark I di University of Manchester (UK): sebuah program permainan naskah yang ditulis oleh Christopher Strachey dan program permainan catur yang ditulis oleh Dietrich Prinz. John McCarthy membuat istilah "kecerdasan buatan " pada konferensi pertama yang disediakan untuk pokok persoalan ini, pada 1956. Dia juga menemukan bahasa pemrograman Lisp. Alan Turing memperkenalkan "Turing test" sebagai sebuah cara untuk mengoperasionalkan test perilaku cerdas.

Selama tahun 1960-an dan 1970-an, Joel Moses mendemonstrasikan kekuatan pertimbangan simbolis untuk mengintegrasikan masalah di dalam program Maccsima, program berbasis pengetahuan yang sukses pertama kali dalam bidang matematika. Marvin Minsky dan Seymour Papert menerbitkan *Perceptrons*, yang mendemostrasikan batas jaringan syaraf sederhana dan Alain Colmerauer mengembangkan bahasa komputer Prolog. Ted Shortliffe mendemonstrasikan kekuatan sistem berbasis aturan untuk representasi pengetahuan dan inferensi dalam diagnosa dan terapi medis yang kadangkala disebut sebagai sistem pakar pertama. Hans Moravec mengembangkan kendaraan terkendali komputer pertama untuk mengatasi jalan berintang yang kusut secara mandiri. Pada tahun 1980-an, jaringan syaraf digunakan secara meluas dengan algoritma perambatan balik, pertama kali diterangkan oleh Paul John Werbos pada 1974. Pada tahun 1982, para ahli fisika seperti Hopfield menggunakan teknik-teknik statistika untuk menganalisis sifat-sifat penyimpanan dan optimasi pada jaringan syaraf. Para ahli psikologi, David Rumelhart dan Geoff Hinton, melanjutkan penelitian mengenai model jaringan syaraf pada memori. Pada tahun 1985-an sedikitnya empat kelompok riset menemukan kembali algoritma pembelajaran propagansi balik (*Back-Propagation learning*). Algoritma ini berhasil diimplementasikan ke dalam ilmu komputer dan psikologi. Tahun 1990-an ditandai perolehan besar dalam berbagai bidang AI dan demonstrasi berbagai macam aplikasi.

## 2.2. Jaringan Syaraf Tiruan

Jaringan syaraf tiruan merupakan salah satu kecerdasan buatan, yang merupakan representasi tiruan dari otak manusia yang selalu mencoba mensimulasikan proses pembelajaran pada otak manusia tersebut. Model ini merupakan model kotak hitam (*blackboxmodel*), sehingga dalam penerapannya tidak membutuhkan pengetahuan yang kompleks antar berbagai aspek dalam melakukan proses pemodelan faktor-faktor tersebut.



Gambar 1. Jaringan Syaraf Tiruan *Backpropagation* dengan satu *hiddenlayer*

Effendi, dkk pada tahun 2008 memprediksi PJK dengan faktor resiko dengan variabel input yang berjumlah 9 unit sel dan data yang dilatih 40 data rekam medis pasien dan diperoleh Jaringan Syaraf dapat mengenali 80% data baru sesuai dengan target dan 20% data baru tidak sesuai dengan target [1]. Lestari pada tahun 2014 melakukan penelitian dengan menggunakan algoritma klasifikasi *nearest neighbor* (k-nn) untuk mendeteksi penyakit jantung dengan nilai akurasi 70% selain itu dia juga menyatakan bahwa komparasi terhadap algoritma atau metode data mining lainnya dalam mendeteksi penyakit jantung, perlu dilakukan untuk mengetahui algoritma mana yang lebih akurat dan efisien, sehingga dapat ditentukan algoritma yang tepat untuk mendeteksi penyakit jantung [6].

### 2.3. Propagasi Balik (*Backpropagation*)

Backpropagation merupakan algoritma pembelajaran yang terawasi dan biasanya dan biasanya digunakan untuk perceptron banyak lapisan untuk mengubah bobot-bobot yang terhubung dengan neuron-neuron yang ada pada lapisan tersembunyinya. Algoritma ini menggunakan error output untuk mengubah nilai bobot-bobotnya dalam arah mundur (*backward*). Untuk mendapatkan error ini, tahap perambatan maju (*forward propagation*) harus dikerjakan terlebih dahulu. Perambatan maju, neuron-neuron diaktifkan dengan menggunakan fungsi aktivasi yang dapat dideferensiasikan, seperti:

$$a. \text{ Sigmoid, } y = f(x) = \frac{1}{1+e^{-x}}$$

$$b. \text{ Tansig, } y = f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

$$c. \text{ Purelin, } y = f(x) = x$$

dapat digambarkan arsitekturnya pada gambar 1.

Adapun algoritma *training backpropagation* adalah sebagai berikut :

*Step 0* : Inisialisasi nilai bobot dengan nilai acak yang kecil.

*Step 1* : Selama kondisi berhenti masih belum terpenuhi, kerjakan *step 2 - 9*.

*Step 2* : Untuk tiap pasangan pelatihan (s:t), kerjakan *step 3 - 8*.

#### ***Feedforward*** :

*Step 3* : Setiap unit *input* ( $X_i, i=1, \dots, n$ ) menerima sinyal *input*  $x_i$  dan menyebarkan sinyal itu ke seluruh unit pada *layer* berikutnya (*hidden layer*)

*Step 4* : Setiap unit dalam ( $Z_j, j=1, \dots, p$ ) hitung nilai *input* dengan menggunakan nilai bobotnya

$$z\_in_j = \hat{w}_{0j} + \sum_{i=1}^n x_i \hat{w}_{ij}$$

Selanjutnya hitung nilai *output* dengan menggunakan fungsi aktivasi yang dipilih

$$z_j = f(z\_in_j)$$

Hasil fungsi tersebut dikirim ke semua unit pada *layer* berikutnya (unit *output*).

*Step 5* : Untuk tiap unit *output* ( $Y_k, k=1, \dots, m$ ) hitung nilai *input* dengan menggunakan nilai bobotnya

$$y\_in_k = w_{0k} + \sum_{j=1}^p z_j w_{jk}$$

Kemudian hitung nilai *output* dengan menggunakan fungsi aktivasi

$$y_k = f(y\_in_k)$$

#### ***Backpropagation Of Error***

*Step 6* : Setiap unit *output* ( $Y_k, k=1, \dots, m$ ) menerima pola target yang bersesuaian dengan pola *input* dan kemudian hitung error informasi

$$u_k = (t_k - y_k) f'(y_{in_k})$$

Kemudian hitung koreksi nilai bobot yang akan digunakan untuk memperbaharui nilai bobot  $w_{jk}$  :

$$\Delta w_{jk} = \tau u_k z_j$$

Hitung koreksi nilai bias yang akan digunakan untuk memperbaharui nilai  $w_{0k}$  :

$$\Delta w_{0k} = \tau u_k$$

dan nilai  $u_k$  dikirim ke unit pada *layer* sebelumnya.

*Step 7* : Setiap unit dalam ( $Z_j, j=1, \dots, p$ ) hitung delta *input* yang berasal dari unit pada *layer* di atasnya :

$$u_{in_j} = \sum_{k=1}^m u_k w_{jk}$$

Kemudian nilai tersebut dikalikan dengan nilai turunan dari fungsi aktivasi untuk menghitung error informasi :

$$u_j = u_{in_j} f'(z_{in_j})$$

Hitung koreksi nilai bobot yang kemudian digunakan untuk memperbaharui nilai  $\hat{ij}$

$$\Delta \hat{ij} = \tau u_j x_i$$

dan hitung nilai koreksi bias yang kemudian digunakan untuk memperbaharui  $\hat{oj}$

$$\Delta \hat{oj} = \tau u_j$$

#### **Update nilai bobot dan bias :**

*Step 8* : Setiap unit *output* ( $Y_k, k=1, \dots, m$ ) update bias dan bobotnya ( $j=0, \dots, p$ )

$$w_{jk}(\text{new}) = w_{jk}(\text{old}) + \Delta w_{jk}$$

Setiap unit *hidden* ( $Z_j, j=1, \dots, p$ ) update bias dan bobotnya ( $i=0, \dots, n$ )

$$\hat{ij}(\text{new}) = \hat{ij}(\text{old}) + \Delta \hat{ij}$$

*Step 9* : Uji: apakah kondisi berhenti sudah terpenuhi? [2]

## **2. Hasil dan Pembahasan**

### **3.1. Data Input**

Populasi dari penelitian ini adalah pasien yang sedang menjalani perawatan di RSUD Dr. M. Haulussy Ambon sepanjang bulan Juni 2015 yang berjumlah 20 orang, dimana responden merupakan penderita (*case*) Jantung koroner. Data untuk kelompok *case* merupakan data primer maupun sekunder, diperoleh menggunakan media komunikasi berupa kuesioner dan data rekam medik pasien yang terdiagnosis menderita Jantung koroner yang berumur 15 tahun keatas dan sedang menjalani perawatan di IRA (Instalasi Rawat Inap) pada saat survei dilakukan. variabel data yang diperoleh bersesuaian dengan tabel berikut

**Tabel 1. Daftar kriteria dalam deteksi penyakit jantung**

Kriteria	Keterangan	Bobot
Jenis Kelamin (Laki-Laki; Perempuan)	Umur < 30	0.1
	Umur > 30 <50	0.3
	Umur > 50 <70	0.7
	Umur >70	0.8
Merokok	Tidak	0.1
	Masa Lalu	0.3
	Ya	0.6
Kelebihan berat badan	Ya	0.8
	Tidak	0.1
Pengguna alkohol	Tidak	0.1
	Masa Lalu	0.3
	Ya	0.6
Pola makan garam yang tinggi	Ya	0.9
	Tidak	0.1
Pola makan lemak yang tinggi	Ya	0.9
	Tidak	0.1
Olah raga	Ya	0.1
	Tidak	0.6
Keturunan	Ya	0.7
	Tidak	0.1
Tekanan Darah	Normal (130/89)	0.1
	Rendah (< 119/79)	0.8
	Tinggi (> 200/160)	0.9
Kholestrol	Sangat Tinggi > 200	0.9
	Tinggi 160 - 200	0.8
	Normal < 160	0.1
Tekanan Gula	Tinggi >120 dan <400	0.5
	Normal 90 - 120	0.1
	Rendah <90	0.4
Tekanan Jantung	Rendah <60 bpm	0.9
	Normal 60 bpm - 100 bpm	0.1
	Tinggi > 100 bpm	0.9
Kebiasaan sehari- hari	Malas	0.7
	Tidak	0.1

Sumber : *European Journal of Scientific Research* ISSN 1450-216X  
Vol. 31 No. 4 (2009), pp. 642-656

Pada permasalahan ini arsitektur Jaringan Syaraf Tiruan yang digunakan adalah Jaringan Syaraf Tiruan dengan banyak lapisan (*multi layer*) dengan algoritma *Backpropagation*, yang terdiri dari:

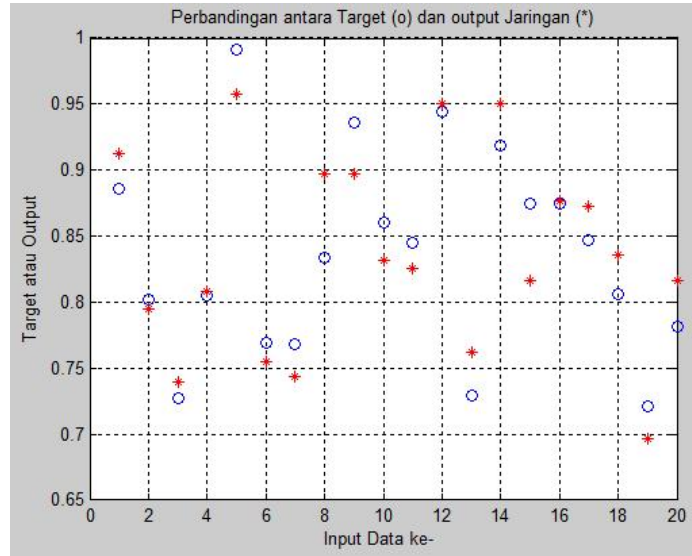
- Lapisan masukan (*input*) dengan 13 simpul ( $x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12}, x_{13}$ ).
- Lapisan tersembunyi (*Hidden*) dengan jumlah simpul ditentukan oleh pengguna ( $Z_1, Z_2$ ).
- Lapisan keluaran (*Output*) dengan 1 simpul ( $Y$ ).

Jaringan syaraf yang akan dibangun adalah algoritma propagasi balik (*Backpropagation*) dengan fungsi aktivasi *Sigmoid*. Fungsi aktivasi dalam jaringan syaraf tiruan dipakai untuk menentukan keluaran suatu *Neuron*.

### 3.2. Hasil Pengujian Sistem

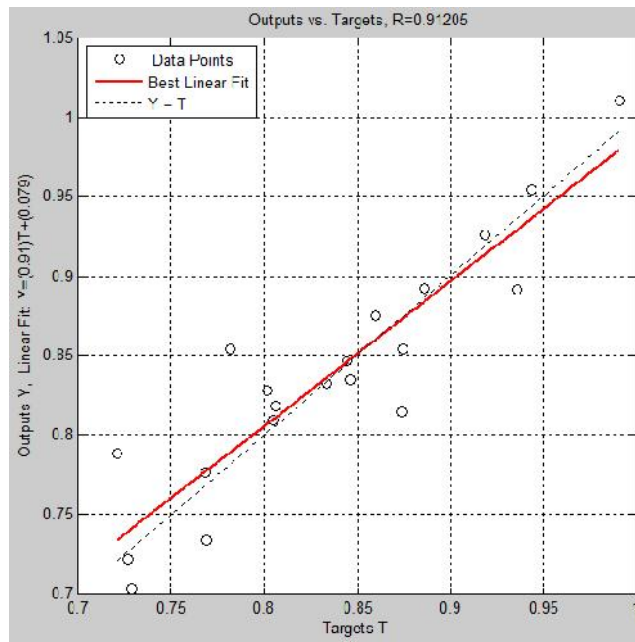
Data diuji kedalam pola jaringan 13-2-1, 13-3-1, 13-4-1, 13-5-1, dan 13-6-1 dimana 13 unit input masukkan, unit *hidden layer* yang ditentukan secara acak 2 sampai 6 dan 1 unit output dengan fungsi aktivasi sigmoid. Berikut ini merupakan Plot Target vs Output pada tipe 13-6-1 dimana terlihat bahwa data memiliki error yang relatif kecil antara 0,0009 pada data ke-8 sampai 0,0727 pada data terakhir yaitu data ke-

20, diperoleh persamaan regresi pada plot:  $Y = 0,91T + 0,079$  dengan nilai koefisien korelasi ( $r$ ) sebesar 0,91205, hasil ini merupakan hasil terbaik dari ke lima arsitektur yang diuji bahkan kecepatan perhitungan pun lebih cepat dengan 55 iterasi.



**Gambar. 2. Plot Target vs Output pada tipe 13-6-1**

Berikut ini adalah salah satu hasil plot jaringan dengan tipe 13-6-1 yang merupakan tipe arsitektur terbaik dalam mendeteksi pengidap penyakit jantung (Gambar 2).



**Gambar 2. Plot regresi hasil pengujian jaringan 13-6-1**

Pengujian juga dilakukan terhadap 4 tipe yang lainnya yang menghasilkan keluaran tidak lebih baik dari tipe 13-6-1. Hasil lengkap pengujian 5 tipe arsitektur diperoleh rata-rata error minimum ada pada tipe 13-6-1, hasil lengkapnya pada Tabel 2 berikut.

**Tabel 2. Pengujian Sistem**

Data ke	Tipe arsitektur				
	13-2-1	13-3-1	13-4-1	13-5-1	13-6-1
1	0.0017	0.0352	0.0090	0.0318	0.0060
2	0.0847	0.0124	0.0333	0.0598	0.0264
3	0.0000	0.0771	0.0361	0.0020	0.0053
4	0.0055	0.0003	0.0021	0.0054	0.0038
5	0.1049	0.0075	0.0162	0.0047	0.0187
6	0.0002	0.0337	0.0042	0.0018	0.0350
7	0.0019	0.0346	0.0038	0.0246	0.0076
8	0.0029	0.0138	0.0032	0.0353	0.0009
9	0.0527	0.0196	0.0433	0.0426	0.0445
10	0.0265	0.0121	0.0109	0.0101	0.0153
11	0.0417	0.0355	0.0021	0.0048	0.0020
12	0.0649	0.1404	0.0037	0.0187	0.0103
13	0.0036	0.0766	0.0094	0.0004	0.0256
14	0.0325	0.0196	0.0044	0.0447	0.0072
15	0.0106	0.0624	0.0885	0.0451	0.0203
16	0.0176	0.0710	0.0087	0.0226	0.0593
17	0.0058	0.0350	0.0004	0.0400	0.0120
18	0.0004	0.0029	0.0225	0.0346	0.0121
19	0.0034	0.0810	0.0812	0.0219	0.0668
20	0.1036	0.0306	0.0045	0.0479	0.0727

Dari kelima tipe arsitektur yang digunakan penulis dalam proses pengujian sistem Jaringan Syaraf Tiruan dengan algoritma *Backpropagation* dengan menggunakan *matlab* dengan sampel data sebanyak 20 orang data rekam medis, maka penulis menyimpulkan lebih baik menggunakan model 13-6-1. Dari hasil yang diperoleh model ini lebih selektif dalam mengolah data serta *epoch* (pengulangan) lebih sedikit digunakan untuk memperoleh *error* minimum.

**Tabel 3. Tabel Perbandingan Pengujian Jaringan**

Tipe arsitektur	b	r	m	iterasi
13-2-1	0.3036	0.7986	0.6369	316
13-3-1	0.3972	0.7066	0.5244	187
13-4-1	0.2031	0.9081	0.7586	128
13-5-1	0.1392	0.9103	0.8334	88
13-6-1	0.9077	0.9120	0.9077	55

Ket: persamaan regresi:  $Y=bT+m$  dimana T adalah target, b titik perpotongan dengan sumbu y, r merupakan koefisien korelasi sedangkan m adalah gradien garis hasil regresi linier

### 3. Kesimpulan

Diperoleh konfigurasi jaringan yang terbaik adalah tipe dengan 13 input dan 6 hidden layer dengan 1 output (13-6-1) dengan bobot terbaik yang sudah diperoleh dengan kemampuan jaringan mendeteksi seseorang terkena penyakit jantung koroner dengan skala error 0,0009 sampai 0,0727 yang dipandang sangat kecil error yang diperoleh dengan persamaan regresi yang terbentuk adalah  $y= 0,91T+0,079$  dengan koefisien korelasi sebesar 0,91205. Sedangkan tingkat ketepatan tiap tipe arsitektur jaringan tergambar pada tabel 2. Berdasarkan hal tersebut Maka dapat diambil kesimpulan bahwa penggunaan JST tipe 13-6-1 dalam memprediksi penyakit jantung koroner terbukti memiliki nilai akurasi cukup tinggi dengan rentang error minimum sehingga dapat digunakan dalam prediksi PJK.



## Daftar Pustaka

- [1] Effendy N., Subagja, Faisal A., Prediksi penyakit jantung koroner (PJK) berdasarkan faktor risiko menggunakan jaringan syaraf tiruan backpropagation, Seminar Nasional Aplikasi Teknologi Informasi 2008 (SNATI 2008), Yogyakarta, 21 Juni 2008.
- [2] Fauset, L., Fundamentals of Neural Networks, New Jersey: Prentice Hall, 1994.
- [3] Kemenkes RI. 2014. *Info Datin: Situasi Kesehatan Jantung*.
- [4] Kusumadewi, S., 2004, "Membangun jaringan Syaraf Tiruan Menggunakan Matlab & Excel Link". Yogyakarta: Graha Ilmu.
- [5] Larandipa L. A, Pontia F.T, Triyanto D., 2013, Aplikasi Jaringan Syaraf Tiruan Untuk Diagnosa Penyakit Jantung Koroner (Pjk) Dengan Metode Backpropagation, Jurnal Coding Sistem Komputer Universitas Tanjungpura Volume 01 No. 1 (2013), hal 11 – 19 ISSN : 2338-493x
- [6] Lestari M., 2014, Penerapan algoritma klasifikasi nearest neighbor (k-nn) untuk mendeteksi penyakit jantung, jurnal Faktor Exacta 7(4): 366-371, 2014
- [7] Wahyuni E. G., Pridjodiprodjo, W., 2013, Prototype Sistem Pakar untuk Mendeteksi Tingkat Resiko Penyakit Jantung Koroner dengan metode *Dempster-Shafer* (Studi Kasus: RS. PKU Muhammadiyah Yogyakarta), IJCCS, Vol.7, No.2, July 2013, pp. 133~144,



## MODEL REGULASI PADA PROSES BIODEGRADASI *POLYETHYLENE TEREPHTHALATE (PET)*

**Taufan Talib**

Program Studi Matematika, Universitas Halmahera  
Jl. Wari Raya, Tobelo, Indonesia  
e-mail:taufan.talib@gmail.com

---

### Abstrak

Tim *International Genetically Engineered Machine (IGEM)* Indonesia merancang sebuah sistem biodegradasi limbah *Polyethylene Terephthalate (PET)* menggunakan *LC-Cutinase* yang diproduksi oleh bakteri *Escherchia Coli (E-Coli)* dalam sistem *whole cell biocatalys*. Paper ini menghasilkan model matematika kontinu menggunakan Persamaan Diferensial Biasa (PDB) dan Fungsi Hill dari modul regulasi. Kemudian model tersebut diuji dengan melakukan analisis titik tetap dan nilai eigen. Hasil menunjukkan model sudah dapat merepresentasikan fenomena biologi yang terjadi pada modul regulasi.

*Kata Kunci:* Biodegradasi, modul regulasi, PET.

## REGULATION MODEL ON BIODEGRAGATION PROCESS OF *POLYETHYLENE TEREPHTHALATE (PET)*

### Abstract

*International Genetically Engineered Machine (IGEM)* Team – Indonesia designed a biodegradation system of *Polyethylene Terephthalate (PET)* residue using *LC-Cutinase* which were produced by *Escherchia Coli (E-Coli)* in *whole cell biocatalyst* system. The result presented in this paper is a continue mathematical model using Ordinary Differential Equation and Hill function from regulation model. The model then tested using fixed point analysis and eigen value. The result shows that model is able to represent biological phenomen which happened on regulation model.

*Keywords:* Biodegradation, PET, regulation modul.

---

### 1. Pendahuluan

Selama lebih dari 50 tahun produksi dan konsumsi plastik di dunia terus meningkat, sehingga mengakibatkan pencemaran lingkungan juga ikut meningkat, serta menciptakan risiko serius bagi alam dan manusia. Pada tahun 2013, 290 juta ton plastik diproduksi di dunia dan digunakan sebagai wadah atau kemasan dalam kehidupan sehari-hari. Sifatnya yang ringan dan awet menjadi bahan serbaguna yang dipakai untuk membuat berbagai peralatan dalam kehidupan kita. Namun demikian, plastik adalah salah satu bahan yang tidak ramah lingkungan karena membutuhkan waktu yang sangat lama untuk terurai [1].

Jenis plastik yang dipakai sebagai bahan baku botol plastik adalah jenis *Polyethylene Terephthalate (PET)*. Meskipun tingkat daur ulang PET telah meningkat dari 7% menjadi 30% dalam beberapa tahun terakhir, namun sebagian besar botol plastik masih dibuang di tempat pembuangan sampah dan terus mencemari lingkungan [2].

Tim IGEM ITB 2014 merancang sebuah sistem degradasi limbah PET menggunakan bakteri *Escherchia Coli (E-Coli)*. Bakteri akan menghasilkan enzim *LC-Cutinase* pada modul regulasi. Model matematika yang dibangun dapat merepresentasikan proses regulasi, sehingga dinamika yang terjadi mudah dipahami untuk mendesain sistem yang lebih efektif dan efisien.

## 2. Modul Regulasi pada Sistem Biodegradasi PET

Pada modul regulasi, *LC-Cutinase* diproduksi terus menerus hingga bakteri mengalami stres. Saat kondisi stres, akan terjadi salah pelipatan protein sehingga menghasilkan *Inclusion Body*. *Inclusion Body* tidak dapat mendegradasi PET dan akan menumpuk di dalam sel, sehingga modul regulasi dirancang untuk memperbaiki atau memulihkan inefisiensi sistem biodegradasi. Ketika *Inclusion Body* muncul, promotor *inducible* dari *Inclusion body (PibpAB)* akan mengaktifasi produksi mRNA TetR dan menghasilkan TetR. Kemudian TetR akan menghentikan ekspresi *LC-Cutinase* dengan cara menekan produksi mRNA.

Produksi mRNA yang ditekan akan berkurang, sehingga mengakibatkan banyaknya konsentrasi mRNA yang ditranskripsi ikut berkurang. Hal ini berujung pada produksi *LC-Cutinase* dan penumpukan *Inclusion Body* turut berkurang. Proses ini akan terus berlangsung selama bakteri masih dalam kondisi stres. Setelah pulih, bakteri dapat kembali memproduksi *LC-Cutinase* [3].

## 3. Modul Regulasi

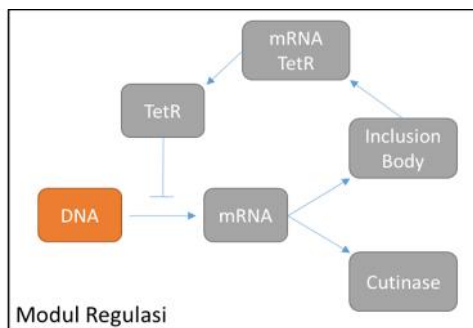
### 3.1. Asumsi

Asumsi yang digunakan dalam model regulasi, yaitu:

- i. Pengamatan untuk satu unit sel
- ii. Tidak ada perubahan suhu, pH, dan tekanan selama proses berlangsung
- iii. Tidak melibatkan faktor pembelahan sel
- iv. Bakteri sudah mulai mengekspresikan *LC-Cutinase* dan *Inclusion Body* (dalam jumlah yang sangat kecil) sebelum direaksikan dengan PET

### 3.2. Model Matematika

Berikut interaksi antar komponen biokimia dalam mendesain bakteri tersebut pada model regulasi



Gambar 1. Proses Modul Regulasi

Gambar 1 menjelaskan bahwa mRNA dihasilkan dari proses transkripsi DNA, setelah ditranslasikan menjadi enzim *LC-Cutinase* atau *Inclusion Body*. Misalkan sebanyak  $x$  mRNA ditranslasi menjadi *LC-Cutinase*, sedangkan sisanya  $(1 - x)$  mRNA ditranslasi menjadi *Inclusion Body*, dengan  $0 < x < 1$ . Model matematika untuk laju konsentrasi *LC-Cutinase* dan *Inclusion Body* sebagai berikut.

$$\frac{dC}{dt} = r_c x M - \chi_c C \tag{1}$$

$$\frac{dI}{dt} = r_i (1 - x) M - \chi_i I \tag{2}$$

Dalam hal ini, semakin banyak *Inclusion Body* akan menyebabkan semakin sedikit *LC-Cutinase*. Dengan demikian dipilih

$$x = \frac{1}{1 + I}$$

Substitusikan  $x$  ke persamaan (1)-(2), sehingga diperoleh model laju konsentrasi *LC-Cutinase* dan *Inclusion Body*.

$$\frac{dC}{dt} = \frac{r_c M}{1+I} - \chi_c C \quad (3)$$

$$\frac{dI}{dt} = \frac{r_i MI}{1+I} - \chi_i I \quad (4)$$

Berikutnya produksi mRNA TetR diaktivasi oleh *Inclusion Body* yang selanjutnya ditranslasikan menjadi TetR. Enzim TetR ini akan menekan atau menghentikan proses transkripsi DNA. Sel akan berhenti memproduksi enzim *LC-Cutinase* maupun *Inclusion Body*, sehingga konsentrasi *Inclusion Body* akan berkurang demikian pula dengan konsentrasi mRNA TetR dan TetR. Akibatnya proses transkripsi MRNA yang menghasilkan LC-Cutinase akan meningkat hingga pada suatu waktu tertentu mulai muncul *Inclusion Body* dan siklus diatas berulang kembali.

Model matematika untuk laju konsentrasi mRNA dan mRNA TetR mengikuti persamaan Hill. Sementara itu konsentrasi TetR mengikuti model penambahan (produksi) dan pengurangan (degradasi), sehingga dapat dibuat model matematika untuk laju konsentrasi mRNA, mRNA TetR, dan TetR.

$$\frac{dM}{dt} = \frac{r_m}{1+\left(\frac{R}{K}\right)^n} - \chi_m M \quad (5)$$

$$\frac{dMr}{dt} = \frac{r_{mr} I}{K_{mr}^n + I^n} - \chi_{mr} Mr \quad (6)$$

$$\frac{dR}{dt} = r_r Mr - \chi_r R \quad (7)$$

Selanjutnya variabel dan parameter yang digunakan dalam persamaan diatas, dengan nilai parameter yang dipilih untuk memperoleh hasil simulasi yang beresilasi.

**Tabel 1. Variabel Model Regulasi**

Variabel	Deskripsi	Satuan
$M$	Konsentrasi mRNA	$\sim M$
$C$	Konsentrasi <i>LC-Cutinase</i>	$\sim M$
$I$	Konsentrasi <i>Inclusion Body</i>	$\sim M$
$Mr$	Konsentrasi mRNA TetR	$\sim M$
$R$	Konsentrasi mRNA TetR	$\sim M$

**Tabel 2. Parameter Model Regulasi**

Parameter	Deskripsi	Nilai
$K$	Koefisien aktivasi untuk mRNA	1
$K_{mr}$	Koefisien aktivasi untuk mRNA TetR	0.7
$n$	Koefisien nilai Hill	3
$\Gamma_m$	Laju produksi mRNA	0.08
$\Gamma_c$	Laju produksi <i>LC-Cutinase</i>	0.001
$\Gamma_i$	Laju produksi <i>Inclusion Body</i>	0.005
$\Gamma_{mr}$	Laju produksi mRNA TetR	0.073
$\Gamma_r$	Laju produksi TetR	0.1
$\chi_m$	Laju degradasi mRNA	0.005
$\chi_c$	Laju degradasi <i>LC-Cutinase</i>	0.5
$\chi_i$	Laju degradasi <i>Inclusion Body</i>	0.009
$\chi_{mr}$	Laju degradasi mRNA TetR	0.03
$\chi_r$	Laju degradasi mRNA TetR	0.03

#### 4. Analisis dan Simulasi Model

##### 4.1. Analisis

Analisis kestabilan model regulasi dilakukan untuk mRNA, *Inclusion Body*, mRNA TetR dan TetR. Sedangkan *LC-Cutinase* tidak dilibatkan dalam analisis ini, dikarenakan tidak mempengaruhi hasil analisis tersebut. Berikutnya agar sistem biodegradasi yang dibangun lebih efisien dalam mencegah munculnya *Inclusion Body*, maka sistem tersebut harus memperlihatkan perilaku osilasi. Oleh karena itu dengan pemodelan ingin melihat, apakah perilaku tersebut dapat muncul atau tidak?

##### i. Titik Tetap

Dari model matematika yang telah dibentuk pada di persamaan (4)-(7) didapat dua buah titik tetap berikut.

$$\left( I = 0, M = \frac{\Gamma_m}{\chi_m}, Mr = 0, R = 0 \right) \quad (8)$$

$$\left( I = \frac{\Gamma_i M - \chi_i}{\chi_i}, M = \frac{\Gamma_m}{\left(1 + \left(\frac{R}{K}\right)^n\right) \chi_m}, Mr = \frac{\Gamma_m I}{\left(K_{mr}^n + I^n\right) \chi_{mr}}, R = \frac{\Gamma_r Mr}{\chi_r} \right) \quad (9)$$

Analisis titik tetap pertama (8) mudah dilakukan, tetapi tidak demikian dengan titik tetap yang kedua (9). Oleh karena itu agar lebih mudah menganalisis perilaku titik tetap yang kedua, nilai parameter pada Tabel 2 disubstitusikan pada model. Selanjutnya titik tetap yang dianalisis lebih lanjut adalah titik tetap yang **eksis**, yaitu berupa bilangan *real* dan non-negatif. Titik tetap yang diperoleh sebagai berikut.

- $\{I = 0, M = 16, Mr = 0, R = 0\}$
- $\{I = 0.081849, M = 1.947328, Mr = 0.579733, R = 1.932442\}$
- $\{I = 2.659325, M = 6.586785, Mr = 0.3379165, R = 1.126388\}$
- $\{I = 7.868997, M = 15.96419, Mr = 0.039270, R = 0.130898\}$

## ii. Analisis Kestabilan

Uji kestabilan dari titik tetap yang eksis, dilakukan dengan menggunakan matriks *Jacobian* berikut

$$J = \begin{vmatrix} -x_m & 0 & 0 & -\frac{r_m \left(\frac{R}{K}\right)^n}{\left(1 + \left(\frac{R}{K}\right)^n\right)^2 R} \\ \frac{r_i I}{1+I} & \frac{r_i M}{1+I} - \frac{r_i IM}{(1+I)^2} - x_i & 0 & 0 \\ 0 & 0 & -x_{mr} & 0 \\ 0 & 0 & r_r & -x_r \end{vmatrix} \quad (10)$$

Untuk mendapatkan nilai eigen pada titik tetap pertama (8), substitusikan titik tetap pertama (8) ke matriks *J* (10), sehingga diperoleh:

$$\lambda_1 = -x_m, \lambda_2 = \frac{r_i r_m - x_i x_m}{x_m}, \lambda_3 = -x_{mr}, \lambda_4 = -x_r$$

Nilai eigen pertama, ketiga dan keempat sudah bernilai negatif, sedangkan untuk nilai eigen kedua belum diketahui. Akibatnya didapat syarat pada titik tetap pertama ini, yaitu

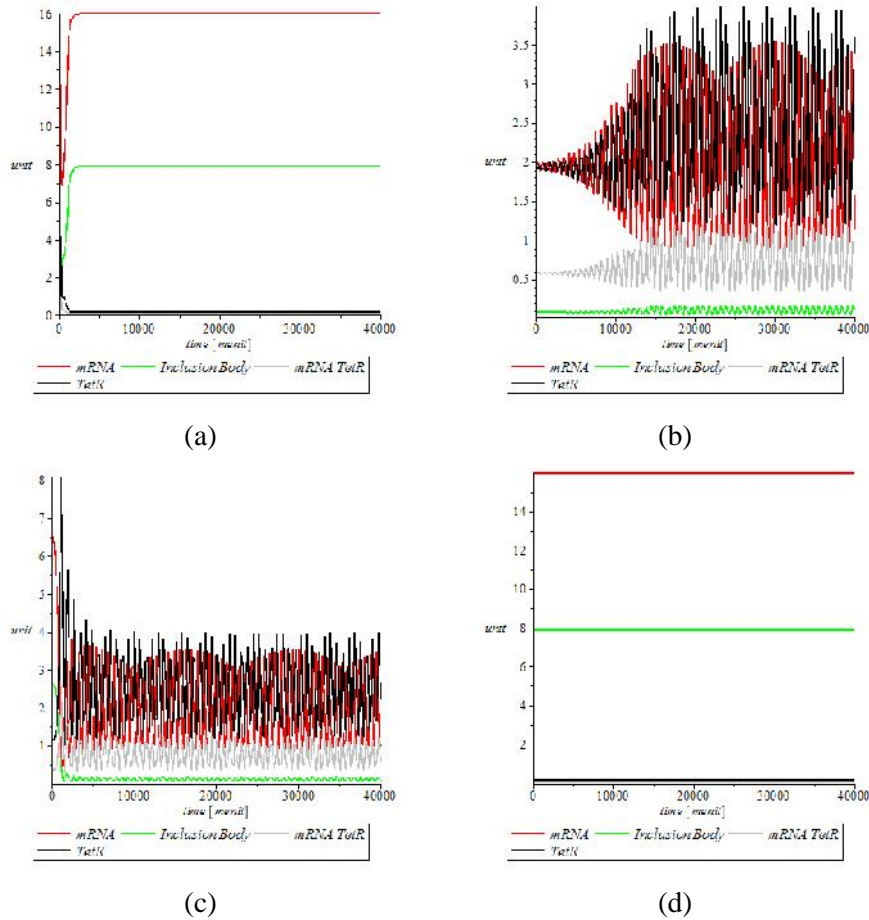
- $r_i r_m \leq x_i x_m$ , mengakibatkan model stabil
- $r_i r_m > x_i x_m$ , mengakibatkan model tidak stabil

Untuk titik tetap yang bernilai real dan **eksis** diatas, diperoleh nilai eigen seperti pada Tabel 3 :

Tabel 3. Kestabilan Titik Tetap			
No	Titik Tetap	Nilai Eigen	Kestabilan
1	$I = 0, M = 16$	$-0.005, -0.03$	Tidak stabil
	$Mr = 0, R = 0$	$-0.03, 0.071$	
2	$\{I = 0.081849, M = 1.947328,$	$0.000314 \pm 0.009561$	Tidak stabil
	$Mr = 0.579733, R = 1.932442\}$	$-0.033154 \pm 0.0097561$	
3	$\{I = 2.659325, M = 6.586785,$	$0.004934, -0.040685$	Tidak stabil
	$Mr = 0.3379165, R = 1.126388\}$	$-0.017895 \pm 0.0150391$	
4	$\{I = 7.868997, M = 15.96419,$	$-0.004738, -0.008332$	Stabil
	$Mr = 0.039270, R = 0.130898\}$	$-0.02896, -0.030955$	

iii. Perilaku Dinamik

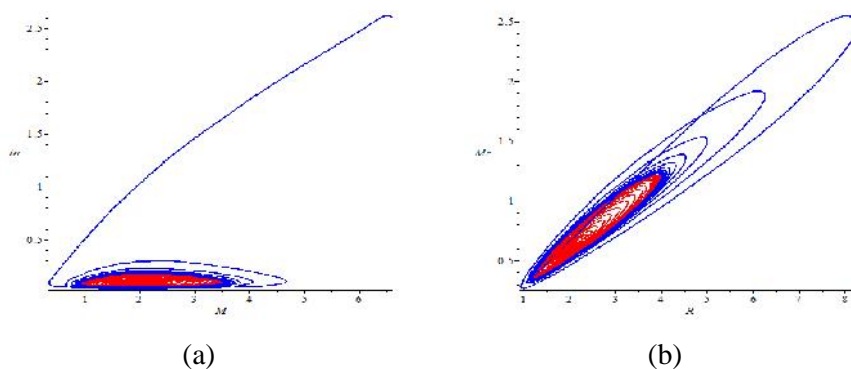
Perilaku dinamik dari model regulasi dengan nilai awal disekitar titik tetapnya.



Gambar 2. Perilaku dinamik nilai awal (a) disekitar titik tetap 1, (b) disekitar titik tetap 2, (c) di sekitar titik tetap 3, (d) di sekitar titik tetap 4

Pada Gambar 2(a), nilai awal yang diambil adalah disekitar titik tetap 1. Dari Gambar tersebut terlihat bahwa sistem regulasi menuju titik tetap 4. Sedangkan pada Gambar 2(b), nilai awal yang diambil adalah disekitar titik tetap 2. Dari gambar tersebut terlihat bahwa sistem regulasi menghasilkan perilaku osilasi dengan amplitudo dan periode tetap. Pada Gambar 2(c), nilai awal yang diambil adalah disekitar titik tetap 3. Dari gambar tersebut terlihat bahwa sistem regulasi menghasilkan perilaku osilasi dengan amplitudo dan periode tetap. Pada Gambar 2(d), nilai awal yang diambil adalah disekitar titik tetap 4. Dari Gambar tersebut terlihat bahwa sistem regulasi stabil. Berikut diperlihatkan trajektori pada bidang fasa untuk nilai awal di sekitar titik tetap 2.





**Gambar 3. Bidang Fasa (a) antara *Inclusion Body* dengan mRNA, (b) antara mRNA TetR dengan TetR**

Gambar 3(a) memperlihatkan perilaku antara *Inclusion Body* dengan mRNA, dimana simulasi dengan nilai awal disekitar titik tetap 3 menghasilkan solusi periodic berbentuk spiral yang bergerak ke dalam. Akan tetapi jika nilai awal yang dipakai adalah disekitar titik tetap 2, diperoleh trajektori yang beresilasi dengan bentuk spiral yang bergerak keluar, tetapi tidak pernah melewati daerah trajektori biru. Demikian juga gambar 3(b) untuk perilaku antara mRNA TetR dan TetR memperlihatkan keadaan yang sama. Hal ini menunjukkan perilaku dinamik sistem beresilasi dengan amplitudo dan periode yang menuju nilai yang sama. Ini menunjukkan eksistensi dari *limit cycle*.

**iv. Analisis Parameter dan Perilaku Nilai Awal**

Analisis paramater dilakukan untuk melihat seberapa jauh parameter dapat divariasikan sehingga model masih beresilasi. Analisis parameter dilakukan dengan cara menentukan kapan sistem mengalami bifurkasi hopf, dengan melihat perilaku nilai eigen dari matriks *Jacobian* untuk tiap titik tetap yang ada pada Tabel 3. Analisis dilakukan dengan memvariasikan satu parameter, tetapi menjaga parameter lainnya tetap sama seperti di Tabel 2. Selanjutnya dilihat seberapa jauh parameter dapat divariasikan.

Untuk analisis parameter yang dilakukan pada titik tetap pertama tidak mempengaruhi nilai parameter yang ada. Hal ini disebabkan oleh syarat  $r_i r_m \leq x_i x_m$  mengakibatkan sistem menjadi stabil. Sedangkan apabila dipakai syarat  $r_i r_m > x_i x_m$  sistem menjadi tidak stabil.

Selanjutnya analisis parameter yang dilakukan pada titik tetap kedua, diperoleh pada tabel dibawah ini.

**Tabel 4. Selang parameter untuk titik tetap 2 yang masih beresilasi**

Parameter	Nilai	Parameter	Nilai
$r_m$	[0.04, 0.56]	$x_m$	[0.0025, 0.0086]
$r_i$	[0.0025, 0.035]	$x_i$	[0.0079, 0.067]
$r_{mr}$	[0.05, $\infty$ ]	$x_{mr}$	[0.001, 0.035]
$r_r$	[0.05, $\infty$ ]	$x_r$	[0.001, 0.035]

Tabel 4 memperlihatkan sistem masih beresilasi pada selang tiap parameter. Apabila digunakan nilai parameter diluar dari selang tersebut, maka model akan menjadi stabil dengan menghasilkan nilai eigen negatif. Selanjutnya selang nilai paramater pada titik tetap 2 dan 3, tapi model masih dapat beresilasi.

**Tabel 5. Selang parameter untuk titik tetap 3 dan 4 yang masih beresilasi**

Parameter	Nilai	Parameter	Nilai
$r_m$	[0.046, 4]	$x_m$	[0.0008, 0.0088]
$r_i$	[0.0029, 0.25]	$x_i$	[0.0016, 0.015]
$r_{mr}$	[0.019, 0.28 ]	$x_{mr}$	[0.008, 0.11]
$r_r$	[0.02, 0.029 ]	$x_r$	[0.008, 0.11]

Tabel 5 menjelaskan tentang nilai parameter pada selang diatas masih menghasilkan nilai titik tetap yang *real*, sedangkan diluar dari selang tersebut nilai titik tetap menjadi kompleks. Jadi nilai parameter yang dapat dipakai berada pada Tabel 4 -5, sehingga memperoleh hasil simulasi yang beresilasi untuk model regulasi. Selanjutnya perilaku nilai awal yang dapat merepresentasikan fenomena biologi pada model regulasi.

Untuk melihat perilaku tersebut, berikan nilai awal di salah satu variabel dengan nilai awal variabel lainnya sama dengan nol (kecuali *Inclusion Body* dalam jumlah yang sangat sedikit). berikut nilai awal yang dapat diberikan:

**Tabel 6. Nilai Awal yang Dapat Diberikan**

	mRNA	<i>Inclusion Body</i>	mRNA TetR	TetR
Nilai awal 1	[0, 7.3]	0.00001	0	0
Nilai awal 2	0	[0.00001, 3.158]	0	0
Nilai awal 3	0	0.00001	[0, 13.4]	0
Nilai awal 4	0	0.00001	0	[0, 119,76]

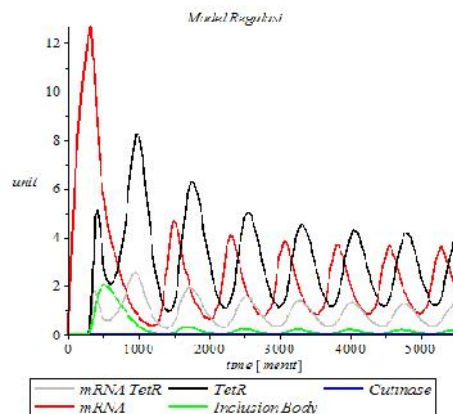
Perilaku nilai awal pada Tabel 6, menunjukkan bahwa model regulasi masih dapat beresilasi, apabila diberikan nilai awal yang berbeda dari tabel diatas maka model akan stabil.

#### 4.2. Simulasi Model

Hasil simulasi numerik pada model regulasi dengan nilai parameter pada Tabel 2 dan nilai awal yang diberikan sesuai keadaan biologi dimana konsentrasi awal dari model regulasi bernilai nol (kecuali *Inclusion Body* dalam jumlah yang sedikit), dikarenakan DNA belum mentranskripsi mRNA. Hasil simulasi diperoleh dengan nilai awal ( $I = 0.00001$ ,  $C = 0$ ,  $M = 0$ ,  $Mr = 0$ ,  $R = 0$ ) ditunjukkan dari gambar berikut.

Terlihat dari Gambar 3, model regulasi, ketika konsentrasi *LC-Cutinase* meningkat maka mulai muncul *Inclusion Body*. Ketika konsentrasi *Inclusion Body* mulai mengalami peningkatan, maka muncul mRNA TetR yang diikuti pula oleh TetR. Saat TetR mulai muncul, mRNA mengalami penurunan dikarenakan TetR menghentikan proses transkripsi dari DNA menjadi mRNA.

Jika mRNA turun, maka hal serupa terjadi pada *LC-Cutinase*. Saat mRNA terhenti maka *Inclusion Body* akan ikut berkurang dan begitupun dengan mRNA TetR. Penurunan TetR akan mengakibatkan mRNA kembali naik, dan keadaan ini berulang seterusnya. Hal ini sudah dapat merepresentasikan fenomena biologi dimana menghasilkan simulasi yang beresilasi.



**Gambar 4. Hasil simulasi model regulasi**

#### 5. Kesimpulan

Model regulasi pada sistem biodegradasi PET, serta analisis dan simulasi modelnya telah dijelaskan. dapat disimpulkan bahwa fenomena pada modul regulasi sudah dapat merepresentasikan hasil eksperimen, Hal ini terlihat dari model regulasi menghasilkan simulasi yang beresilasi.

## Daftar Pustaka

1. IGEM, ITB. *2014.igem.org*. [Online] 2014. [http://2014.igem.org/Team:ITB\\_Indonesia](http://2014.igem.org/Team:ITB_Indonesia).
2. Gourmelon, Gaelle. *www.worldwatch.org*. [Online] 2015. <http://www.worldwatch.org/global-plastic-production-rises-recycling-lags-0>.
3. IGEM, UC Davis,. *2012.igem.org*. [Online] 2012. [http://2012.igem.org/Team:UC\\_Davis/Project/Catalyst](http://2012.igem.org/Team:UC_Davis/Project/Catalyst).
4. Anton, H. dan Rorres, C. *Elementary Linear Algebra*. 10th. s.l. : Wiley, 2010.
5. Brauer, Fred dan Castillo-Chaves, Carlos. *Mathematical Model in Population Biology and Epidemiology*. 2nd. s.l. : Springer, 2010.



## PERAMALAN JUMLAH KENDARAAN DI DKI JAKARTA DENGAN JARINGAN BACKPROPAGATION

Alwi Smith<sup>1</sup>, Zumrotus Sya'diyah<sup>2</sup>

<sup>1</sup>Fakultas Keguruan dan Ilmu Pendidikan Universitas Pattimura

<sup>1</sup>Fakultas Keguruan dan Ilmu Pendidikan Universitas Darussalam Ambon

Jl. Ir. M. Putuhena, Kampus Unpatti, Poka-Ambon, Indonesia

e-mail: <sup>1</sup>alwi.smith1960@gmail.com, <sup>2</sup>zuma.yakuza@gmail.com

---

### Abstrak

Kemacetan di ibukota DKI Jakarta tidak dapat dihindari, terutama pada titik-titik persimpangan baik di jalan-jalan protokol hingga di jalan lingkungan. Semakin hari, kemacetan di Jakarta semakin parah. Menurut sebuah penelitian, kemacetan tersebut membuat masyarakat Jakarta mengalami kerugian hingga Rp 48 triliun per tahun [1]. Dalam makalah ini akan dibahas mengenai prediksi jumlah kendaraan pada tahun 2017. Prediksi ini akan dilakukan dengan menggunakan jaringan syaraf tiruan, yaitu metode backpropagation. Metode ini digunakan karena keunggulannya dalam learning rate. Learning rate sangat berguna dalam menentukan prediksi dengan eror yang kecil. Prediksi jumlah kendaraan ini akan dilakukan pada kendaraan bermotor, mobil pribadi dan kendaraan umum. Sehingga dari hasil prediksi ini akan dapat ditentukan langkah-langkah yang tepat untuk menekan laju pertumbuhan jumlah kendaraan. Dalam pembahasannya nanti akan digunakan Matlab 2009a.

*Kata Kunci* : jaringan syaraf tiruan, metode backpropagation..

## FORECASTING NUMBER OF VEHICLES IN DKI JAKARTA WITH BACKPROPAGATION NETWORK

### Abstract

Congestion in the capital city of Jakarta can not be spared, imagine at the intersection points both in the streets protocol up on the road environment. The more the day, the congestion in Jakarta getting worse. According to a study, the congestion that caused the people of Jakarta to lose up to Rp 48 trillion per year [1]. In this paper will discuss about the prediction of the number of vehicles in 2017. This prediction will be done by using artificial neural network, the method of backpropagation. This method is used because of its superiority in learning. Level of learning is very useful in determining predictions with small errors. Predicted number of these vehicles will be done on motor vehicles, private cars and public transport. From the results of this preview will be determined the appropriate steps to suppress the growth rate of the number of vehicles. In the discussion will be used Matlab 2009a.

*Keywords*: artificial neural network, backpropagation method.

---

### 1. Pendahuluan

Kemacetan lalu lintas Jakarta yang semakin akut ini membuat pemerintah Provinsi DKI Jakarta melakukan berbagai upaya untuk mengatasinya. Namun upaya ini seakan sia-sia jika melihat data terbaru jumlah kendaraan di Ibukota. Oleh karena itu, dibutuhkan prediksi yang akurat untuk mengetahui jumlah kendaraan tiap tahun agar dapat dilakukan tindakan preventif untuk menyesuaikan antara kebutuhan masyarakat atas kendaraan dengan kendaraan umum yang ada. Selain itu, dapat dilakukan juga pembatasan pada kepemilikan kendaraan. Hal ini diharapkan dapat membantu pemerintah dalam melakukan upaya penanganan terhadap kemacetan yang selalu terjadi dan bahkan bertambah parah tiap tahunnya.

Dalam makalah ini akan dibahas mengenai prediksi jumlah kendaraan pada tahun 2017. Prediksi ini akan dilakukan dengan menggunakan jaringan syaraf tiruan, yaitu metode *backpropagation*. Metode ini digunakan karena keunggulannya dalam *learning rate*. *Learning rate* sangat berguna dalam menentukan prediksi dengan eror yang kecil. Prediksi jumlah kendaraan ini akan dilakukan pada kendaraan bermotor, mobil pribadi dan kendaraan umum. Sehingga dari hasil prediksi ini akan dapat ditentukan langkah-langkah yang tepat untuk menekan laju pertumbuhan jumlah kendaraan. Dalam pembahasannya nanti akan digunakan Matlab 2009a.

### 1.1. Jaringan Saraf Tiruan [2]

Jaringan syaraf tiruan (*artificial neural network*) adalah pemrosesan sistem informasi pada karakteristik tertentu dalam keadan yang berhubungan dengan jaringan syaraf biologi. Jaringan syaraf tiruan dibangun berdasarkan generalisasi dari model matematika pada manusia atau syaraf biologi, didasarkan pada asumsi:

- 1) Pemrosesan informasi terjadi pada banyak element yang disebut *neuron*.
- 2) Sinyal berjalan diantara neuron yang terkoneksi jaringan.
- 3) Masing masing jaringan koneksi dihubungkan dengan bobot, yang mana didalam jaringan syaraf khusus, melipatgandakan transmisi sinyal.
- 4) Masing masing neuron mempergunakan fungsi aktifasi (biasanya nonlinier) ke jaringan inputnya untuk menentukan sinyal output.

Jaringan syaraf dikarakteristikan oleh :

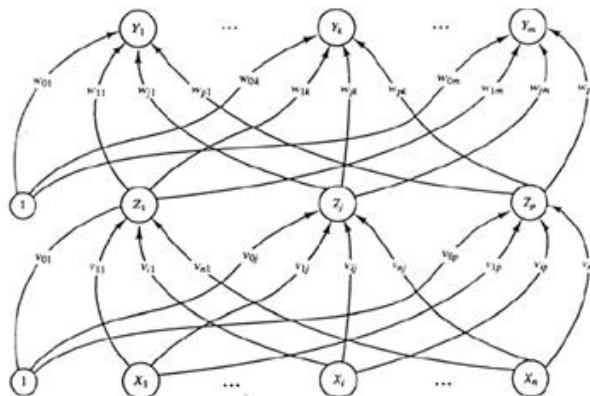
- 1) Polanya terkoneksi diantara *neuron* (disebut arsitektur)
- 2) Metode untuk menentukan bobot dari koneksi (disebut training atau learning, algoritma)
- 3) Fungsi aktifasi.

### 1.2. Backpropagation [2]

*Backpropagation* melatih jaringan untuk mendapatkan keseimbangan antara kemampuan jaringan untuk mengenali pola yang digunakan selama proses training serta kemampuan jaringan untuk memberikan respon yang benar terhadap pola inputan yang serupa (tapi tak sama) dengan pola yang dipakai selama *training*.

#### 1.2.1 Arsitektur

*Backpropagation* memiliki beberapa unit yang ada di dalam satu atau lebih hidden layer.



**Gambar 1. Arsitektur backpropagation**

Gambar 1 merupakan arsitektur *backpropagation* dengan  $n$  buah inputan (ditambah sebuah bias), sebuah hidden layer yang terdiri dari  $p$  unit (ditambah sebuah bias), serta  $m$  buah unit output. Dimana  $v_{ji}$  merupakan bobot garis dari unit input  $x_i$  ke unit hidden layer  $z_j$  ( $v_{j0}$  merupakan bobot garis yang menghubungkan bias di unit inputan ke unit *hidden layer*  $z_j$ ).  $w_{ji}$  merupakan bobot dari unit hidden layer  $z_j$  ke unit output  $y_k$  ( $w_{k0}$  merupakan bobot garis yang menghubungkan bias di unit *hidden layer* ke unit output  $z_k$ ).

### 1.2.2 Fungsi Aktivasi

Fungsi aktivasi untuk jaringan *backpropagation* memiliki beberapa karakteristik yang penting. Fungsi aktivasi tersebut kontinu, dapat diturunkan dan tidak turun secara monoton. Untuk efisiensi perhitungan, turunan dari fungsi aktivasi tersebut juga mudah dihitung. Fungsi aktivasi yang paling umum digunakan adalah nilai dari turunannya (pada nilai tertentu dari variabel independen) dapat diekspresikan dalam bentuk nilai dari fungsi aktivasi tersebut (pada nilai variabel independen). Fungsi aktivasi yang paling sering digunakan adalah fungsi sigmoid biner. Sedangkan fungsi lain yang juga cukup sering dipakai adalah fungsi sigmoid bipolar. Penjelasan dari kedua fungsi tersebut diberikan sebagai berikut:

1. Fungsi sigmoid biner untuk interval (0,1)

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

$$f'(x) = f(x)[1 - f_1(x)]$$

2. Fungsi sigmoid bipolar dengan interval (-1,1)

$$f(x) = \frac{2}{1 + e^{-x}} - 1$$

$$f'(x) = \frac{1}{2}[1 + f_2(x)][1 - f_2(x)]$$

### 1.2.3 Algoritma *Backpropagation*

Step 0 : Inisialisasi bobot semua bobot dengan bilangan acak kecil

Bobot bias input ( $v_{0j}$ ) = bilangan acak dari  $-\beta$  dan  $\beta$ , dengan

Bobot input ( $v_{ij}$ ) = bilangan acak dari -0.5 dan 0.5

Bobot bias hidden ( $w_{0k}$ ) dan bobot hidden ( $w_{jk}$ ) = bilangan acak dari -1 dan 1.

Step 1 : Jika kondisi penghentian belum terpenuhi, lakukan Step 2-9

Step 2 : Untuk setiap pasang data pelatihan, lakukan langkah 3-8

Step 3 : Tiap unit input ( $X_i, i=1, \dots, n$ ) menerima sinyal dan meneruskannya ke unit hidden

Step 4 : Hitung semua output di unit hidden ( $Z_j, j=1, \dots, p$ )

$$z\_in_j = v_{0j} + \sum_{i=1}^n x_i v_{ij}$$

$$z_j = f(z\_in_j)$$

Step 5 : Hitung semua output di unit output ( $Y_k, k=1, \dots, m$ )

$$y\_in_k = w_{0k} + \sum_{j=1}^p z_j w_{jk}$$

$$y_k = f(y\_in_k)$$

Step 6 : Hitung  $\delta$  unit output berdasarkan error di setiap unit output  $Y_k$

$$u_k = (t_k - y_k) f'(y\_in_k)$$

Hitung suku perubahan bobot hidden dan bobot bias *hidden* dengan *learning rate*  $\alpha$

$$\Delta w_{jk} = \alpha u_k z_j$$

$$\Delta w_{0k} = \alpha u_k$$

Step 7 : Hitung  $\delta$  unit *hidden* berdasarkan *error* di setiap unit *hidden*  $Z_j$

Hitung suku perubahan bobot input dan bobot bias input dengan *learning rate*  $\alpha$

$$u_{in_j} = \sum_{k=1}^m u_k w_{jk}$$

$$u_j = u_{in_j} f'(z_{in_j})$$

$$\Delta v_{ij} = \tau u_j x_i$$

$$\Delta v_{0_j} = \tau u_j$$

Step 8 : Hitung semua perubahan bobot

$$w_{jk}(\text{baru}) = w_{jk}(\text{lama}) + \Delta w_{jk}$$

$$v_{ij}(\text{baru}) = v_{ij}(\text{lama}) + \Delta v_{ij}$$

Step 9 : Test kondisi penghentian

## 2. Hasil dan Pembahasan

### 2.1. Analisa Data

Sebagaimana telah disebutkan sebelumnya bahwa data yang akan digunakan dalam proses peramalan atau prediksi ini adalah data jumlah kendaraan di Jakarta selama sepuluh tahun terakhir. Data ini akan digunakan untuk mengetahui kisaran yang akurat tentang jumlah kendaraan Data yang digunakan dalam proses prediksi adalah sebagai berikut:

**Tabel 1. Data Jumlah Kendaraan Polda Metro Jaya [3]**

Tahun	Jenis Kendaraan Bermotor				Jumlah
	Sepeda Motor	Mobil Penumpang	Mobil Beban	Mobil Bis Penumpang	
2007	1.813.136	1.130.496	347.443	253.648	3.544.723
2008	2.257.194	1.195.871	366.221	254.849	4.074.135
2009	3.316.900	1.529.824	464.748	315.652	5.627.124
2010	3.940.700	1.645.306	488.517	316.396	6.390.919
2011	4.647.435	1.766.801	499.581	316.502	7.230.319
2012	5.310.068	1.835.653	504.727	317.050	7.967.498
2013	5.947.173	1.916.469	518.991	318.332	8.727.965
2014	6.765.723	2.034.943	538.731	308.528	9.647.925
2015	7.084.753	2.054.254	507.410	308.941	9.955.358
2016	8.484.384	2.285.802	564.776	322.440	11.657.402

(Sumber: www.komisikepolisianindonesia.com, 2016)

Tampak pada tabel bahwa kenaikan jumlah kendaraan di Jakarta sangat signifikan. Tapi, tiap tahun mempunyai tingkat kenaikan yang berbeda-beda. Hal inilah yang membuat prediksi atas data selanjutnya yang akurat menjadi sulit didapatkan. Oleh karena itu, akan digunakan metode backpropagation untuk meramalkan data berikutnya. Program akan belajar untuk mengetahui model kenaikan jumlah kendaraan yang berubah dan berbeda tiap tahunnya.

### 2.2. Hasil dan Representasi Prediksi

Pada metode *backpropagation*, langkah yang pertama dilakukan adalah menentukan bobot awal. Pemilihan bobot awal akan mempengaruhi apakah jaringan mencapai minimum global (atau hanya lokal) dari *error*, dan jika itu terjadi, seberapa cepat akan konvergen. *Update* bobot antara dua unit bergantung pada turunan fungsi aktivasi dari unit batas atas dan unit batas bawah. Untuk alasan inilah, sangat penting untuk menghindari pemilihan bobot awal yang akan membuat aktivasi atau turunan dari aktivasi tersebut bernilai nol. Nilai dari bobot awal tidak harus terlalu besar, atau sinyal *input* awal ke tiap unit *hidden* atau *output* akan jatuh pada daerah dimana turunan dari fungsi sigmoid memiliki nilai yang sangat kecil (disebut *area saturation*). Selain itu, apabila bobot awal terlalu kecil, *input* jaringan untuk unit *hidden* atau *output* akan mendekati nol, yang akan menyebabkan proses *learning* yang sangat lambat.



Prosedur umum adalah untuk memberi bobot awal (dan bias) dengan nilai acak antara -0.5 dan 0.5 (atau antara -1 dan 1 atau interval lain yang sesuai). Nilai tersebut bisa negatif maupun positif karena nilai akhir dari bobot setelah *training* mungkin sesuai dengan tanda tersebut.

Pada metode ini, proses training tidak dilakukan pada seluruh data, melainkan hanya sebagian saja (biasanya setengah dari data keseluruhan). Sedangkan sisa data yang lain digunakan dalam proses testing. Dalam training, bobot akan diperbarui di setiap iterasi sedemikian hingga diperoleh bobot yang sesuai dengan model data. Sedangkan proses *testing* akan menentukan seberapa akurat prediksi yang telah dilakukan. Hal inilah yang membuat metode ini unggul dari metode yang lain.

Bobot input yang dipakai adalah data yang bernilai (-0.5, 0.5). Bobot ini akan diinputkan ke program dalam bentuk matriks yang ukurannya sesuai dengan jumlah data yang akan diproses dan jumlah hidden layer yang diinginkan. Untuk jaringan syaraf tiruan dengan lebih dari 1 unit *hidden*, hanya modifikasi kecil dari algoritma *backpropagation* pada bab 2.2.2. Perhitungan dari U diulang untuk setiap *hidden layer*. Hasil teoritis yang telah banyak ditunjukkan pada menyimpulkan bahwa satu *hidden layer* cukup bagi sebuah jaringan *backpropagation* untuk menetapkan setiap pemetaan kontinu dari pola *input* ke pola *output* pada sebarang derajat ketepatan. Bagaimanapun juga, 2 *hidden layer* mungkin dapat membuat *training* lebih mudah untuk beberapa situasi. Sedangkan bobot awal untuk bobot input diambil nilai  $[-S, S]$ , dengan  $S = 0.7(p)^{1/n} = 0.7\sqrt[n]{p}$ . Dan bobot awal untuk bobot hidden diambil nilai (-0.5,0.5).

Dari penjelasan tersebut, maka dalam makalah ini akan digunakan bobot input data yang bernilai (0.5,0.5), jumlah data adalah 10 dan banyak *hidden layer* adalah 1. Oleh karena itu, dapat diambil suatu inisialisasi bobot berupa matriks berukuran  $5 \times 1$  yang dinyatakan dalam tabel berikut:

**Tabel 2. Bobot Input**

Bobot input
0.2
0.2
0.2
0.2
0.2

Karena data yang dimiliki bernilai sangat besar, maka perlu dilakukan konversi dengan rumus berikut:

$$x\_baru = \frac{x\_lama - x\_min}{x\_maks - x\_min}$$

Konversi ini pada akhir peramalan atau prediksi akan dikembalikan lagi dengan invers dari persamaan tersebut sebagai berikut:

$$y\_pred = y\_lama * (y\_maks - y\_min) + y\_min$$

Sehingga, konversi ini tidak akan mengubah data yang dihasilkan. Dari proses konversi pada data input diperoleh:

**Tabel 3. Hasil Konversi data Inputan**

Tahun	Jenis Kendaraan Bermotor				Jumlah
	Sepeda Motor	Mobil Penumpang	Mobil Beban	Mobil Bis	
2007	0	0	0	0	0
2008	0.066563	0.056587	0.086402	0.017458	0.065257
2009	0.22541	0.345647	0.539748	0.901326	0.256685
2010	0.318915	0.445605	0.649114	0.912141	0.350833
2011	0.424853	0.550768	0.700023	0.913682	0.454301
2012	0.52418	0.610364	0.7237	0.921648	0.545168
2013	0.61968	0.680316	0.789332	0.940284	0.638906
2014	0.742378	0.782864	0.880161	0.797767	0.752304
2015	0.7902	0.799579	0.723765	0.803771	0.7902
2016	1	1	1	1	1

Data pada Tabel 3 adalah data inputan untuk program. Setelah running, yaitu dengan jumlah hidden layer adalah 1, *learning rate* sebesar 0.5, banyak iterasi adalah 3 (telah mencapai kekonvergenan, artinya bobot tidak terbaru lagi), bobot bias input adalah 0.7, bobot bias *hidden* adalah 0.5, dan bobot *hidden layer*

(berukuran 1x1) sebesar 0.2, diperoleh data hasil *running* program sebagaimana dijelaskan dalam sub bab berikut ini.

### 2.2.1 Prediksi Jumlah Kendaraan Bermotor

Setelah dilakukan *running* program terhadap data jumlah kendaraan bermotor selama sepuluh tahun terakhir, diperoleh data-data sebagai berikut:

**Tabel 4. Bobot Awal dan Akhir layer**

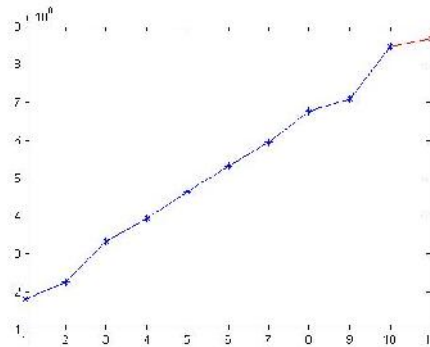
Bobot Awal Input	Bobot akhir Input ( $\times 10^5$ )
0.7	1.0879
0.2	0.0000
0.2	0.0724
0.2	0.2452
0.2	0.3470
0.2	0.4622

a) Bobot Awal dan Akhir *layer* input

Bobot Awal <i>Hidden</i>	Bobot Akhir <i>Hidden</i> ( $\times 10^7$ )
0.5	5.0388
0.2	4.9625

b) Bobot Awal dan Akhir *hidden layer*

Sehingga diperoleh data prediksi jumlah kendaraan bermotor pada tahun 2017 adalah sebesar  $8.6895 \times 10^6$ . Hal ini digambarkan dengan garis merah pada grafik berikut:



**Grafik 1. Hasil Prediksi Jumlah Kendaraan Bermotor**

### 2.2.2 Prediksi Jumlah Mobil Penumpang

Setelah dilakukan *running* program terhadap data jumlah mobil penumpang selama sepuluh tahun terakhir, diperoleh data-data sebagai berikut:

**Tabel 5. Bobot Awal dan Akhir layer**

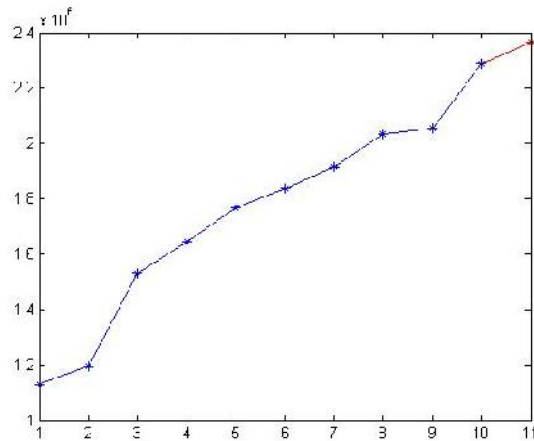
Bobot Awal Input	Bobot akhir Input ( $\times 10^4$ )
0.7	3.6429
0.2	0.0000
0.2	0.2062
0.2	1.2591
0.2	1.6233
0.2	2.0064

a) Bobot Awal dan Akhir *layer* input

Bobot Awal <i>Hidden</i>	Bobot Akhir <i>Hidden</i> ( $\times 10^7$ )
0.5	1.5191
0.2	1.4940

b) Bobot Awal dan Akhir *hidden layer*

Sehingga diperoleh data prediksi jumlah mobil penumpang pada tahun 2017 adalah sebesar  $2.3665 \times 10^6$ . Hal ini digambarkan dengan garis merah pada grafik berikut:



**Grafik 2. Hasil Prediksi Jumlah Mobil Penumpang**

### 2.2.3 Prediksi Jumlah Mobil Beban

Setelah dilakukan *running* program terhadap data jumlah mobil beban selama sepuluh tahun terakhir, diperoleh data-data sebagai berikut:

**Tabel 6. Bobot Awal dan Akhir layer**

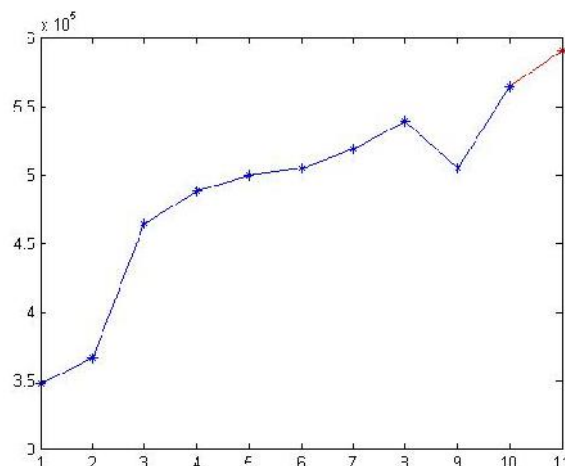
Bobot Awal Input	Bobot akhir Input ( $\times 10^3$ )
0.7	9.4811
0.2	0.0002
0.2	0.8193
0.2	5.1172
0.2	6.1541
0.2	6.6367

a) Bobot Awal dan Akhir *layer* input

Bobot Awal <i>Hidden</i>	Bobot Akhir <i>Hidden</i> ( $\times 10^6$ )
0.5	3.9479
0.2	3.8847

b) Bobot Awal dan Akhir *hidden layer*

Sehingga diperoleh data prediksi jumlah mobil beban pada tahun 2017 adalah sebesar  $5.9050 \times 10^5$ . Hal ini digambarkan dengan garis merah pada grafik berikut:



**Grafik 3. Hasil Prediksi Jumlah Mobil Beban**

### 2.2.4 Prediksi Jumlah Mobil Bis

Setelah dilakukan running program terhadap data jumlah mobil bis selama sepuluh tahun terakhir, diperoleh data-data sebagai berikut:

**Tabel 7. Bobot Awal dan Akhir layer**

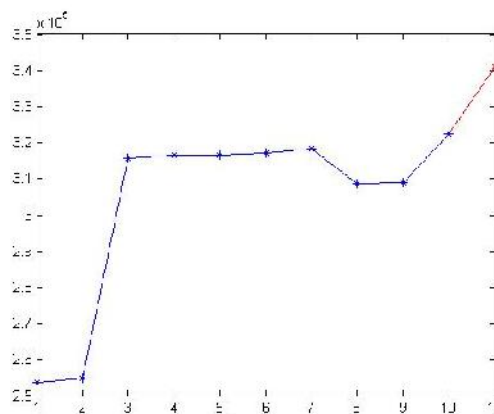
Bobot Awal Input	Bobot akhir Input ( $\times 10^3$ )
0.7	5.4923
0.2	0.0002
0.2	0.0961
0.2	4.9499
0.2	5.0093
0.2	5.0177

a) Bobot Awal dan Akhir layer input

Bobot Awal Hidden	Bobot Akhir Hidden ( $\times 10^6$ )
0.5	2.3629
0.2	2.3276

b) Bobot Awal dan Akhir hiddenlayer

Sehingga diperoleh data prediksi jumlah mobil beban pada tahun 2017 adalah sebesar  $3.4102 \times 10^5$ . Hal ini digambarkan dengan garis merah pada grafik berikut:



**Grafik 4. Hasil Prediksi Jumlah Mobil Bis**

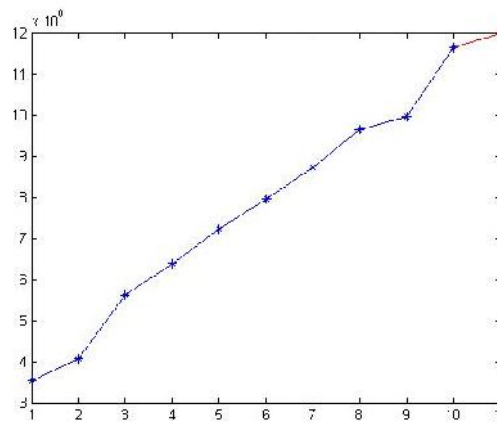
### 2.2.5 Prediksi Jumlah Total Seluruh Kendaraan

Setelah dilakukan running program terhadap data jumlah total seluruh kendaraan selama sepuluh tahun terakhir, diperoleh data-data sebagai berikut:

**Tabel 8. Prediksi jumlah Total kendaraan tahun 2017**

	Prediksi kenaikan jumlah tiap jenis kendaraan
Sepeda Motor	8.689.500
Mobil Penumpang	2.366.500
Mobil Beban	590.500
Mobil Bis	341.020
Jumlah Total	11.987.520

Sehingga diperoleh data prediksi jumlah total seluruh kendaraan pada tahun 2017 adalah sebesar 11.987.520. Hal ini digambarkan dengan garis merah pada grafik berikut:



**Grafik 5. Hasil Prediksi Jumlah Total Kendaraan**

### 3. Kesimpulan

Dari penggunaan program menggunakan metode *backpropagation*, diperoleh hasil prediksi sebagai berikut:

	Prediksi kenaikan jumlah tiap jenis kendaraan
Sepeda Motor	8.689.500
Mobil Penumpang	2.366.500
Mobil Beban	590.500
Mobil Bis	341.020
Jumlah Total	11.987.520

Tabel menunjukkan bahwa jumlah sepeda motor naik sebanyak 205.116 unit dari data september 2016, mobil penumpang naik 80.698 unit, mobil beban naik sebanyak 25.724 unit, mobil bis naik 18.580 unit, sedangkan total jumlah kendaraan akan mengalami kenaikan pula sebesar 330.118 unit.

Kenaikan ini harus diatasi dengan pengurangan jumlah kendaraan. Hal ini dapat dilakukan dengan mengadakan uji kelayakan kendaraan dimana kendaraan yang sudah tak layak operasi harus segera diberhentikan ijin beroperasinya. Selain itu, pengurangan tingkat kenaikan jumlah kendaraan dapat dilakukan juga dengan mengurangi pembelian kendaraan dengan cara kredit. Hal ini dikarenakan oleh pemberian kredit dengan bunga yang terlalu rendah meningkatkan keinginan dan daya beli masyarakat atas kendaraan bermotor. Dan kasus ini menyumbangkan pengaruh besar bagi meroketnya jumlah kendaraan. Cara lain yang juga dapat dilakukan adalah membatasi kepemilikan kendaraan bagi tiap keluarga. Hal ini disebabkan oleh tingginya tingkat konsumsi masyarakat Indonesia atas barang mewah, terutama mobil.

### Daftar Pustaka

1. Fausett, Laurence. *Fundamentals of Neural Network: Architectures, Fundamentals, and Applications*. Newe Jersey : Prentice-Hall, Inc., 1994.
2. Susanto. *www.Detik News.com*. [Online] 26 November 2008. [Dikutip: 23 Juni 2015.] [www.Detik News.com//kemacetan\\_susanto\\_26NOV2008](http://www.Detik News.com//kemacetan_susanto_26NOV2008).
3. Kompolnas. *www.komisikepolisianindonesia.com*. [Online] Juni 2016. [www.komisikepolisianindonesia.com/data\\_kendaraan\\_2016](http://www.komisikepolisianindonesia.com/data_kendaraan_2016).



## PREDIKSI PENGGUNAAN *BANDWIDTH* MENGGUNAKAN *ELMAN RECURRENT NEURAL NETWORK*

Jefri Radjabaycolle<sup>1</sup>, Reza Pulungan<sup>2</sup>

<sup>1,2</sup>Jurusan Ilmu Komputer dan Elektronika, FMIPA UGM, Yogyakarta  
e-mail: <sup>1</sup>jeffiradja@gmail.com, <sup>2</sup>pulungan@ugm.ac.id

---

### Abstrak

Jaringan Syaraf Tiruan (JST) sering dipakai dalam menyelesaikan permasalahan tertentu seperti prediksi, klasifikasi, dan pengolahan data. Berdasarkan hal tersebut, dalam penelitian ini mencoba menerapkan JST untuk menangani permasalahan dalam prediksi penggunaan *bandwidth*. Sistem yang dikembangkan dapat digunakan untuk memprediksi penggunaan *bandwidth* dengan menerapkan *Elman Recurrent Neural Network* (ERNN). Struktur Elman dipilih karena dapat membuat iterasi jauh lebih cepat sehingga memudahkan proses konvergensi. Vektor input yang digunakan menggunakan *windows size*. Hasil penelitian dengan menggunakan target *error* sebesar 0.001 menunjukkan nilai MSE terkecil yaitu pada *windows size* 11 dengan nilai 0.002833. Kemudian dengan menggunakan 13 *neuron* pada *hidden layer* diperoleh nilai *error* paling optimal (*minimum error*) sebesar 0.003725.

*Kata kunci* : Prediksi, *Bandwidth*, jaringan syaraf tiruan (JST), *elman recurrent neural network*

## PREDICTIONS OF BANDWIDTH USING ELMAN RECURRENT NEURAL NETWORK

### Abstract

Artificial Neural Network (ANN) can be used to solve specific problems such as prediction, classification, and data processing. Accordingly, in this study tries to apply neural networks to deal with the problem of prediction bandwidth usage. The system developed can be used to predict the bandwidth usage by applying Elman Recurrent Neural Network (ERNN). Elman structure chosen because it can make much faster iterations so as to facilitate the process of convergence. Input vector used to use windows size. The results using a target error of 0001 showed the smallest MSE value is on the windows size 11 with a value of 0.002833. Then, using 13 neurons in the hidden Layer error values obtained most optimal (*minimum error*) amounted to 0.003725.

*Keywords* : Prediction, Bandwidth, artificial neural network (ANN), elman recurrent neural network

---

### 1. Pendahuluan

*Bandwidth* merupakan ukuran jumlah data yang dapat melakukan perjalanan lebih dari satu sistem komunikasi yang dialokasikan dalam rentang waktu atau disebut juga sebagai kecepatan data. *Bandwidth* juga memiliki arti bahwa semakin besar *bandwidth* bisa menghasilkan komunikasi yang lebih cepat. Ketersediaan *bandwidth* jaringan merupakan faktor penting dalam memilih layanan web. Pada dasarnya besarnya kebutuhan *bandwidth* mempresentasikan kapasitas dari koneksi, semakin tinggi kebutuhan *bandwidth*, umumnya akan diikuti oleh kinerja yang lebih baik. Salah satu solusi yang paling efektif untuk mengatasinya adalah dengan mengelola pemakaian *bandwidth* yang menghasilkan suatu kualitas layanan lalu lintas aliran data yang baik dan berkualitas.

Kemampuan untuk mengantisipasi kebutuhan *bandwidth* sangat penting untuk layanan yang efisien dan pengambilan keputusan cerdas dalam menghadapi perkembangan lalu lintas yang cepat dan perubahan pola lalu lintas [1]. Pada dasarnya besarnya kebutuhan *bandwidth* mempresentasikan kapasitas dari koneksi, semakin tinggi kebutuhan *bandwidth*, umumnya akan diikuti oleh kinerja yang lebih baik, meskipun kinerja keseluruhan juga tergantung pada faktor-faktor lain, misalnya *latency* yaitu waktu tunda antara masa sebuah

perangkat meminta akses ke jaringan dan masa perangkat itu memberi izin untuk melakukan transmisi .

Khususnya yang terjadi di lingkungan kampus Universitas Pattimura, terdapat beberapa keluhan dari pengguna jaringan internet. Mahasiswa maupun karyawan dan dosen di Universitas Pattimura mengeluhkan adanya koneksi internet yang dirasa kurang sesuai dengan apa yang diharapkan, misalnya dari segi kecepatan ataupun kestabilan koneksi. Kecepatan koneksi yang didapatkan oleh pengguna tidak hanya dipengaruhi oleh alokasi *bandwidth* yang diberikan oleh pihak kampus tetapi dipengaruhi juga oleh alokasi *bandwidth* yang disediakan oleh penyedia layanan *server* yang diakses atau dituju oleh pengguna. Dari pihak administrator jaringan di Universitas Pattimura, diperoleh informasi total *bandwidth* yang dimiliki pihak kampus dengan pembagian alokasi *bandwidth* berdasarkan jumlah pengguna pada setiap departemen atau bagian yang ada. Secara garis besar ada beberapa faktor yang harus dipertimbangkan dalam menentukan alokasi *bandwidth* yaitu jumlah departemen beserta karyawan yang ada dalam setiap departemen, jumlah dosen, jumlah mahasiswa, serta alokasi khusus yang diperlukan untuk layanan *server* untuk mendukung kegiatan belajar mengajar di kampus. Permasalahan yang muncul dari pembagian *bandwidth* yang sudah dilaksanakan adalah sudah tepatkah pembagian *bandwidth* yang diimplementasikan untuk saat ini, perlu dilakukan perubahan atau tidak untuk pembagian *bandwidth* nya sehingga biaya yang dikeluarkan untuk sewa *bandwidth* dapat menjadi efektif dan efisien.

Berangkat dari permasalahan yang sudah disebutkan diatas, diperlukan adanya analisa lebih mendalam mengenai berapa kebutuhan *bandwidth* yang harus disediakan oleh pihak kampus sehingga layanan koneksi internet maupun *Local Area Network* (LAN) dapat berjalan dan mendukung semua aktifitas yang memerlukan kondisi *network* dengan layanan yang sesuai. Untuk membantu pihak manajemen menentukan alokasi yang tepat untuk sewa *bandwidth* maka dibutuhkan suatu sistem yang dapat memprediksi penggunaan *bandwidth* menggunakan jaringan syaraf tiruan dengan algoritma *Elman Recurrent Neural Network*. Struktur Elman dipilih karena dapat membuat iterasi jauh lebih cepat dan konvergensi akan menjadi lebih cepat yang disebabkan oleh jaringan *Elman* memiliki *feedback* yang menawarkan proses pembelajaran yang lebih cepat pada jaringan [2]. Kemampuan belajar yang dimilikinya dapat dilatih untuk mempelajari dan menganalisa pola data masa lalu dan berusaha mencari suatu formula atau fungsi yang akan menghubungkan pola data masa lalu dengan keluaran yang diinginkan pada saat ini [3].

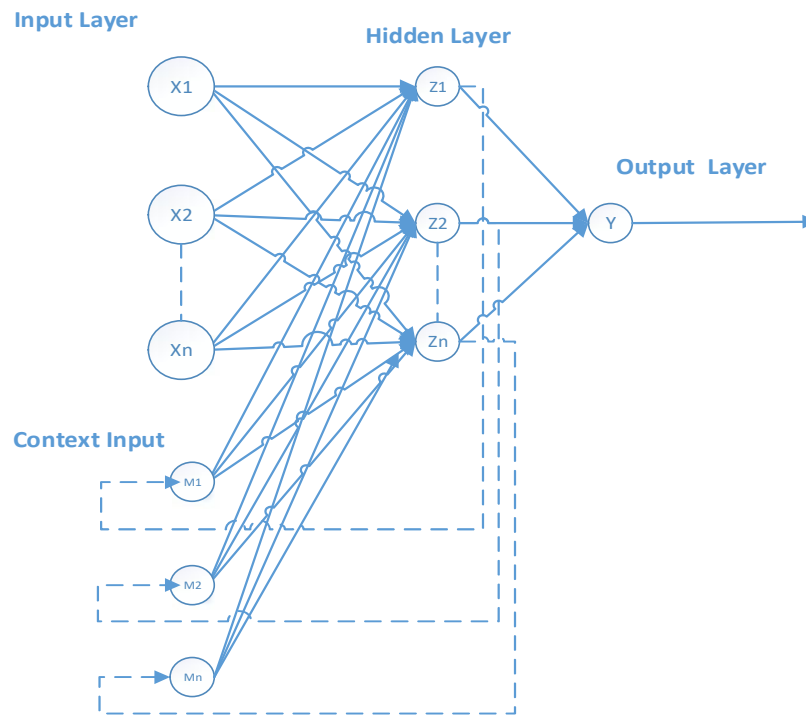
## 2. Metode Penelitian

### 2.1 Arsitektur ERRN (*Elman Recurrent Neural Network*)

*Recurrent Neural Network* adalah salah satu bagian dari model *artificial neural network* yang mempunyai *feedback* dari keluaran *hidden Layer* ke masukan *input Layer* . *Recurrent Neural Network* mempunyai struktur dan algoritma pelatihan yang lebih kompleks dibandingkan *feed-forward neural network*. Pada *Recurrent Neural Network*, *output* dari *network* digunakan kembali sebagai *input network*. *Elman Recurrent Neural* disebut *partial recurrent neural network* karena *recurrent weight* adalah tetap [4]. *Network* ditambahkan di *layer context* sebagai tambahan *layer* proses. Biasanya *network* ini merupakan *feedforward neural network*. Pelatihan dilakukan pada hubungan *forward*. Sedangkan hubungan *backward* dari *output Layer* ke *input Layer* tidak dapat dilakukan pelatihan. *Layer context* digunakan untuk mengingat status terakhir dari *hidden layer* . *Output* dari *network* tergantung dari status sebelum maupun status *network* pada saat ini. Kemampuan dalam mengingat status terakhir menjadikan *network* ini memiliki memori yang dinamis. Arsitektur *Elman Neural Network* hampir sama dengan arsitektur *feedforward backpropagation*, namun ditambah dengan *layer context* untuk menampung hasil *output* dari *hidden layer* . *Layer* akan di *update* tidak hanya pada *input* jaringan saja tapi juga dengan aktivasi *forward propagation* sebelumnya.

*Elman Recurrent Neural Network* merupakan variasi dari *Multi Layer Perceptron*. Akan tetapi pada *Elman Recurrent Neural Network* terdapat beberapa *node* yang posisinya berdekatan dengan *input layer* yang berhubungan dengan *hidden layer*. *Node-node* tersebut mengandung isi dari salah satu *layer* yang telah dilatih sebelumnya. Pada prinsipnya, masukan disebarkan secara *feed forward* yang kemudian diberikan suatu *learning rule*. Jaringan jenis dapat memelihara suatu urutan keadaan dan mengijinkannya untuk melakukan beberapa pekerjaan sekaligus, contohnya seperti *sequence prediction* yang berada diluar kemampuan *Multi Layer Perceptron* [4]. Masukan tidak hanya nilai dari luar jaringan, tetapi ditambah dengan nilai keluaran dari *neuron* tersembunyi dari propagasi sebelumnya. Gambar 1 merupakan struktur *Elman Recurrent Network*.



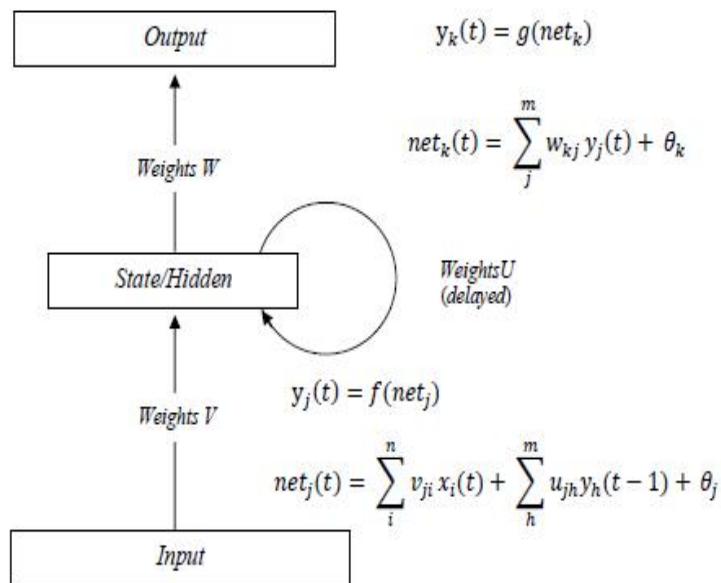


**Gambar 1.** Arsitektur *Elman Recurrent Neural Network*

Model jaringan syaraf tiruan untuk prediksi pemakaian dan penggunaan *bandwidth* menggunakan struktur jaringan syaraf tiruan 3 layer /lapisan. Struktur jaringan syaraf tiruan 3 layer terdiri dari sebuah *layer input*, sebuah *layer hidden*, sebuah *layer output* dan *context input*.

**2.2 Algoritma Elman Recurrent Neural Network**

Jaringan *Elman vector input* akan dipropagasikan melalui *layer bobot* dan akan dikombinasikan dengan aktifasi sebelumnya memlaui *layer bobot recurrent* tambahan yaitu  $U$  seperti pada Gambar 2 dimana  $U$  adalah bobot delay.



**Gambar 2.** Jaringan Elman

Setiap kali sebuah pattern disajikan, unit akan menghitung aktifasinya seperti halnya pada jaringan *feedforward*. Adapun tahapan dalam algoritma *JST Recurrent Elman* adalah sebagai berikut :

**Tahap 1 :** *Training* pola input atau vector saat time ( $t$ ) seperti persamaan (1)

$$y_t(t) = f(n_j) \quad (1)$$

dengan jaringan *recurrent* seperti pada persamaan (2)

$$n_j(t) = \sum_i^n v_j x_i(t) + \sum_h^m u_{jh} y_h(t-1) + \theta_j \quad (2)$$

**Tahap 2 :** *Output* jaringan ditentukan oleh lapisan *context* dan satu set *output* berbobot ( $w$ ) seperti persamaan (3), yang kemudian dapat dijabarkan menjadi persamaan (4).

$$y_k(t) = g(n_k(t)) \quad (3)$$

$$n_k(t) = \sum_j^m w_k y_j(t) \theta_k \quad (4)$$

**Tahap 3 :** *Back propagation* untuk jaringan *recurrent*, Setiap bobot dimodifikasi, dimana fungsi biaya (kesalahan) sehubungan dengan bobot dihitung dan kemudian disesuaikan. Fungsi biaya yang digunakan dalam penelitian ini adalah *Mean Square Error* (MSE) pada persamaan (5)

$$M = \frac{1}{n} \sum_p^n \sum_k^m (d_p - y_p)^2 \quad (5)$$

keterangan :

$d$  : ouput yang diinginkan

$n$  : jumlah total sampel

$m$  : jumlah *node output*

**Tahap 4 :** Menurut *gradient descent*, setiap perubahan bobot dalam jaringan harus sebanding dengan *gradient* negatif dari biaya sehubungan dengan tertentu yang tertarik untuk dimodifikasi.  $\eta$  adalah sebuah *learning rate* seperti pada persamaan (6).

$$\Delta w = -\eta \frac{\partial}{\partial} \quad (6)$$

**Tahap 5 :** Menggunakan aturan rantai dari diferensiasi, bobot berkaitan dengan *error* pada *output* dapat dinyatakan dengan persamaan (7).

$$\delta_p = -\frac{\partial}{\partial y_p} \frac{\partial y_p}{\partial n_p} = (d_p - y_p) g'(y_p) \quad (7)$$

Sedangkan *error* pada *hidden layer* dinyatakan dengan persamaan (8)

$$\delta_p = -\left( \sum_k^m \frac{\partial}{\partial y_p} \frac{\partial y_p}{\partial n_p} \frac{\partial n_p}{\partial y_p} \right) \frac{\partial y_p}{\partial n_p} = \sum_{k=1}^m \delta_p w_k f'(y_p) \quad (8)$$

Dimana  $m$  adalah *neuron* pada lapisan *output*.

**Tahap 6 :** Perubahan bobot untuk *output* dinyatakan dengan persamaan (9). Sedangkan persamaan (10) untuk perubahan bobot *input*.

$$\Delta w_k = \eta \sum_p^n \delta_p y_p \quad (9)$$

$$\Delta v_j = \eta \sum_p^n \delta_p y_p \quad (10)$$

**Tahap 7 :** Berdasarkan komponen waktu, perubahan bobot *recurrent* dapat dinyatakan seperti persamaan (11).

$$\Delta u_{jh} = \eta \sum_p^n \delta_p (t) y_{ph}(t-1) \quad (11)$$

**Tahap 8 :** Pelatihan akan berhenti apabila  $error < target\ error$ .

## 2.3 Teori Prediksi

Secara umum pengertian prediksi adalah tafsiran. Namun dengan menggunakan teknik-teknik tertentu maka prediksi bukan hanya sekedar tafsiran. Ada beberapa definisi tentang prediksi, diantaranya :

1. Prediksi diartikan sebagai penggunaan teknik-teknik statistik dalam bentuk gambaran masa depan berdasarkan pengolahan angka-angka historis.
2. Prediksi merupakan bagian integral dari kegiatan pengambilan keputusan manajemen.
3. Prediksi adalah peramalan, rencana, atau estimasi kejadian masa depan yang tidak pasti. Selain itu prediksi juga dapat diartikan sebagai penggunaan teknik-teknik statistik dalam membentuk gambaran masa depan berdasarkan pengolahan angka-angka historis.

Metode prediksi merupakan cara memperkirakan apa yang akan terjadi pada masa depan secara sistematis dan pragmatis atas dasar data yang relevan pada masa yang lalu, sehingga dengan demikian metode prediksi diharapkan dapat memberikan objektivitas yang lebih besar. Selain itu metode prediksi dapat memberikan cara pengerjaan yang teratur dan terarah, dengan demikian dapat dimungkinkannya penggunaan teknik penganalisaan yang lebih maju. Dengan penggunaan teknik-teknik tersebut maka diharapkan dapat memberikan tingkat kepercayaan dan keyakinan yang lebih besar, karena dapat diuji penyimpangan atau deviasi yang terjadi secara ilmiah.

### 2.3.1 Jenis Prediksi

Berdasarkan sifatnya, prediksi dibedakan atas dua macam yaitu :

- a. Prediksi kualitatif adalah prediksi yang didasarkan atas pendapat suatu pihak, dan datanya tidak bisa direpresentasikan secara tegas menjadi suatu angka atau nilai. Hasil prediksi yang dibuat sangat bergantung pada orang yang menyusunnya. Hal ini penting karena hasil prediksi tersebut ditentukan berdasarkan pemikiran yang instuisi, pendapat dan pengetahuan serta pengalaman penyusunnya.
- b. Prediksi kuantitatif adalah prediksi yang didasarkan atas data kuantitatif masa lalu (data historis) dan dapat dibuat dalam bentuk angka yang biasa disebut sebagai data *time series*

Hasil prediksi yang dibuat sangat bergantung pada metode yang dipergunakan dalam prediksi tersebut. Baik tidaknya metode yang dipergunakan ditentukan oleh perbedaan atau penyimpangan antara hasil ramalan dengan kenyataan yang terjadi. Semakin penyimpangan antara hasil ramalan dengan kenyataan yang akan terjadi maka semakin baik pula metode yang digunakan.

### 2.3.2 Pengukuran Prediksi

Teknik prediksi tidak selamanya selalu tepat karena teknik prediksi yang digunakan belum tentu sesuai dengan sifat datanya, atau disebabkan oleh kondisi di luar bisnis yang mengharuskan bisnis itu menyesuaikan diri. Oleh karena itu, perlu diadakan pengawasan prediksi sehingga dapat diketahui sesuai atau tidaknya teknik prediksi yang digunakan. Sehingga dapat dipilih dan ditentukan teknik prediksi yang lebih sesuai dengan cara menentukan batas toleransi prediksi atas penyimpangan yang terjadi.

Pada prinsipnya, pengawasan prediksi dilakukan dengan membandingkan hasil prediksi dengan kenyataan yang terjadi. Penggunaan teknik prediksi yang menghasilkan penyimpangan terkecil adalah teknik prediksi yang paling sesuai untuk digunakan.

Besarnya *error* prediksi dihitung dengan mengurangi data riil dengan besarnya ramalan pada persamaan (12).

$$Error(E) = X_t - F_t \quad (12)$$

Keterangan

$X_t$  = Data riil periode ke- $t$

$F_t$  = Ramalan Periode ke- $t$

## 2. 4 Proses Jaringan Syaraf Tiruan

Penerapan jaringan syaraf tiruan dalam permasalahan prediksi penggunaan *bandwidth* meliputi penentuan komponen-komponen jaringan syaraf tiruan yang digunakan dengan langkah-langkah sebagai berikut [5] :

1. Menentukan variabel masukan.
2. Jumlah masukan ditentukan berdasarkan data jumlah penggunaan *bandwidth* di Universitas Pattimura.
3. Menentukan jumlah *neuron* pada *hidden layer* dan *context layer* .
4. Banyaknya jumlah *neuron* yang digunakan pada *hidden layer* dan *context layer* akan menentukan kinerja sistem ketika memprediksi penggunaan *bandwidth*. Hal ini disebabkan karena semakin banyak *neuron* dalam *hidden layer* dan *context layer* maka akan membuat *neuron* tersebut memiliki beban yang semakin sedikit dalam menampung data-data yang harus dipelajari. *Hidden layer* dan *context layer* ditentukan secara *trial* dan *error* sampai didapatkan hasil yang terbaik.
5. Menentukan parameter pembelajaran, yaitu maksimum *epoch*, toleransi *error* dan *learning rate*.
6. Menentukan fungsi bobot yang akan digunakan berdasarkan fungsi aktivasi. Penentuan nilai bobot minimum dan maksimum ditentukan atas dasar  $0 < n < 1$  yang merupakan jangkauan dari fungsi. Fungsi ini digunakan untuk jaringan syaraf tiruan yang dilatih dengan menggunakan metode *Elman Recurrent*. Fungsi *sigmoid biner* mempunyai nilai antara 0 sampai 1. Fungsi ini sering digunakan jaringan syaraf tiruan yang membutuhkan nilai *output* yang terletak pada interval 0 sampai 1.
7. Menentukan *output*.

Nilai *output* ditentukan berdasarkan hasil yang diinginkan yaitu nilai berkisar antara 0 sampai 1 yang merepresentasikan nilai prediksi *bandwidth*, yang selanjutnya dinormalisasikan supaya menjadi nilai prediksi aktual.

## 3. Hasil dan Pembahasan

### 3.1 Analisis Preprocessing Data

*Preprocessing* data bertujuan untuk memilih data yang digunakan sebagai data penelitian, namun sebelumnya data tersebut di normalisasi terlebih dahulu. Dimana hasil normalisasi data tersebut selanjutnya di representasikan kedalam *sliding windows* dengan *windows size* sebesar lima. Proses *sliding windows* mengakibatkan data penelitian di reduksi karena proses penentuan data awal menjadi berubah. Total data

penelitian menjadi 82 vektor data, pada Tabel 1 menunjukkan contoh penerapan *windows size* pada data penelitian.

**Tabel 1. Data penelitian dengan *windows size* 5**

$x_1'$	$x_2'$	$x_3'$	$x_4'$	$x_5'$	$y'$
0.313	0.246267	0.291533	0.245333	0.207067	0.2968
0.246267	0.291533	0.245333	0.207067	0.2968	0.252533
0.291533	0.245333	0.207067	0.2968	0.252533	0.142733
0.245333	0.207067	0.2968	0.252533	0.142733	0.3354
0.207067	0.2968	0.252533	0.142733	0.3354	0.2636
...	...	...	...	...	...

Data penelitian dengan *sliding windows* 5 selanjutnya dikelompokkan menjadi dua kelompok, yaitu data *training* sebanyak 80% dari keseluruhan data dan data *testing* sebanyak 20% dari data penelitian. Dari data keseluruhan yang berjumlah 82 vektor data maka didapatkan data yang menjadi data *training* set adalah sejumlah 67 data dan data yang menjadi data *testing* set sejumlah 15 data.

### 3.2 Training dengan nilai *window size*

Percobaan pertama untuk mendapatkan nilai *windows size* yang tepat yang menjadi parameter pengujian agar didapatkan hasil yang memiliki tingkat akurasi yang tinggi. Terdapat beberapa percobaan yang dilakukan, dan hasil yang dijadikan sebagai acuan awal adalah jumlah *epoch* maksimal yaitu 100.000 dan 1.000.000. Pelatihan akan berhenti jika telah mendapatkan  $MSE < target\ error$  atau telah mencapai maksimum *epoch* yang ditentukan. Parameter *training* untuk pengujian nilai *windows size* dapat dilihat pada Tabel 2.

**Tabel 2. Parameter *training* nilai *windows size* dengan *epoch* 100.000**

Parameter	Nilai	Keterangan
Target error (MSE)	0,001	Target proses berhenti
Learning rate	0,1	Kecepatan pembelajaran sistem
Jumlah Hidden	3	Jumlah lapisan tersembunyi
Maksimum epoch	100.000	Jumlah iterasi
Transfer Function	Sigmoid Binner	
Bobot	Random (0,1)	

Hasil training dari percobaan ini adalah nilai dari *windows size* yang digunakan untuk mencari nilai MSE terkecil dengan *epoch* proses yang telah ditentukan. Nilai MSE terkecil yang didapatkan adalah 0.003277 dengan jumlah epoch sebanyak 100.000 *epoch*, hasil ini didapatkan pada nilai *input/windows size* delapan.

### 3.3 Training dengan jumlah neuron pada *hidden layer*

Jumlah *neuron* sangat mempengaruhi pada keakuratan hasil prediksi terhadap data yang digunakan dalam jaringan *Elman Recurrent Neural Network*. Jumlah *neuron* pada *hidden layer* yang cocok digunakan untuk prediksi *bandwidth* diperoleh melalui proses percobaan dengan melakukan beberapa kali *training*. Sebagai acuan awal *learning rate* yang digunakan adalah 0.01 dan 0.02. Target error adalah 0,001 dengan *epoch* maksimal sebanyak 1.000.000 dimana proses akan berhenti apabila nilai fungsi kinerja kurang dari atau sama dengan target error yang telah ditentukan atau telah mencapai nilai maksimum *epoch*.

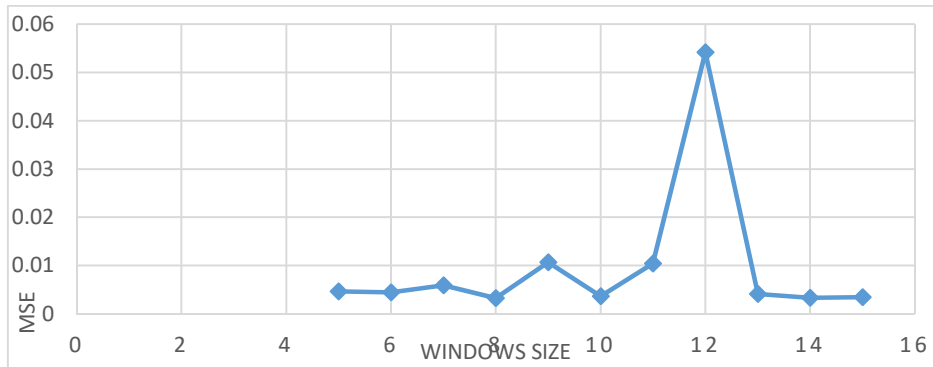
**Tabel 3. Parameter pengujian jumlah neuron dengan target error 0.001**

Parameter	Nilai	Keterangan
Target error (MSE)	0,001	Target proses berhenti
Learning rate	0,1	Kecepatan pembelajaran sistem
Jumlah Hidden	3,5,7,10,15,20, 25, 30	Jumlah lapisan tersembunyi
Maksimum epoch	100.000	Jumlah iterasi
Transfer Function	Sigmoid Binner	
Bobot	Random (0,1)	

Hasil percobaan yang didapat dengan menggunakan jumlah *neuron* yang berbeda-beda pada *hidden layer*, hasil yang didapat menunjukkan bahwa dengan jumlah *neuron* sebanyak tigabelas unit menghasilkan nilai *error* (MSE) terkecil dari seluruh percobaan yaitu sebesar 0.003725. Proses berikutnya adalah menguji hasil training dengan menggunakan jumlah *neuron* yang berbeda-beda pada tiap-tiap percobaan.

### 3.4 Analisis hasil training dengan *windows size*

Proses *training* yang dilakukan dengan menggunakan parameter *windows size* menunjukkan bahwa dengan jumlah *node input* 8 menghasilkan nilai MSE terkecil yaitu 0.003277. Berdasarkan Tabel 3 dibuatlah grafik perbandingan antara jumlah *neuron input* terhadap nilai MSE seperti pada Gambar 2.

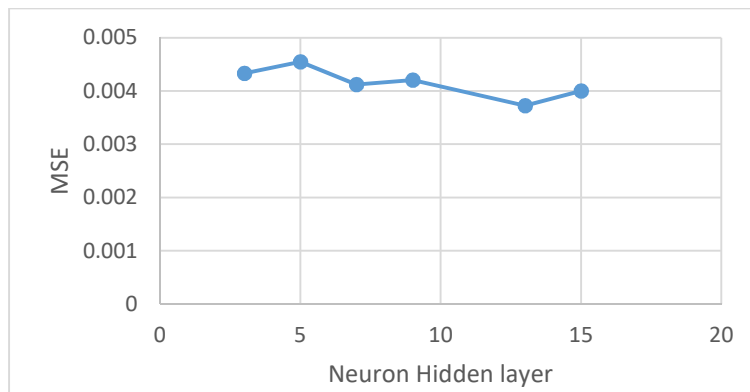


Gambar 2. Grafik perbandingan *windows size* dan nilai MSE (target 0.001)

Pada saat *training* dengan target 0.001 nilai *epoch* maksimum yang ditetapkan adalah 100.000 nilai MSE mengalami peningkatan pada saat jumlah *neuron input* 12 yaitu 0.054172.

### 3.5 Training dengan jumlah *hidden layer*

Pengujian dengan menggunakan parameter jumlah *hidden Layer* dengan parameter yang ditunjukkan pada Tabel 2. Berdasarkan Tabel 2 dibuatlah grafik perbandingan antara jumlah *windows size* terhadap nilai MSE seperti pada Gambar 3.



Gambar 3. Grafik perbandingan *neuron hidden layer* dan nilai MSE (target 0.001)

Pada saat *training* dengan *learning rate* 0.01 diperoleh konfigurasi terbaik dengan nilai MSE terkecil yaitu pada jumlah *neuron hidden layer* 13 yaitu sebesar 0.003725.

### 3.5 Analisis Hasil *Testing*

Pada penelitian ini data *testing* yang digunakan sebesar 20% dari total data yang digunakan, sehingga jumlah data yang digunakan sebanyak 15 vektor data. Data yang digunakan untuk *testing* merupakan data

baru yang tidak diikutsertakan dalam proses *training*. Tingkat akurasi hasil proses *testing* sangat dipengaruhi oleh bobot hasil *training*, yang menunjukkan kemampuan jaringan dalam mengenali pola-pola yang dilatih. Berdasarkan hasil percobaan yang dilakukan, dengan target toleransi *error* sebesar 0.001 dengan nilai masing-masing parameter sama dengan nilai yang terdapat pada proses *training* untuk jumlah *neuron hidden layer*.

Hasil *testing* dengan jumlah data sebanyak 15 data menghasilkan nilai MSE yang berbeda seperti yang ditunjukkan pada Tabel 4. Berdasarkan hasil *testing* diperoleh nilai MSE terkecil yaitu pada jumlah *neuron hidden layer* 13 yaitu sebesar 0.002422.

**Tabel 4. Hasil testing dengan target error 0.001**

<i>Neuron Hidden</i>	MSE
3	0.002513
5	0.002510
7	0.002526
9	0.002625
13	0.002422
15	0.002813

#### 4. Kesimpulan

Berdasarkan hasil penelitian dan pembahasan yang telah dilakukan, maka diperoleh kesimpulan sebagai berikut, sistem yang dikembangkan mampu mengenali pola dan dapat melakukan prediksi dalam hal penggunaan *bandwidth* dengan menggunakan metode jaringan saraf tiruan *elman*. Hasil *training* dengan menggunakan *windows zise* 8 pada maksimum *epoch* 100.000 diperoleh nilai MSE terkecil sebesar 0.003277. Hasil *training* dengan menggunakan *windows zise* 11 pada maksimum *epoch* 1.000.000 diperoleh nilai MSE terkecil sebesar 0.002833. Kemudian hasil *training* untuk jumlah *neuron* pada *hidden layer* diperoleh nilai MSE terkecil yaitu pada jumlah *neuron* 13 sebesar 0.003725. Hasil *testing* dengan menggunakan parameter pada percobaan pada jumlah *neuron hidden layer* 13 diperoleh nilai MSE terkecil yaitu sebesar 0.002422.

#### Daftar Pustaka

- [1] R. R. T. d. I. Muarifah, "Implementasi Mikrotik Sebagai Manajemen Bandwidth .," in *IST AKPRIND*, Yogyakarta, 2012.
- [2] T. L. H. F. Pasila, F, "Elman Neural Network Application With Accelerated Lma Training For East Java-Bali Electrical Load Time Series Data Forecasting," 2009.
- [3] I. G. R. a. S. A. Mekongga, "The Prediction of Bandwidth On Need Computer Network Through Artificial Neural network Method of Backpropagation," vol. 02, p. 98–107., 2012.
- [4] M. S. S. A. R. Aziz, "Enhancement of Particle Swarm Optimization in Elman Recurrent Network with bounded Vmax Function," 2009.
- [5] Y. Noorviani, "Penerapan Elman Recurrent Neural network untuk Diagnosis Gangguan Autis Pada Anak," *Tesis, Universitas Gadjah Mada Yogyakarta*, 2011.
- [6] O. Abdalla, "A comparison of feed-forward back-propagation and radial basis artificial neural networks," *A Monte Carlo study. In Proceedings 2010 International Symposium on Information Technology - Engineering Technology*, p. 994–998, 2010.







## KINERJA DIAGRAM KONTROL W DAN DIAGRAM KONTROL G

M. Stivo Noya Van Delsen<sup>1</sup>, Mozart Winston Talakua<sup>2</sup>

<sup>1,2</sup> Jurusan Matematika FMIPA UNPATTI

Jl. Ir. M. Putuhena, Kampus Unpatti, Poka-Ambon, Maluku

e-mail: <sup>1</sup>marlonnvd@gmail.com, <sup>2</sup>ocat1615@yahoo.com

---

### ABSTRAK

Pengontrolan terhadap variabilitas proses merupakan salah satu hal yang sangat penting dalam bidang industri. Sehingga, saat ini dikembangkan diagram kontrol yang khusus untuk memonitor variabilitas proses. Diagram kontrol multivariat yang digunakan untuk memonitor variabilitas proses antara lain diagram diagram kontrol W dan diagram kontrol G. Salah satu cara yang dapat digunakan untuk melihat kinerja diagram kontrol adalah dengan menggunakan *Average Run Length* (ARL). Sehingga pada penelitian ini akan dilihat kinerja dari diagram kontrol W dan diagram kontrol G, kemudian membandingkan kinerja diagram kontrol W dan G dengan menggunakan ARL. Hasil dari ARL menunjukkan bahwa nilai ARL yang diperoleh diagram kontrol G lebih kecil dari diagram kontrol W. Berdasarkan hasil tersebut dapat disimpulkan bahwa kinerja diagram kontrol G lebih baik dibandingkan dengan diagram kontrol W, namun untuk pergeseran varian yang besar diagram kontrol W dapat digunakan sebagai pendamping diagram kontrol G.

Kata Kunci : *Average Run Length*, Diagram Kontrol G, Diagram Kontrol W

## PERFORMANCE OF W CONTROL CHART AND G CONTROL CHART

### ABSTRACT

Monitoring of variability process is the important in industry. So, today has been developed multivariate control charts which are usefull for monitoring vaiabilty process. Multivariate control charts used to monitor variability process, among others W control chart and G control chart. Performance of control charts can be seen with *Average Run Length* (ARL). So in this study will be seen the performance of the W control chart and G control chart, and then compare the performance W control chart and G control chart by ARL. Result of the ARL shows that the value's G control chart is smaller than the W control chart. Based on the results can be conclude that the performance of G control chart is better than the W control chart, but for the large of variance shift the W control chart can be used with G control chart as companion.

Keywords : *Average run length*, control chart G, control chart W

---

### 1. Pendahuluan

Penerapan ilmu statistika hingga saat ini sangat berkembang pesat. Salah satu bidang yang merasakan dampak tersebut adalah bidang industri. Penerapan statistika dalam bidang industri lebih dikenal dengan statistika industri. Metode yang dapat digunakan dalam statistika industri adalah *Statistical Process Control* (SPC). SPC digunakan untuk menganalisis dan memperbaiki suatu proses dengan menggunakan diagram kontrol (*Control Chart*).

Diagram kontrol adalah sebuah grafik yang memberi gambaran tentang perilaku sebuah proses. Berdasarkan banyaknya karakteristik kualitas yang diamati, diagram kontrol terdiri dari diagram kontrol univariat dan diagram kontrol multivariat. Namun dalam mengamati atau memonitor perilaku sebuah proses banyaknya karakteristik kualitas yang digunakan lebih dari satu, ini mengakibatkan diagram kontrol multivariat lebih sering digunakan dibandingkan dengan diagram kontrol univariat. Selain itu, dalam memonitor proses dapat dilakukan melalui memonitor mean proses dan memonitor variabilitas proses [6].

Hingga saat ini, diagram kontrol multivariat untuk memonitor variabilitas proses terus berkembang. Diagram kontrol ini pertama kali dikembangkan oleh Alt (1985) yang mengembangkan diagram kontrol W berdasarkan *likelihood ratio test* W dan diagram kontrol |S| yang berdasar pada *sample generalized variance*. Kemudian pada tahun 2002 dikembangkan diagram kontrol G yang diperoleh dari perluasan konsep statistik G oleh Levinson, Holmes dan Mergen. Pada tahun 2005, Djauhari mengembangkan diagram kontrol |S| menjadi diagram kontrol *improved generalized sample* atau diagram kontrol *Impreved* |S|. Setelah itu Sindelar (2007) mengembangkan diagram kontrol berdasarkan matriks korelasi yang dikenal dengan diagram kontrol  $W_R$ .

Dalam perkembangannya kinerja diagram kontrol juga selalu menjadi perhatian khusus. Untuk melihat kinerja suatu diagram kontrol dapat menggunakan *Average Run Length* (ARL). Rakhmawati (2011) menggunakan nilai ARL untuk membandingkan kinerja diagram kontrol W dan diagram kontrol  $W_R$ . Pada penelitian tersebut dihasilkan bahwa diagram kontrol W mempunyai kinerja yang lebih baik dari diagram kontrol  $W_R$ . Kemudian pada tahun 2013, Noya Van Delsen dan Mashuri menggunakan nilai ARL untuk membandingkan diagram kontrol G dengan diagram kontrol |S|, yang menghasilkan bahwa kinerja diagram kontrol G lebih baik dibanding diagram kontrol |S|. Menindaklanjuti kedua penelitian tersebut, maka penelitian ini bertujuan untuk membandingkan kinerja diagram kontrol G dan diagram kontrol W dengan menggunakan ARL.

## 2. Tinjauan Pustaka

Tahun 1985 diketahui sebagai awal mula munculnya diagram kontrol multivariat untuk memonitor variabilitas proses oleh Alt dan Smith. Alt dan Smith mengembangkan diagram kontrol berdasarkan statistik *Likelihood Ratio Test* dan *Sample Generalized Variance*, atau yang lebih dikenal dengan diagram kontrol W dan diagram kontrol |S|. Diagram kontrol berikut dikembangkan berdasarkan pada statistik G oleh Levinson, Holmes dan Mergen (2002). Kemudian Djauhari (2005) mengembangkan diagram kontrol |S|, yang dikenal dengan diagram kontrol *Improved* |S| dan Sindelar (2007) mengembangkan diagram kontrol berdasarkan matriks korelasi atau diagram kontrol  $W_R$ .

Menindaklanjuti penelitian Sindelar, Rakhmawati (2011) membandingkan kinerja diagram kontrol W dengan kinerja diagram kontrol  $W_R$ . Kemudian Noya Van Delsen dan Mashuri (2013) membandingkan kinerja diagram kontrol G dengan diagram kontrol *Improved* |S|.

### 2.1 Diagram Kontrol G

Levinson, Holmes dan Mergen (2002) mengembangkan diagram kontrol multivariat untuk mengontrol variabilitas proses dengan berdasar pada statistik G, yang disebut diagram kontrol G. Levinson, Holmes dan Mergen memperluas statistik G untuk membuat diagram kontrol G. Diagram kontrol G dikhususkan untuk mengontrol variabilitas proses multivariat pada pengamatan subgroup.

Untuk mendeskripsikan diagram kontrol G dari masing-masing subgroup ke- $m$ , dengan ukuran masing-masing subgroup  $n$ , Levinson, Holmes dan Mergen (2002) mendeskripsikan diagram kontrol G sebagai berikut:

$$G = M \times m = \ln \left[ \frac{|S|^{v_1+v_2}}{|S_1|^{v_1} |S_{2,i}|^{v_2}} \right] \times \left( 1 - \left[ \frac{1}{v_1} + \frac{1}{v_2} - \frac{1}{v_1+v_2} \right] \times \left[ \frac{2p^2+3p-1}{6(p+1)} \right] \right)$$

dimana

$$S = \frac{v_1 S_1 + v_2 S_{2,i}}{v_1 + v_2} \quad S_{2,i} = \begin{bmatrix} s_{i,1}^2 & s_{i,12} & \cdots & s_{i,1j} & \cdots & s_{i,1p} \\ & s_{i,2}^2 & \cdots & s_{i,2j} & \cdots & s_{i,2p} \\ & & \ddots & \vdots & \vdots & \vdots \\ & & & s_{i,j}^2 & \cdots & s_{i,jp} \\ & & & & \ddots & \vdots \\ & & & & & s_{i,p}^2 \end{bmatrix}; \quad \begin{matrix} i = 1, 2, \dots, m \\ j = 1, 2, \dots, p \end{matrix}$$

$$S_1 = \begin{bmatrix} \frac{1}{m} \sum_{i=1}^m S_{i,1}^2 & \frac{1}{m} \sum_{i=1}^m S_{i,12} & \dots & \frac{1}{m} \sum_{i=1}^m S_{i,1j} & \dots & \frac{1}{m} \sum_{i=1}^m S_{i,1p} \\ & \frac{1}{m} \sum_{i=1}^m S_{i,2}^2 & \dots & \frac{1}{m} \sum_{i=1}^m S_{i,2j} & \dots & \frac{1}{m} \sum_{i=1}^m S_{i,2p} \\ & & \ddots & \vdots & \vdots & \vdots \\ & & & \frac{1}{m} \sum_{i=1}^m S_{i,j}^2 & \dots & \frac{1}{m} \sum_{i=1}^m S_{i,jp} \\ & & & & \ddots & \vdots \\ & & & & & \frac{1}{m} \sum_{i=1}^m S_{i,p}^2 \end{bmatrix} = \frac{1}{m} \sum_{i=1}^m S_{2,i}$$

dengan,

$$s_{i,j}^2 = \frac{1}{n-1} \sum_{k=1}^n (x_{ijk} - \bar{x}_{ij})^2 \qquad s_{i,jj^*} = \frac{1}{n-1} \sum_{k=1}^n (x_{ijk} - \bar{x}_{ij})(x_{ij^*k} - \bar{x}_{ij^*}); j \neq j^*$$

$$v_1 = m(n-1) \qquad v_2 = n-1$$

keterangan

$p$  = Banyaknya karakteristik kualitas

$m$  = Banyaknya subgrup

$n$  = Banyaknya data pada setiap subgrup

$S_{2,i}$  = Matriks kovarian subgrup ke- $i$

$S_1$  = Rata-rata dari matriks kovarian subgrup

$v_1$  = Derajat bebas untuk  $S_1$

$v_2$  = Derajat bebas untuk  $S_{2,i}$

$S_{i,j}^2$  = Variansi variabel ke- $j$  pada subgrup ke- $i$

$S_{i,jj^*}$  = Kovariansi antara variabel ke- $j$  dengan variabel ke- $j^*$

$x_{ijk}$  = Merupakan data ke- $k$  variabel ke- $j$  pada subgrup ke- $i$

$\bar{x}_{ij}$  = Merupakan rata-rata variabel ke- $j$  pada subgrup ke- $i$

Menurut Kramer dan Janses (1969) statistik G berdistribusi chi-square dengan derajat kebebasan  $\frac{p(p+1)}{2}$ . Sehingga diagram kontrol G mengikuti distribusi chi-square dengan derajat kebebasan  $\frac{p(p+1)}{2}$ .

Maka batas kontrol untuk diagram kontrol G adalah :

$$[LCL, UCL] = \left[ t_{\frac{p(p+1)}{2}, \frac{\alpha}{2}}^2, t_{\frac{p(p+1)}{2}, 1-\frac{\alpha}{2}}^2 \right]$$

dengan  $\alpha$  adalah taraf kepercayaan yang diinginkan. Menurut Levinson, Holmes dan Mergen (2002) jika statistik G yang diperoleh berada di atas UCL atau di bawah LCL, maka dapat disimpulkan proses produksi tidak terkendali.

Diagram kontrol G digunakan secara khusus untuk mengontrol apakah terjadi perubahan dalam matriks kovarian. Perubahan proses pada matriks kovarian ditunjukkan dengan adanya pengamatan yang berada di luar batas kontrol oleh diagram kontrol G. Sehingga mengakibatkan proses produksi tidak terkontrol.

## 2.2 Diagram Kontrol W

Variabilitas proses multivariat dapat dinyatakan dalam bentuk matriks kovariansi  $S$  berukuran  $p \times p$ , dimana elemen diagonal utama merupakan nilai variabilitas dari masing-masing karakteristik kualitas dan elemen selain diagonal utama merupakan nilai kovariansi antar karakteristik kualitas. Terdapat dua komponen yang dapat digunakan untuk mengukur variabilitas total dari sekumpulan data multivariat. Komponen yang pertama adalah determinan matriks kovariansi ( $|S|$ ) yang biasa disebut dengan *generalized variance*. Operasi determinan dipilih karena pada operasi determinan selain melibatkan diagonal utama juga melibatkan elemen selain diagonal utama. Akar kuadrat dari komponen pertama ini proporsional dengan luasan atau volume yang dibangkitkan oleh data. Komponen yang kedua adalah *trace* dari matriks kovariansi ( $\text{tr}(S)$ ) yang merupakan jumlahan dari variabilitas masing-masing karakteristik kualitas.

Diagram kontrol ini dikembangkan oleh Alt pada tahun 1985 dengan mengasumsikan bahwa matriks kovariansi sebenarnya,  $\Sigma$ , tidak diketahui (diestimasi dari sampel *in control* yang besar), perbandingan dibuat antara matriks kovariansi sampel dari proses dan matriks kovariansi yang tidak diketahui. Perbandingan tersebut adalah serangkaian uji signifikansi dalam bentuk sebagai berikut (Alt, 1985):

$$H_0 : \Sigma = \Sigma_0$$

$$H_1 : \Sigma \neq \Sigma_0$$

dimana  $\Sigma$  dan  $\Sigma_0$  adalah matriks kovariansi populasi dan matriks kovariansi *in control*. Statistik tersebut adalah rasio dari estimator maximum likelihood untuk distribusi  $N_p(\bar{x}, \Sigma)$  dalam bentuk sebagai berikut :

$$\Omega = \{(\bar{x}, \Sigma) : -\infty < \bar{x} < \infty, \Sigma \text{ adalah definit positif}\}$$

$$\check{S} = \{(\bar{x}, \Sigma) : -\infty < \bar{x} < \infty, \Sigma = \Sigma_0\}$$

Hal ini telah dijelaskan (Anderson, 1984) bahwa estimator maximum likelihood untuk kasus multivariat normal adalah sebagai berikut :

$$\hat{\bar{x}}_{\Omega} = \bar{x} \quad \hat{\Sigma}_{\Omega} = \frac{1}{n} A_i$$

$$\hat{\bar{x}}_{\check{S}} = \bar{x} \quad \hat{\Sigma}_{\check{S}} = \Sigma_0$$

dimana  $\bar{x}$  adalah vektor *mean*,  $n$  adalah ukuran sampel, dan  $A_i = (n-1)S_i$  sehingga fungsi likelihood berdasar pada distribusi normal multivariat adalah

$$L(\hat{\Omega}) = \left(\frac{2f}{n-1}\right)^{\frac{pn}{2}} |A_i|^{\frac{n}{2}} e^{-\frac{pn}{2}}$$

$$L(\check{S}) = (2f)^{\frac{pn}{2}} |\Sigma_0|^{\frac{n}{2}} e^{-\frac{1}{2}\text{tr}(\Sigma_0^{-1} A_i)}$$

Rasio likelihoodnya adalah sebagai berikut :

$$\Lambda(x) = \left(\frac{1}{n-1}\right)^{\frac{pn}{2}} e^{\frac{pn}{2}} \left|\Sigma_0^{-1} A_i\right|^{\frac{n}{2}} e^{-\frac{1}{2}\text{tr}(\Sigma_0^{-1} A_i)}$$

Rasio log likelihoodnya adalah sebagai berikut :

$$W_i = -2 \ln(\Lambda) = -pn + pn \ln(n) - n \ln\left(\left|\Sigma_0^{-1} A_i\right|\right) + \text{tr}\left(\Sigma_0^{-1} A_i\right)$$

Statistik uji tersebut dihitung dan digambarkan untuk setiap sampel ke- $i$ . dimana  $i = 1, 2, \dots, n$ , yang akan

ditolak jika nilainya lebih dari  $t_{\left(r, \frac{p(p+1)}{2}\right)}^2$ . Jika plot nilai  $W_i$  lebih dari *upper control limit* (UCL)  $t_{\left(r, \frac{p(p+1)}{2}\right)}^2$ , maka proses dapat dikatakan tidak terkontrol.

### 2.3 Average Run Length (ARL)

ARL merupakan rata-rata pengamatan yang harus diplot pada diagram kontrol sebelum sampai terindikasi kondisi *out of control*. Berdasarkan definisi ini maka ARL berfungsi untuk mengukur efektifitas kinerja diagram kontrol dalam mendeteksi perubahan pada suatu proses. Penentuan ARL dapat dilakukan dengan 3 pendekatan, yaitu integral, simulasi dan Rantai Markov (Wen Lu dan Reynolds, 2001). Nilai ARL terbagi menjadi 2 yaitu  $ARL_0$  (ARL *in control*) dan  $ARL_1$  (ARL *out of control*) dengan rumus sebagai berikut:

$$ARL_0 = \frac{1}{P(\text{Tolak } H_0 | H_0 \text{ benar})} = \frac{1}{r}$$

$$ARL_1 = \frac{1}{P(\text{Terima } H_0 | H_0 \text{ Benar})} = \frac{1}{1-s}$$

dengan  $H_0$  merupakan proses dalam kondisi *in control*. Sehingga  $\alpha$  disebut tipe kesalahan I pada uji hipotesis, yang berarti probabilitas memutuskan bahwa proses dalam kondisi *out of control* namun kenyataannya proses dalam kondisi *in control*. Sedangkan  $s$  adalah tipe kesalahan II pada uji hipotesis, yang berarti probabilitas memutuskan bahwa proses dalam kondisi *in control* namun pada kenyataannya proses dalam kondisi *out of control*, sehingga  $1-s$  adalah probabilitas yang memutuskan bahwa proses dalam kondisi *out of control* sebagai proses yang *out of control*. Maka  $ARL_0$  dapat diartikan sebagai rata-rata titik pengamatan yang harus diplot sampai ditemukannya pengamatan yang *out of control*, pada saat proses berada pada kondisi *in control*. Sedangkan  $ARL_1$  dapat diartikan sebagai rata-rata titik pengamatan yang diplot sampai ditemukannya pengamatan yang *out of control* pada proses dalam kondisi *out of control*. Nilai ARL yang diperoleh dengan pendekatan hasil simulasi yang didefinisikan sebagai nilai rata-rata *Run Length* (RL) dari semua replikasi dalam simulasi. RL merupakan jumlah titik pengamatan hingga ditemukannya *out of control* yang pertama untuk masing-masing replikasi. Sehingga jika nilai ARL yang dihasilkan semakin kecil, maka sampel yang dibutuhkan untuk memberikan sinyal perubahan proses pun semakin kecil atau dengan kata lain semakin kecil nilai ARL maka semakin cepat pula diagram diagram kontrol dapat mendeteksi peubahan proses, sehingga diagram kontrol tersebut semakin efektif untuk mendeteksi perubahan proses.

## 3. Hasil dan Pembahasan

### 3.1 Perbandingan Kinerja Diagram Kontrol G dan W Berdasarkan ARL

Kinerja diagram kontrol dapat diukur dari seberapa cepat diagram kontrol tersebut mendeteksi sinyal *out of control*. Sinyal *out of control* dapat dideteksi dengan menggunakan nilai ARL.

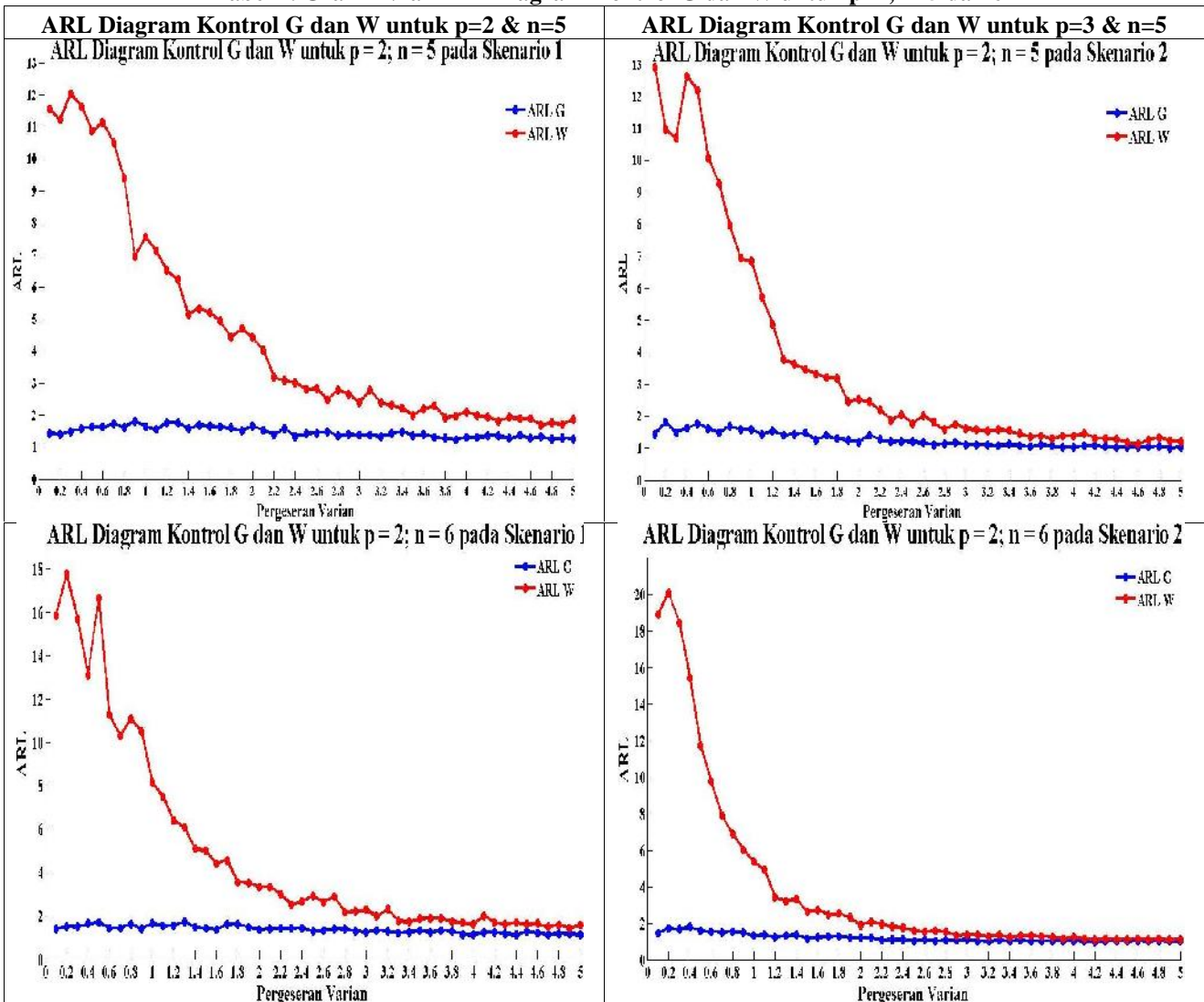
Pada penelitian ini akan digunakan karakteristik kualitas ( $p$ ) sebanyak 2, 3, dan 4, sedangkan ukuran subgroup ( $n$ ) sebanyak 5 dan 6, serta alfa sebesar 0,0027. Dalam rangka membantu mendeteksi sinyal *out of control* dibuat skenario untuk melihat kinerja diagram kontrol. Skenario yang dibuat yaitu melakukan pergeseran dalam matriks kovarian. Pergeseran dibuat pada nilai variansi dalam matriks kovarian. Sehingga skenario yang dibuat bergantung banyaknya karakteristik kualitas yang digunakan. Jika karakteristik kualitas yang digunakan sebanyak 2, maka matriks kovarian yang terbentuk berukuran  $2 \times 2$ , sehingga skenario yang dibuat adalah 2. Jika karakteristik kualitas yang digunakan sebanyak 3, maka matriks kovarian yang terbentuk berukuran  $3 \times 3$ , sehingga skenario yang dibuat adalah 3. Demikian juga hal ini berlaku untuk karakteristik kualitas sebanyak 4, maka skenario yang digunakan sebanyak 4.

**3.2 ARL Diagram Kontrol G dan W untuk  $p=2$  dan  $n=5$  dan 6.**

Berdasarkan hasil simulasi pada Tabel 1, dapat dilihat grafik nilai ARL diagram kontrol G dan diagram kontrol W untuk  $p=2$  dan  $n=5$ . Baik pada skenario 1 maupun skenario 2, nilai ARL diagram kontrol G lebih kecil dibandingkan nilai ARL diagram kontrol W. Bahkan untuk pergeseran varian yang sangat kecil sekalipun diagram kontrol G sangat baik dalam mendeteksi sinyal *out of control*. Sedangkan untuk diagram kontrol W nilai ARL yang dibutuhkan untuk pergeseran varian yang kecil sangatlah besar. Selain itu grafik nilai ARL diagram kontrol W yang ditunjukkan juga cenderung menurun tetapi nilai ARL-nya masih lebih besar dibandingkan dengan nilai ARL diagram kontrol G. Ini ditunjukkan dengan interval nilai ARL untuk diagram kontrol G yang berkisar 1,24 sampai 1,82 untuk skenario 1 dan 1,01 sampai 1,83 untuk skenario 2. Sedangkan nilai ARL untuk diagram kontrol W berkisar antara 1,7 sampai 12,05 untuk skenario 1 dan 1,13 sampai 12,9 untuk skenario 2.

Sedangkan untuk  $p=2$  dan  $n=6$  grafik menunjukkan bahwa nilai ARL untuk diagram kontrol G lebih statis dan lebih kecil dibanding ARL diagram kontrol W. Bahkan untuk pergeseran yang kecil nilai ARL diagram kontrol G yang dihasilkan jauh lebih kecil dibanding diagram kontrol W. Ini diperlihatkan oleh nilai ARL diagram kontrol G dan W yang masing-masing berkisar antara 1,12 sampai 1,72 dan 1,44 sampai 17,79 untuk skenario 1 sedangkan skenario 2 berkisar antara 1 sampai 1,79 dan 1,1 sampai 20,12. Sehingga untuk  $p=2$  dan  $n=6$ , dapat disimpulkan bahwa kinerja diagram kontrol G lebih baik dibanding diagram kontrol W.

**Tabel 1. Grafik Nilai ARL Diagram Kontrol G dan W untuk  $p=2, n=5$  dan 6**

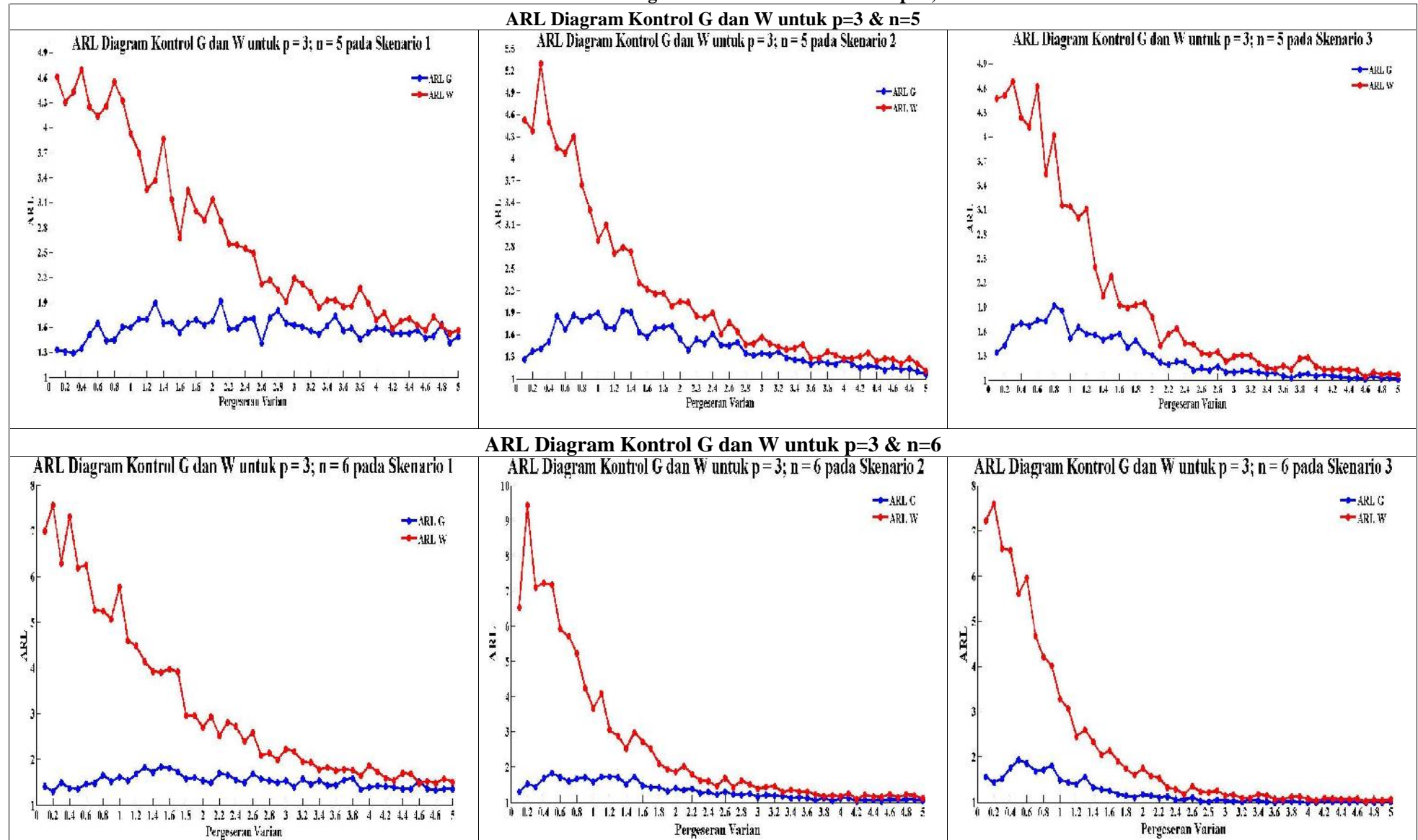


### 3.3 ARL Diagram Kontrol G dan W untuk $p=3$ dan $n=5$ dan 6.

Pada  $p=3$  dan  $n=5$  akan terdapat 3 skenario, demikian juga untuk  $p=3$  dan  $n=6$  akan terdapat 3 skenario. Dari hasil yang disajikan pada Tabel 2, maka grafik nilai ARL untuk  $p=3$  dan  $n=5$  menunjukkan bahwa diagram kontrol G mengalami fluktuatif dan cenderung menurun. Sedangkan diagram kontrol W cenderung menurun dari pergeseran varian yang kecil ke pergeseran varian yang besar. Selain itu dapat dilihat juga bahwa nilai ARL untuk diagram kontrol G juga lebih kecil dibanding diagram kontrol W. Ini ditunjukkan oleh kisaran nilai ARL untuk diagram kontrol G pada skenario 1, 2 dan 3 adalah 1,29 sampai 1,92; 1,07 sampai 1,93 dan 1,01 sampai 1,92. Sedangkan ARL untuk diagram kontrol W pada skenario 1, 2 dan 3 berkisar antara 1,53 sampai 4,7; 1,11 sampai 5,29 dan 1,04 sampai 4,68. Ini berarti bahwa pada  $p=3$  dan  $n=5$  kinerja diagram kontrol G lebih baik dibanding kinerja diagram kontrol W.



Tabel 2. Grafik Nilai ARL Diagram Kontrol G dan W untuk  $p=3, n=5$  dan 6



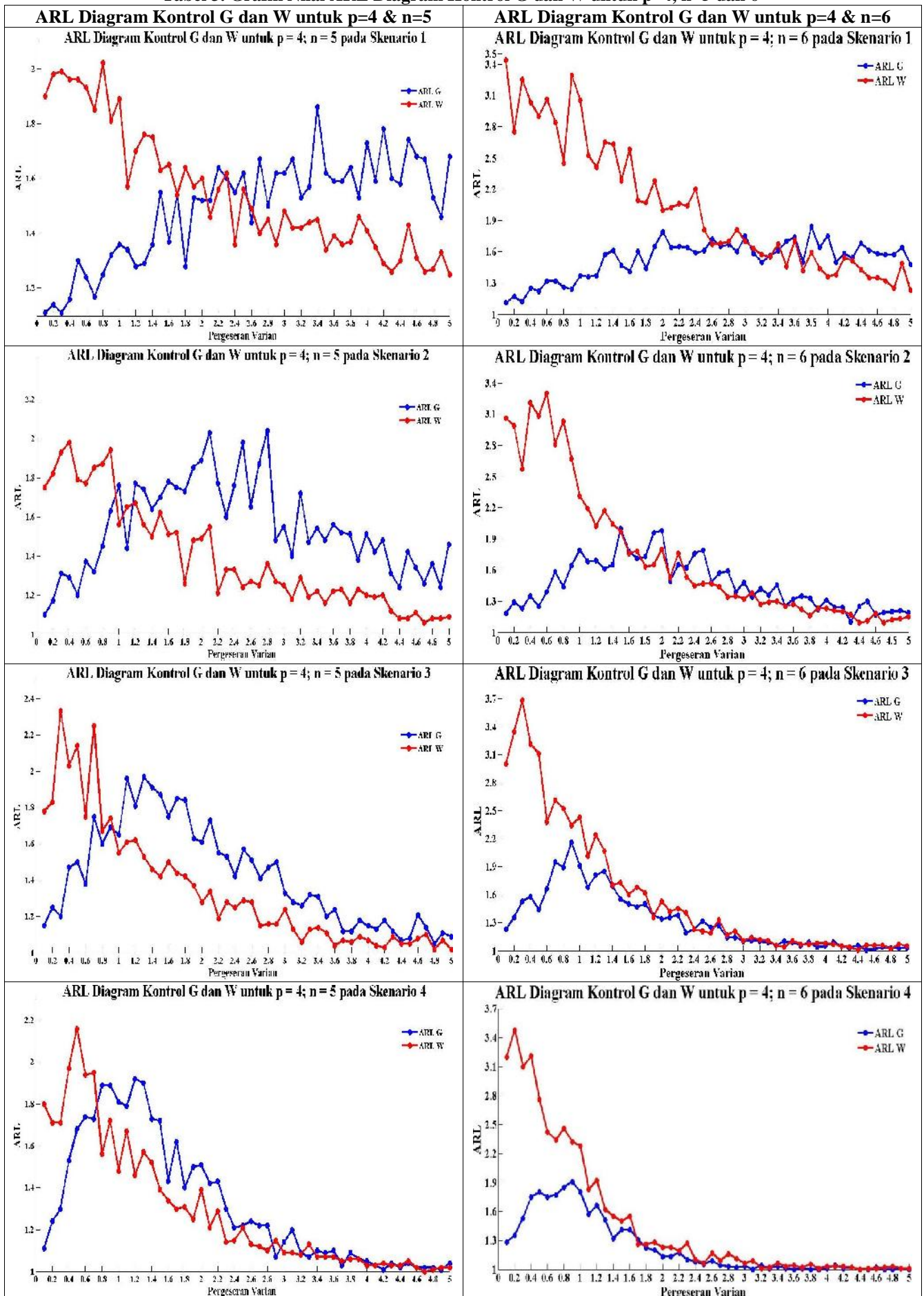
Sedangkan untuk hasil  $p=3$  dan  $n=6$  diperlihatkan juga dalam Tabel 2, yang menunjukkan nilai-nilai ARL untuk setiap pergeseran varian mempunyai pola yang sama dengan  $p=3$  dan  $n=5$ . Diagram kontrol G mampu dengan cepat mendeteksi pergeseran varian yang kecil, dan grafik nilai ARLnya juga mengalami fluktuatif serta cenderung statis. Sedangkan diagram kontrol W, grafik nilai ARL-nya juga mengalami fluktuatif dan cenderung menurun. Namun untuk pergeseran varian yang kecil, nilai ARL diagram kontrol W sangat besar jika dibanding dengan ARL diagram kontrol G. Selain itu dapat dilihat untuk setiap pergeseran dalam setiap skenario nilai ARL diagram kontrol G lebih kecil dibanding ARL diagram kontrol W. Ini berarti kinerja diagram kontrol G lebih baik dibanding diagram kontrol W untuk  $p=3$  dan  $n=6$ .

### 3.4 ARL Diagram Kontrol G dan W untuk $p=4$ dan $n=5$ dan 6.

Hasil dari grafik nilai ARL diagram kontrol G dan W  $p=4$  dan  $n=5$  dan 6 diperlihatkan pada Tabel 3, dengan jumlah skenario sebanyak 4 untuk  $n=5$  dan 4 skenario untuk  $n=6$ . Berdasarkan hasil untuk  $p=4$  dan  $n=5$  yang diperlihatkan pada skenario 1, 2, 3 dan 4, nilai ARL diagram kontrol G untuk pergeseran varian yang kecil masih lebih kecil dibanding diagram kontrol W. Sedangkan untuk pergeseran varian yang besar, nilai ARL diagram kontrol W masih lebih kecil dibanding ARL diagram kontrol G. Ini diperlihatkan dengan kisaran nilai ARL untuk masing-masing diagram kontrol pada seluruh pergeseran varian. ARL diagram kontrol G pada skenario 1, 2, 3 dan 4 berkisar antara 1,11 sampai 1,86; 1,1 sampai 2,04; 1,05 sampai 1,97 dan 1,01 sampai 1,92. Sedangkan ARL diagram kontrol W berkisar antara 1,25 sampai 2,02 untuk skenario 1; 1,06 sampai 1,98 untuk skenario 2; 1,02 sampai 2,33 untuk skenario 3 dan skenario 4 berkisar 1 sampai 2,16. Ini berarti pada  $p=4$  dan  $n=5$  maka kinerja diagram kontrol G baik jika terjadi pergeseran varian yang kecil, sedangkan jika terjadi pergeseran varian yang besar maka kinerja diagram kontrol W lebih baik dibanding diagram kontrol G.

Hasil diatas berbeda pada grafik nilai ARL diagram kontrol G dan W pada  $p=4$  dan  $n=6$ . Skenario 1 dan 2 dari hasil pada  $p=4$  dan  $n=6$  sebanding dengan hasil pada  $p=4$  dan  $n=5$ , namun skenario 3 dan 4 memiliki hasil yang berbeda dengan hasil sebelumnya. Pada skenario 1 dan 2 untuk pergeseran varian yang kecil nilai ARL untuk diagram kontrol G lebih kecil dibanding diagram kontrol W, tetapi menghasilkan hasil sebaliknya untuk pergeseran varian yang besar. Sedangkan pada skenario 3 dan 4, khususnya untuk pergeseran varian yang kecil ARL diagram kontrol G lebih kecil dibanding diagram kontrol W. Tetapi pada pergeseran varian yang besar nilai ARL dari kedua diagram hampir sama. Ini menandakan bahwa kinerja diagram kontrol G sangat baik digunakan jika terjadi pergeseran varian yang kecil, namun untuk pergeseran varian-varian yang besar dapat juga digunakan diagram kontrol W sebagai pendamping dari diagram kontrol G.

Tabel 3. Grafik Nilai ARL Diagram Kontrol G dan W untuk  $p=4, n=5$  dan  $6$



#### 4. Kesimpulan

Berdasarkan pembahasan yang telah diuraikan sebelumnya, maka kesimpulan yang dapat diperoleh adalah :

- 1) Diagram kontrol G sangat baik dalam mendeteksi perubahan proses, walaupun perubahan proses yang terjadi sangat kecil.
- 2) Diagram kontrol W dapat digunakan sebagai pembanding diagram kontrol G untuk mendeteksi perubahan proses yang besar.
- 3) Kinerja diagram kontrol G lebih baik dibanding kinerja diagram kontrol W, namun untuk perubahan proses yang besar diagram kontrol W dapat digunakan sebagai pendamping diagram kontrol G.

#### Daftar Pustaka

- [1] F. B. Alt dan S. J. N. E, Multivariate quality control In Encyclopedia of Statistical Scinces Kotz, New York: John Wiley & Sons, 1985, pp. 110-122.
- [2] T. Anderson, An Introduction to Multivariate Statistical Analysis, New York: John Wiley & Sons, Inc, 1984.
- [3] Djauhari, M. A., "Improved Monitoring of Multivariate Process Variability. *Journal of Quality Technology*," vol. 37, 2005, pp. 32-39.
- [4] Kramer, C.Y. dan Jensen, D.R, "Fundamentals of Multivariate Analysis-Part II: Inference About Two Treatments" *Journal of Quality Technology*, vol. 1, pp. 189-204, 1969.
- [5] Levinson, William A., Holmes, D. S. and Mergen, A. E, Variation Chart for Multivariate Processes *Quality Engineering*, 2002, pp. 539-545.
- [6] Montgomery, D. C, "Introduction to Statistical Quality Control," 6th Edition penyunt, New York, John Wiley & Sons, Inc, 2009.
- [7] Noya Van Delsen, M dan Mashuri, M, "Perbandingan Kinerja Diagram Kontrol G dan Diagram Kontrol Improved |S|," Surabaya.
- [8] Rakhmawati, Dwi Yuli, "Perbandingan Kinerja Diagram Kontrol Multivariat untuk Variabilitas Berdasarkan Matriks Kovariansi dan Matriks Korelasi," Surabaya, 2011.
- [9] Sindelar, M.F, "Multivariate Statistical Process Control For Corellation Matrices," Tesis Ph.D, 2007.
- [10] Wen Lu, C. dan Reynolds, M. R, " Cusum Charts for Monitoring an Autocorrelated Process," *Journal of Quality Technology*, pp. 316-334., 2001.

# PEDOMAN PENULISAN



jurnal ilmu matematika dan terapan

**Barekeng** terbit dua kali dalam setahun yaitu Bulan Maret dan Desember. **Barekeng** menerima naskah dalam bentuk hasil penelitian, catatan penelitian (*note*) atau artikel ulasan balik (*review/ minireview*) dan ulasan (*feature*) baik dalam bahasa Indonesia maupun dalam bahasa Inggris yang berkaitan dengan bidang Matematika dan Terapannya. Naskah yang dikirimkan merupakan naskah asli yang belum pernah diterbitkan di media manapun.

## PENGIRIMAN NASKAH

Naskah dikirimkan kepada:

**Redaksi Barekeng**

**Jurusan Matematika**

**Fakultas MIPA**

**Universitas Pattimura**

**Jl. Ir. M. Putuhena, Poka-Ambon**

Email: [jurnalbarekeng@gmail.com](mailto:jurnalbarekeng@gmail.com)

Naskah yang dikirimkan harus dalam bentuk naskah cetak (*hard copy*) dan naskah lunak (*soft copy*), disertai dengan alamat korespondensi lengkap dan alamat *email* yang dapat dihubungi.

**Naskah cetak (*hard copy*):**

Naskah cetak dikirim sebanyak satu eksemplar dengan format pengetikan menggunakan *Microsoft Word* seperti berikut:

- Naskah diketik 1 spasi pada kertas HVS Ukuran A4 dengan batas tepi 2 cm dan font *Times New Roman* berukuran 11 point.
- Jumlah halaman maksimum 12 halaman termasuk Lampiran (Gambar dan Tabel). Setiap halaman diberi nomor secara berurutan pada tepi kanan atas. Untuk keterangan Lampiran: Tipe huruf Times New Roman berukuran 10 point.
- Persamaan matematika (*equations*) dapat diketik dengan menggunakan *MS Equations* atau *MathType* dengan tipe huruf *Cambria* atau Times New Roman berukuran 11 point.

**Naskah lunak (*soft copy*):**

Naskah lunak harus dalam format *Microsoft Word* dan dikirimkan dalam bentuk disk (CD, DVD), *flashdisk*, atau *attachment email*.

## SUSUNAN NASKAH

- a. Judul ditulis dalam Bahasa Indonesia dan Bahasa Inggris.
- b. Nama Lengkap Penulis (tanpa gelar).
- c. Nama Lembaga atau Institusi, disertai Alamat Lengkap dengan nomor kode pos. Untuk korespondensi dilengkapi No. Telp., fax dan *email*.
- d. Judul Ringkas (*Running Title*) (jika diperlukan).
- e. Abstrak (*Abstract*) dalam Bahasa Indonesia dan Bahasa Inggris.
- f. Kata Kunci (*Keywords*) dalam Bahasa Indonesia dan Bahasa Inggris.
- g. Pendahuluan (*Introduction*) meliputi latar belakang, masalah dan tujuan penelitian.

- h. Tinjauan Pustaka meliputi ulasan (*Review*) penelitian dari beberapa literatur serta teori-teori dasar yang mendukung penelitian.
- i. Metode Penelitian (*Methods and Materials*) meliputi bahan, cara, dan analisis dalam penelitian (jika ada).
- j. Hasil dan Pembahasan (*Results and Discussion*) ditulis secara berkesinambungan dalam satu rangkaian naskah penulisan.
- k. Kesimpulan (*Conclusion*)
- l. Ucapan Terima Kasih (*Acknowledgements*) (Jika diperlukan)
- m. Daftar Pustaka ditulis memakai sistem indeks sesuai *reference style* IEEE. Di bawah ini beberapa contoh penulisan sumber acuan:

**Jurnal:**

- [1] K. R. Gabriel, "The Biplot Graphic Display of Matrices with Application to Principal Component Analysis," *Biometrika*, vol. 58, pp. 453-467, 1997.

**Buku:**

- [2] D. Rosadi, *Ekonometrika & Analisis Runtun Waktu Terapan dengan Eviews (Aplikasi untuk bidang ekonomi, bisnis, dan keuangan)*, Yogyakarta: Andi Offset, 2012.

**Skripsi/ Tesis/ Disertasi:**

- [3] M. Apri, "Model Biaya Total Jaringan Pipa Transmisi Gas dan Optimasinya," Departemen Matematika ITB, Bandung, 2002.

**Informasi dari Internet:**

- [4] G. Skye, "Transformation," 8 Desember 2012. [Online]. Available: <http://www.livelove.co.uk/ap/>. [Diakses 4 Oktober 2014].

- n. Lampiran meliputi Gambar dan Tabel beserta keterangannya (jika diperlukan).

## CATATAN (NOTE)

- Naskah harus dikirimkan ke redaksi selambat-lambatnya 2 (dua) bulan sebelum bulan penerbitan jurnal (Maret dan Desember).
- Naskah akan dinilai oleh tim penilai yang relevan sebelum diterbitkan dan tim redaksi berhak merubah struktur naskah tanpa merubah isi naskah.
- Naskah dapat diterima atau ditolak. Naskah ditolak, jika tidak memenuhi kriteria penulisan, pelanggaran hak cipta, kualitas rendah, dan tidak menanggapi korespondensi redaksi. Pengumuman naskah ditolak atau diterima paling lambat 1 (satu) bulan setelah naskah terkirim.
- Penulis atau penulis pertama memperoleh 1 (satu) eksemplar jurnal yang sudah diterbitkan.

**B**arekeng

jurnal ilmu matematika dan terapan

ISSN 1978 - 7227



9 771978 722737

© Jurusan Matematika FMIPA UNPATTI 2016