

# ***HUMAN MACHINE INTERFACE SISTEM KONTROL AUTOPILOT UGV (UNMANNED GROUND VEHICLE)***

Muarif Fandhi A<sup>\*)</sup>, Aris Triwiyatno, dan Budi Setiyono

Jurusan Teknik Elektro, Universitas Diponegoro Semarang  
Jl. Prof. Sudharto, SH, Kampus UNDIP Tembalang, Semarang 50275, Indonesia

<sup>\*)</sup>*E-mail: muarif.fandhi.akhmad@gmail.com*

## **Abstrak**

Unmanned Ground Vehicle (UGV) merupakan perangkat mekanik yang bergerak di atas permukaan tanah dan berfungsi sebagai sarana membawa, mengangkut sesuatu, atau melakukan pemetaan lokasi dan dikendalikan dari jarak jauh. UGV banyak digunakan didalam berbagai medan yang sulit ditempuh ataupun berbahaya bagi keselamatan manusia, misal untuk lokasi bencana alam, radiasi, ataupun untuk menjinakkan bom dalam dunia militer. Pada tugas akhir ini dilakukan perancangan dan pembuatan human machine interface untuk sistem kontrol jarak jauh serta untuk menampilkan pergerakan UGV sesuai dengan titik target yang telah ditentukan. Data yang diperoleh akan ditampilkan kedalam komputer menggunakan visual studio. Data hasil pengukuran dikirim secara nirkabel dari mikrokontroler ke komputer dengan menggunakan modul YS-1020UA. Hasil pengujian menunjukkan bahwa pergerakan UGV dapat ditampilkan dengan baik melalui interface pada monitor laptop. Data yang diterima ditampilkan dalam bentuk grafik maupun data dalam bentuk tabel. Hasil pengujian interface mampu mengendalikan UGV untuk berpindah dari satu titik target ke titik target lainnya, selain itu UGV dapat dikendalikan secara manual maupun otomatis yang dapat diatur melalui interface.

*Kata kunci: Human Machine Interface, UGV, GPS, YS-1020UA, Visual Studio*

## **Abstract**

Unmanned Ground Vehicle (UGV) is a mechanical device that moves on the ground and serves as a means of carrying, transporting something, or mapping the location and controlled remotely. UGV is widely used in a variety of terrain difficult to reach or dangerous for human safety, eg for the location of a natural disaster, radiation, or to defuse the bomb in the military. This final project was designed and manufactured human machine interface for remote control system and displayed the UGV movement according to target poin that have been determined. UGV movement data obtained from GPS sensor and encoder. Received data would be showed on computer used visual studio. Measurement data was sent wirelessly from the microcontroller to the computer using the YS-1020UA module. The results showed that the movement of UGV can be displayed on the interface on the laptop. Received Data was displayed in the form of graphs and data in tables form. UGV can be remotely controlled to move from one target to another, moreover it can also be controlled manually or automatically which can be set throug the interface.

*Keywords: Human Machine Interface,UGV, GPS, YS-1020UA, Visual Studio*

## **1. Pendahuluan**

Dalam era globalisasi ini, perkembangan teknologi yang menerapkan ilmu-ilmu elektronika dan sistem kontrol telah semakin banyak dan populer. Hal ini dapat diketahui dengan banyaknya peralatan yang menerapkan perpaduan dari ilmu dan sistem kontrol tersebut, salah satunya pada sistem navigasi robot khususnya pengoperasian UGV (*Unmanned Ground Vehicle*). UGV merupakan perangkat mekanik yang dioperasikan baik itu secara manual maupun otomatis di atas permukaan tanah untuk

membawa atau mengangkut sesuatu tanpa adanya kontak secara langsung oleh manusia[1]. Dalam beberapa pekerjaan yang berkaitan dengan pengawasan maupun pengamatan ke daerah yang sulit dijangkau sistem UGV dapat membantu mempermudah karena UGV dapat di kendalikan dari jarak jauh[2].

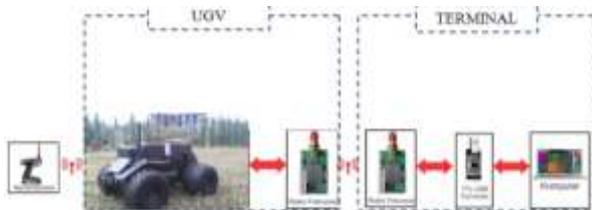
Sistem yang dapat digunakan untuk mengatasi kesulitan dalam pengawasan dan pengamatan pekerjaan pada UGV adalah sistem kontrol *autopilot*. Sistem kontrol *autopilot* memiliki kemampuan lebih dalam pengambilan

keputusan dan penyelesaian misi, sistem ini mampu untuk menyelesaikan tugas secara mandiri dalam lingkungan yang sulit[3]. Sensor yang biasanya digunakan dalam sistem autopilot antara lain adalah GPS, kompas, enkoder, magnetometer, kamera dan sebagainya. Dengan menggunakan GPS, posisi robot seperti garis lintang dan bujur dapat diketahui. GPS mengirimkan data tersebut menggunakan antarmuka komunikasi serial.

Pada sistem *autopilot human machine interface* digunakan sebagai ruang pengendali. *Human machine interface* (HMI) adalah sistem yang menghubungkan antara manusia dan teknologi mesin[4]. Prinsip HMI sering digunakan pada sistem kontrol modern karena operator dapat melihat atau mengontrol proses pada *plant* secara langsung dari jarak jauh. HMI menampilkan data pada operator dan menyediakan *input* dan *output* kontrol bagi operator dalam berbagai bentuk, termasuk grafik, tabel, *touch screen*, dan lain sebagainya[5]. HMI pada sistem autopilot UGV ini digunakan sebagai visualisasi data yang dikirim dari mikrokontroler serta memberikan input kepada mikrokontroler untuk menjalankan UGV secara autopilot menuju ke titik target yang telah ditentukan.

## 2. Metode

### 2.1. Mekanisme Autopilot pada UGV



Gambar 1. Diagram blok sistem kontrol autopilot pada UGV

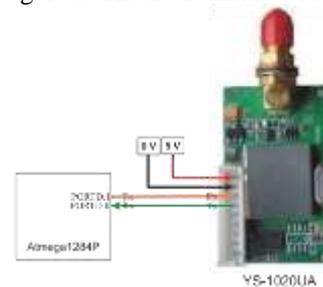
Sistem mekanisme *autopilot* pada UGV ini terdiri dari berbagai macam komponen antara lain sensor GPS, modul kompas digital, kontroler, perangkat UGV, dan komputer. Kontroler yang digunakan untuk mengatur pergerakan pada sistem ini adalah sebuah mikrokontroler. Mikrokontroler bertugas untuk mengambil data berupa posisi lintang, bujur, dan orientasi UGV dari GPS dan sensor kompas sebagai data input pemrosesan algoritma kontrol autopilot sehingga didapat sinyal kontrol yang akan diberikan pada aktuatur berupa motor DC dan motor servo. Selain sebagai kontroler, mikrokontroler juga

bertugas untuk mengirimkan data GPS ke komputer. Komputer memiliki peranan untuk menentukan posisi titik awal UGV dan memberikan titik target yang diinginkan untuk kemudian dikirimkan ke mikrokontroler melalui komunikasi serial dan monitoring data-data GPS, seperti koordinat posisi lintang dan bujur, orientasi, serta kecepatan UGV. Komputer juga dapat menerima

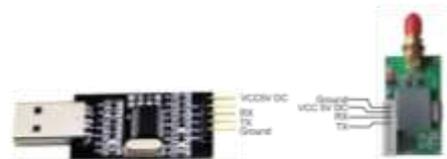
masukan dari *joystick* untuk pengendalian UGV secara manual.

### 2.2 Perancangan Modul Nirkabel

Proses pengiriman data dari UGV ke komputer atau dari komputer ke UGV membutuhkan media perantara. Media perantara yang digunakan pada tugas akhir ini adalah modul YS-1020UA. Modul YS-1020UA merupakan modul elektronika yang menghasilkan gelombang radio dengan frekuensi sekitar 433 MHz. Modul YS-1020UA dihubungkan dengan masing masing perangkat yaitu komputer dan mikrokontroler. Selain merancang komunikasi antara modul YS-1020UA dengan mikrokontroler, modul YS-1020UA juga perlu terkoneksi dengan komputer. Agar komputer dapat terhubung dengan modul YS-1020UA, maka diperlukan suatu modul rangkaian sebagai antarmuka komunikasi serial.



Gambar 2. Konfigurasi pin antara mikrokontroler dengan modul YS-1020UA



Gambar 3. Konfigurasi pin antara modul YS-1020UA dengan converter USB to RS232 TTL PL2303HX

### 2.3 Human Machine Interface

HMI merupakan perangkat lunak antarmuka berupa Graphical User Interface (GUI) berbasis komputer yang menjadi penghubung antara operator dengan mesin atau peralatan yang dikendalikan serta bertindak pada level pengawasan. Jadi secara umum HMI memiliki fungsi untuk memudahkan operator dalam mengawasi proses dan melakukan pengendalian terhadap *plant* dalam proses pengendalian.

#### 2.3.1 Microsoft Visual Studio

Visual studio merupakan sebuah bahasa pemrograman yang dibuat oleh microsoft yang dapat digunakan dalam

pembuatan program aplikasi. Visual studio memiliki komponen-komponen utama, yaitu:

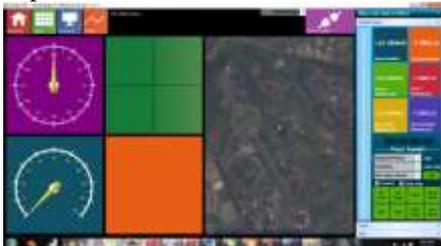
- a. *Title Bar*, menunjukkan nama *project* (program aplikasi yang sedang dibuat).
- b. *Menu Bar*, berisi menu – menu utama yang dimiliki Visual Studio dan pada masing – masing menu terdapat sub – sub menu yang lebih spesifik.
- c. *Form*, tempat untuk merancang aplikasi yang sedang kita buat.
- d. *Toolbox*, terdiri dari beberapa *class object* (alat) yang digunakan dalam pembuatan aplikasi.
- e. *Project Explorer Windows*, jendela untuk menampilkan *project-project*, *form-form*, atau modul-modul yang terlihat dalam pembuatan aplikasi.
- f. *Properti Windows*, Jendela untuk menampilkan dan mengubah *property-property* yang dimiliki oleh sebuah *object*.
- g. *Code Windows*, Jendela yang digunakan untuk menampilkan atau mengetikkan kode program.
- h. *Form Layout Windows*, jendela yang menunjukkan posisi relative form terhadap layar monitor.

### 2.3.2 Perancangan Perangkat Lunak pada Visual C# 2010

Perangkat lunak Visual C# 2010 menyediakan beberapa fasilitas yang dapat digunakan untuk perancangan sistem pada tugas akhir ini, antara lain integrasi data GPS dengan Google Map, komunikasi serial antara mikrokontroler dengan komputer baik itu dalam menerima data maupun mengirim perintah dari dan ke mikrokontroler, penentuan titik awal serta titik target posisi UGV, visualisasi data yang dikirim oleh mikrokontroler dalam bentuk grafik, dan pembuatan tabel untuk penampungan data tersebut secara sementara. Secara umum perancangan perangkat lunak pada Visual C# 2010 seperti pada Gambar 3.22 terdiri atas :

#### A. Konversi data dari GPS ke *Google Maps*

Format data GPS yang dikirim oleh mikrokontroler sudah dalam format *dd.dddd* sehingga data dapat langsung diintegrasikan dengan format data lintang dan bujur pada *Google Maps*.



Gambar 4. Tampilan beranda pada Visual C# 2010



Gambar 5. Tampilan *Maps* pada *Interface*

#### B. Tampilan Pembacaan Sudut Orientasi dan Kecepatan UGV pada *Interface*

Pembacaan orientasi memungkinkan pengguna untuk melihat posisi dari UGV. Nilai orientasi didapat dari GPS melalui pembacaan dari mikrokontroler.



Gambar 6. Tampilan Sudut Orientasi UGV pada *Interface*



Gambar 7. Tampilan Kecepatan UGV pada *Interface*

#### C. Program komunikasi antara komputer dan mikrokontroler

Adapun isi dari program ini bertujuan untuk menangani data GPS dan kompas digital yang diterima komputer. Pada Visual C# 2010, komponen komunikasi serial sudah disediakan. Agar komputer dengan mikrokontroler bisa saling berkomunikasi maka perlu dilakukan pengaturan terlebih dahulu dengan mengatur *properties* dari *serial port*.

Data yang digunakan untuk menyimpan data yang dikirim dari mikrokontroler berupa bujur, lintang, bujur awal, lintang awal, bujur target, lintang target, error x, error y, error jarak, error orientasi, data x, data y, orientasi, *crisp motor*, *crisp servo*, status GPS dan status mode kontrol.



Gambar 8. Pengaturan port serial program C#.

Data yang dikirim dari serial mikrokontroler terdapat beberapa data yang sebenarnya juga didapat dari user interface atau komputer, namun dalam hal ini sengaja dikirim agar dapat memastikan kepada user apakah data yang telah dikirim dari komputer benar-benar telah diterima oleh mikrokontroler secara lengkap atau tidak.

D. Penentuan titik awal dan titik target UGV

Program ini bertujuan untuk menentukan titik awal acuan posisi UGV dan titik target yang akan dituju oleh UGV. Untuk menentukan titik awal UGV perlu dibuat *event buttonAwal\_Click* pada tombol tersebut. Titik awal berupa posisi lintang dan bujur UGV akan disimpan ke dalam variabel *BujurZero* dan *LintangZero* yang memiliki tipe data *string*. Agar simbol berupa titik awal muncul pada peta, *class* dengan nama *GMarkerGoogle* dan *GMarkerRect* perlu diinisialisasi sehingga didapatkan objek *m* dan *mBorders*.



Gambar 9. Tombol Titik Awal dan Target pada Interface

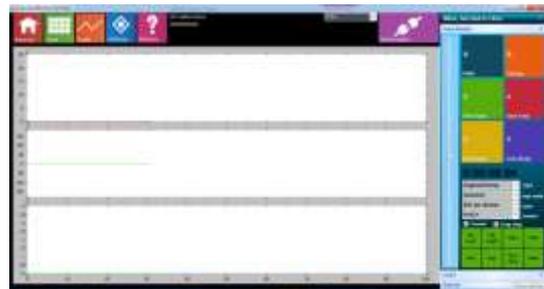
Untuk menentukan titik target dibuat *event buttonTarget\_Click* pada tombol tersebut. Titik target berupa posisi lintang dan bujur yang akan dituju disimpan ke dalam variabel *targetLintang* dan *targetBujur* dengan tipe data *string*. Variabel *targetLintang* dan *targetBujur* merupakan selisih antara posisi bujur dan lintang yang telah ditentukan lewat peta dengan variabel titik awal yang telah diberikan pada saat penentuan titik awal UGV..

E. Program untuk menampilkan grafik dan tabel data

Program ini bertujuan untuk menampilkan data-data yang diterima oleh komputer dapat ditampilkan dalam bentuk tabel dan grafik. Untuk bisa menampilkan data dalam grafik, variable pada grafik yang telah didapat melalui komunikasi serial perlu dikonversi terlebih dahulu dari tipe data *string* menjadi *double*.



Gambar 10. Tampilan tab Tabel pada interface



Gambar 11. Tampilan tab Grafik pada Interface

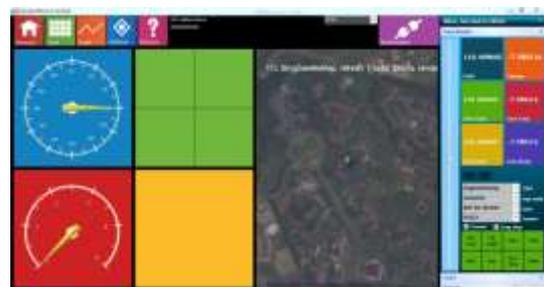
### 3. Hasil dan Analisa

Pengujian yang akan dilakukan pada penelitian ini meliputi pengujian sistem yaitu berupa pengujian tampilan *interface* serta cara mengoperasikan UGV melalui *interface*.

#### 3.1 Pengujian Sistem Interface

Pengujian sistem interface meliputi cara mengoperasikan UGV melalui *interface*. Sebelum mengoperasikan UGV terlebih dahulu menjalankan program *interface*.

##### 3.1.1 Tampilan Halaman Awal Interface



Gambar 12. Tampilan Halaman Awal Interface

Perangkat USB to RS232 TTL PL2303HX dihubungkan ke komputer. Kemudian pengguna memilih *port* yang digunakan dengan melihat pada *device manager* yang terdapat pada komputer, lalu memilih *port* yang sama yang ada didalam *interface*. *Port* yang telah dipilih kemudian pengguna menekan tombol sambungkan.

Tombol ini digunakan untuk menyambungkan modul YS-1020UA yang terhubung dengan komputer dengan modul YS-1020UA yang terhubung mikrokontroler pada UGV.



**Gambar 13.** Port pada *Interface*



**Gambar 14.** Tombol Sambungkan



**Gambar 15.** Tampilan pergerakan UGV dengan data GPS pada *Interface*

Pada Gambar 15 terlihat bahwa setelah tombol start ditekan maka peta akan langsung memperbarui lokasi sesuai dengan masukan data GPS yang ada pada UGV. Data GPS pada *interface* ditandai dengan titik berwarna merah. Gambar diatas menunjukkan tampilan pergerakan UGV berdasarkan pada data GPS.

### 3.1.2 Pengujian *Interface* pada *Autopilot* UGV

Pengujian dilakukan dengan menentukan titik awal dan titik target UGV. Titik awal UGV didapat dari data GPS dimana pada *interface* data GPS disimbolkan dengan titik berwarna merah. Titik target ditentukan sendiri oleh pengguna dengan baik itu satu target atau banyak target. Titik awal UGV ditunjukkan dengan warna hijau sedangkan titik target ditunjukkan dengan warna merah. Setelah target ditentukan maka selanjutnya yang harus dilakukan adalah memilih mode *fuzzy* sebagai kontrol yang digunakan untuk menjalankan UGV menuju target yang ditentukan.



**Gambar 16.** Titik Awal dan Titik Target UGV

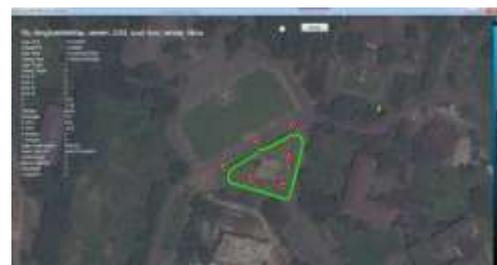
Gambar 16 menunjukkan pergerakan UGV dari titik awal yang ditunjukkan dengan warna hijau menuju ke titik target yang ditunjukkan dengan warna merah. Pengguna dapat memilih data tracking UGV yang ingin ditampilkan. Terdapat 3 pilihan data yang pertama data GPS, kedua data dari *rotary encoder* dan ketiga data *complementary filter*. Tampilan *interface* pada Gambar 17 menunjukkan bahwa data berasal dari GPS yang terdapat pada UGV. Tampilan *interface* pada Gambar 18 menunjukkan data pergerakan UGV didapat dari *rotary encoder* yang terdapat pada UGV. Sedangkan tampilan *interface* pada Gambar 19 menunjukkan pergerakan UGV yang didapatkan dari data gabungan enkoder dan GPS.



**Gambar 17.** Tampilan Data GPS



**Gambar 18.** Tampilan Data *Rotary Encoder*



**Gambar 19.** Tampilan Data *Complementary Filter*

### 3.1.3 Pengujian Radio Frekuensi pada Interface

Pengujian dilakukan dengan menjalankan UGV yang telah terintegrasi didalamnya radio frekuensi dan mengirimkan data berupa increment variabel, kemudian UGV dijalankan menjauhi terminal. Pada terminal radio frekuensi lainnya dibaca dengan menggunakan komputer. Pengamatan data dilakukan sampai data yang terkirim mulai hilang atau radio frekuensi tidak menerima data secara utuh. Dari pengujian yang dilakukan didapatkan jarak aman yang mampu ditempuh oleh radio frekuensi yang digunakan sejauh 80 meter.

### 3.2 Hasil Pengujian Interface

Proses pengujian *interface* meliputi tampilan UGV dalam menuju target secara *autopilot*. Caranya dengan menentukan titik awal dan titik target pada *interface*. Pada pengujian kali ini ditentukan UGV bergerak menuju satu target, dua target dan tiga target. Selain itu juga di tampilkan gambar grafik dari masing-masing pengujian.



Gambar 20. Tampilan *Interface* pengujian UGV menuju 1 target



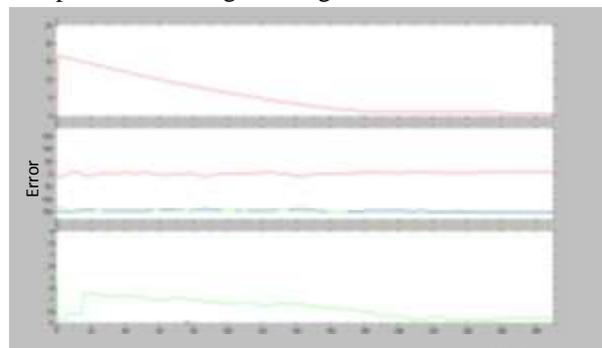
Gambar 21. Tampilan *Interface* Pengujian UGV menuju 2 target



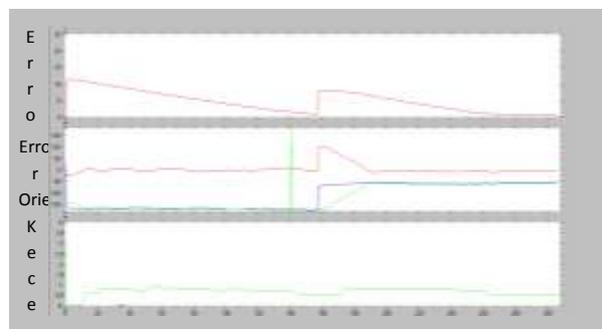
Gambar 22. Tampilan *Interface* Pengujian UGV menuju 3 target

Gambar 20 menunjukkan tampilan *interface* saat UGV bergerak menuju 1 target. Gambar 21 menunjukkan tampilan *interface* saat UGV bergerak dengan 2 target. Gambar 22 menunjukkan tampilan *interface* saat UGV bergerak menuju 3 target. Data yang didapat untuk mengetahui pergerakan UGV pada gambar diatas menggunakan data dari *complementary filter*.

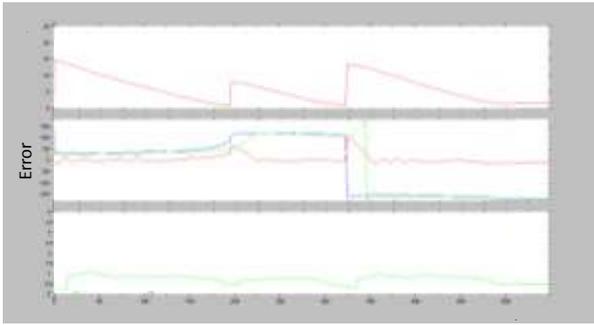
Gambar 23 menampilkan grafik error jarak, error orientasi dan kecepatan UGV dengan 1 target. Gambar 24 menampilkan grafik error jarak, error orientasi dan kecepatan UGV dengan 2 target. Sedangkan gambar 25 menampilkan grafik error jarak, error orientasi dan kecepatan UGV dengan 3 target.



Gambar 23. Tampilan Grafik error jarak, error orientasi dan kecepatan UGV dengan 1 target



Gambar 24. Tampilan Grafik error jarak, error orientasi dan kecepatan UGV dengan 2 target



Gambar 25. Tampilan Grafik error jarak, error orientasi dan kecepatan UGV dengan 3 target

#### 4. Kesimpulan

Dari pengujian yang telah dilakukan disimpulkan bahwa interface mampu menampilkan pergerakan UGV menuju berbagai target. Hal ini dapat dilihat pada gambar bahwa UGV mampu mengikuti target yang telah diberikan baik. Jarak maksimal *interface* mampu menerima data dari UGV sejauh 80 meter. Selain itu interface mampu menampilkan grafik pergerakan UGV saat menuju target. Untuk pengembangan selanjutnya pada UGV dapat ditambahkan kamera sehingga pemgguna dapat melihat medan secara langsung.

#### Referensi

- [1]. Goge, Douglas W., *A Brief History of Unmanned Ground Vehicle (UGV) Development Efforts*, Unmanned System Magazine, United States of America, 1995.
- [2]. Martinic, Gary, *The proliferation, diversity and utility of ground-based robotic technologies*, Canadian Military Journal, Vol. 14, No. 4, Kanada 2014
- [3]. Miaolei Zhou dan Shanbo He, *Research of Autonomous Navigation Strategy for an Outdoor Mobile Robot*, College of Communication Engineering, Jilin University, Changchun, 2014
- [4]. Nurpulaela, Lela, *Perancangan Human Machine Interface Dengan VB6.0 Untuk Visualisasi dan Monitoring Kecepatan Motor AC 3Phase pada PLC OMRON CJ1M*, Teknik Elektro, STT Bina Tunggal, Bekasi, 2012
- [5]. Laksono, Teguh Pudar Mei, *Sistem Scada Water Level Control Menggunakan Software Wonderware Intouch*, Teknik Elektro, Universitas Negeri Semarang, Semarang, 2013
- [6]. Wahyu, Fahmi, *Desain Kontrol Autopilot menggunakan GPS pada Kapal*, Jurusan Teknik Elektronika, PENS-ITS, Surabaya, 2011.
- [7]. Putra, Agfianto Eko, *Belajar Mikrokontroler Teori & Aplikasi*, Gava Media Yogyakarta, 2006.
- [8]. Rohde, Mitchell M., Victor E. Perlin, Karl D. Iagnemma, Robert M. Lupa, Steven M. Rohde, James Overholt, dan Graham Fiorani, *Semi Autonommous UGV Control with Intuitive Interface*, Robotic Mobility Group, Massachusetts Institute of Technology, United State of America.2008.
- [9]. Stefan, Jeff, *Navigating with GPS*, Circuit Cellar Magazine, USA, 2000.
- [10]. Sodri, *Pengendalian Miniatur Instalasi Listrik Menggunakan Mikrokontroler Arduino Mega Terintegrasi HMI*, Teknik Elektro, Universitas Pendidikan Indonesia, Jakarta, 2013.
- [11]. Kuswandi, Son., *Kendali Cerdas Teori dan Aplikasi praktisnya*, Penerbit Andi, Yogyakarta, 2007.
- [12]. Midian, Jason, *Perancangan Sistem Antarmuka Berbasis HMI Untuk Aplikasi Mesin Etching PCB*, Departemen Teknik Elektro, Universitas Indonesia, Jakarta, 2009.
- [13]. Zulkha, Dimasyqi, *Rancang bangun sistem pengendali kecepatan motor dc dengan menggunakan metode pengendali PID pada mobil listrik (zec-01)*, Jurusan Teknik Fisika – Fakultas Teknologi Industri ITS Surabaya, 2012
- [14]. Luthfa, M Fairuz, *Perancangan sistem pemantauan posisi untuk pejalan kaki menggunakan fusi data mems sensor accelerometer, magnetometer dan GPS*, Tugas Akhir Teknik Elektro Universitas Diponegoro, Semarang, 2014