

«THESIS TITLE»

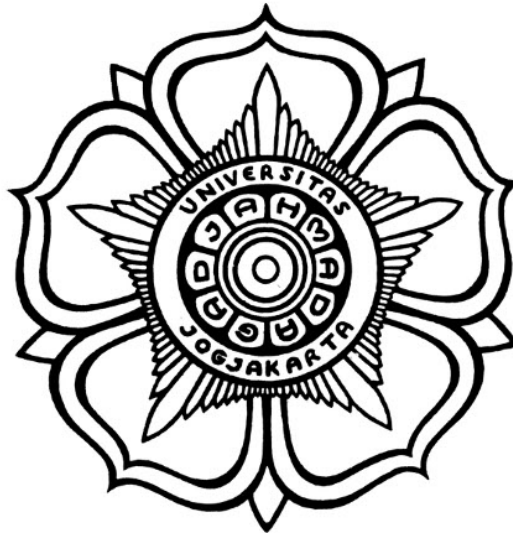
Pendadaran/Tesis/Ringkasan Tesis*

submitted in partial fulfillment of the requirements
for the degree of master

«Program name»Program

«Major»

Department of Electrical Engineering and Information Technology



proposed by

«AUTHOR»

«NIM»

To

**GRADUATE PROGRAM
FACULTY OF ENGINEERING
UNIVERSITAS GADJAH MADA
YOGYAKARTA
«year submit»**

THESIS

«THESIS TITLE»

Written by

«AUTHOR»

«NIM»

Has been presented
on «Exam date»

Thesis Examiners

Supervisor I

Examiners

«Supervisor 1»

«Examiner 1»

Supervisor II

«Supervisor 2»

«Examiner 2»

This thesis has been accepted as a partial fulfillment
for the degree of Master

Date:

Program Coordinator of «Program name»

«Program coordinator»

NIP. «NIP»

Head of Department

Department of Electrical Engineering and Information Technology

«Head of the department»

NIP. «NIP»

STATEMENT

Saya yang bertanda tangan di bawah ini :

Nama : Nama
NIM : NIM
Tahun terdaftar : 2017
Program Studi : S2 Teknik Elektro
Fakultas : Teknik

Menyatakan bahwa dalam dokumen ilmiah Skripsi/Tesis/Disertasi*** ini tidak terdapat bagian dari karya ilmiah lain yang telah diajukan untuk memperoleh gelar akademik di suatu lembaga Pendidikan Tinggi, dan juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang/lembaga lain, kecuali yang secara tertulis disitasi dalam dokumen ini dan disebutkan sumbernya secara lengkap dalam daftar pustaka.

Dengan demikian saya menyatakan bahwa dokumen ilmiah ini bebas dari unsur-unsur plagiasi dan apabila dokumen ilmiah Skripsi/Tesis/Disertasi*** ini di kemudian hari terbukti merupakan plagiasi dari hasil karya penulis lain dan/atau dengan sengaja mengajukan karya atau pendapat yang merupakan hasil karya penulis lain, maka penulis bersedia menerima sanksi akademik dan/atau sanksi hukum yang berlaku.

Yogyakarta, 13 Maret 2020

Canggih
NIM

PREFACE

[SAMPLE]

Puji syukur ke hadirat Allah SWT yang telah melimpahkan rahmat dan barokah-Nya sehingga penulis dapat menyelesaikan tesis dengan judul "...". Laporan tesis ini disusun untuk memenuhi salah satu syarat dalam memperoleh gelar Master of Engineering (M.Eng.) pada Program Studi S2 Teknik Elektro Fakultas Teknik Universitas Gadjah Mada Yogyakarta.

Dalam melakukan penelitian dan penyusunan laporan tesis ini penulis telah mendapatkan banyak dukungan dan bantuan dari berbagai pihak. Penulis mengucapkan terima kasih yang tak terhingga kepada:

1. <nama pembimbing utama + gelar> selaku dosen pembimbing utama, dan <nama pembimbing pendamping + gelar> selaku dosen pembimbing pendamping, yang telah dengan penuh kesabaran dan ketulusan memberikan ilmu dan bimbingan terbaik kepada penulis.
2. <nama kaprodi + gelar> selaku Ketua Departemen Teknik Elektro dan Teknologi Informasi dan <nama kaminat + gelar> selaku Ketua Program Studi S2 Teknik Elektro Fakultas Teknik Universitas Gadjah Mada yang memberikan izin kepada penulis untuk belajar.
3. Para Dosen Program Studi S2 Teknik Elektro Fakultas Teknik Universitas Gadjah Mada yang telah memberikan bekal ilmu kepada penulis.
4. Para Karyawan/wati Program Studi S2 Teknik Elektro Fakultas Teknik Universitas Gadjah Mada yang telah membantu penulis dalam proses belajar.
5.dst

Penulis menyadari sepenuhnya bahwa laporan tesis ini masih jauh dari sempurna, untuk itu semua jenis saran, kritik dan masukan yang bersifat membangun sangat penulis harapkan. Akhir kata, semoga tulisan ini dapat memberikan manfaat dan memberikan wawasan tambahan bagi para pembaca dan khususnya bagi penulis sendiri.

Yogyakarta, 13 Maret 2017

Canggih

NOMENCLATURE AND ABBREVIATION

[SAMPLE]

b	=	bias
$K(x_i, x_j)$	=	fungsi kernel
y	=	kelas keluaran
C	=	parameter untuk mengendalikan besarnya pertukaran antara penalti variabel slack dengan ukuran margin
L_D	=	persamaan Lagrange dual
L_P	=	persamaan Lagrange primal
\mathbf{w}	=	vektor bobot
\mathbf{x}	=	vektor masukan
ANFIS	=	Adaptive Network Fuzzy Inference System
ANSI	=	American National Standards Institute
DAG	=	Directed Acyclic Graph
DDAG	=	Decision Directed Acyclic Graph
HIS	=	Hue Saturation Intensity
QP	=	Quadratic Programming
RBF	=	Radial Basis Function
RGB	=	Red Green Blue
SV	=	Support Vector
SVM	=	Support Vector Machines

ABSTRACT

Servomotor uses feedback controller to control the speed or the position, or both. Typically, the PID controller is used and has evolved into more recent approaches like the hybrid with fuzzy logic controller (FLC) or neural network (NN). Many tuning methods for PID controller have been developed, and one of them is based on natural evolution, the genetic algorithm (GA). The significant drawback of GA is that the optimization process needs too many iterations and too long duration. In this thesis, a new optimization GA-based algorithm that emanates from modification of conventional GA to reduce the iterations number and the duration time, namely, semi-parallel operation genetic algorithm (SPOGA) is proposed. The aim of the algorithm is to improve a controller performance when used for a DC servomotor application.

The servomotor's transfer function is obtained via system identification and is modelled using MATLAB commands. The model is used in the simulation of speed and position control and the performance of relevant conventional, fuzzy, and hybrid controllers are compared for various predefined conditions. The best controller is then selected to be optimized using SPOGA. Next, the performance comparison of GA and SPOGA is conducted based on the maximum value of parallel functions obtained. The SPOGA is then used to optimize the selected controllers and the performance comparisons of the controllers were conducted.

Detailed performance comparisons of controllers for a DC servomotor speed and position control under seven predefined conditions is presented. As compared to conventional GA, SPOGA performs better in reducing the number of test runs with the same results. The findings demonstrate the effectiveness of the hybrid-fuzzy controller for speed and position control of a DC servomotor, and confirm the ability of SPOGA as an optimization algorithm for the hybrid-fuzzy controller.

Keywords :control, fuzzy, genetic algorithms, servomotor

INTISARI

Dokumen ini merupakan format panduan bagi penulis untuk menulis Tesis yang siap disahkan oleh pembimbing maupun Program Studi.. Para penulis harus mengikuti petunjuk yang diberikan dalam template ini. Anda dapat menggunakan dokumen ini baik sebagai petunjuk penulisan dan sebagai template di mana Anda dapat mengetik teks Anda sendiri. Tuliskan intisari dalam bahasa Indonesia.

Kata kunci – Letakkan kata kunci Anda di sini, kata kunci dipisahkan dengan koma. Istilah dengan bahasa Indonesia.

CONTENTS

ENDORSEMENT PAGE	ii
STATEMENT.....	iii
PREFACE.....	iv
NOMENCLATURE AND ABBREVIATION	v
ABSTRACT	vi
INTISARI.....	vii
CONTENTS	viii
LIST OF FIGURES	ix
LIST OF TABLES.....	x
CHAPTER I Perintah-perintah dasar	1
1.1 Penggunaan Sitasi.....	1
1.2 Penulisan Gambar.....	1
1.3 Penulisan Tabel	1
1.4 Penulisan formula.....	2
1.5 Contoh list.....	2
1.6 Contoh <i>highlight</i>	2
CHAPTER II Blok beda halaman	4
2.1 Membuat algoritma terpisah.....	4
2.2 Membuat tabel terpisah	4
2.3 Menulis formula terpisah halaman	5
CHAPTER III «Chapter name».....	6
CHAPTER IV «Chapter name».....	7
CHAPTER V «Chapter name».....	8
REFERENCES	9
APPENDIX	L-1
L.1 Sample algorithm	L-1
L.2 Sample Python code	L-2
L.3 Sample Matlab code	L-3

LIST OF FIGURES

Figure 1.1	Contoh gambar.....	1
------------	--------------------	---

LIST OF TABLES

Table 1.1	tabel ini	1
Table 1.2	tabel ini	2
Table 2.1	Contoh tabel panjang	4

CHAPTER I

PERINTAH-PERINTAH DASAR

1.1 Penggunaan Sitasi

Contoh penggunaan sitasi [1, 2] [3] [4] [5] [6, 7]

1.2 Penulisan Gambar

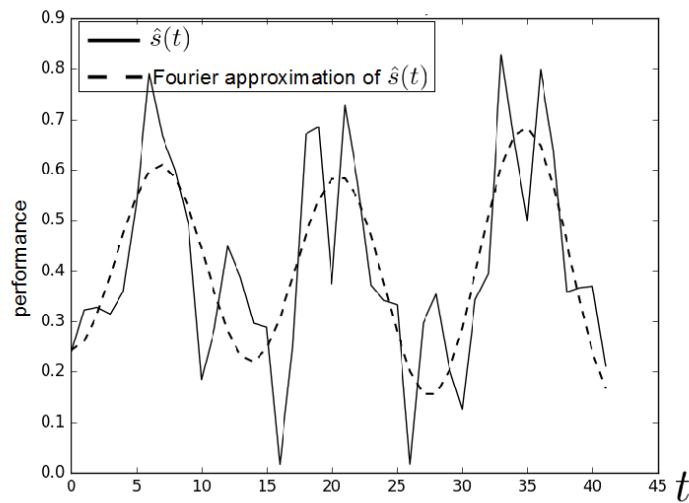


Figure 1.1. Contoh gambar.

Contoh gambar terlihat pada Gambar 1.1. Gambar diambil dari [7].

1.3 Penulisan Tabel

Table 1.1. tabel ini

ID	Tinggi Badan (cm)	Berat Badan (kg)
A23	173	62
A25	185	78
A10	162	70

Contoh penulisan tabel bisa dilihat pada Tabel 1.1.

Contoh penulisan tabel dengan nama kolom sangat panjang dengan bantuan *tabulary* bisa dilihat pada Tabel 1.2.

Table 1.2. tabel ini

ID	Tinggi Badan Mahasiswa di Universitas Gadjah Mada (cm)	Berat Badan Mahasiswa di Universitas Gadjah Mada(kg)
A23	173	62
A25	185	78
A10	162	70

1.4 Penulisan formula

Contoh penulisan formula

$$L_{\psi_z} = \{t_i \mid v_z(t_i) \leq \psi_z\} \quad (1.1)$$

Contoh penulisan secara *inline*: $PV = nRT$. Untuk kasus-kasus tertentu, kita membutuhkan perintah "mathit" dalam penulisan formula untuk menghindari adanya jeda saat penulisan formula.

Contoh formula **tanpa** menggunakan "mathit": $PVA = RTD$

Contoh formula **dengan** menggunakan "mathit": $PVA = RTD$

Untuk mengutip persamaan dapat juga menggunakan (1.2). Persamaan dapat ditulis secara sentral pada file "equations/equations.tex".

$$\min_{\mathbf{x}} \frac{1}{2} \mathbf{x}^T Q \mathbf{x} + \mathbf{c}^T \mathbf{x} \quad (1.2)$$

1.5 Contoh list

Berikut contoh penggunaan list

1. First item
2. Second item
3. Third item

1.6 Contoh *highlight*

Berikut contoh penggunaan **highlight kuning menggunakan** `\hly{}`. Penggunaan *highlight* tidak disarankan untuk dua paragraf sekaligus karena dapat mengakibatkan kerusakan dalam penomoran.

Gunakan perintah `\hly{}` yang terpisah untuk tiap paragrafnya seperti ini.

Penggunaan tanda *curly bracket* { } di dalam *highlight* seperti penggunaan
\textit{} diperbolehkan, namun penghapusan otomatis oleh
"utilities/remove_highlight.ipynb" tidak mendukung penghapusan
nested bracket seperti itu. Alternatifnya gunakan pemisahan *for the italic text* seperti ini.

CHAPTER II

BLOK BEDA HALAMAN

2.1 Membuat algoritma terpisah

Untuk membuat algoritma terpisah seperti pada contoh berikut, kita dapat memanfaatkan perintah *algstore* dan *algrestore* yang terdapat pada paket *algcompatible*. Pada dasarnya, kita membuat dua blok algoritma dimana blok pertama kita simpan menggunakan *algstore* dan kemudian di-restore menggunakan *algrestore* pada algoritma kedua. Perintah tersebut dimaksudkan agar terdapat kesinamungan antara kedua blok yang sejatinya adalah satu blok.

Algorithm 1 Contoh algorima

```
1: procedure CREATESET( $v$ )  
2:   Create new set containing  $v$   
3: end procedure
```

Pada blok algoritma kedua, tidak perlu ditambahkan caption dan label, karena sudah menjadi satu bagian dalam blok pertama. Pembagian algoritma menjadi dua bagian ini berguna jika kita ingin menjelaskan bagian-bagian dari sebuah algoritma, maupun untuk memisah algoritma panjang dalam beberapa halaman.

```
4: procedure CONCATSET( $v$ )  
5:   Create new set containing  $v$   
6: end procedure
```

2.2 Membuat tabel terpisah

Untuk membuat tabel panjang yang melebihi satu halaman, kita dapat mengganti kombinasi *table* + *tabular* menjadi *longtable* dengan contoh sebagai berikut.

Table 2.1. Contoh tabel panjang

header 1	header 2
foo	bar
foo	bar
foo	bar
foo	bar
foo	bar
foo	bar

foo	bar
foo	bar
foo	bar
foo	bar
foo	bar

2.3 Menulis formula terpisah halaman

Terkadang kita butuh untuk menuliskan rangkaian formula dalam jumlah besar sehingga melewati batas satu halaman. Solusi yang digunakan bisa saja dengan memindahkan satu blok formula tersebut pada halaman yang baru atau memisah rangkaian formula menjadi dua bagian untuk masing-masing halaman. Cara yang pertama mungkin akan menghasilkan alur yang berbeda karena ruang kosong pada halaman pertama akan diisi oleh teks selanjutnya. Sehingga di sini kita dapat memanfaatkan *align* yang sudah diatur dengan mode *allowdisplaybreaks*. Penggunaan *align* ini memungkinkan satu rangkaian formula terpisah berbeda halaman.

Contoh sederhana dapat digambarkan sebagai berikut.

$$\begin{aligned}
 x &= y^2 \\
 x &= y^3 \\
 a + b &= c \\
 x &= y - 2 \\
 a + b &= d + e \\
 x^2 + 3 &= y \\
 a(x) &= 2x \\
 b_i &= 5x \\
 10x^2 &= 9x \\
 2x^2 + 3x + 2 &= 0 \\
 5x - 2 &= 0 \\
 d &= \log x \\
 y &= \sin x
 \end{aligned} \tag{2.1}$$

CHAPTER III
«CHAPTER NAME»

CHAPTER IV
«CHAPTER NAME»

CHAPTER V
«CHAPTER NAME»

REFERENCES

- [1] L. E. Nugroho, "E-book as a platform for exploratory learning interactions," *International Journal of Emerging Technologies in Learning (iJET)*, vol. 11, no. 01, pp. 62–65, 2016. [Online]. Available: <http://www.online-journals.org/index.php/i-jet/article/view/5011>
- [2] P. I. Santosa, "User?s preference of web page length," *International Journal of Research and Reviews in Computer Science*, pp. 180–185, 2011.
- [3] N. A. Setiawan, "Fuzzy decision support system for coronary artery disease diagnosis based on rough set theory," *International Journal of Rough Sets and Data Analysis (IJRSDA)*, vol. 1, no. 1, pp. 65–80, 2014.
- [4] C. P. Wibowo, P. Thumwarin, and T. Matsuura, "On-line signature verification based on forward and backward variances of signature," in *Information and Communication Technology, Electronic and Electrical Engineering (JICTEE), 2014 4th Joint International Conference on*. IEEE, 2014, pp. 1–5.
- [5] D. A. Marenda, A. Nasikun, and C. P. Wibowo, "Digitory, a smart way of learning islamic history in digital era," *arXiv preprint arXiv:1607.07790*, 2016.
- [6] S. Wibirama, S. Tungjitkusolmun, and C. Pintavirooj, "Dual-camera acquisition for accurate measurement of three-dimensional eye movements," *IEEJ Transactions on Electrical and Electronic Engineering*, vol. 8, no. 3, pp. 238–246, 2013.
- [7] C. P. Wibowo, "Clustering seasonal performances of soccer teams based on situational score line," *Communications in Science and Technology*, vol. 1, no. 1, 2016.

APPENDIX

L.1 Sample algorithm

Algorithm 2 Kruskal's Algorithm

```
1: procedure MAKESET( $v$ )
2:   Create new set containing  $v$ 
3: end procedure
4:
5: function FINDSET( $v$ )
6:   return a set containing  $v$ 
7: end function
8:
9: procedure UNION( $u, v$ )
10:  Unites the set that contain  $u$  and  $v$  into a new set
11: end procedure
12:
13: function KRUSKAL( $V, E, w$ )
14:    $A \leftarrow \{\}$ 
15:   for each vertex  $v$  in  $V$  do
16:     MakeSet( $v$ )
17:   end for
18:   Arrange  $E$  in increasing costs, ordered by  $w$ 
19:   for each  $(u, v)$  taken from the sorted list do
20:     if FindSet( $u$ )  $\neq$  FindSet( $v$ ) then
21:        $A \leftarrow A \cup \{(u, v)\}$ 
22:       Union( $u, v$ )
23:     end if
24:   end for
25:   return  $A$ 
26: end function
```

L.2 Sample Python code

```
1 import numpy as np
2
3 def incmatrix(genl1, genl2):
4     m = len(genl1)
5     n = len(genl2)
6     M = None #to become the incidence matrix
7     VT = np.zeros((n*m,1), int) #dummy variable
8
9     #compute the bitwise xor matrix
10    M1 = bitxormatrix(genl1)
11    M2 = np.triu(bitxormatrix(genl2),1)
12
13    for i in range(m-1):
14        for j in range(i+1, m):
15            [r,c] = np.where(M2 == M1[i,j])
16            for k in range(len(r)):
17                VT[(i)*n + r[k]] = 1;
18                VT[(i)*n + c[k]] = 1;
19                VT[(j)*n + r[k]] = 1;
20                VT[(j)*n + c[k]] = 1;
21
22    if M is None:
23        M = np.copy(VT)
24    else:
25        M = np.concatenate((M, VT), 1)
26
27    VT = np.zeros((n*m,1), int)
28
29    return M
```

L.3 Sample Matlab code

```
1 function X = BitXorMatrix(A,B)
2 %function to compute the sum without charge of two vectors
3
4 %convert elements into unsigned integers
5 A = uint8(A);
6 B = uint8(B);
7
8 m1 = length(A);
9 m2 = length(B);
10 X = uint8(zeros(m1, m2));
11 for n1=1:m1
12     for n2=1:m2
13         X(n1, n2) = bitxor(A(n1), B(n2));
14     end
15 end
```