

**PENGEMBANGAN *FIRMWARE TRACKER* BUS KAMPUS DENGAN  
MODUL GPS PADA PLATFORM STM32**

**SKRIPSI**



Disusun oleh:

**AIRLANGGA RASYAD FIDIYANTO  
19/443562/TK/48758**

**PROGRAM STUDI TEKNIK ELEKTRO  
DEPARTEMEN TEKNIK ELEKTRO DAN TEKNOLOGI INFORMASI  
FAKULTAS TEKNIK UNIVERSITAS GADJAH MADA  
YOGYAKARTA  
2023**

## **HALAMAN PENGESAHAN**

### **PENGEMBANGAN *FIRMWARE TRACKER* BUS KAMPUS DENGAN MODUL GPS PADA PLATFORM STM32**

#### **SKRIPSI**

Diajukan Sebagai Salah Satu Syarat untuk Memperoleh  
Gelar Sarjana Teknik Program S-1  
pada Departemen Teknik Elektro dan Teknologi Informasi  
Fakultas Teknik  
Universitas Gadjah Mada

Disusun oleh:

**AIRLANGGA RASYAD FIDIYANTO**  
**19/443562/TK/48758**

Telah disetujui dan disahkan

Pada tanggal . . . . .

Pembimbing I

Pembimbing II

**I Wayan Mustika, S.T., M.Eng., Ph.D.**  
**NIP. 198109212014041001**

**Ir. Agus Bejo, S.T., M.Eng., D.Eng., IPM.**  
**198001012015041002**

## **HALAMAN PERSEMBAHAN**

**[SAMPLE]**

This is page of dedication

## **PERNYATAAN BEBAS PLAGIASI**

Saya yang bertanda tangan di bawah ini :

Nama : Airlangga Rasyad Fidiyanto  
NIM : 19/443562/TK/48758  
Tahun terdaftar : 2019  
Program Studi : S1 Teknik Elektro  
Fakultas : Teknik

Menyatakan bahwa dalam dokumen ilmiah Skripsi 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 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

Airlangga Rasyad Fidiyanto  
19/443562/TK/48758

## PRAKATA

Puji syukur ke hadirat Allah SWT yang telah melimpahkan rahmat dan barokah-Nya sehingga penulis dapat menyelesaikan skripsi dengan judul Pengembangan *Firmware Tracker* Bus Kampus dengan Modul GPS pada Platform STM32. Laporan skripsi ini disusun untuk memenuhi salah satu syarat dalam memperoleh gelar Sarjana Teknik (S.T.) pada Program Studi S1 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. I Wayan Mustika, S.T., M.Eng., Ph.D. selaku dosen pembimbing utama, dan Ir. Agus Bejo, S.T., M.Eng., D.Eng., IPM. selaku dosen pembimbing pendamping, yang telah dengan penuh kesabaran dan ketulusan memberikan ilmu dan bimbingan terbaik kepada penulis.
2. Ir. Hanung Adi Nugroho, S.T., M.E., Ph.D., IPM. selaku Ketua Departemen Teknik Elektro dan Teknologi Informasi dan Ir. Adha Imam Cahyadi, S.T., M.Eng., D.Eng., IPM. selaku Ketua Program Studi S1 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

Airlangga Rasyad Fidiyanto

## ARTI LAMBANG DAN SINGKATAN

### [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.



## DAFTAR ISI

HALAMAN PENGESAHAN .....	ii
HALAMAN PERSEMBAHAN .....	iii
PERNYATAAN BEBAS PLAGIASI .....	iv
PRAKATA .....	v
ARTI LAMBANG DAN SINGKATAN .....	vi
ABSTRACT .....	vii
INTISARI .....	viii
DAFTAR ISI .....	ix
DAFTAR GAMBAR .....	x
DAFTAR TABEL .....	xi
BAB I PENDAHULUAN .....	1
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	2
1.3 Batasan Tugas Akhir .....	2
1.4 Tujuan Tugas Akhir .....	2
1.5 Manfaat Tugas Akhir .....	2
1.6 Sistematika Penulisan .....	2
BAB II TINJAUAN PUSTAKA .....	3
2.1 Tinjauan Pustaka .....	3
2.1.1 Teknologi Pelacak .....	3
2.1.2 Transmisi Data Nirkabel .....	3
2.2 Dasar Teori .....	4
2.2.1 STM32 Nucleo-WL55JC1 .....	4
2.2.2 Modul GNSS Teseo LIV3FL .....	5
2.2.3 Antena GPS .....	5
2.2.4 Modul LoRa(?) .....	5
BAB III METODE TUGAS AKHIR .....	6
3.1 Alat dan Bahan Tugas Akhir .....	6
3.2 Alur Tugas Akhir .....	6
3.3 Studi Literatur .....	6
3.4 Analisis Kebutuhan Sistem .....	6
3.5 Perancangan <i>End Node</i> .....	6
3.6 Perancangan API dan Basis Data .....	6
3.7 Perancangan Web Penampil .....	6
BAB IV «Chapter name» .....	7
BAB V «Chapter name» .....	8
DAFTAR PUSTAKA .....	9
LAMPIRAN .....	L-1
L.1 Sample algorithm .....	L-1
L.2 Sample Python code .....	L-2
L.3 Sample Matlab code .....	L-3

## DAFTAR GAMBAR

Gambar 2.1	<i>Development Board</i> STM32 Nucleo-WL55JC1 .....	4
------------	---	---

## **DAFTAR TABEL**

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Salah satu moda transportasi dalam kota paling populer di Indonesia adalah bus. Daerah Istimewa Yogyakarta juga telah menyediakan dua buah layanan bus publik seperti Trans Jogja dan Teman Bus. Salah satu faktor yang membuat penggunaan bus cukup populer adalah cakupan wilayahnya luas dan juga murah [1]. Selain itu, lalu lintas yang padat dan lahan parkir yang terbatas juga menjadi motivasi beberapa orang untuk menggunakan transportasi publik. Jika peningkatan jumlah pada suatu daerah sangat tinggi maka dibutuhkan fasilitas transportasi umum yang layak seperti bus [2].

Pada awal bulan Maret tahun 2022, Rektor Universitas Gadjah Mada, Prof. Ir. Panut Mulyono, M.Eng., D.Eng., meluncurkan dua buah bus listrik untuk transportasi internal kampus. Kedua bus ini adalah inovasi dari UGM untuk memudahkan mobilisasi mahasiswa di area kampus seluas 183.36 hektar ini dan mengurangi penggunaan energi fosil secara bersamaan. Setiap bus akan memutari UGM sebanyak sepuluh kali dengan setiap putaran membutuhkan satu jam. Dengan adanya fasilitas bus kampus Trans Gadjah Mada diharapkan dapat membuat lingkungan kampus menjadi lebih nyaman dan kondusif.

Salah satu masalah yang banyak dikeluhkan oleh civitas akademika UGM adalah ketidakpastian waktu kedatangan Trans Gadjah Mada. Meskipun sudah diberikan jadwal estimasi kedatangan bus, terkadang waktu kedatangan bus tidak sesuai dikarenakan faktor cuaca, lalu lintas di sekitar Jalan Persatuan yang padat, dan faktor lainnya. Oleh karena itu, dibutuhkan suatu sistem untuk melacak posisi dari bus kampus agar calon penumpang dapat mengestimasi kapan bus yang akan ditumpangi datang.

Masalah serupa juga terjadi di India. Berdasarkan [3], masyarakat India hanya mengetahui waktu kedatangan berdasarkan jadwal saja tanpa mengetahui posisi terbaru dari bus yang akan ditumpangi. Penelitian yang dilakukan oleh [4] menunjukkan bahwa sistem pelacak berbasis GPS telah diimplementasikan di beberapa negara, tetapi belum diimplementasikan di Indonesia, khususnya di lingkungan Universitas Gadjah Mada.

*Global Positioning System* atau GPS adalah teknologi sistem navigasi berbasis satelit yang dapat menunjukkan posisi secara akurat. Dengan adanya sistem pelacak lokasi bus Trans Gadjah Mada berbasis GPS, diharapkan dapat membantu civitas akademika Universitas Gadjah Mada untuk mengestimasi waktu kedatangan bus.

Sesuai pemaparan di atas, tujuan dari penelitian ini adalah untuk mengembangkan suatu sistem IoT untuk melacak lokasi terbaru dari bus Trans Gadjah Mada. Tujuan pengembangan ini adalah memastikan data dari sensor adalah benar dan dapat dikirimkan ke *server* tanpa kehilangan atau kesalahan data.

- 1.2 Rumusan Masalah**
- 1.3 Batasan Tugas Akhir**
- 1.4 Tujuan Tugas Akhir**
- 1.5 Manfaat Tugas Akhir**
- 1.6 Sistematika Penulisan**

**BAB I**

**BAB II**

**BAB III**

**BAB VI**

**BAB V**

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Tinjauan Pustaka**

##### **2.1.1 Teknologi Pelacak**

Penelitian sebelumnya telah mengembangkan berbagai sistem pelacak aset dengan berbagai macam pendekatan pada perangkat keras maupun perangkat lunak untuk berbagai aplikasi. Sebagai contoh, [5] merancang suatu sistem untuk melacak dompet dengan menggunakan TK-102 GPS *Tracker*.

Tim peneliti dari *Vidyalankar Institute of Technology* telah merancang suatu sistem yang dapat mendeteksi lokasi dari kendaraan dan juga emisi CO yang dihasilkan. Pada sistem yang dirancang, digunakan *development board* Arduino Uno yang berbasis mikrokontroler ATmega328. Ketika kandungan gas CO sudah melebihi ambang batas, sistem akan memutuskan pengiriman bahan bakar dan kemudian mengirimkan data koordinat dari modul GPS ke *server* Apache yang telah dirancang [6].

Sebuah sistem *speedometer* telah dirancang oleh [7]. Sistem tersebut menggunakan modul GPS untuk menghitung kecepatan dan koordinat lokasi kendaraan. Data kecepatan kendaraan didapat dari menghitung waktu yang dibutuhkan oleh kendaraan untuk berpindah dari satu titik ke titik lainnya. Data yang didapat dikirimkan dengan API Adafruit IO menggunakan modul SIM808.

Penelitian yang dilakukan oleh [8] dari *University of London* menggunakan mikrokontroler AT89S52 dari keluarga 8051. Digunakan modul GPS M-89 yang diatur untuk menerima isyarat transmisi satelit pada frekuensi 1575.42 MHz. Data yang diterima akan ditampilkan pada layar LCD dan dikirimkan dengan modul GSM. Kemudian, data yang telah diterima akan ditampilkan pada situs web.

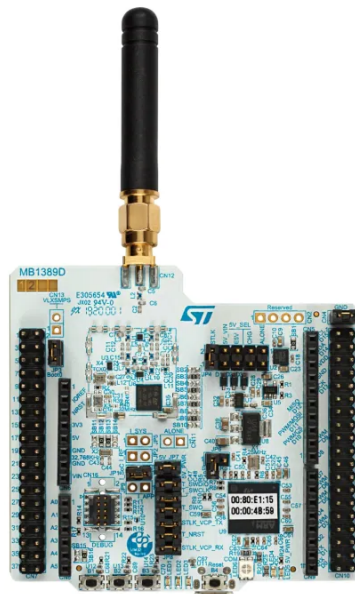
##### **2.1.2 Transmisi Data Nirkabel**

Sistem dapat dikembangkan dengan banyak cara, tetapi pada penelitian ini sistem yang akan dirancang hanya fokus untuk melacak lokasi dari kendaraan, dalam kasus ini adalah bus kampus "Trans Gajah Mada"

## 2.2 Dasar Teori

### 2.2.1 STM32 Nucleo-WL55JC1

STM32 Nucleo-WL55JC1 adalah *development board* berbasis mikrokontroler STM32WL55 yang dikembangkan oleh ST Microelectronics. Mikrokontroler yang digunakan adalah STM32WL55 *dual-core* Arm Cortex-M4/M0+ dengan *clock speed* 48 MHz [9].



Gambar 2.1. *Development Board* STM32 Nucleo-WL55JC1

Perangkat ini sudah terintegrasi dengan STLINK-V3E, sehingga tidak dibutuhkan perangkat tambahan untuk memprogram dan melakukan *debugging* pada perangkat [10]. Selain itu, perangkat ini juga mendukung penggunaan *expansion board* Arduino dan ST morpho. *Development board* STM32 Nucleo-WL55JC1 ditunjukkan oleh Gambar 2.1.

Mikrokontroler STM32WL55 memiliki *clock speed* 48 MHz jika dibandingkan dengan Arduino Mega yang hanya 16 MHz. Selain itu, STM32WL55 memiliki SRAM dengan kapasitas 64 KB atau delapan kali lipat dari yang dimiliki oleh Arduino Mega [11].

Karena performa tinggi dengan konsumsi daya rendah, maka digunakan mikrokontroler STM32WL55. Selain itu, STM32 juga memiliki komunitas yang tidak kalah luas dengan komunitas Arduino dan ESP-32.

**2.2.2 Modul GNSS Teseo LIV3FL**

**2.2.3 Antena GPS**

**2.2.4 Modul LoRa(?)**



## **BAB III**

### **METODE TUGAS AKHIR**

#### **3.1 Alat dan Bahan Tugas Akhir**

Dalam penelitian ini digunakan beberapa perangkat keras dan perangkat lunak. Perangkat keras yang digunakan dalam penelitian ini adalah sebagai berikut

1. Apple MacBook Pro dengan prosesor Apple M1 Pro 16-*core* dan memori terpadu 16 GB menggunakan sistem operasi macOS Ventura
2. STM32 Nucleo-WL55JC1 berbasis ARM Cortex-M0 dan ARM Cortex-M4
3. Modul GNSS Teseo LIV3FL
4. Modul antena Taoglas xxxx
5. Kabel *jumper* untuk purwarupa alat

Selanjutnya, perangkat lunak dan pustaka yang diunakan adalah sebagai berikut

1. STM32CubeIDE sebagai IDE pada *end node*
2. Visual Studio Code sebagai IDE untuk merancang situs jaringan penampil
3. Amazon Web Service untuk merancang API meliputi yang Lambda, API *Gateway*, dan RDS
4. MySQL sebagai basis data untuk pengolahan data
5. Postman untuk menguji API yang telah dirancang

#### **3.2 Alur Tugas Akhir**

#### **3.3 Studi Literatur**

#### **3.4 Analisis Kebutuhan Sistem**

#### **3.5 Perancangan *End Node***

#### **3.6 Perancangan API dan Basis Data**

#### **3.7 Perancangan Web Penampil**

**BAB IV**  
**«CHAPTER NAME»**

**BAB V**  
**«CHAPTER NAME»**

## DAFTAR PUSTAKA

- [1] M. M. Rohani, D. C. Wijeyesekera, and A. T. A. Karim, "Bus Operation, Quality Service and The Role of Bus Provider and Driver," *Procedia Engineering*, vol. 53, pp. 167–178, 2013.
- [2] A. Sutandi, "Pentingnya Transportasi Umum Untuk Kepentingan Publik," *Jurnal Administrasi Publik*, vol. 12, no. 1, pp. 19–34, 2015.
- [3] S. H. Sutar, R. Koul, and R. Suryavanshi, "Integration of Smart Phone and IOT for development of smart public transportation system," *2016 International Conference on Internet of Things and Applications, IOTA 2016*, pp. 73–78, 2016.
- [4] M. Sneha, C. N. Urs, S. Chatterji, M. S. Srivatsa, K. J. Pareekshith, and H. A. Kashyap, "Darideepa: A mobile application for bus notification system," *Proceedings of 2014 International Conference on Contemporary Computing and Informatics, IC3I 2014*, pp. 724–727, 2014.
- [5] H. M. Ekhsan, M. A. A. Zainudin, and J. N. Hamid, "Mobile App for Wallet Tracking using GPS Tracker," *2021 6th IEEE International Conference on Recent Advances and Innovations in Engineering, ICRAIE 2021*, vol. 2021, pp. 47–50, 2022.
- [6] K. H. Asha, K. Abhijna, S. Tabassum, and S. Shaur, "An Intelligent Air Pollution Vehicle Tracker System Using Smoke Sensor and GPS," *Cognitive Science and Technology*, pp. 399–410, 2022.
- [7] A. Najmurokhman, Kusnandar, A. Daelami, U. Komarudin, and M. Imanudin, "Design and Implementation of Vehicle Speed Recorder using GPS Tracker and Internet-of-Things Platform," *ICAICST 2021 - 2021 International Conference on Artificial Intelligence and Computer Science Technology*, pp. 152–156, 2021.
- [8] M. Mukhtar, "GPS based Advanced Vehicle Tracking and Vehicle Control System," *International Journal of Intelligent Systems and Applications*, vol. 7, no. 3, pp. 1–12, 2015.
- [9] STMicroelectronics, "STM32WL55JC," jul 2022.
- [10] STMicroelectronics, "Nucleo-WL55JC," oct 2022.
- [11] STMicroelectronics, "Multiprotocol LPWAN dual core 32-bit Arm® Cortex®-M4/M0+ LoRa®, (G)FSK, (G)MSK, BPSK, up to 256KB flash, 64KB SRAM," 2022.

## LAMPIRAN

### L.1 Sample algorithm

---

**Algorithm 1** 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
```