



TUGAS AKHIR - TM091486

SYSTEM STRATEGY MONITORING PADA SAPUANGIN SURYA II

**Nadhira Raafianti
NRP 2110 100 030**

**Dosen Pembimbing
Dr. Muhammad Nur Yuniarto, ST**

**JURUSAN TEKNIK MESIN
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya 2014**



FINAL PROJECT - TM 091486

SYSTEM STRATEGY MONITORING FOR SAPUANGIN SURYA II

**Nadhira Raafianti
NRP 2110 100 030**

**Advisor
Dr. Muhammad Nur Yuniarto, ST**

**DEPARTMENT OF MECHANICAL ENGINEERING
Faculty of Industrial Technology
Sepuluh Nopember Institute of Technology
Surabaya 2014**

SYSTEM STRATEGY MONITORING PADA SAPUANGIN SURYA II

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Teknik
pada
Bidang Studi Otomasi Industri
Jurusan Teknik Mesin
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember

Oleh :

Nadhira Raafianti
NRP. 2110 100 030

Disetujui oleh Tim Penguji Tugas Akhir :

1. Dr. M. Nur Yuniarto
NIP. 197506301998021001 (Pembimbing)
2. Arif Wahyudi, ST, MT, PhD
NIP. 197303222001121001 (Penguji I)
3. M. Khoirul Effendi, ST, MSc, Eng
NIP. 198204142010121001 (Penguji II)
4. Dr. Eng. Sutikno, ST, MT
NIP. 197407032000031001 (Penguji III)

SURABAYA
Juli 2014

SYSTEM STRATEGY MONITORING PADA SAPUANGIN SURYA II

Nama Mahasiswa : Nadhira Raafianti
NRP : 2110100030
Jurusan : Teknik Mesin FTI-ITS
Dosen Pembimbing : Dr. M. Nur Yuniarto

ABSTRAK

Sektor transportasi khususnya kendaraan bermotor tercatat sebagai pengonsumsi minyak bumi terbesar, yakni 47 persen BBM di Indonesia (LIPI, 2003). Sebagai wujud peran serta ITS dalam melaksanakan pengembangan bidang energi untuk mengatasi masalah krisis energi di Indonesia, dibentuk ITS Solar Car Racing Team. Tim ini telah membuat mobil bertenaga surya, SapuAngin Surya. Salah satu evaluasi dari mobil pada tahun 2013 adalah kurangnya manajemen energi. Dari kurangnya manajemen energi tersebut konsumsi energi pada mobil SapuAngin Surya tidak seimbang dan menyebabkan baterai yang digunakan cepat habis. Terdapat beberapa cara untuk mengatur asupan energi dari mobil. Salah satu caranya adalah dengan membuat sebuah program System Strategy Monitoring untuk mobil SapuAngin Surya II yang akan mengikuti kompetisi World Solar Challenge 2015 mendatang.

Tugas akhir ini adalah membahas System Strategy Monitoring untuk SapuAngin Surya menggunakan software Matlab. Input yang dibutuhkan untuk pembuatan System Strategy Monitoring ini berasal dari data dan juga langsung berasal dari mobil surya. Data dari mobil surya tersebut berupa data serial yang akan ditransfer dari mobil surya ke komputer menggunakan modul Wireless RF yaitu Xbee dan di convert ke USB menggunakan FTDI sebagai serial to USB converter nya. Lalu data tersebut akan dibaca oleh Matlab dan dimasukkan ke dalam perumusan selanjutnya. Setelah semua data yang didapat telah dimasukkan dalam perumusan matematis, maka selanjutnya akan

dibuat program untuk menentukan strategi balap dan monitoring nya.

Dari pembuatan System Strategy Monitoring untuk mobil SapuAngin Surya II ini dihasilkan sistem untuk monitoring kondisi dan kinerja mobil secara real-time. Selain itu juga didapatkan hasil berupa tampilan untuk monitoring kinerja mobil secara real time selama kompetisi berlangsung. Monitoring ini juga akan menjadi acuan strategi selama perlombaan World Solar Challenge 2015 berlangsung.

Kata kunci : Solar Car, System Strategy Monitoring, Konsumsi Energi, Profil Kecepatan.

SYSTEM STRATEGY MONITORING FOR SAPUANGIN SURYA II

Name : Nadhira Raafianti
NRP : 2110100030
Department : Mechanical Engineering FTI-ITS
Advisor : Dr. M. Nur Yuniarto

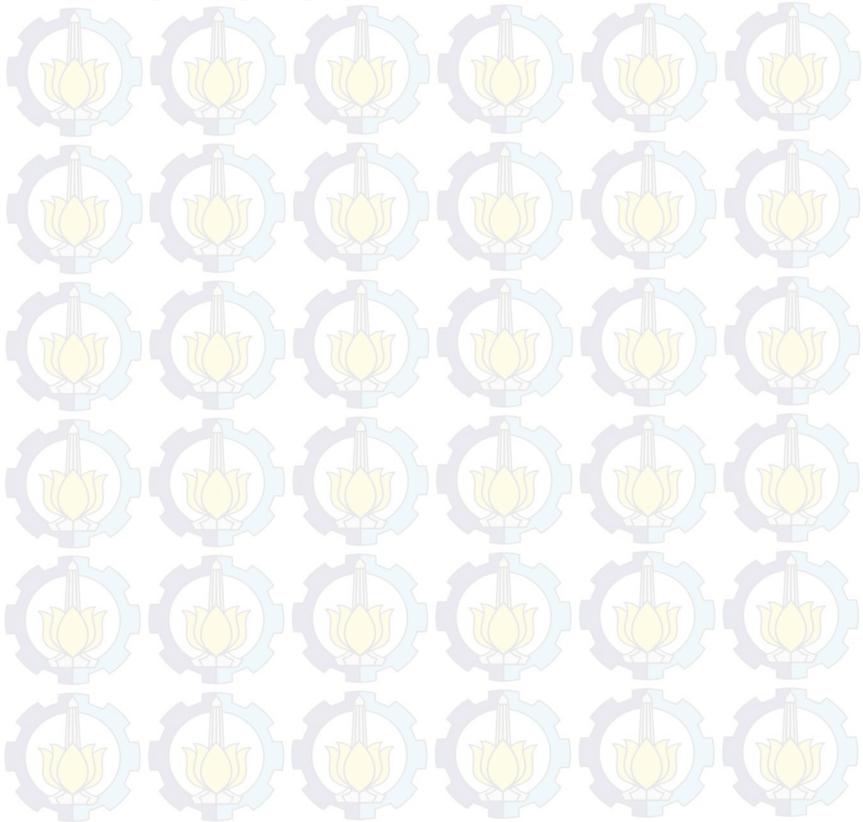
ABSTRACT

Transportation sector, especially motor vehicles, noted as the highest fuel consumer on earth as it consumed 47% fuel in Indonesia (LIPI, 2003). As ITS' participation in developing renewable energy to anticipate the energy crisis in Indonesia, ITS Solar Car Racing Team were formed and created a solar-powered electric vehicle, SapuAngin Surya. One of the evaluations from the car is that it has less energy management. From that bad management of energy, it caused the unbalanced energy consumption on SapuAngin Surya and it also caused the batteries drained in a short time. There's many way to manage the energy consumption of the car. One of them is to make a System Strategy Monitoring software for SapuAngin Surya II that will compete in World Solar Challenge 2015.

This research is to make a software named System Strategy Monitoring for SapuAngin Surya II using Matlab. Inputs that needed for this System Strategy Monitoring are datas that followed from solar car. The inputs are serial datas that transferred from solar car into the computer using a wireless RF module named Xbee, and the datas are converted into USB using FTDI as the serial to USB converter. Then the datas will be read in Matlab and will be proceed into the next mathematics calculations. After all the datas needed were available and proceeded into the mathematics calculations, then System Strategy Monitoring software was ready to designed and made.

From the making of System Strategy Monitoring for SapuAngin Surya II, it resulted a system for monitoring the condition and performance of the solar car in real-time. In addition, the results obtained was a display for monitoring the performance of the solar car in real-time condition during the competition. This monitoring will also be a reference strategy for the upcoming World Solar Challenge 2015.

Keyword : Solar Car, System Strategy Monitoring, energy consumption, speed profile.



KATA PENGANTAR

Alhamdulillah, puji syukur kehadirat Allah Subhanahu wa Ta'ala yang telah melimpahkan rahmat-Nya sehingga Tugas Akhir yang saya kerjakan ini dapat terselesaikan dengan baik.

Laporan Tugas Akhir ini saya susun sesuai dengan bidang studi saya Otomasi Industri dengan judul ***System Strategy Monitoring Pada Sapuangin Surya II***. Saya juga tidak lupa mengucapkan terima kasih kepada.

1. Bapak Muhammad Nur Yuniarto selaku dosen pembimbing Tugas Akhir.
2. Bapak Arif Wahyudi, ST, MT, PhD., Bapak M. Khoirul Effendi, ST, MSc, Eng., dan Bapak Dr. Eng. Sutikno, ST, MT selaku dosen penguji Seminar Proposal dan Sidang Tugas Akhir.
3. Bapak Wawan Aries Widodo selaku dosen wali yang selalu memberi motivasi.
4. Orang tua, adik-adik, Sekar, Naura, Nadia, Sasha, Amira, Neysa dan keluarga penulis yang telah memberikan motivasi.
5. Seluruh keluarga Lembaga Bengkel Mahasiswa Mesin, Pimen, Tegar, Oky, Ateng, Pindra, Opi, Aji, Hisyam, Botol, Tenyom, Genjik, Satpam, Mas El, Mas Cete, dan lain lain yang telah menjadi keluarga kedua.
6. Seluruh keluarga ITS *Solar Car Racing Team*, Mba Pentol, Mas Tono, Mas Squid, Mas Grangsang, Aufar, Husen, Jordy, Seno, Monyet, Ainun, dan lain lain yang selalu menyemangati.
7. Seluruh keluarga laboratorium Otomasi Industri.
8. Iqo, Elsa, Dea, sahabat dan *partner* terbaik.
9. Seluruh teman-teman dari Ikasmariagitma, Arida, Laras, Avatar, Ardi, Fira, dan lain lain yang selalu memberi dukungan.
10. Semua pihak yang telah membantu atas terselesaikannya laporan Tugas Akhir ini, yang tidak bisa disebutkan satu-

persatu. Semoga keikhlasan dan kebaikan yang telah diberikan mendapatkan balasan dari Allah SWT

Karena keterbatasan waktu dan kemampuan, saya menyadari laporan Tugas Akhir ini masih terdapat kekurangan, kesalahan, dan keterbatasan sehingga masih belum sempurna. Untuk itu saya mengharapkan kritik dan saran demi kesempurnaan Tugas Akhir ini. Semoga Tugas Akhir ini dapat bermanfaat bagi saya pribadi dan bangsa.

Surabaya, Juli 2014

Penulis

DAFTAR ISI

Halaman Judul	i
Halaman Pengesahan	ii
Abstrak	iii
Abstract	v
Kata Pengantar	vii
Daftar Isi	ix
Daftar Gambar	xiii
Daftar Tabel	xvii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan Penelitian	3
1.5 Manfaat Penelitian	3
BAB II TINJAUAN PUSTAKA DAN DASAR TEORI	5
2.1 Pendahuluan	5
2.2 Sistem Strategi Tipikal	5
2.3 Parameter-Parameter selama Perlombaan	6
2.3.1 Konsumsi Energi	7
2.3.2 Rute Perlombaan	7
2.3.3 Manajemen Baterai	10
2.3.4 <i>Solar Energy</i>	12
2.3.5 Model Dinamis dari Mobil SapuAngin Surya	14
2.3.6 Profil Kecepatan Mobil SapuAngin Surya	16
2.4 <i>Graphical User Interfaces</i> (GUI)	18
2.4.1 GUI pada Matlab	19
2.4.2 <i>System Strategy Monitoring</i> dengan GUI Matlab	20
2.4.3 Komunikasi Perangkat Keras Menggunakan Perangkat Lunak Matlab	21
2.4.4 Konversi Data	22
2.5 <i>System Strategy Monitoring</i> pada SapuAngin Surya II	23
BAB III METODOLOGI PENELITIAN	27
3.1 <i>Flowchart</i> Penelitian	28
3.1.1 Studi Literatur	29

3.1.2	Perumusan Masalah.....	29
3.1.3	Pengumpulan Data.....	30
3.1.4	Perumusan dan Perhitungan Matematis.....	30
3.1.5	Pembuatan dan Simulasi <i>Software System Strategy Monitoring</i>	30
3.1.6	Analisa dan Pembahasan.....	31
3.1.7	Kesimpulan dan Saran.....	31
3.2	<i>Flowchart</i> Pembuatan <i>System Strategy Monitoring</i>	32
3.2.1	<i>Data Input</i>	33
3.2.2	Proses: Persiapan Peralatan (Xbee-FTDI dan Xbee-Arduino).....	34
3.2.3	Proses: <i>Connect</i> Xbee Dengan Komputer.....	34
3.2.4	Proses: Konfigurasi Xbee Pada X-CTU.....	35
3.2.5	Proses: Konfigurasi Xbee Pada Arduino.....	37
3.2.6	Proses: Konfigurasi Xbee Pada Matlab.....	37
3.2.7	Proses: Komunikasi Antar Xbee Dengan Matlab.....	38
3.2.8	Proses: Menampilkan Data Secara <i>Real-Time</i> Pada Matlab.....	38
3.2.9	Proses: Membuat GUI Pada Matlab.....	39
3.2.10	Proses: Menampilkan Grafik Secara <i>Real-Time</i> Pada GUI Matlab.....	40
3.2.11	Proses: Pengambilan Data Secara <i>Real-Time</i> Menggunakan <i>Hardware</i> Yang Telah Dirangkai.....	40
3.2.12	<i>Data Output</i>	40
3.3	Data Pendukung.....	41
3.3.1	Spesifikasi SapuAngin Surya II.....	41
3.3.2	<i>Drive Cycle</i> Selama Perlombaan.....	42
3.3.2.1	<i>Drive Cycle</i> Pada Hari Pertama.....	42
3.3.2.2	<i>Drive Cycle</i> Pada Hari Kedua.....	43
3.3.2.3	<i>Drive Cycle</i> Pada Hari Ketiga.....	44
3.3.2.4	<i>Drive Cycle</i> Pada Hari Keempat.....	46
BAB IV PERHITUNGAN DAN ANALISA DATA.....		49
4.1	Perhitungan dan Model Dari Potensiometer.....	49
4.2	Analisa <i>System Strategy Monitoring</i>	61
4.3	Analisa Data.....	66

4.3.1 Grafik Kecepatan Fungsi Waktu Hari Pertama.....	66
4.3.2 Grafik Kecepatan Fungsi Waktu Hari Kedua.....	68
4.3.3 Grafik Kecepatan Fungsi Waktu Hari Ketiga.....	71
4.3.4 Grafik Kecepatan Fungsi Waktu Hari Keempat.....	74
4.3.5 Grafik Konsumsi Energi Hari Pertama.....	77
4.3.6 Grafik Konsumsi Energi Hari Kedua.....	78
4.3.7 Grafik Konsumsi Energi Hari Ketiga.....	79
4.3.8 Grafik Konsumsi Energi Hari Keempat.....	80
BAB V KESIMPULAN DAN SARAN.....	83
5.1 Kesimpulan.....	83
5.2 Saran.....	85
DAFTAR PUSTAKA.....	87
LAMPIRAN.....	89



DAFTAR GAMBAR

Gambar 2.1	<i>Power System</i> dari Mobil Surya	5
Gambar 2.2	Sistem Strategi Mobil Surya Tipikal	6
Gambar 2.3	Grafik <i>Road Elevation</i>	10
Gambar 2.4	GUIDE <i>Quick Start</i> pada Matlab	19
Gambar 2.5	<i>Stanford Solar Car Team Real Time Monitoring</i>	20
Gambar 2.6	Antarmuka Serial Menggunakan Matlab	22
Gambar 2.7	Sistem Pada Mobil Tenaga Surya	23
Gambar 3.1	<i>Framework Pengerjaan System Strategy Monitoring</i>	27
Gambar 3.2	<i>Flowchart</i> Penelitian	28
Gambar 3.3	<i>Flowchart</i> Pembuatan <i>System Strategy Monitoring</i>	32
Gambar 3.4	Modul Xbee Pro S3B	34
Gambar 3.5	FT232RL Pada Xbee USB <i>Adapter</i>	35
Gambar 3.6	Menyambungkan FT232RL ke Komputer	35
Gambar 3.7	Jendela Konfigurasi Xbee Pada X-CTU	36
Gambar 3.8	Konfigurasi Serial/USB <i>Port Parameters</i>	37
Gambar 3.9	Grafik <i>Real-Time</i> Pada Matlab	38
Gambar 3.10	<i>Layout System Strategy Monitoring</i> Pada SapuAngin Surya II	39
Gambar 3.11	SapuAngin Surya Generasi Kedua	41
Gambar 3.12	Grafik <i>Drive Cycle</i> Pada Hari Pertama	42
Gambar 3.13	Grafik <i>Drive Cycle</i> Pada Hari Kedua	43
Gambar 3.14	Grafik <i>Drive Cycle</i> Pada Hari Ketiga	45
Gambar 3.15	Grafik <i>Drive Cycle</i> Pada Hari Keempat	46
Gambar 4.1	Pemasangan Rangkaian Potensiometer	49
Gambar 4.2	Blok Diagram Sistem Pada Potensiometer	50
Gambar 4.3	<i>Transfer Function</i> Pada <i>Preamplifier</i>	51
Gambar 4.4	<i>Transfer Function</i> Pada <i>Power Amplifier</i>	51
Gambar 4.5	<i>Transfer Function</i> Pada <i>Motor and Wheel</i>	52
Gambar 4.6	Blok Diagram Pada Sistem	53
Gambar 4.7	Blok Diagram Pada Simulink	54

Gambar 4.8	<i>Ramp Response</i> Pada Sistem.....	54
Gambar 4.9	Grafik Perbandingan <i>Time to Accelerate</i> dan Kecepatan.....	57
Gambar 4.10	Pemodelan Putaran Potensiometer Menggunakan <i>Software</i> MATLAB/Simulink.....	57
Gambar 4.11	<i>Block Parameter</i> Untuk Blok <i>Query Instruments</i>	58
Gambar 4.12	<i>Block Parameter</i> untuk Blok <i>Unit Delay</i>	59
Gambar 4.13	Pemodelan Konsumsi Energi Menggunakan <i>Software</i> MATLAB/Simulink.....	60
Gambar 4.14	Alur Komunikasi Serial Pada <i>System Strategy Monitoring</i>	61
Gambar 4.15	<i>System Strategy Monitoring</i> Pada SapuAngin Surya II.....	62
Gambar 4.16	<i>Monitoring</i> Panel Surya Pada <i>System Strategy Monitoring</i>	62
Gambar 4.17	<i>Monitoring</i> Motor Pada <i>System Strategy Monitoring</i>	63
Gambar 4.18	<i>Monitoring</i> Baterai Pada <i>System Strategy Monitoring</i>	63
Gambar 4.19	Grafik Konsumsi Energi dan Profil Kecepatan pada <i>System Strategy Monitoring</i>	64
Gambar 4.20	Tampilan Grafik Kecepatan Hari Pertama pada <i>System Strategy Monitoring</i>	66
Gambar 4.21	Grafik Kecepatan Fungsi Waktu Hari Pertama.....	67
Gambar 4.22	Grafik Kecepatan Fungsi Waktu Hari Pertama Pada Detik 0-5000.....	68
Gambar 4.23	Tampilan Grafik Kecepatan Hari Kedua pada <i>System Strategy Monitoring</i>	69
Gambar 4.24	Grafik Kecepatan Fungsi Waktu Hari Kedua.....	69
Gambar 4.25	Grafik Kecepatan Fungsi Waktu Hari Kedua Pada Detik 0-5000.....	70
Gambar 4.26	Tampilan Grafik Kecepatan Hari Ketiga pada <i>System Strategy Monitoring</i>	71
Gambar 4.27	Grafik Kecepatan Fungsi Waktu Hari Ketiga.....	72

Gambar 4.28	Grafik Kecepatan Fungsi Waktu Hari Ketiga Pada Detik 0-5000.....	73
Gambar 4.29	Tampilan Grafik Kecepatan Hari Keempat pada <i>System Strategy Monitoring</i>	74
Gambar 4.30	Grafik Kecepatan Fungsi Waktu Hari Keempat.....	74
Gambar 4.31	Grafik Kecepatan Fungsi Waktu Hari Keempat Pada Detik 0-5000.....	75
Gambar 4.32	Grafik Perbandingan Konsumsi Energi Referensi dan Aktual Hari Pertama.....	77
Gambar 4.33	Grafik Perbandingan Konsumsi Energi Referensi dan Aktual Hari Kedua.....	78
Gambar 4.34	Grafik Perbandingan Konsumsi Energi Referensi dan Aktual Hari Ketiga.....	79
Gambar 4.35	Grafik Perbandingan Konsumsi Energi Referensi dan Aktual Hari Keempat.....	80



DAFTAR TABEL

Tabel 2.1	<i>Land Marks</i> untuk <i>World Solar Challenge</i> 2013	8
Tabel 2.2	Instruksi Konversi Data	23
Tabel 4.1	Parameter Pada Sistem	50
Tabel 4.2	Nilai Parameter-Parameter Pada Sistem	52
Tabel 4.3	Tabel Perhitungan Kecepatan dan <i>Time to Accelerate</i>	56



BAB I PENDAHULUAN

1.1 Latar Belakang

Sektor transportasi khususnya kendaraan bermotor tercatat sebagai pengonsumsi minyak bumi terbesar, yakni 47 persen BBM di Indonesia (LIPI, 2003). Jumlah total kendaraan bermotor di Indonesia pada tahun 2012 berdasarkan data Badan Pusat Statistik (BPS) lebih dari 94 juta kendaraan. Tingkat pertumbuhan penambahan jumlah kendaraan sebesar 5-7% per tahun atau sebesar 3-4 juta per tahun. Dengan jumlah kendaraan bermotor tersebut, berjuta-juta liter BBM dikonsumsi per hari untuk memenuhi kebutuhan. Sebagai wujud peran serta ITS dalam melaksanakan pengembangan bidang energi untuk mengatasi masalah krisis energi di Indonesia, dibentuk ITS *Solar Car Racing Team* yang telah mengikuti kompetisi mobil surya tingkat Internasional di Australia yaitu *World Solar Challenge 2013*. Dalam kompetisi *World Solar Challenge*, tiap tim diharuskan membuat sebuah mobil listrik menggunakan energi surya untuk berjalan dan dilombakan dalam rute yang telah ditetapkan.

Pada kompetisi yang telah diikuti oleh SapuAngin Surya pada tahun 2013 yang lalu, terdapat banyak evaluasi selama perlombaan. Salah satunya adalah kurangnya persiapan strategi selama *race* berlangsung, sehingga jarak tempuh dari mobil SapuAngin Surya yang diperoleh kurang maksimal. Hal tersebut disebabkan oleh kurangnya manajemen energi yang baik. Dari kurangnya manajemen energi tersebut konsumsi energi pada mobil SapuAngin Surya tidak seimbang dan menyebabkan baterai yang digunakan cepat habis. Konsumsi energi yang tidak seimbang itu juga mempengaruhi jarak tempuh dari mobil.

Asupan dan keluaran dari energi pada mobil sangat dipengaruhi oleh banyak parameter. Terdapat beberapa cara untuk mengatur asupan energi dari mobil. Salah satu caranya adalah dengan membuat sebuah program *System Strategy Monitoring* untuk mobil SapuAngin Surya II yang akan mengikuti kompetisi

World Solar Challenge 2015 mendatang. Pembuatan *System Strategy Monitoring* ini akan menghasilkan sistem dan tampilan *monitoring* kondisi mobil secara *real-time* selama kompetisi berlangsung. Dari sistem *monitoring* tersebut maka dapat dianalisa kondisi mobil secara *real-time* sehingga dapat menjadi acuan tim untuk menyusun strategi *race* yang direkomendasikan selama kompetisi berlangsung.

1.2 Rumusan Masalah

Masalah yang ada dalam tugas akhir ini adalah bagaimana cara untuk membuat program *System Strategy Monitoring* untuk *monitoring* kondisi dari mobil SapuAngin Surya II selama kompetisi *World Solar Challenge 2015* akan berlangsung. Untuk itu maka dapat ditentukan perumusan masalahnya, yaitu:

1. Bagaimana langkah pembuatan software *System Strategy Monitoring* untuk SapuAngin Surya agar didapatkan sistem *monitoring* secara *real-time* selama kompetisi *World Solar Challenge* mendatang.
2. Bagaimana langkah pembuatan software *System Strategy Monitoring* untuk SapuAngin Surya agar didapatkan hasil berupa tampilan untuk *monitoring* kinerja mobil secara *real-time* selama kompetisi *World Solar Challenge* berlangsung.

1.3 Batasan Masalah

Dalam penelitian ini, permasalahan-permasalahan yang ada dibatasi pada beberapa hal yaitu:

1. Rancangan dan properti dari SapuAngin Surya II diperoleh berdasarkan data.
2. Kondisi lintasan yang digunakan berdasar pada lintasan pada *World Solar Challenge 2013* yang lalu.
3. Pembuatan *System Strategy Monitoring* menggunakan software MATLAB (Simulink dan GUIDE).

1.4 Tujuan Penelitian

Tujuan dari penulisan tugas akhir ini adalah:

1. Mendapatkan sistem *monitoring* secara *real-time* selama kompetisi *World Solar Challenge* mendatang dari *System Strategy Monitoring* yang telah dirancang.
2. Mendapatkan tampilan untuk *monitoring* kinerja mobil secara *real-time* selama kompetisi *World Solar Challenge* berlangsung.

1.5 Manfaat Penelitian

Tugas akhir ini memiliki manfaat antara lain:

1. Menjadi acuan dalam perlombaan *World Solar Challenge* 2015.
2. Menjadi acuan strategi selama perlombaan berlangsung.
3. Membantu tim dalam memprediksi target harian dari mobil SapuAngin Surya II.

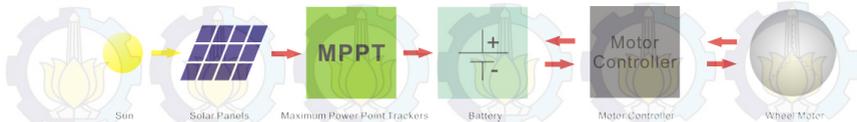


BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Pendahuluan

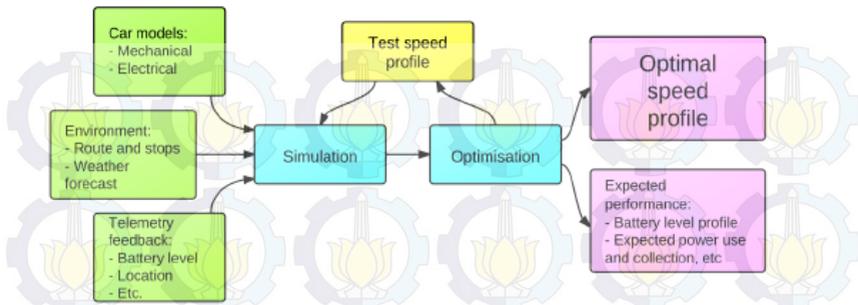
Pada dasarnya prinsip kerja mobil surya hampir sama dengan mobil listrik, hanya berbeda dalam beberapa hal. Perbedaan yang paling utama yaitu pada mobil listrik pengisian dayanya dilakukan menggunakan sumber listrik berupa listrik PLN sedangkan pada mobil surya pengisian dayanya dilakukan oleh sel surya yang mengubah energi surya menjadi energi listrik. Mobil listrik terdiri dari dua subsistem dalam susunan sistemnya, motor listrik sebagai sistem penggerak dan *platform* kendaraan. Komponen utama dalam mobil listrik meliputi: sumber energi listrik, *system control* sebagai pusat control, dan *power converter* sebagai perangkat yang mengkonversi sumber energi listrik dengan kebutuhan variabel dari mobil listrik itu sendiri (Bambang Sri Kaloko et al, 2011).



Gambar 2.1 Power System dari Mobil Surya
(Alexandra Boulgakov, 2012)

2.2 Sistem Strategi Tipikal

Tujuan utama dari pembuatan *strategy monitoring* adalah untuk meminimalisir waktu yang akan digunakan mobil surya untuk menempuh jarak sepanjang 3,000 kilometer dan memonitor konsumsinya agar tidak melebihi dari total energi yang tersedia. Strategi ideal ini dapat dideskripsikan dengan profil kecepatan rekomendasi sepanjang rute perlombaan yang akan dijalankan oleh mobil agar dapat mengkonservasi energi dan dapat menempuh waktu yang seminimal mungkin (Alexandra Boulgakov, 2012).



Gambar 2.2 Sistem Strategi Mobil Surya Tipikal
(Alexandra Boulgakov, 2012)

Sebelum dapat mengkalkulasi strategi yang diinginkan, maka sistem harus dilengkapi dengan data-data yang mencukupi, antara lain:

- Model mekanis dan elektris dari mobil, untuk mengestimasi penggunaan dan pengumpulan energi pada mobil.
- Rute yang akan dilalui oleh mobil (menyesuaikan dengan regulasi perlombaan).

2.3 Parameter-Parameter selama Perlombaan

Pembuatan *System Strategy Monitoring* ini merupakan sebuah kontribusi untuk proyek dari ITS *Solar Car Racing Team* yang bertujuan untuk berpartisipasi dalam kompetisi *World Solar Challenge 2015*, sebuah perlombaan untuk kendaraan bertenaga energi surya. Tujuan dari perlombaan ini adalah untuk mengarahkan mobil surya dari seluruh peserta untuk *race* dari Australia bagian utara ke selatan secepat mungkin. Kompetisi yang diadakan setiap dua tahun sekali ini telah diselenggarakan sejak tahun 1987. Pesaing-pesaing yang berasal dari universitas-universitas dan perusahaan-perusahaan dari seluruh dunia berpartisipasi dan berlomba-lomba mengembangkan mobil bertenaga surya yang inovatif (World Solar Challenge, 2013).

Kendaraan-kendaraan yang berpartisipasi didukung secara sepenuhnya oleh tenaga surya. Energi dari radiasi matahari yang digunakan harus dikonversi oleh sel-sel fotovoltaik selama perlombaan. Kendaraan dapat menggunakan baterai sebagai media penyimpanan energi. Komputer eksternal untuk kendaraan dapat digunakan untuk mengoptimalkan penggunaan mereka selama perlombaan. Tujuan karya ini adalah untuk mengembangkan dan menguji metode untuk tujuan ini. Pekerjaan ini dilakukan untuk kendaraan dengan parameter-parameter yang telah diberikan (Alexandra Boulgakov, 2012).

2.3.1 Konsumsi Energi

Pada mobil surya, konsumsi energi terbesar ada pada kegiatan mekanik seperti *rolling*, *aero* dan *braking*. Energi yang digunakan pada mobil berasal sepenuhnya dari tenaga matahari, dan akan disimpan di dalam baterai. Energi tersebut akan digunakan untuk menggerakkan kendaraan. Dari 100 % energi yang diperoleh dari sinar matahari, sebesar 75% nya merupakan energi yang digunakan oleh motor untuk menggerakkan mobil. Sebesar 12% merupakan energi yang digunakan di sistem mekanik kendaraan seperti *rolling resistance*, *aerodynamics* dan *braking*. Dan 13% sisanya adalah pengaruh eksternal, yaitu dari kecepatan angin, tingkat kekasaran permukaan mobil dan permukaan jalan, elevasi dari jalanan, dan lain-lain (Bambang Sri Kaloko et al, 2011).

2.3.2 Rute Perlombaan

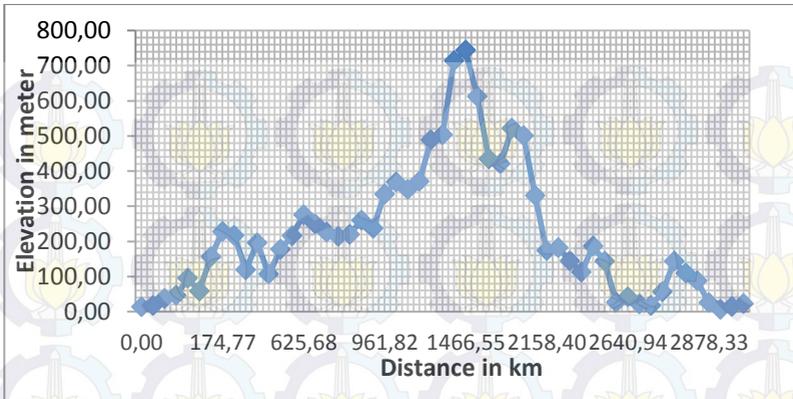
Untuk tujuan mengoptimalkan mobil surya, terdapat 56 *land marks* (kota besar, kota kecil, pemberhentian, dan lain-lain) yang akan diidentifikasi sepanjang rute yang sesuai dengan rute dari *World Solar Challenge 2013*. *Land marks* ini didata beserta dengan jarak yang dihitung mulai *start* hingga *finish* beserta juga dengan *latitude*, *longitude* dan elevasi nya.

Tabel 2.1 Land Marks untuk World Solar Challenge 2013 (Wong, K. & Rizos, C., 1996)

No	Name	Latitude	Length	Height	Distance
1	Darwin	12.40	130.92	13.41	0.00
2	Palmerston	12.47	130.97	17.07	24.96
3	HW36	12.55	131.07	37.49	34.52
4	HW34	12.67	131.07	46.02	48.37
5	Crocodile Farm	12.97	131.10	94.49	83.01
6	Adelaide River	13.23	131.10	58.52	113.02
7	Douglas Hot Spr	13.53	131.42	156.36	157.82
8	Emerald Springs	13.62	131.62	227.38	174.77
9	Pine Creek	13.80	131.83	216.71	205.94
10	X1	14.18	132.03	118.57	269.87
11	X2	14.30	132.08	195.38	290.00
12	Katherine	14.45	132.25	106.98	316.01
13	Venn	14.57	132.50	176.78	343.99
14	Daly Water	16.25	133.38	215.49	590.03
15	X3	16.55	133.35	275.23	625.68
16	Dunmussa	16.65	133.40	249.02	637.41
17	Henry Walker	17.13	133.45	225.86	693.18
18	Causeway Dam	17.22	133.45	213.36	703.64
19	Elliot	17.55	133.53	219.46	742.06
20	X4	17.72	133.63	259.69	762.12
21	X5	17.85	133.68	236.22	777.32
22	HW66	19.43	134.20	333.45	961.82
23	Tennant Creek	19.63	134.18	368.81	987.67
24	Warumungu	19.90	134.20	347.78	1021.06
25	X6	20.20	134.22	370.03	1058.16
26	Ankweley	21.57	133.72	488.59	1228.96
27	Barrow Creek	21.62	133.85	503.22	1237.52
28	Harry Greek	23.22	133.75	713.23	1435.39
29	X12	23.47	133.83	743.71	1466.55
30	Alice Springs	23.68	133.87	611.73	1492.58

No	Name	Latitude	Length	Height	Distance
31	X13	24.73	133.17	434.04	1626.02
32	<u>Eridunda</u>	25.17	133.17	420.62	1679.95
33	<u>Kulgera</u>	25.85	133.28	521.51	1763.51
34	X7	26.08	133.18	500.18	1792.21
35	Mania	27.32	133.65	330.40	1946.39
36	Cooper Pedy	28.97	134.75	173.74	2158.40
37	X8	29.62	135.15	181.97	2241.61
38	Glendambo	30.97	135.73	143.87	2410.93
39	X9	31.23	136.40	112.47	2456.52
40	<u>Nurrungu</u>	31.25	136.80	185.93	2478.46
41	<u>Mount Gunson</u>	31.50	137.98	142.95	2548.29
42	<u>Baroota</u>	32.17	137.97	26.52	2618.84
43	<u>Hesso</u>	32.20	137.58	41.45	2640.94
44	Port Augusta	32.48	137.75	21.34	2671.93
45	B89	33.17	138.07	16.76	2746.84
46	Wamer Town	33.22	138.12	56.08	2752.84
47	X10	33.30	138.17	143.26	2784.77
48	Snow Town	33.52	138.20	109.42	2800.83
49	<u>Lochiel</u>	33.92	138.15	87.78	2826.98
50	<u>Beaufort</u>	34.07	138.98	25.30	2878.33
51	Port Wakefield	34.13	138.22	6.40	2923.64
52	Dublin	34.45	138.35	14.63	2958.20
53	End	34.83	138.58	22.86	3000.51

Dari data yang didapat dari tabel tersebut, didapatkan grafik yang menggambarkan elevasi pada sepanjang jalan rute perlombaan dari *plotting* elevasi versus jarak.



Gambar 2.3 Grafik *Road Elevation*

Dari grafik diatas, dapat dilihat bahwa elevasi pada rute perlombaan akan mengalami kenaikan dan akan mencapai puncaknya pada 743.71 meter dan akan mengalami penurunan kembali hingga akhir perjalanan. Dengan asumsi target jarak yang akan dicapai per hari nya sejauh 750-850 km, maka dapat diasumsikan bahwa hari tersulit adalah hari kedua dimana pada hari kedua mobil surya akan mengalami kenaikan elevasi yang cukup signifikan. Maka dari itu, manajemen konsumsi dan pengisian baterai harian harus direncanakan untuk mendapatkan energi yang cukup besar agar dapat mengatasi *trouble* yang tidak diharapkan selama perjalanan.

2.3.3 Manajemen Baterai

Sesuai dengan peraturan perlombaan untuk *Challenger Class*, selama perlombaan berlangsung baterai pada mobil surya tidak dapat diisi ulang kecuali dari sinar matahari yang didapat selama *race*. Oleh karena itu, manajemen penggunaan baterai sangat diperhatikan dalam perlombaan ini.

Berdasarkan dari rute perlombaan yang telah dibahas di sub bab sebelumnya, maka dapat diketahui tingkat kesulitan yang akan dihadapi per hari nya. Dari tingkat kesulitan pada jalanan didapat bahwa hari terberat adalah hari kedua dimana mobil akan menempuh rute dengan kenaikan elevasi yang signifikan. Tingkat kesulitan pada jalanan akan mempengaruhi konsumsi energi. Konsumsi energi tersebut nantinya akan mempengaruhi kinerja dari mobil surya, sehingga akan diperkirakan akumulasi penggunaan baterai selama *race* untuk *manage* konsumsi energi pada baterai. Konsumsi baterai yang diakumulasikan adalah 25% untuk hari pertama, 35% untuk hari kedua, 10% untuk hari ketiga, dan 20% untuk hari terakhir dan menyisakan 10% sisa di akhir dari perjalanan.

Berikut merupakan *state equation* pada baterai:

$$\dot{B}(t) = -\eta_{bd}(P_c(t) - P_s(t)) \quad (2.1)$$

$$\dot{B}(t) = \eta_{bt}P_s(t) \quad (2.2)$$

Dimana:

- Persamaan (1)= *State equation* untuk baterai selama *discharging*
- Persamaan (2)= *State equation* untuk baterai selama *charging*
- P_c = Energi yang dikonsumsi oleh mobil surya
- P_s = Energi yang didapat dari panel surya

Pemberhentian selama 30 menit pada *control stop* di tengah hari akan menambah 12-17% energi pada baterai, sedangkan *charging* yang dilakukan saat jam berakhirnya *race* (dari pukul 05:00 sore hingga terbenamnya matahari) dan sebelum jam dimulainya kembali *race* (dari terbitnya matahari hingga pukul 08:00 pagi) dapat menambah 5-7% energi pada baterai. Sehingga, terdapat kurang lebih 20% energi tambahan untuk baterai (Moustafa Elshafei, 2013).

2.3.4 Solar Energy

Perhitungan dari radiasi matahari yang terpancar di permukaan bumi pada tempat dan waktu tertentu dapat digunakan untuk memperkirakan energi surya yang tersedia untuk digunakan pada mobil surya selama *race* berlangsung (Moustafa Elshafei, 2013).

Terdapat beberapa parameter-parameter input utama yang digunakan dalam perhitungan energi surya, yaitu:

- Tanggal
- Waktu pada saat ini
- Zona Waktu (dalam satuan jam, tergantung pada Greenwich Timur atau Barat)
- *Latitude* (Φ) yaitu lokasi sudut utara atau selatan dari garis khatulistiwa, $-90^\circ \leq \Phi \leq 90^\circ$
- *Longitude*, yaitu derajat bujur (timur atau barat)

Energi surya yang dihitung diasumsikan dalam kondisi cerah dan tidak mempertimbangkan kondisi cuaca yang sebenarnya.

Alur proses perhitungan energi surya tersebut yaitu:

- 1.) Pertama, hitung hari dalam tahun

$$B = \frac{(N-1) \times 360}{365} \quad (2.3)$$

Dimana N merupakan hari ke-N pada 1 tahun

- 2.) Mencari meridian standar

$$L_{st} = (\text{Perbedaan waktu zona waktu dalam jam}) \times 15 \quad (2.4)$$

Jika berada pada daerah timur, maka:

$$L_{st} = 360 - L_{st} \quad (2.5)$$

- 3.) Menghitung *solar local time* (T_s)

$$T_s = \text{Standard Time} + 4(L_{st} - \text{Longitude}) + E \quad (2.6)$$

$$E = 229.2(0.00075 + 0.001868 \cos(B) - 0.032077 \sin(B) - 0.014615 \cos(2B) - 0.04089 \sin(2B)) \quad (2.7)$$

Dimana E merupakan persamaan waktu (*equation of time*) dalam satuan menit.

- 4.) Menghitung ω yaitu sudut waktu (*hour angle*), yaitu *solar time* dalam sudut. Dihitung sebagai 15 derajat per jam yang diukur dari tengah hari. Misalnya adalah pada pukul 10:30 menjadi -22.5 derajat.

$$\omega = \{(12:00 - T_s) * 15 \text{ per hour}\} \quad (2.8)$$

jika berada pada pagi atau siang hari, dan

$$\omega = (T_s * 15 \text{ per hour}) \quad (2.9)$$

untuk lainnya.

- 5.) Menghitung sudut deklinasi matahari

$$\delta = 23.45 \sin\left(360 \times \frac{284+N}{365}\right) \quad (2.10)$$

Dimana δ merupakan sudut deklinasi dengan acuan bidang pada ekuator (tergantung dari hari pada tahun tertentu)

Dengan *solar flux*:

$$G_{s0} = 1367 \text{ watt}/m^2 \quad (2.11)$$

Jika kita menyertakan radiasi ekstra terestrial, maka total energi radiasi yang mencapai bumi adalah:

$$G_{s1} = G_{s0} \left(1 + 0.33 \times \cos\left(\frac{360N}{365}\right)\right) \quad (2.12)$$

Dan energi surya harian rata-rata adalah:

$$H_d = \frac{24 \times 3600}{\pi} G_{s1} (\cos(\phi) \cos(\delta) \sin(\omega_s) + \frac{\pi \times \omega_s}{180} \sin(\phi) \sin(\delta)) \quad (2.13)$$

Dimana ω_s merupakan derajat sudut waktu tenggelamnya matahari.

$$\cos(\omega_s) = (-\tan(\phi) \tan(\delta)) \quad (2.14)$$

Sudut waktu terbitnya matahari merupakan kebalikan/hasil negatif dari sudut waktu tenggelamnya matahari.

Jam *daylight* (jam pada saat siang hari) diberikan dalam persamaan:

$$N_{hrs} = \left(\frac{2}{15}\right) \cos^{-1}(-\tan(\phi) \tan(\delta)) \quad (2.15)$$

Radiasi yang diterima antara perbedaan waktu yang ditentukan ω_1 dan ω_2 ; $\omega_2 > \omega_1$

$$I_h = \frac{24 \times 3600}{\pi} G_{s1} (\cos(\phi) \cos(\delta) (\sin(\omega_2) - \sin(\omega_1)) + \frac{\pi \times (\omega_2 - \omega_1)}{180} \sin(\phi) \sin(\delta)) \quad (2.16)$$

Persamaan I_h diatas memberikan hasil berupa perhitungan energi surya yang akan didapatkan pada jangka waktu yang telah ditentukan pada suatu luasan per satuan m^2 . Energi surya aktual yang didapatkan oleh mobil surya diberikan dalam persamaan:

$$E_s(t_2, t_1) = \eta_s \times A_s \times I_h \quad (2.17)$$

Dimana:

- o η_s adalah efisiensi dari panel surya
- o A_s adalah luasan dari panel surya

2.3.5 Model Dinamis dari Mobil SapuAngin Surya

Model dinamis dari mobil surya yang akan dibuat dapat dijelaskan dengan persamaan di bawah ini (Moustafa Elshafei, 2013):

$$m\ddot{x} + \lambda_a x^2 + C_r mg \cos(\theta(x)) + mg \sin(\theta(x)) = F(t) \quad (2.18)$$

Dimana:

- m = massa dari mobil (termasuk massa pengemudi yang berada di dalam mobil)
- λ_a = koefisien *aerodynamic resistance*
- C_r = *roll constant*

- θ = sudut inklinasi pada jalan

Selama perjalanan, mobil akan berjalan dalam kecepatan yang relatif konstan. Maka dari itu persamaan diatas dapat disederhanakan menjadi:

$$\lambda_a x^2 + C_r mg \cos(\theta(x)) + mg \sin(\theta(x)) = F(t) \quad (2.19)$$

Lalu dikalikan dengan profil kecepatan:

$$\lambda_a x^3 + C_r mg \dot{x} \cos(\theta(x)) + \dot{x} mg \sin(\theta(x)) = P_c(t) = P_s(t) + P_b(t) \quad (2.20)$$

Dimana:

- P_s merupakan energi yang didapat dari panel surya
- P_b merupakan energi yang didapat dari media penyimpanan/baterai

Rute perjalanan terbagi dalam beberapa *land marks*. Tiap bagian akan diidentifikasi dari posisi pada GPS nya, jaraknya dari titik start dan juga elevasinya. Maka waktu yang digunakan untuk mobil berjalan dari satu titik ke titik lain dapat dirumuskan sebagai berikut:

$$t_i = \frac{Ms(i) - Ms(i-1)}{v(i)} = \frac{d_i}{v_i} \quad (2.21)$$

Dimana:

- $v(i)$ merupakan kecepatan mobil antar 2 *land marks* (antara $Ms(i)$ dan $Ms(i-1)$)
- t_i merupakan waktu yang digunakan untuk mobil berjalan dari satu titik ke titik lain (antar 2 *land marks*) dengan kecepatan $v(i)$
- d_i merupakan jarak antar *land marks* (antara $Ms(i)$ dan $Ms(i-1)$)

Dengan mengintegrasikan persamaan diatas terhadap waktu, akan didapatkan persamaan yang baru yaitu:

$$t_i(\lambda_a v_i^3 + C_r mg \cos(\theta_i) v_i + mg \sin(\theta_i) v_i) = E_c(i) = E_s(i) + E_b(i) \quad (2.22)$$

$$f_i(v_i) = E_c(i) \quad (2.23)$$

$$\tan(\theta_i) = \frac{(h_{i+1} - h_i)}{d_i} \quad (2.24)$$

Dimana:

- h_i merupakan elevasi pada *land mark* ke- i
- $E_s(i)$ merupakan energi surya yang tersedia
- $E_b(i)$ merupakan energi pada baterai yang drop selama perjalanan dari $Ms(i)$ ke $Ms(i - 1)$

2.3.6 Profil Kecepatan Mobil SapuAngin Surya

Perumusan untuk profil kecepatan dari mobil SapuAngin Surya didapatkan dari pengelolaan energi yang tersedia pada mobil (yang berasal dari energi surya) dan juga dari konsumsi baterai. Kecepatan yang optimal didapatkan dari pemanfaatan energi surya yang tersedia dan ketersediaan energi yang dialokasikan oleh baterai, dan akan digunakan untuk memaksimalkan jarak tempuh mobil surya sejak *race* dimulai hingga berakhir tiap harinya. Langkah ini dapat dilakukan secara dinamis di setiap waktu dan setiap titik sepanjang rute perjalanan untuk terus menyesuaikan berdasarkan kondisi dan status baterai pada saat itu.

Terdapat beberapa asumsi yang harus ditetapkan:

- Kondisi cuaca dianggap cerah
- Waktu berakhirnya perjalanan pada tiap harinya adalah pukul 17:00 waktu setempat

- Pengaruh kecepatan angin diabaikan
- Baterai tidak discharge selama mengemudi
- Pemberhentian mobil di control stop sekitar pukul 12:30 waktu setempat dan selama 30 menit. Selama waktu ini baterai dapat discharge
- Baterai tidak boleh dihabiskan hingga kurang dari 5% dari kapasitas penuh (5 KWH)

Pendekatan untuk memecahkan masalah optimasi ini bergantung terhadap penentuan *cost function* yang dapat dievaluasi untuk setiap kecepatan yang diberikan antar daerah (land marks) hingga berakhirnya hari perlombaan. Kemudian akan digunakan salah satu teknik optimasi berbasis pencarian konvensional untuk menemukan kecepatan yang optimal agar meminimalisir *cost function* (Moustafa Elshafei, 2013).

$V = [V_1, V_2, \dots, V_{Nms}]$ merupakan profil kecepatan dari posisi mobil pada saat itu hingga berakhirnya hari, dan i_f merupakan index dari *land mark* yang terakhir dilihat sebelum mobil berhenti pada pukul 5:00 sore pada waktu setempat.

Jarak yang ditempuh hingga $T_f = 5:00 \text{ pm}$ diberikan dalam persamaan berikut:

$$t_i = \frac{d_i}{v_i}; T_i = \sum_{j=0}^i t_j; i = 1, 2, 3, \dots, i_f; T_{i_f} < T_f \quad (2.25)$$

$$D(V) = \sum_{i=1}^{i_f} d_i + (T_f - T_{i_f})v_{i_f+1} \quad (2.26)$$

Dimana t_0 merupakan waktu *start* (atau bisa juga merupakan waktu pada saat itu juga).

Persamaan kedua adalah persamaan yang menggambarkan jarak $D(V)$, yaitu jarak yang ditempuh dari posisi atau *land mark* terakhir hingga berhentinya mobil pada pukul 5:00 sore pada waktu setempat. *Cost function* $J(V)$ akan dijelaskan sebagai berikut:

$$J(V) = W_1 \text{abs} \left(B(T_f) - B(dy) \right) + \frac{W_2}{D(V)} + J_2 \quad (2.27)$$

Dimana:

- W_1 dan W_2 adalah berat yang sesuai
- $B(dy)$ adalah status baterai yang ditargetkan saat waktu berakhirnya race
- B merupakan status baterai yang direkomendasikan berdasarkan konsumsi baterai yang telah diakumulasikan ($B=[0.75, 0.40, 0.30, 0.10]$)
- J_2 merupakan penalti tambahan jika status baterai terlalu rendah (melebihi batas yang diasumsikan, yaitu 5% dari kapasitas penuh) atau jika energi dari sinar matahari tidak sepenuhnya digunakan secara optimal
- $B(T_f)$ merupakan status baterai saat jam berakhirnya race pada hari T_f yang dihitung dengan cara mengurangi konsumsi baterai selama perjalanan menggunakan persamaan yang telah dibahas pada sub bab sebelumnya dan juga penambahan *recharging* baterai selama jeda istirahat 30 menit.

$$B(T_f) = \sum_{i=1}^{i_f+1} (E_c(i) - E_s(i)) + E_r \quad (2.28)$$

Meskipun terdapat jeda waktu tambahan untuk pengisian baterai selama waktu pemberhentian (antara jam 5:00 sore hingga matahari terbenam dan antara matahari terbit hingga jam 08:00 pagi), perubahan status baterai akan disertakan pada status baterai untuk hari berikutnya.

2.4 Graphical User Interfaces (GUI)

Graphical User Interface (GUI) adalah suatu sistem *interface* pada komputer yang memberikan atau memfasilitasi tampilan pilihan pada layar, yang biasanya berbentuk ikon (simbol gambar) atau menu (daftar karakter alfanumerik) sebagai

sarana yang dapat digunakan pengguna untuk memberikan perintah melalui *device* input.

2.4.1 GUI pada Matlab

Untuk orang awam, penggunaan Matlab sebagai program untuk penyelesaian masalah matematis memang menjadi kendala walaupun dengan listing sederhana sekalipun. Namun dalam penggunaannya, Matlab GUI juga dapat memudahkan kita dan orang lain untuk menggunakan program Matlab yang telah dibuat dari listing menjadi sebuah jendela interaktif yang dapat dipahami dengan mudah.



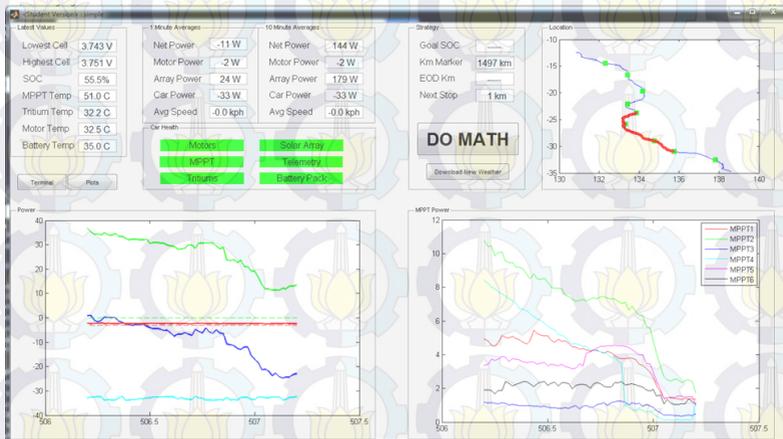
Gambar 2.4 GUIDE *Quick Start* pada Matlab

GUIDE atau GUI Builder merupakan sebuah Graphical User Interface (GUI) yang dibangun dengan objek grafis seperti tombol (pushbutton), edit, slider, text, combo, sumbu (axes), maupun menu dan lain-lain untuk kita gunakan. Sebagai contoh, ketika menggerakkan slider, maka kita dapat melihat perubahan sebuah nilai. Kemudian ketika kita menekan tombol OK, maka aplikasi akan dijalankan. Aplikasi yang menggunakan GUI umumnya lebih mudah dipelajari dan digunakan karena orang

yang menjalankannya tidak perlu mengetahui perintah yang ada dan bagaimana perintah bekerja (www.repository.usu.ac.id on Monday, 17th Feb 2014).

2.4.2 System Strategy Monitoring dengan GUI Matlab

Pembuatan *System Strategy Monitoring* dengan GUI Matlab akan mampu merepresentasikan performa dari desain kendaraan. Dan yang paling penting adalah membantu dalam pengembangan dan evaluasi desain mobil agar mampu mencapai hasil yang lebih baik. Dengan parameter dan kondisi kerja yang telah ditentukan dan menyesuaikan dengan lomba pada tahun sebelumnya, simulasi digunakan untuk menentukan profil kecepatan yang direkomendasikan selama perlombaan berlangsung. Selain itu, dari hasil simulasi dan pembuatan *interface* ini juga dapat diketahui aspek-aspek yang berpengaruh terhadap konsumsi energi sehingga dapat digunakan sebagai bahan pertimbangan jika dilakukan desain ulang kendaraan (Alexandra Boulgakov, 2012).



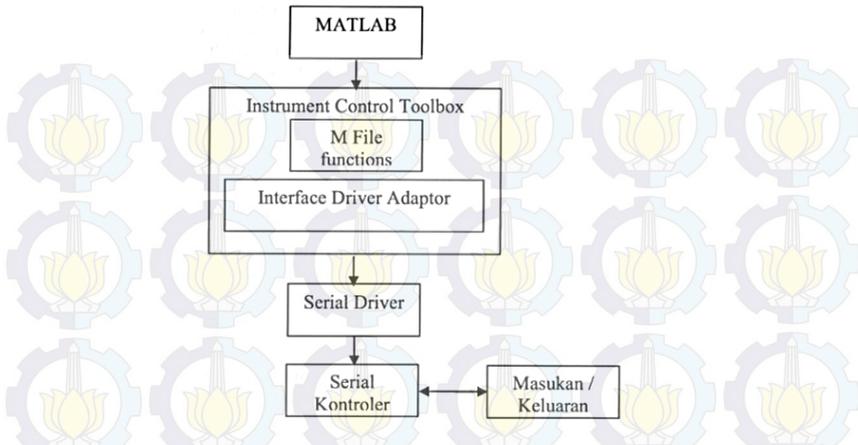
Gambar 2.5 Stanford Solar Car Team Real Time Monitoring (<http://solarcar.stanford.edu/>)

Seperti yang terlihat, gambar diatas merupakan salah satu contoh dari *Real Time Monitoring* yang dibuat dan digunakan oleh Universitas Stanford. *Real Time Monitoring* ini digunakan oleh *Stanford Solar Car* pada World Solar Challenge 2013 yang lalu.

2.4.3 Komunikasi Perangkat Keras Menggunakan Perangkat Lunak Matlab

Teknik antarmuka antara komputer dengan instrumen tidak akan terlepas dari peran perangkat lunak pada komputer untuk menerima dan mengirim data antar komputer dengan instrumen tersebut agar dapat terjadi komunikasi. Komunikasi yang sering dipakai adalah komunikasi data serial dan data paralel. Metode komunikasi paralel tidak mengenal perubahan waktu dalam pengirimannya. Data yang dikirimkan sebanyak n -bit dilakukan secara bersamaan. Pada komunikasi serial pengiriman data dilakukan berdasarkan waktu, artinya pengiriman data n -bit dilakukan secara sekuensial per bit sehingga membentuk satu data yang berupa informasi.

Pada program Matlab terdapat fasilitas pemrograman untuk melakukan teknik antarmuka ini, diantaranya komunikasi GPIB, VISA standar, Serial dan Interfacing melalui media soundcard pada PC. Oleh karena itu para pengguna perangkat lunak Matlab saat ini sudah dapat melakukan akses langsung dengan perangkat keras yang ada di luar komputer. Tujuan akses ke perangkat keras di luar komputer dimaksudkan untuk mengambil data dan mengirimkan data dari atau keluar komputer. Dengan demikian diharapkan data-data yang berasal dari luar PC dapat diambil untuk diolah lebih lanjut dengan perangkat lunak Matlab untuk keperluan simulasi, akuisisi data, pengendalian dan sebagainya (Sukarman, 2006).



Gambar 2.6 Antarmuka Serial Menggunakan Matlab
(Sukarman, 2006)

2.4.4 Konversi Data

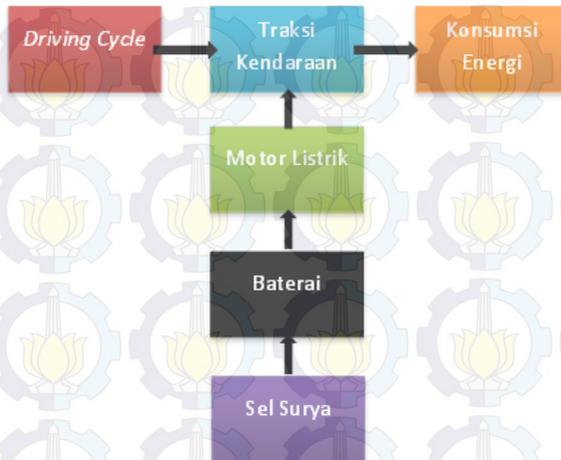
Komunikasi data yang berhubungan dengan komputer tidak terlepas dari komunikasi data digital, artinya seluruh data yang dikirimkan harus sudah dalam kondisi logika "0" dan "1". Komunikasi serial yang terjadi pada komputer menggunakan 8-bit data digital. Oleh karena itu data desimal yang akan dikirim menggunakan komunikasi serial harus dikonversi menjadi data biner 8-bit. Cara mengkonversi data pada program Matlab dapat dilakukan dengan beberapa cara seperti ditunjukkan dalam tabel berikut:

Tabel 2.2 Instruksi Konversi Data (Sukarman, 2006)

Instruksi Konversi	Keterangan
Dec2bin(D,N)	Konversikan dari decimal ke biner D=value, N=bit yang dipakai
Bin2dec('B')	Konversikan nilai biner ke decimal dengan B= value string
Dec2hex(D,N)	Konversikan bilangan heksa ke decimal dengan D=value , N= digit yang dipakai
Hex2dec(H)	Konversikan dari decimal ke heksa dengan H=value string
Dec2base(D)	Konversikan dari decimal ke basis
Base2dec(B)	Konversikan dari basis ke desimal
Dec2base(D)	Konversikan dari decimal ke basis

2.5 System Strategy Monitoring pada SapuAngin Surya II

System Strategy Monitoring yang akan dibuat merupakan sebuah *interface* yang akan menampilkan suatu jendela interaktif untuk mengatur strategi mobil surya. *Interface* ini akan menampilkan output berupa profil kecepatan yang direkomendasikan berdasarkan hasil perhitungan. Selain itu, *interface* ini juga menampilkan jarak tempuh yang ditargetkan.

**Gambar 2.7** Sistem Pada Mobil Tenaga Surya

Dari diagram diatas, dapat dilihat bahwa energi input yang berasal dari energi matahari akan diserap oleh panel-panel surya. Setelah itu, energi tersebut akan disimpan pada baterai. Kemudian energi tersebut diberikan ke motor listrik yang akan menggerakkan mobil surya. Energi yang didapatkan dari panel surya terlebih dahulu digunakan untuk menyimpan energi pada baterai. Jika baterai belum terisi penuh, maka energi yang digunakan untuk menggerakkan motor listrik langsung berasal dari panel surya. Hal ini dikarenakan baterai digunakan untuk menanggulangi cuaca yang mendung ataupun kondisi lintasan yang berat sehingga membutuhkan daya lebih. Daya yang lebih tersebut diambil dari daya yang simpanan pada baterai. Oleh karena itu, energi yang digunakan untuk menggerakkan motor sangat bergantung terhadap energi yang didapatkan panel surya dari matahari langsung.

$$\text{Energi Input} = \text{Energi Output}$$

$$\begin{aligned} \text{Energi Input} &= \text{Energi Baterai} + \text{Energi Panel Surya} \\ &= \text{Energi Output} \end{aligned}$$

Input dari *System Strategy Monitoring* ini adalah data-data yang didapatkan dari BMS (*Battery Management System*), *joulemeter* dan *motor controller*. Data yang didapat dari BMS meliputi *state of charging*, temperatur baterai, temperatur MPPT, dan lain lain (mengikuti dari data yang dapat dimonitoring sesuai dengan tipe BMS yang digunakan). Data yang didapat dari *joulemeter* meliputi energi yang didapat pada panel surya. Sedangkan data yang didapat dari *motor controller* adalah voltase dan arus yang nantinya akan diperhitungkan menjadi daya. *System Strategy Monitoring* ini akan menampilkan semua data yang didapatkan dari BMS, *joulemeter* dan *controller* tersebut secara *real time*. Jika energi yang disimpan dan energi yang digunakan mobil dapat dimonitor, maka akan dapat dilakukan perhitungan terhadap asupan energi harian mobil. Dari

perhitungan asupan energi ini akan didapatkan profil kecepatan dan estimasi jarak tempuh harian.

Hasil output dari *System Strategy Monitoring* ini berupa grafik profil kecepatan yang direkomendasikan per harinya. Lalu terdapat juga hasil output berupa grafik jarak tempuh yang direkomendasikan per harinya. Dan yang terakhir adalah hasil output berupa grafik status baterai per harinya. Status baterai yang dimaksud adalah daya yang tersisa dalam penyimpanan baterai.

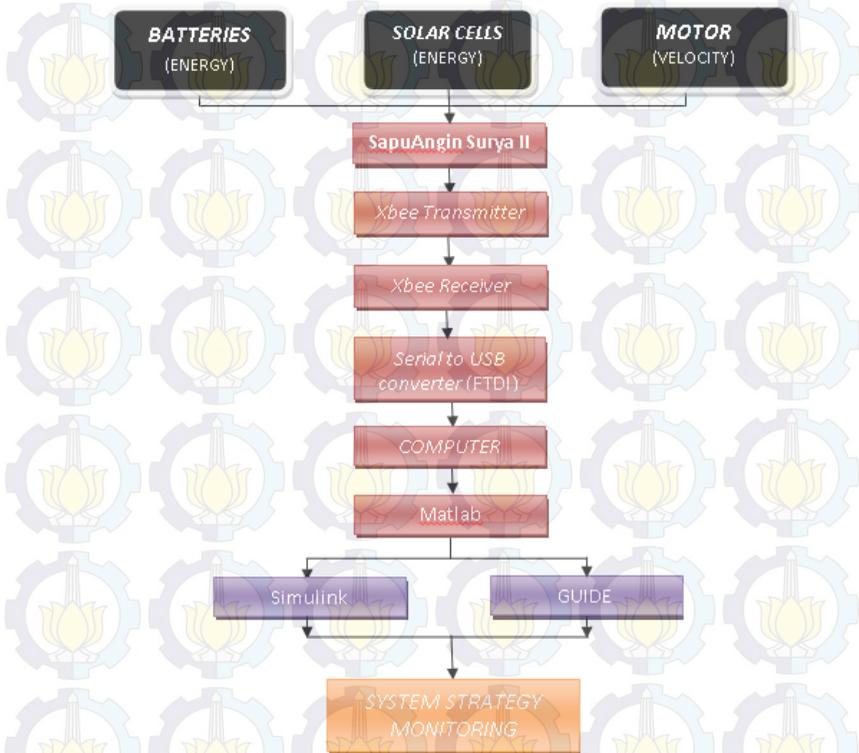
Dari grafik kecepatan, jarak tempuh dan status baterai ideal yang didapatkan, maka tim akan dapat menyusun strategi perlombaan. Karena grafik yang didapatkan itu maka tim dapat membandingkan keadaan ideal dengan keadaan aktual mobil pada saat itu. Oleh karena itu tim dapat menentukan langkah yang akan dilakukan berikutnya.

System Strategy Monitoring merupakan sebuah GUI yang akan ditampilkan menggunakan laptop oleh tim yang berada pada kendaraan pemantau mobil surya. Sehingga, data yang didapat oleh *System Strategy Monitoring* ini diambil dari kondisi mobil saat sedang berjalan. Oleh karena itu, data dari mobil (dari BMS, *joule meter* dan *motor controller*) dikirimkan oleh *transmitter* pada mobil surya dan diterima di laptop pengguna oleh *receiver* menggunakan *wireless* (teknologi nirkabel). Penggunaan *wireless* ini pun tidak mengkonsumsi banyak energi dari mobil surya, sehingga penggunaan energi untuk *wireless* ini dapat diabaikan.



BAB III METODOLOGI PENELITIAN

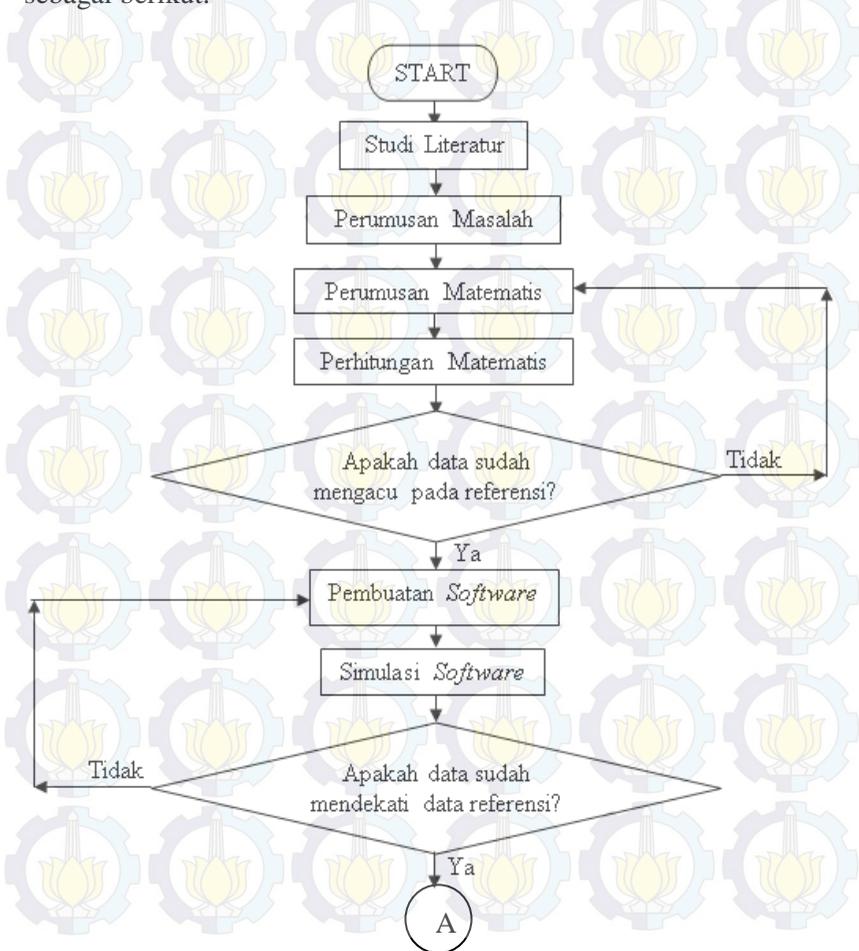
System Strategy Monitoring yang akan dibuat merupakan suatu program yang akan memudahkan *monitoring* mobil SapuAngin Surya II. Selain *monitoring* energi, *System Strategy Monitoring* ini juga akan menghasilkan rekomendasi terhadap profil kecepatan dari SapuAngin Surya II. Dibawah ini merupakan gambaran dari *System Strategy Monitoring*.



Gambar 3.1 Framework Pengerjaan *System Strategy Monitoring*

3.1 Flowchart Penelitian

Sebagai suatu proses yang terstruktur, dalam pelaksanaannya pembuatan *software* pengoptimisasi ini dilakukan dengan langkah-langkah yang sistematis. Langkah pembuatan *software* pada tugas akhir ini dijelaskan dalam beberapa tahapan sebagai berikut.





Gambar 3.2 *Flowchart* Penelitian

3.1.1 Studi Literatur

Studi literatur dilakukan dengan mempelajari permasalahan yang timbul serta langkah-langkah dan metode yang dapat dilakukan untuk menyelesaikan permasalahan. Studi pustaka dilakukan lewat buku, jurnal, dan berbagai artikel dalam internet.

3.1.2 Perumusan Masalah

Fokus permasalahan yang telah dirumuskan adalah sebagai berikut:

1. Bagaimana langkah pembuatan software *System Strategy Monitoring* untuk SapuAngin Surya agar didapatkan sistem *monitoring* secara *real-time* selama kompetisi *World Solar Challenge* mendarat.
2. Bagaimana langkah pembuatan software *System Strategy Monitoring* untuk SapuAngin Surya agar didapatkan hasil berupa tampilan untuk *monitoring* kinerja mobil secara *real-time* selama kompetisi *World Solar Challenge* berlangsung.

Dengan properti kendaraan, lintasan, target harian serta referensi yang telah ditentukan, maka semua aspek-aspek tersebut dapat dimasukkan dalam perumusan dan perhitungan matematis. Semua sistem tersebut akan dimodelkan, disimulasikan dan dibuat *monitoring* nya menggunakan Matlab.

3.1.3 Pengumpulan Data

Untuk melanjutkan perumusan, perhitungan dan pemodelan, diperlukan data-data yang mendukung penelitian. Data tersebut merupakan data dari mobil dan juga data yang nantinya didapat dari *hardware* yang telah dirangkai.

3.1.4 Perumusan dan Perhitungan Matematis

Setelah data dari mobil dapat dibaca oleh komputer, maka hal yang dilakukan selanjutnya adalah melakukan perumusan dan perhitungan matematis. Perhitungan ini juga dimasukkan ke dalam pemodelan di Simulink agar didapatkan data output yang sesuai.

Setelah perhitungan matematis, maka data akan dicek dan dianggap valid jika hasil dari perhitungan sudah mendekati hasil daripada data referensi atau acuan yang sesuai dengan hasil dan analisa pada tugas akhir yang disusun oleh Siti Choirun Nisa (Pemodelan Dan Simulasi Mobil Sapu Angin Surya Generasi Kedua Guna Memaksimalkan Performa Di *World Solar Challenge* 2015). Jika hasil masih belum mendekati/terlalu jauh, maka perhitungan dan pemodelan juga pengambilan data akan dikaji ulang.

3.1.5 Pembuatan dan Simulasi *Software System Strategy Monitoring*

Setelah perhitungan yang telah dilakukan, langkah berikutnya adalah pembuatan *software*. *Software* ini dibuat dengan GUI builder pada Matlab (GUIDE). *Software* ini akan

menghasilkan output berupa profil kecepatan yang direkomendasikan sehingga dapat membantu tim dalam menentukan strategi selama perlombaan. Lalu *software* yang sudah jadi akan disimulasikan untuk mengetahui hasil dari output, apakah sudah sesuai atau belum.

Setelah perhitungan matematis, maka data akan dicek dan dianggap valid jika hasil dari output sudah mendekati grafik pada data referensi atau acuan yang sesuai dengan hasil dan analisa pada tugas akhir yang disusun oleh Siti Choirun Nisa (Pemodelan Dan Simulasi Mobil Sapu Angin Surya Generasi Kedua Guna Memaksimalkan Performa Di *World Solar Challenge 2015*). Jika hasil grafik masih belum mendekati/terlalu jauh, maka perhitungan dan pemodelan juga pengambilan data akan dikaji ulang.

3.1.6 Analisa dan Pembahasan

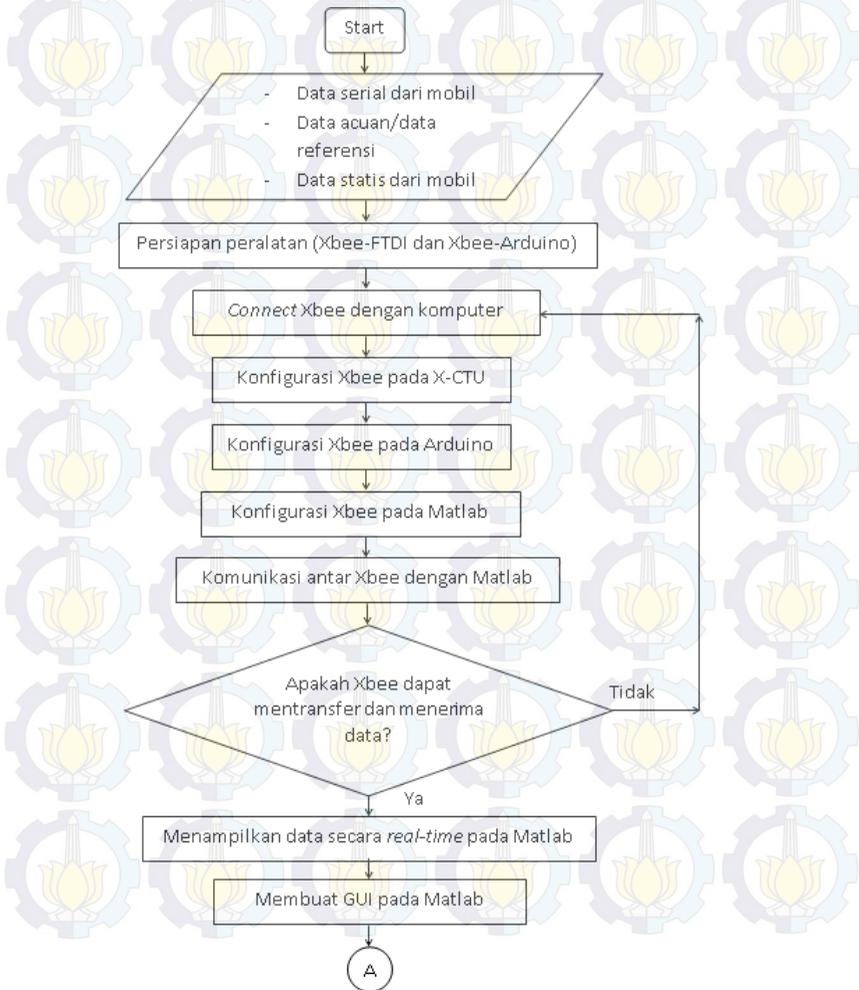
Setelah *software* disimulasikan, maka akan dilakukan analisa mengenai profil kecepatan yang direkomendasikan sesuai dengan hasil yang didapat dari output *software* tersebut. Dari situ akan ditentukan strategi dari tim untuk melakukan *race* selama perlombaan berlangsung.

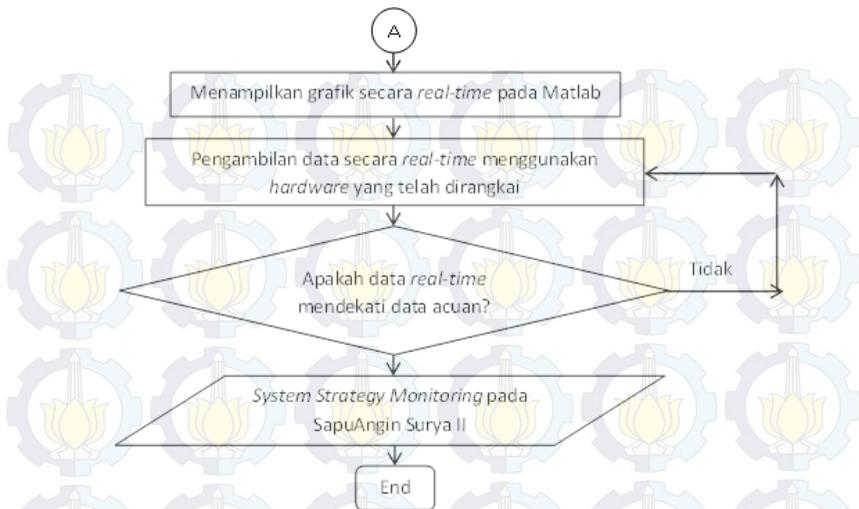
3.1.7 Kesimpulan dan Saran

Pada tahap ini, semua hasil dari penelitian disimpulkan untuk dapat menjawab semua permasalahan. Selanjutnya akan diberikan saran mengenai hasil dari simulasi dan evaluasi. Hal ini penting sebagai acuan dalam penelitian lebih lanjut.

3.2 *Flowchart Pembuatan System Strategy Monitoring*

Sebagai suatu proses yang terstruktur, dalam pelaksanaannya pembuatan *software System Strategy Monitoring* ini dilakukan dengan langkah-langkah sistematis yang dijelaskan sebagai berikut.





Gambar 3.3 Flowchart Pembuatan System Strategy Monitoring

Berikut merupakan penjelasan daripada flowchart pembuatan System Strategy Monitoring:

3.2.1 Data Input

Proses ini merupakan proses pengumpulan data-data yang akan digunakan selama proses penelitian. Yang pertama adalah data yang didapatkan dari hardware. Input dari hardware ini diasumsikan seragam sebagai data serial. Data serial ini nantinya akan didapatkan dari BMS (Battery Management System), solar cell dan motor listrik yang ditransfer menggunakan modul RF. Karena mobil SapuAngin Surya sendiri tersebut masih dalam proses pembuatan, maka data yang akan digunakan merupakan data yang didapat dari putaran potensiometer. Potensiometer tersebut akan dimodelkan sebagai variasi dari bukaan throttle, sehingga tiap putaran dari potensiometer tersebut akan

dirumuskan dalam suatu perhitungan sehingga menyesuaikan dengan kondisi nyata. Yang kedua adalah data dari model statis mobil, seperti massa mobil, koefisien drag mobil, spesifikasi *solar cell*, baterai dan hal-hal yang mempengaruhi konsumsi energi pada mobil. Yang ketiga adalah grafik acuan/referensi dari pemodelan mobil SapuAngin Surya II.

3.2.2 Proses: Persiapan Peralatan (Xbee-FTDI dan Xbee-Arduino)

Proses ini merupakan proses persiapan peralatan yang akan digunakan. Dibutuhkan 2 modul Xbee sebagai *transmitter* dan sebagai *receiver*. *Transmitter* Xbee akan tersambung dengan arduino uno dan *receiver* xbee akan tersambung dengan *adapter* dan komputer. Modul Xbee yang digunakan adalah Xbee Pro S3B.



Gambar 3.4 Modul Xbee Pro S3B

(<http://www.digi.com>)

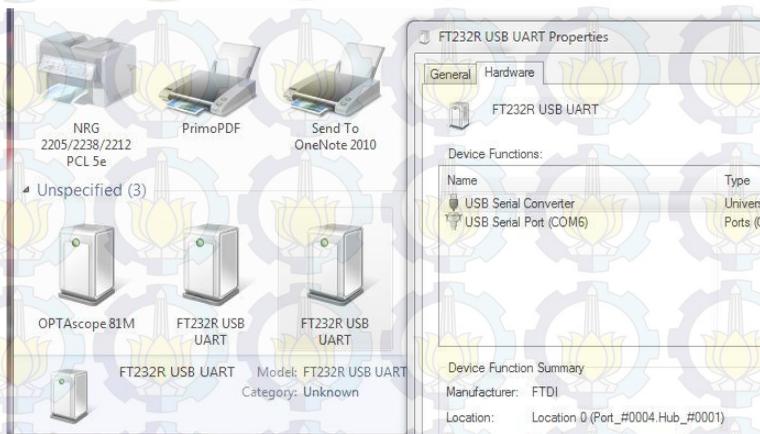
3.2.3 Proses: *Connect* Xbee Dengan Komputer

Untuk menghubungkan Xbee ke komputer, digunakan adapter berupa FTDI (FT232RL). Modul FT232RL adalah modul konversi sinyal USB ke sinyal TTL/UART (*USB-to-TTL Converter*).



Gambar 3.5 FT232RL Pada Xbee USB Adapter

Install driver untuk FTDI pada komputer untuk dapat menyambungkan Xbee dengan komputer.



Gambar 3.6 Menyambungkan FT232RL ke Komputer

3.2.4 Proses: Konfigurasi Xbee Pada X-CTU

X-CTU merupakan sebuah perangkat yang membantu untuk memprogram semua seri dan macam dari Xbee. Setelah *software* ter *install*, pilih *port* USB dimana modul Xbee terhubung. Pada *setting computer* terdapat tombol yang akan

menguji komunikasi komputer dengan modul Xbee dan akan memberikan informasi tentang versi *firmware* jika komunikasi sukses dilaksanakan.



Gambar 3.7 Jendela Konfigurasi Xbee Pada X-CTU

Tiap modul dari Xbee mempunyai konfigurasi tersendiri. Untuk Xbee yang akan digunakan pada penelitian ini, digunakan konfigurasi *default* yang ditetapkan dari software X-CTU tersebut.



Gambar 3.8 Konfigurasi Serial/USB *Port Parameters*

3.2.5 Proses: Konfigurasi Xbee Pada Arduino

Arduino digunakan untuk memberi data yang nantinya akan ditransfer oleh Xbee ke komputer untuk diolah lebih lanjut di Matlab. Hal ini dilakukan dengan memodifikasi kode untuk arduino pada *software* arduino itu sendiri. Arduino yang digunakan adalah arduino UNO. Setelah Xbee telah terkonfigurasi, maka dilakukan *range test* antara xbee yang terhubung dengan arduino dengan Xbee yang terhubung dengan untuk mengetahui apakah kedua modul Xbee (*transmitter* dan *receiver*) sudah terhubung.

3.2.6 Proses: Konfigurasi Xbee Pada Matlab

Setelah Xbee dapat terkonfigurasi dan telah dites untuk mengirim dan menerima data, maka dilakukan konfigurasi pada Matlab untuk dapat menampilkan data yang dikirim dan diterima

oleh xbee. Hal ini dilakukan dengan memodifikasi kode pada Matlab.

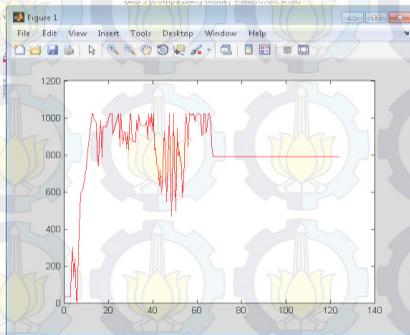
3.2.7 Proses: Komunikasi Antar Xbee Dengan Matlab

Dari hasil pengolahan kode/konfigurasi Xbee pada Matlab, maka data yang telah dikirim oleh modul *transmitter* Xbee dan diterima oleh modul *receiver* Xbee akan terbaca pada program Matlab.

Jika komunikasi tidak berhasil dilakukan, maka akan dilakukan langkah konfigurasi ulang Xbee pada X-CTU hingga konfigurasi ulang Xbee pada Matlab. Hal ini dilakukan agar dapat diketahui penyebab kegagalan komunikasi antar Xbee dengan Matlab sehingga komunikasi dapat berjalan dengan lancar.

3.2.8 Proses: Menampilkan Data Secara *Real-Time* Pada Matlab

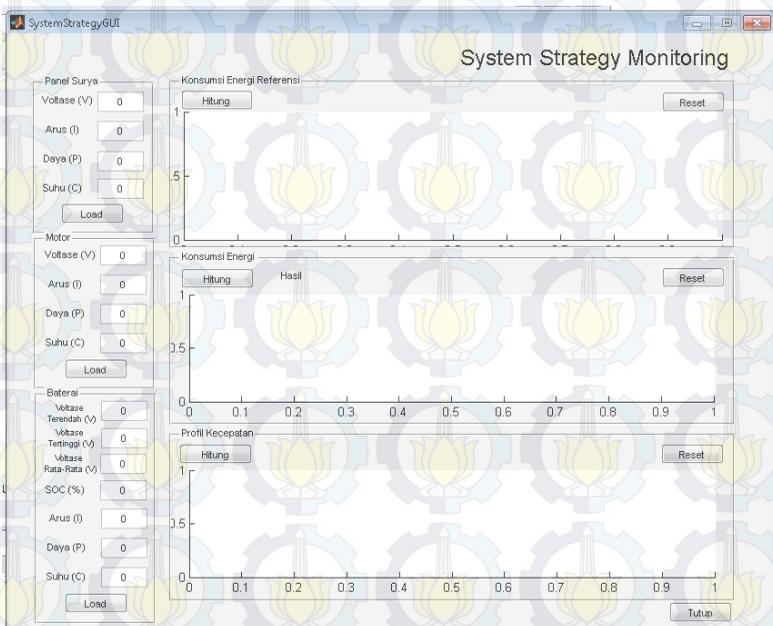
Langkah berikutnya adalah menampilkan data yang diterima oleh modul *receiver* Xbee pada Matlab. Hal ini dilakukan dengan memodifikasi kode pada Matlab, dan hasilnya adalah data *real-time* dari data yang dikirim oleh arduino melalui Xbee.



Gambar 3.9 Grafik *Real-Time* Pada Matlab

3.2.9 Proses: Membuat GUI Pada Matlab

Pembuatan GUI ini dilakukan menggunakan GUIDE pada Matlab. Dalam GUI ini nantinya akan terdapat dua tampilan. Yang pertama adalah tampilan *real-time monitoring*. *Real-time monitoring* ini akan menampilkan data-data dari mobil SapuAngin Surya II melalui Xbee. Dan yang kedua adalah tampilan *strategy monitoring* yang menampilkan grafik-grafik (grafik konsumsi energi, profil kecepatan). *Strategy monitoring* ini akan digunakan sebagai patokan strategi tim dalam kompetisi.



Gambar 3.10 Layout System Strategy Monitoring Pada SapuAngin Surya II

3.2.10 Proses: Menampilkan Grafik Secara *Real-Time* Pada GUI Matlab

Setelah pembuatan GUI, maka hal berikutnya adalah menampilkan grafik secara *real-time* pada GUI Matlab yang telah dirancang. Grafik *real-time* tersebut akan menjadi acuan dari tim selama perlombaan berlangsung.

3.2.11 Proses: Pengambilan Data Secara *Real-Time* Menggunakan *Hardware* Yang Telah Dirangkai

Langkah selanjutnya adalah pengambilan data dimana acuan/referensi akan menjadi acuan dalam pengambilan data *real-time* yang didapat dari *hardware* yang telah dirangkai. Pengambilan data secara *real-time* akan terus mengacu pada data acuan yang telah ditetapkan dari awal hingga akhir pengambilan data. Lalu data yang telah diambil akan dianalisa apakah sudah mendekati data referensi. Jika masih jauh dari data referensi, maka akan dilakukan pengambilan data lagi agar mendekati data acuan tersebut.

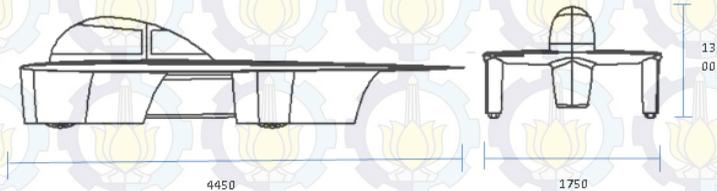
3.2.12 *Data Output*

Data output yang dihasilkan merupakan sebuah software *System Strategy Monitoring* dalam GUI. Software ini dapat menghasilkan dua jenis *output*, yaitu *monitoring* kondisi mobil secara *real time* (kondisi baterai, motor dan *solar cell*) dan juga monitoring strategi berupa konsumsi energi dan kecepatan mobil. Monitoring strategi ini akan mengkalkulasi konsumsi energi dan kecepatan mobil SapuAngin Surya yang direkomendasikan per hari nya dalam bentuk grafik. Data output tersebut didapatkan melalui perhitungan-perhitungan yang telah dirumuskan dan berdasarkan pada asumsi-asumsi yang telah ditentukan.

3.3 Data Pendukung

3.3.1 Spesifikasi SapuAngin Surya II

Berikut merupakan data spesifikasi dari mobil SapuAngin Surya II:



Gambar 3.11 SapuAngin Surya Generasi Kedua

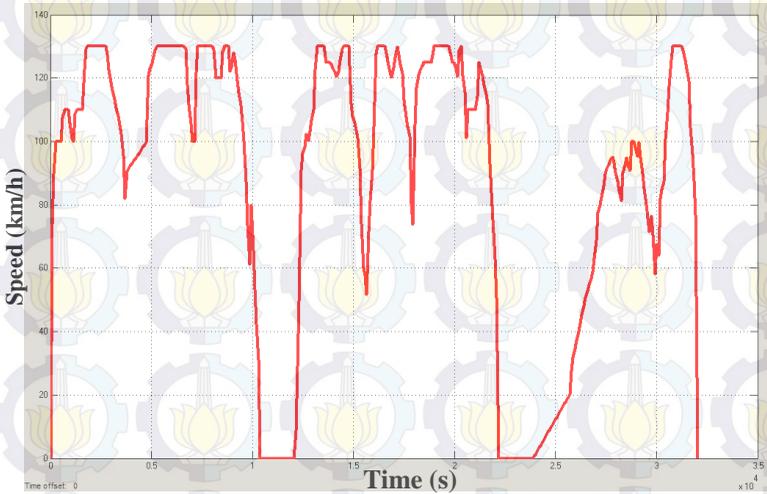
<i>Frontal Area Of Car</i>	:	1,16 m ²
<i>Percent Grade</i>	:	0
<i>Total Mass</i>	:	180 kg
<i>Drivetrain efficiency</i>	:	0,95
<i>Tire rolling resistance coefficient</i>	:	0,0027
<i>Brake and Steering Resistance</i>	:	0,0001
<i>Drag Coefficient</i>	:	0,07
<i>Air Density</i>	:	1,22
<i>Wheel Radius</i>	:	0,4064 m
<i>Final Drive Ratio</i>	:	1
<i>Fixed Gear Ratio</i>	:	1,00
<i>Overall Gear Ratio</i>	:	1

3.3.2 Drive Cycle Selama Perlombaan

Sesuai dengan kondisi nyata, maka ditentukan *drive cycle* selama perlombaan berlangsung. *Drive cycle* ini juga akan menjadi referensi untuk kecepatan ideal.

3.3.2.1 Drive Cycle Pada Hari Pertama

Pada hari pertama didapat *driving cycle* dan waktu tempuh dapat diketahui dengan menggunakan bantuan *Software* Microsoft Office Excel 2010, sehingga didapatkan grafik sebagai berikut.



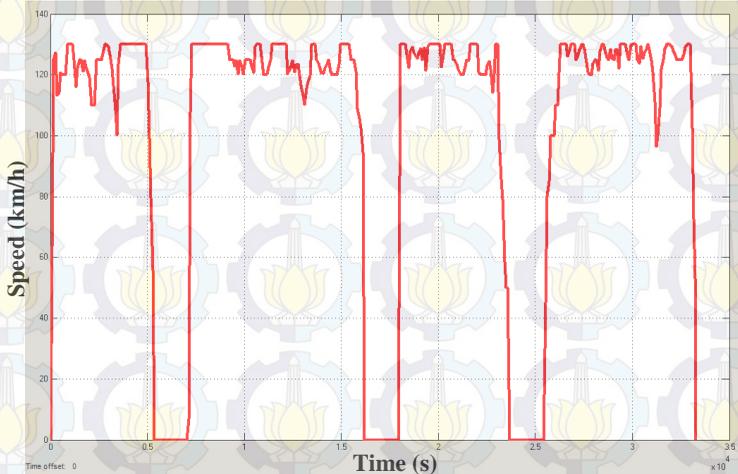
Gambar 3.12 Grafik *Drive Cycle* Pada Hari Pertama

Drive cycle pada hari pertama ditunjukkan pada **Grafik 3.12** diatas, rute yang ditempuh sejauh 796 km dari Darwin menuju ke Tennant Creek selama kurang lebih 32400 detik. Pada hari pertama terdapat dua *control stop* yang harus dilalui. *Control stop* pertama berhasil dicapai pada sekitar detik ke 12000 yaitu sekitar pukul 11:20, sedangkan waktu buka *control stop* pertama

yaitu pukul 11:10 dan waktu tutupnya pukul 13:50 sehingga *control stop* pertama berhasil didapatkan. *Control stop* kedua berhasil dicapai pada detik ke 23893 yaitu sekitar pukul 14:36, sedangkan waktu buka *control stop* kedua yaitu pukul 14:00 dan waktu tutupnya pukul 10:30 hari kedua sehingga *control stop* kedua berhasil didapatkan. Hari pertama waktu lomba diakhiri pukul 17:00 yaitu pada km ke 796 dan bermalam di daerah Renner Springs.

3.3.2.2 Drive Cycle Pada Hari Kedua

Pada hari kedua didapat *driving cycle* dan waktu tempuh dapat diketahui dengan menggunakan bantuan *Software* Microsoft Office Excel 2010, sehingga didapatkan grafik sebagai berikut.



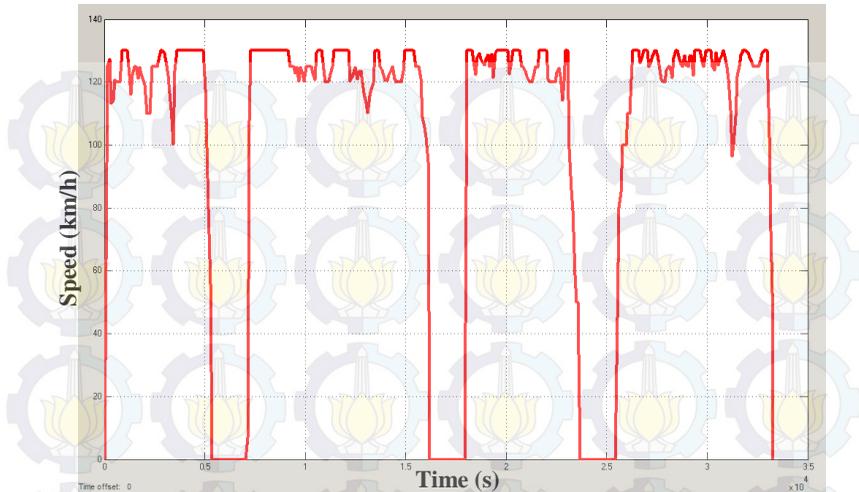
Gambar 3.13 Grafik *Drive Cycle* Pada Hari Kedua

Drive cycle pada hari kedua ditunjukkan pada **Grafik 3.13**, rute yang ditempuh sejauh 970 km dari Tennant Creek menuju ke Kulgera selama kurang lebih 32400 detik. Pada hari

kedua terdapat empat *control stop* yang harus dilalui. *Control stop* pertama berhasil dicapai pada detik ke 7123 yaitu sekitar pukul 10:00, sedangkan waktu buka *control stop* pertama yaitu pukul 09:00 dan waktu tutupnya pukul 17:10 sehingga *control stop* pertama berhasil didapatkan. *Control stop* kedua berhasil dicapai pada detik ke 17989 yaitu sekitar pukul 13:00, sedangkan waktu buka *control stop* kedua yaitu pukul 12:30 dan waktu tutupnya pukul 13:40 hari ketiga sehingga *control stop* kedua berhasil didapatkan. *Control stop* ketiga berhasil dicapai pada detik ke 25455 yaitu sekitar pukul 15:00, sedangkan waktu buka *control stop* ketiga yaitu pukul 15:00 dan waktu tutupnya pukul 08:20 hari ketiga, sehingga *control stop* ketiga berhasil didapatkan. *Control stop* keempat berhasil dicapai pada detik ke 33000 yaitu sekitar pukul 17:00, sedangkan waktu buka *control stop* keempat yaitu pukul 09:00 hari ketiga dan waktu tutupnya pukul 13:30 hari keempat sehingga *control stop* keempat kita lewati untuk memaksimalkan waktu dihari ketiga. Hari kedua waktu lomba diakhiri pukul 17:00 yaitu pada km ke 1766 dan bermalam di daerah Kulgera.

3.3.2.3 Drive Cycle Pada Hari Ketiga

Pada hari ketiga didapat *driving cycle* dan waktu tempuh dapat diketahui dengan menggunakan bantuan *Software* Microsoft Office Excel 2010, sehingga didapatkan grafik sebagai berikut.

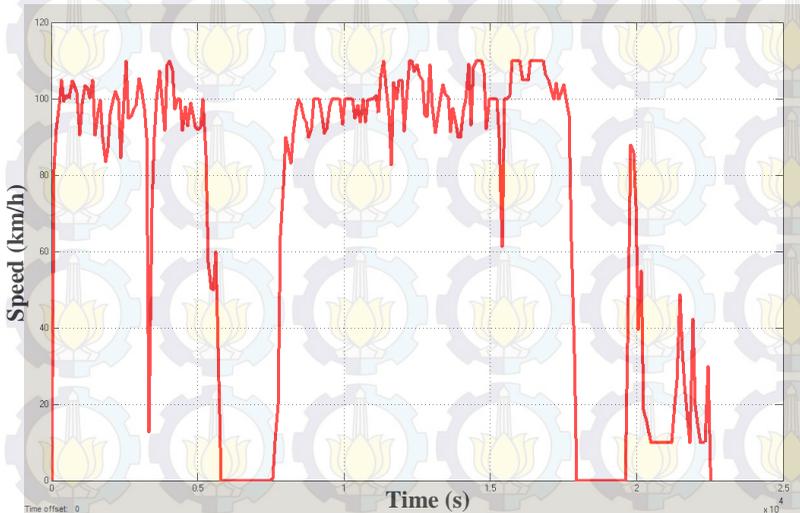


Gambar 3.14 Grafik *Drive Cycle* Pada Hari Ketiga

Drive cycle pada hari ketiga ditunjukkan pada **Grafik 3.14**, rute yang ditempuh sejauh 800 km dari Kulgera menuju ke Glendambo selama kurang lebih 32400 detik. Pada hari ketiga terdapat dua *control stop* yang harus dilalui. *Control stop* pertama berhasil dicapai pada detik ke 16273 yaitu sekitar pukul 13:30, sedangkan waktu buka *control stop* pertama yaitu pukul 13:00 dan waktu tutupnya pukul 11:20 pada hari keempat, sehingga *control stop* pertama berhasil didapatkan. *Control stop* kedua berhasil dicapai pada detik ke 27270 yaitu sekitar pukul 15:30, sedangkan waktu buka *control stop* kedua yaitu pukul 09:20 dan waktu tutupnya pukul 16:00 pada hari keempat, sehingga *control stop* kedua berhasil didapatkan. Hari ketiga waktu lomba diakhiri pukul 17:00 yaitu pada km ke 2566 dan bermalam di daerah Glendambo.

3.3.2.4 Drive Cycle Pada Hari Keempat

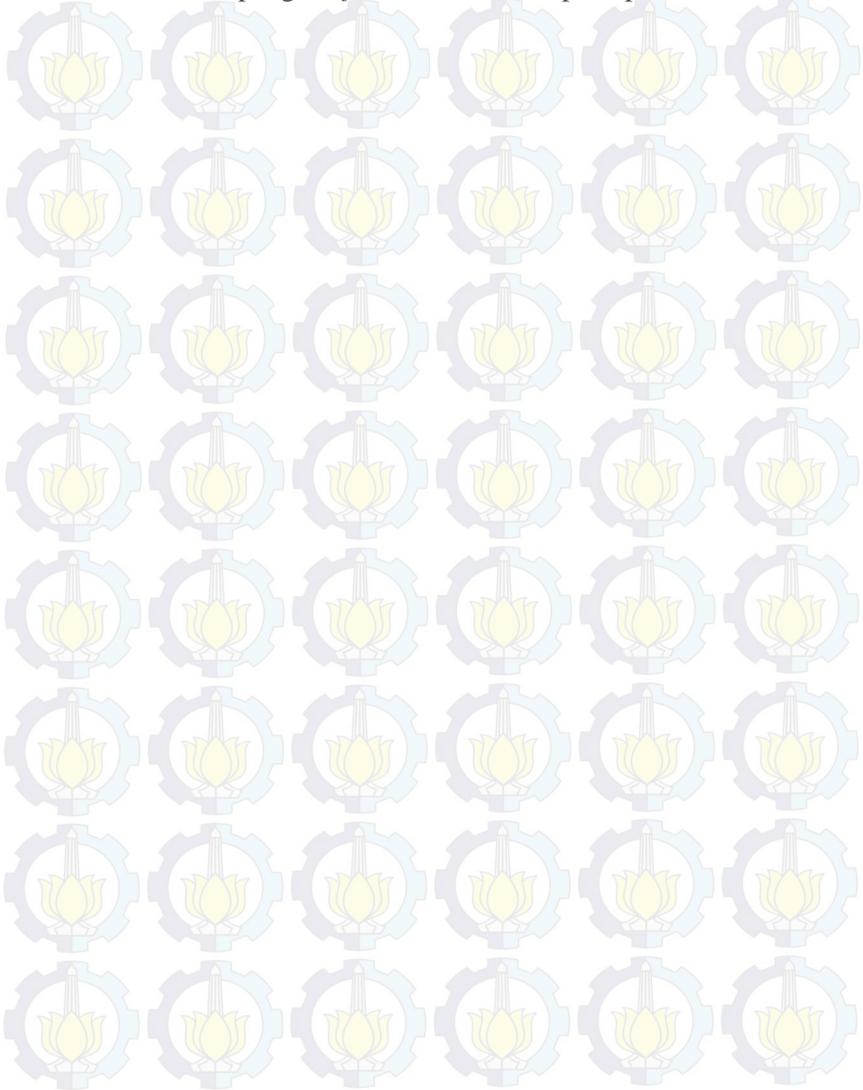
Pada hari keempat didapat *driving cycle* dan waktu tempuh dapat diketahui dengan menggunakan bantuan *Software* Microsoft Office Excel 2010, sehingga didapatkan grafik sebagai berikut.



Gambar 3.15 Grafik *Drive Cycle* Pada Hari Keempat

Drive cycle pada hari keempat ditunjukkan pada **Grafik 3.15**, rute yang ditempuh sejauh 456 km dari Glendambo menuju ke Adelaide selama kurang lebih 22540 detik. Pada hari keempat terdapat dua *control stop* yang harus dilalui. *Control stop* pertama berhasil dicapai pada detik ke 7564 yaitu sekitar pukul 10:10, sedangkan waktu buka *control stop* pertama yaitu pukul 12:00 dan waktu tutupnya pukul 12:50 hari kelima, sehingga *control stop* pertama tidak diambil untuk memaksimalkan waktu tempuh. *Control stop* kedua berhasil dicapai pada detik ke 19674 yaitu sekitar pukul 13:30, sedangkan waktu buka *control stop* kedua yaitu pukul 13:00 dan waktu tutupnya pukul 17:00 hari kelima

sehingga *control stop* kedua berhasil didapatkan. Hari keempat berhasil mencapai garis *finish* di Adelaide pada pukul 14:30.

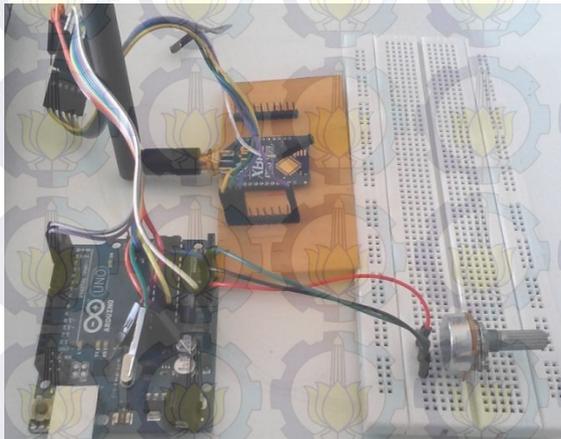




BAB IV PERHITUNGAN DAN ANALISA DATA

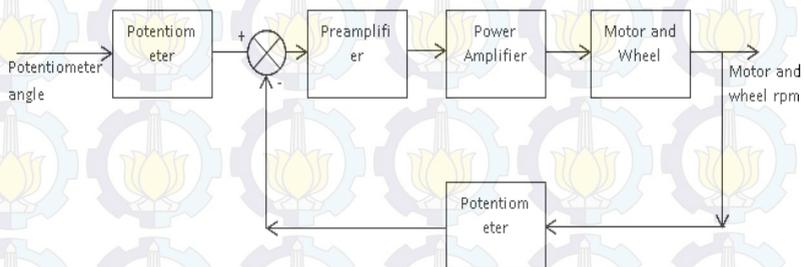
4.1 Perhitungan dan Model Dari Potensiometer

Pada dasarnya pembuatan *System Strategy Monitoring* digunakan untuk *monitoring* kondisi mobil surya secara *real-time* yang mencakup kondisi baterai, panel surya dan motor listrik. Namun karena mobil SapuAngin Surya II masih belum selesai pembuatannya, maka digunakanlah *hardware* lain untuk menggantikan mobil surya tersebut. Pada penelitian yang telah dilakukan, pengambilan data dilakukan secara *real-time* menggunakan alat bantu berupa potensiometer yang terhubung dengan arduino. Potensiometer dimodelkan sebagai *throttle* pada mobil surya. Tiap putaran dari potensiometer tersebut berhubungan dengan bukaan *throttle* pada mobil sehingga akan berpengaruh pada putaran motor listrik. Maka, data yang didapat dari potensiometer akan dikonversikan menjadi kecepatan yang nantinya akan dimonitor.



Gambar 4.1 Pemasangan Rangkaian Potensiometer

Untuk merubah data dari potensiometer menjadi putaran pada motor, maka dibuatlah suatu pemodelan dari potensiometer tersebut. Pemodelan dilakukan menggunakan *software* Matlab Simulink. Berikut merupakan blok diagram dari pemodelan potensiometer tersebut:



Gambar 4.2 Blok Diagram Sistem Pada Potensiometer

Pada pemodelan dari sistem pada potensiometer tersebut terdapat beberapa variabel yang akan digunakan untuk menentukan input, output, dan lain-lain. Variabel-variabel tersebut akan diasumsikan nilainya dan juga menyesuaikan dengan spesifikasi dari motor. Berikut merupakan tabel yang menunjukkan variabel-variabel yang akan digunakan:

Tabel 4.1 Parameter Pada Sistem

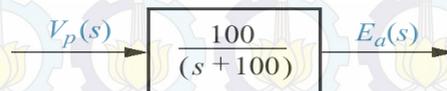
<i>Schematic Parameters</i>	
<i>Parameter</i>	<i>Definition</i>
V	<i>Voltage across potentiometer [Volts]</i>
N	<i>Turns of potentiometer</i>
K	<i>Preamplifier gain</i>
K1	<i>Power amplifier gain</i>
a	<i>Power amplifier pole</i>

Ra	Motor resistance [Ohms]
Ja	Motor inertial constant {kg-m ² }
Da	Motor dampening constant [N-m s/rad]
Kb	Back EMF constant [V-s/rad]
Kt	Motor torque constant [N-m/A]
JL	Load inertial constant [kg-m ²]
DL	Load inertial constant [N-m s/rad]
<i>Block Diagram Parameters</i>	
<i>Parameter</i>	<i>Definition</i>
Kpot	Potentiometer gain
K	Preamplifier gain
K1	Power amplifier gain
a	Power amplifier pole
Km	Motor and wheel gain
am	Motor and wheel pole

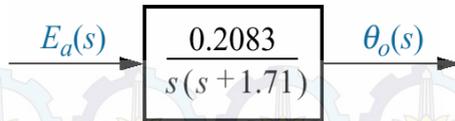
Setelah itu dibuatlah blok diagram pemodelan dari potensiometer tersebut ke dalam Matlab Simulink. *Transfer function* dari tiap subsistem dimasukkan.



Gambar 4.3 *Transfer Function Pada Preamplifier*



Gambar 4.4 *Transfer Function Pada Power Amplifier*



Gambar 4.5 Transfer Function Pada Motor and Wheel

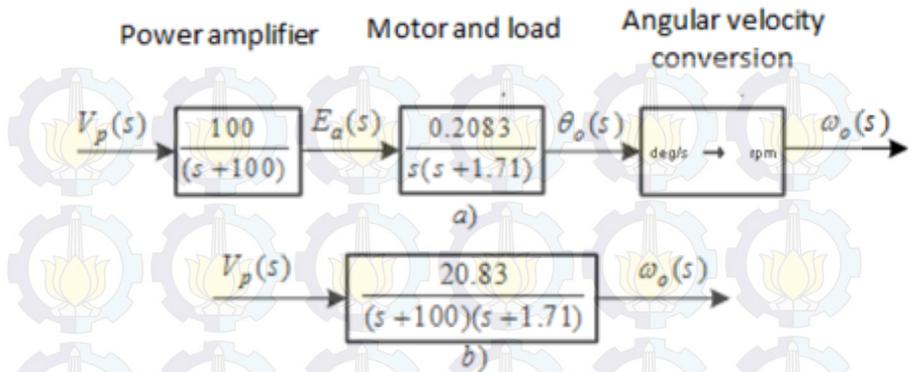
Lalu ditentukan nilai dari parameter-parameter yang terdapat pada sistem tersebut.

Tabel 4.2 Nilai Parameter-Parameter Pada Sistem

Parameter Configuration 1

V	10
n	10
K	—
K_1	100
a	100
R_a	8
J_a	0.02
D_a	0.01
K_b	0.5
K_t	0.5
J_L	1
D_L	1

Setelah itu semua *transfer function* dari tiap subsistem dimasukkan menjadi satu pada blok diagram.



Gambar 4.6 Blok Diagram Pada Sistem

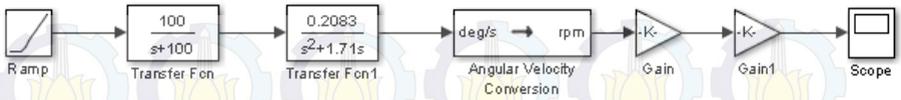
Namun karena output dari blok diagram diatas masih berupa kecepatan angular (dalam satuan rpm), maka dibutuhkan perhitungan untuk mengkonversikan dari satuan rpm menjadi km/jam.

Sesuai dengan perhitungan dari rpm ke km/jam di bawah ini:

$$v = 2\pi r \times RPM \times (60/1000) \text{ km/hr}$$

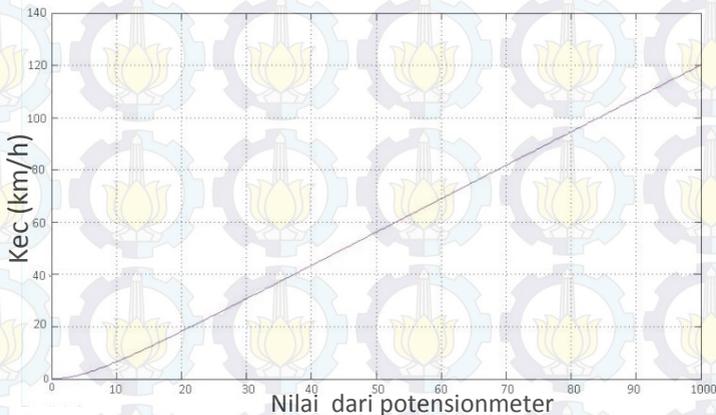
maka output yang dihasilkan dari blok diagram tersebut masih perlu dimasukkan ke dalam perhitungan matematis lagi, dimana r adalah jari-jari roda yang digunakan. Sesuai dengan spesifikasi pada mobil SapuAngin Surya II, maka digunakan nilai r nya adalah 0,4064 m.

Berikut merupakan blok diagram dari sistem dengan output dalam besaran km/h.



Gambar 4.7 Blok Diagram Pada Simulink

Dari blok diagram *open loop* tersebut didapatkan *ramp response* dari scope sebagai berikut:



Gambar 4.8 Ramp Response Pada Sistem

Grafik *ramp response* diatas menggambarkan hubungan antara perubahan voltase dari potensiometer terhadap kecepatan motor. Potensiometer yang akan digunakan akan terhubung pada salah satu pin analog pada arduino. Pin analog pada arduino ini akan membaca input berupa voltase (dari 0 sampai 5 Volt) dan mengkonversikannya ke angka digital yaitu antara 0 (0 Volt) dan 1023 (5 Volt), yaitu pembacaan dengan resolusi 10 bit. Potensiometer yang terhubung dengan tegangan 5 Volt akan memberikan keluaran tegangan antara 0 hingga 5 Volt pada pin bagian tengahnya, tergantung pada sudut diputarnya

potensiometer tersebut. Oleh karena itu keluaran yang terbaca pada Matlab adalah angka digital antara 0 hingga 1023.

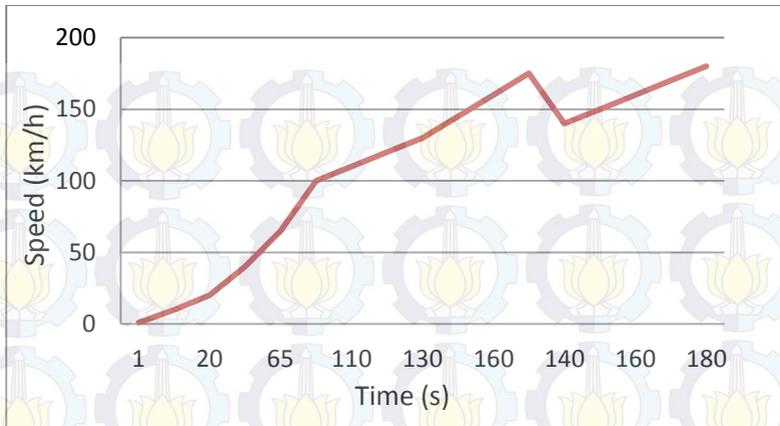
Dari hasil *ramp response* tersebut dapat dilihat fungsi dari potensiometer dan putaran pada motor memiliki perbandingan yang lurus dan linier. Lalu untuk mengkonversikan data pada potensiometer yang diambil secara *real-time* menjadi kecepatan maka dimasukkan blok diagram diatas ke dalam blok diagram *real-time*.

Dalam kenyataannya, mobil membutuhkan waktu untuk untuk mencapai kecepatan tertentu. Hal tersebut disebabkan karena adanya gaya dinamis dari mobil surya itu sendiri. Oleh karena itu, ditentukan waktu untuk mencapai kecepatan tertentu (*time to accelerate to speed*) dan *range* dari kecepatan mobil surya. Untuk memasukkan data tersebut maka dibuatlah tabel perbandingan antara kecepatan pada mobil dengan waktu yang digunakan untuk mencapai kecepatan tersebut (*time to accelerate*). Dengan bantuan *Microsoft Office Excel 2010*, didapatkan tabel perhitungan sebagai berikut.

Tabel 4.3 Tabel Perhitungan Kecepatan dan *Time to Accelerate*

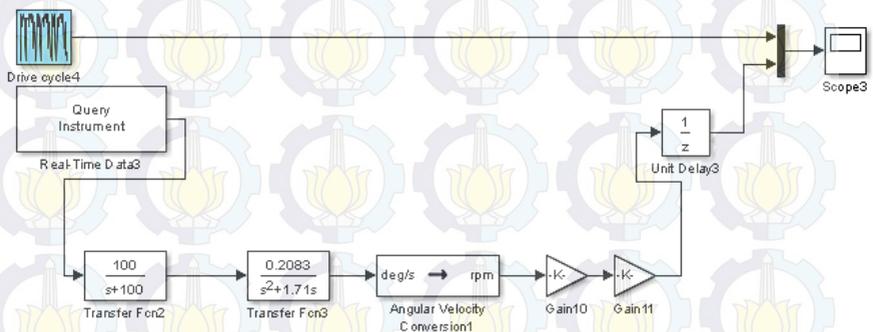
Speed (km/h)	Time to Accelerate to Speed (s)
0	1
10	10
20	20
30	40
40	65
50	100
60	110
70	120
80	130
90	145
100	160
110	175
120	140
130	150
140	160
150	170
160	180

Dari tabel tersebut, maka data yang telah didapatkan akan diplot pada grafik.



Gambar 4.9 Grafik Perbandingan *Time to Accelerate* dan Kecepatan

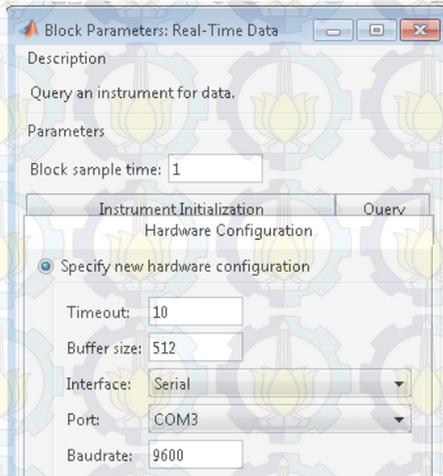
Karena adanya *time to accelerate*, maka terdapat *delay time* pada mobil surya. *Delay time* tersebut akan mempengaruhi profil kecepatan dari mobil surya. Dalam pemodelannya dalam Simulink, *delay time* tersebut ditunjukkan dengan blok *unit delay*.



Gambar 4.10 Pemodelan Putaran Potensiometer Menggunakan Software MATLAB/Simulink

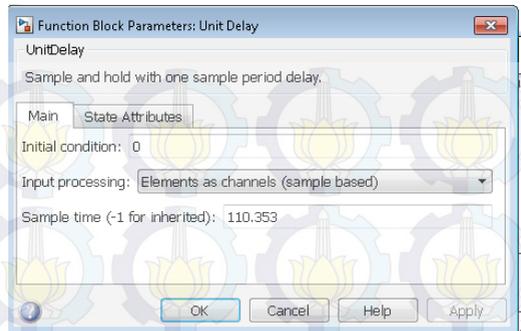
Data dari *driving cycle* per hari nya akan dimodelkan seperti pada Gambar 4.2 diatas tersebut.

Pada blok *query instruments* yang digunakan untuk mengambil data serial dari *hardware* (data dari putaran pada potensiometer), ditetapkan parameternya sebagaimana yang tertera pada gambar dibawah.



Gambar 4.11 Block Parameter Untuk Blok *Query Instruments*

Pada blok *unit delay* yang digunakan untuk memberi *delay time* pada data serial dari *hardware* (data dari putaran pada potensiometer), ditetapkan parameternya sebagaimana yang tertera pada gambar dibawah. *Sample time* pada *unit delay* didapatkan dari rata-rata perubahan waktu terhadap kecepatan (rata-rata *time to accelerate*).



Gambar 4.12 *Block Parameter* untuk Blok *Unit Delay*

Setelah dilakukan pemodelan untuk potensiometer terhadap profil kecepatan, maka akan diambil data secara *real-time* menggunakan potensiometer. Pengambilan data dilakukan untuk mendapatkan profil kecepatan yang direkomendasikan dan menyesuaikan dengan referensi *drive cycle* yang telah ditentukan.

Setelah didapatkan pengambilan data untuk profil kecepatan yang telah dilakukan sebelumnya, maka akan didapatkan *drive cycle* baru. *Drive cycle* baru tersebut merupakan *drive cycle* yang berasal dari data *real-time*. Untuk menghitung konsumsi energi dari *drive cycle* terbaru tersebut, maka *drive cycle* baru akan dimasukkan ke dalam pemodelan yang terdapat pada tugas akhir yang disusun oleh Siti Choirun Nisa (Pemodelan Dan Simulasi Mobil Sapu Angin Surya Generasi Kedua Guna Memaksimalkan Performa Di *World Solar Challenge 2015*).

4.2 Analisa System Strategy Monitoring

Pada *System Strategy Monitoring* yang telah dibuat, digunakan *hardware* yang berfungsi untuk membaca data serial secara *real-time* dari mobil surya ke Matlab menggunakan koneksi *wireless*.

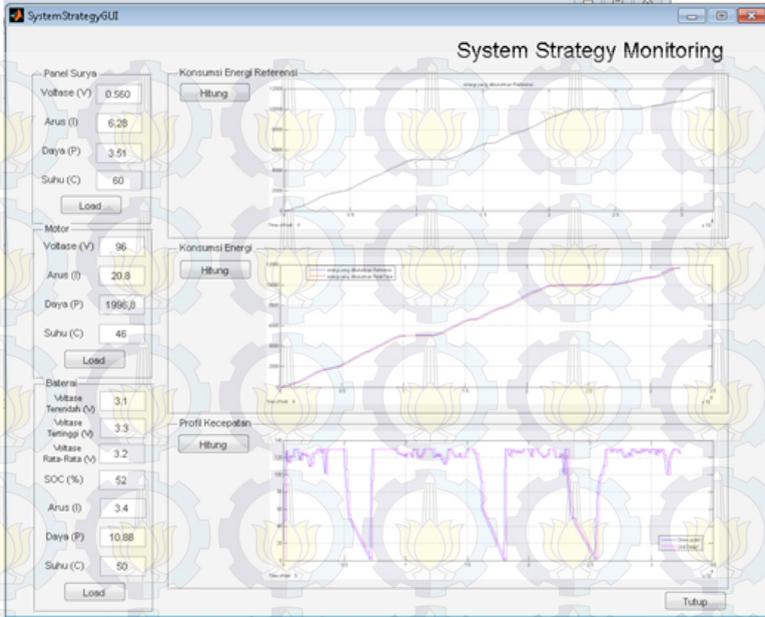


Gambar 4.14 Alur Komunikasi Serial Pada *System Strategy Monitoring*

Hardware yang digunakan adalah sebagai berikut:

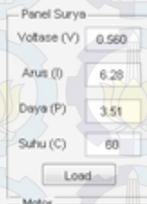
- Arduino UNO
- Transmitter Xbee Pro S3B
- Receiver Xbee Pro S3B
- FTDI sebagai *Serial to USB* converter (*Xbee Adapter*)

Berikut merupakan tampilan dari *System Strategy Monitoring* yang telah dibuat:



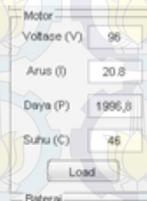
Gambar 4.15 *System Strategy Monitoring* Pada SapuAngin Surya II

Pada *System Strategy Monitoring* yang telah dibuat, terdapat beberapa bagian yang memiliki fungsi yang berbeda.



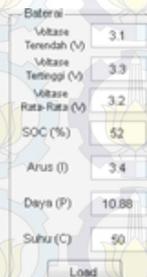
Gambar 4.16 *Monitoring Panel Surya* Pada *System Strategy Monitoring*

Pada *monitoring* panel surya diatas, akan diambil data dari sensor yang dipasang pada panel surya, yaitu sensor arus, voltase dan temperatur. Dari situ akan dapat dimonitor kondisi panel surya pada mobil selama perlobaan berlangsung. Data tersebut akan disalurkan melalui pin-pin pada *microcontroller* dan akan ditransfer melalui Xbee menuju ke *software System Strategy Monitoring* pada komputer.



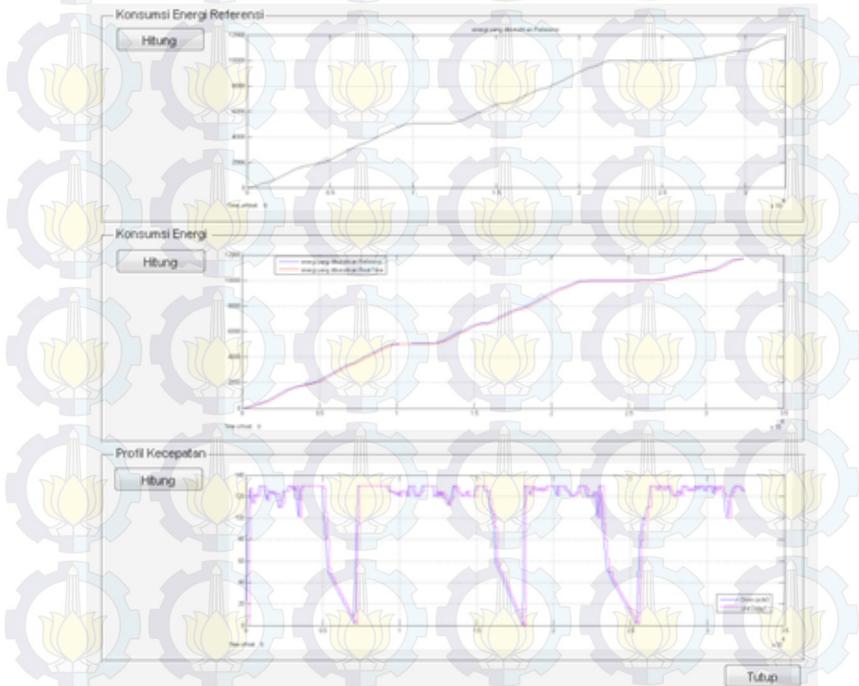
Gambar 4.17 *Monitoring Motor Pada System Strategy Monitoring*

Pada *monitoring* motor diatas, akan diambil data dari sensor yang dipasang pada motor, yaitu sensor arus, voltase dan temperatur. Dari situ akan dapat dimonitor kondisi motor pada mobil selama perlobaan berlangsung. Data tersebut akan disalurkan melalui pin-pin pada *microcontroller* dan akan ditransfer melalui Xbee menuju ke *software System Strategy Monitoring* pada komputer.



Gambar 4.18 *Monitoring Baterai Pada System Strategy Monitoring*

Pada *monitoring* baterai diatas, akan diambil data dari BMS yang dipasang pada mobil surya. BMS tersebut akan memonitor kondisi baterai secara keseluruhan dan juga per *cell* nya. Data yang diambil adalah voltase tertinggi, voltase terendah, voltase rata-rata, *state of charging*, arus, daya dan suhu. Dari situ akan dapat dimonitor kondisi baterai pada mobil selama perlombaan berlangsung. Data tersebut akan disalurkan melalui pin-pin pada *microcontroller* dan akan ditransfer melalui Xbee menuju ke *software System Strategy Monitoring* pada komputer.



Gambar 4.19 Grafik Konsumsi Energi dan Profil Kecepatan pada *System Strategy Monitoring*

Grafik pertama menunjukkan konsumsi energi referensi/ideal yang telah didapatkan dari pemodelan pada tugas

akhir yang disusun oleh Siti Choirun Nisa (Pemodelan Dan Simulasi Mobil Sapu Angin Surya Generasi Kedua Guna Memaksimalkan Performa Di *World Solar Challenge* 2015). Grafik kedua menunjukkan konsumsi energi yang didapat secara *real-time* sehingga dapat dimonitor perbandingan konsumsi energi aktual dan ideal selama perlombaan berlangsung. Sedangkan grafik terakhir merupakan grafik gabungan perbandingan antara profil kecepatan acuan/referensi dengan profil kecepatan yang didapat secara *real-time* langsung dari mobil surya. Grafik profil kecepatan tersebut nantinya akan menjadi acuan untuk *driver* mobil surya untuk dapat mengambil keputusan maupun strategi selama *race* berlangsung.

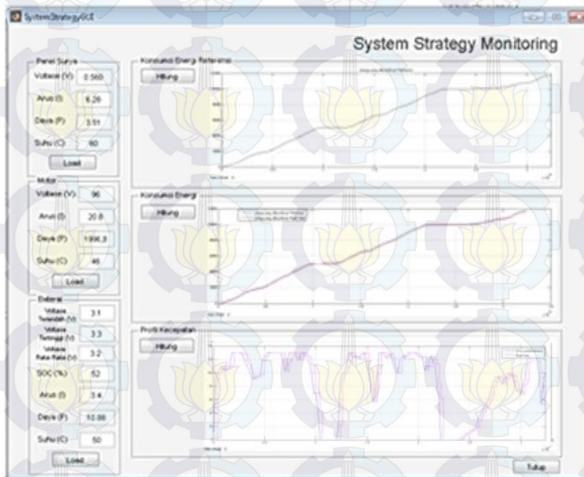
Namun karena belum terselesaikannya mobil Sapu Angin Surya II itu sendiri, maka dalam penelitian ini, data *real-time* yang akan diambil diwakili oleh potensiometer. Potensiometer tersebut akan dimodelkan sebagai *throttle* dimana tiap putaran dari potensiometer itu sendiri akan mewakili bukaan *throttle* pada mobil surya. Bukaan *throttle* sendiri akan mempengaruhi putaran motor listrik sehingga juga akan mempengaruhi kecepatan pada mobil.

4.3 Analisa Data

Untuk mendapatkan profil kecepatan yang direkomendasikan dan mengacu pada referensi, maka diambil data tiap hari nya dan dianalisa untuk didapatkan strategi *race* yang sesuai dengan kondisi yang sebenarnya.

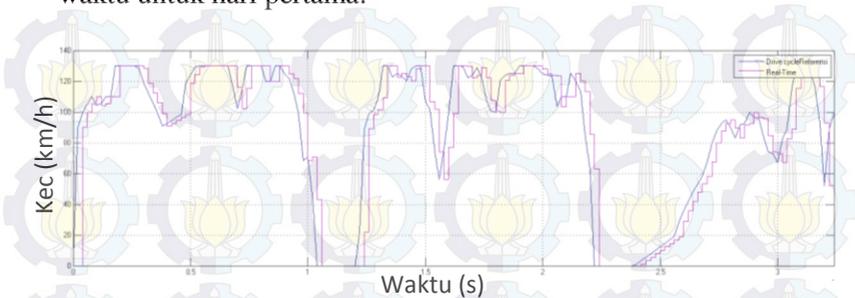
4.3.1 Grafik Kecepatan Fungsi Waktu Hari Pertama

Berikut merupakan grafik kecepatan fungsi waktu pada hari pertama perlombaan. Sumbu x menunjukkan kecepatan (dalam satuan km/jam) dan sumbu y menunjukkan waktu (dalam satuan detik). Grafik yang berwarna biru menunjukkan profil kecepatan ideal/referensi yang didapatkan dari *driving cycle* sedangkan grafik yang berwarna merah muda menunjukkan profil kecepatan secara *real time* yang didapat dari pengambilan data.



Gambar 4.20 Tampilan Grafik Kecepatan Hari Pertama pada *System Strategy Monitoring*

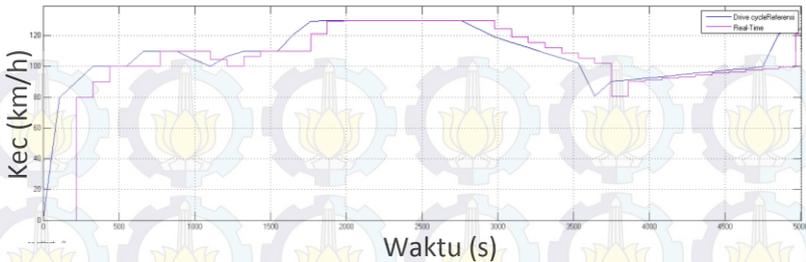
Berikut merupakan tampilan dari grafik kecepatan fungsi waktu untuk hari pertama:



Gambar 4.21 Grafik Kecepatan Fungsi Waktu Hari Pertama

Grafik berwarna biru menunjukkan kecepatan dari Mobil SapuAngin Surya Generasi Kedua selama perlombaan pada hari pertama. Sedangkan grafik berwarna merah muda menunjukkan kecepatan dari Mobil SapuAngin Surya Generasi Kedua selama perlombaan pada hari pertama yang didapat dari pengambilan data dengan mengatur bukaan *throttle* yang diwakili oleh putaran potensiometer. Pada hari pertama waktu yang dibutuhkan selama 32400 detik (9 jam) dan kecepatan didapatkan berdasarkan tabel rute dari World Solar Challenge. Sekaligus sudah disesuaikan dengan peraturan-peraturan pada World Solar Challenge bahwa setiap berhenti di control stop harus berhenti terlebih dahulu selama 30 menit. Kecepatan maksimum nya adalah 130 km/jam dan kecepatan rata-ratanya 100 km/jam.

Strategi *race* yang ditentukan untuk hari pertama ditinjau pada waktu antara 0-5000 detik



Gambar 4.22 Grafik Kecepatan Fungsi Waktu Hari Pertama Pada Detik 0-5000

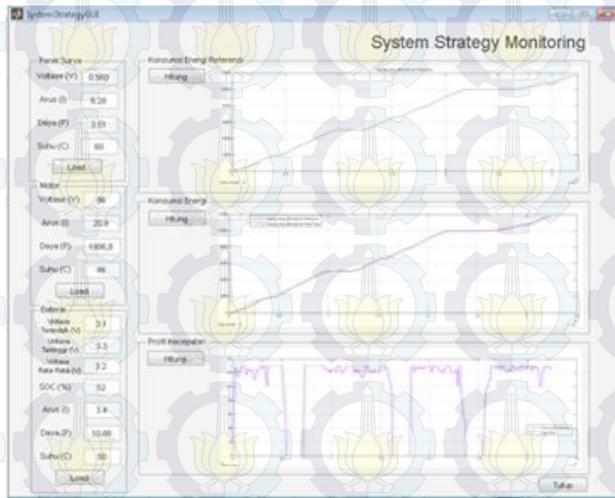
Dapat dilihat bahwa terdapat perbedaan antara grafik referensi dengan grafik aktual. Karena adanya *delay time*, maka grafik aktual memiliki jarak sejak kondisi *starternya*. Hal tersebut dikarenakan mobil membutuhkan *time to accelerate* seperti yang sudah dijelaskan pada subbab sebelumnya.

Pada grafik aktual dapat dilihat bahwa pada sekitar detik ke 200 mobil langsung mengalami percepatan hingga 80 km/jam, hal itu dapat disebabkan karena pemutaran potensiometer yang terlalu besar dalam waktu yang singkat. Lalu pada waktu sisanya, grafik aktual akan terus naik mengikuti profil pada grafik ideal hingga detik ke 1800. Lalu pada detik ke 1800 hingga 3000, kecepatan dijaga konstan di 130 km/jam, kemudian turun perlahan hingga detik ke 3800 dan kecepatan akan naik kembali pada detik ke 3800 hingga detik ke 5000.

4.3.2 Grafik Kecepatan Fungsi Waktu Hari Kedua

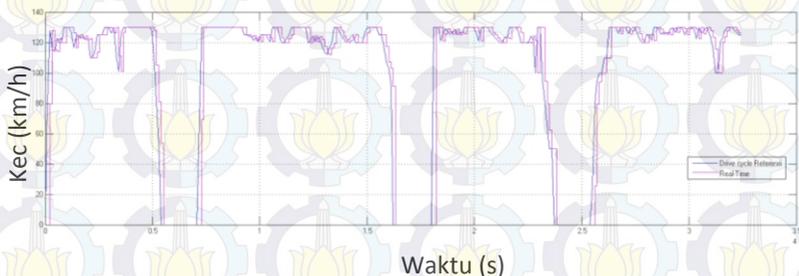
Berikut merupakan grafik kecepatan fungsi waktu pada hari kedua perlombaan. Sumbu x menunjukkan kecepatan (dalam satuan km/jam) dan sumbu y menunjukkan waktu (dalam satuan detik). Grafik yang berwarna biru menunjukkan profil kecepatan ideal/referensi yang didapatkan dari *driving cycle* sedangkan

grafik yang berwarna merah muda menunjukkan profil kecepatan secara *real time* yang didapat dari pengambilan data.



Gambar 4.23 Tampilan Grafik Kecepatan Hari Kedua pada *System Strategy Monitoring*

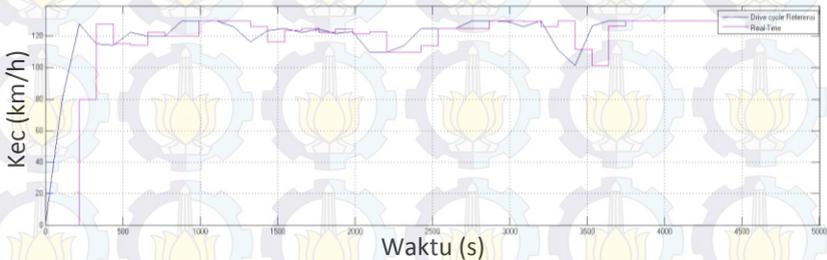
Berikut merupakan tampilan dari grafik kecepatan fungsi waktu untuk hari kedua:



Gambar 4.24 Grafik Kecepatan Fungsi Waktu Hari Kedua

Grafik berwarna biru menunjukkan kecepatan dari Mobil SapuAngin Surya Generasi Kedua selama perlombaan pada hari kedua. Sedangkan grafik berwarna merah muda menunjukkan kecepatan dari Mobil SapuAngin Surya Generasi Kedua selama perlombaan pada hari kedua yang didapat dari pengambilan data dengan mengatur bukaan *throttle* yang diwakilkan oleh putaran potensiometer. Pada hari kedua waktu yang dibutuhkan selama 32400 detik (9 jam) dan kecepatan didapatkan berdasarkan tabel rute dari World Solar Challenge. Sekaligus sudah disesuaikan dengan peraturan-peraturan pada World Solar Challenge bahwa setiap berhenti di control stop harus berhenti terlebih dahulu selama 30 menit. Kecepatan maksimum nya adalah 130 km/jam dan kecepatan rata-ratanya 116 km/jam.

Strategi *race* yang ditentukan untuk hari kedua ditinjau pada waktu antara 0-5000 detik



Gambar 4.25 Grafik Kecepatan Fungsi Waktu Hari Kedua Pada Detik 0-5000

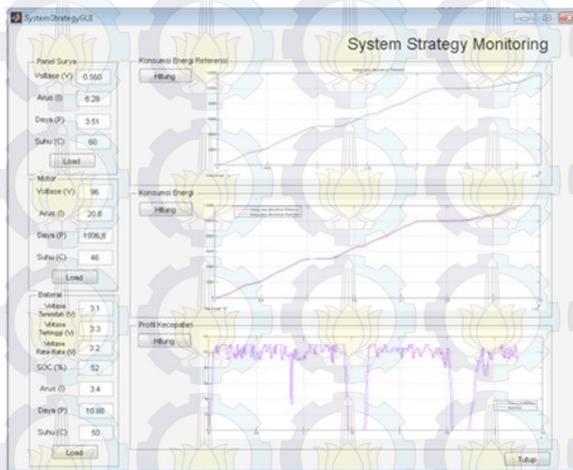
Dapat dilihat bahwa terdapat perbedaan antara grafik referensi dengan grafik aktual. Karena adanya *delay time*, maka grafik aktual memiliki jarak sejak kondisi *starternya*. Hal tersebut dikarenakan mobil membutuhkan *time to accelerate* seperti yang sudah dijelaskan pada subbab sebelumnya.

Pada grafik aktual dapat dilihat bahwa pada sekitar detik ke 200 mobil langsung mengalami percepatan hingga 80 km/jam,

hal itu dapat disebabkan karena pemutaran potensiometer yang terlalu besar dalam waktu yang singkat. Lalu pada waktu sisa nya, grafik aktual akan terus mengikuti profil pada grafik ideal hingga detik ke 3650. Antara detik ke 300 hingga detik ke 3650, perubahan kecepatannya sangat bervariasi, sehingga dapat ditarik strategi *race* nya untuk menjaga kecepatan sebesar 120 km/jam pada detik ke 300 hingga 3650. Lalu pada detik ke 3650 hingga 5000, kecepatan dijaga konstan di 130 km/jam.

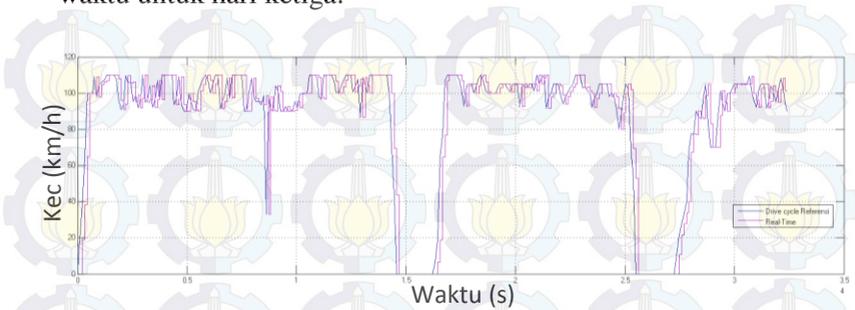
4.3.3 Grafik Kecepatan Fungsi Waktu Hari Ketiga

Berikut merupakan grafik kecepatan fungsi waktu pada hari ketiga perlombaan. Sumbu x menunjukkan kecepatan (dalam satuan km/jam) dan sumbu y menunjukkan waktu (dalam satuan detik). Grafik yang berwarna biru menunjukkan profil kecepatan ideal/referensi yang didapatkan dari *driving cycle* sedangkan grafik yang berwarna merah muda menunjukkan profil kecepatan secara *real time* yang didapat dari pengambilan data.



Gambar 4.26 Tampilan Grafik Kecepatan Hari Ketiga pada *System Strategy Monitoring*

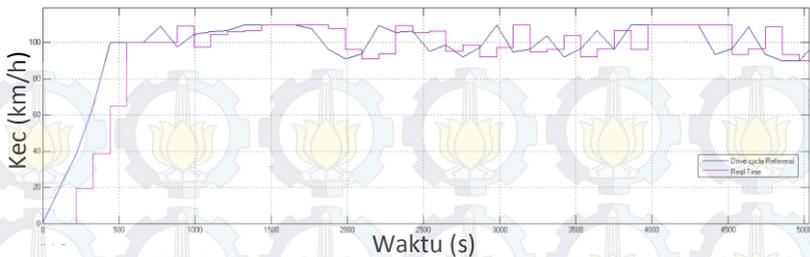
Berikut merupakan tampilan dari grafik kecepatan fungsi waktu untuk hari ketiga:



Gambar 4.27 Grafik Kecepatan Fungsi Waktu Hari Ketiga

Grafik berwarna biru menunjukkan kecepatan dari Mobil SapuAngin Surya Generasi Kedua selama perlombaan pada hari ketiga. Sedangkan grafik berwarna merah muda menunjukkan kecepatan dari Mobil SapuAngin Surya Generasi Kedua selama perlombaan pada hari ketiga yang didapat dari pengambilan data dengan mengatur bukaan *throttle* yang diwakilkan oleh putaran potensiometer. Pada hari ketiga waktu yang dibutuhkan selama 32400 detik (9 jam) dan kecepatan didapatkan berdasarkan tabel rute dari World Solar Challenge. Sekaligus sudah disesuaikan dengan peraturan-peraturan pada World Solar Challenge bahwa setiap berhenti di control stop harus berhenti terlebih dahulu selama 30 menit. Kecepatan maksimum nya adalah 110 km/jam dan kecepatan rata-ratanya 97 km/jam.

Strategi *race* yang ditentukan untuk hari ketiga ditinjau pada waktu antara 0-5000 detik



Gambar 4.28 Grafik Kecepatan Fungsi Waktu Hari Ketiga Pada Detik 0-5000

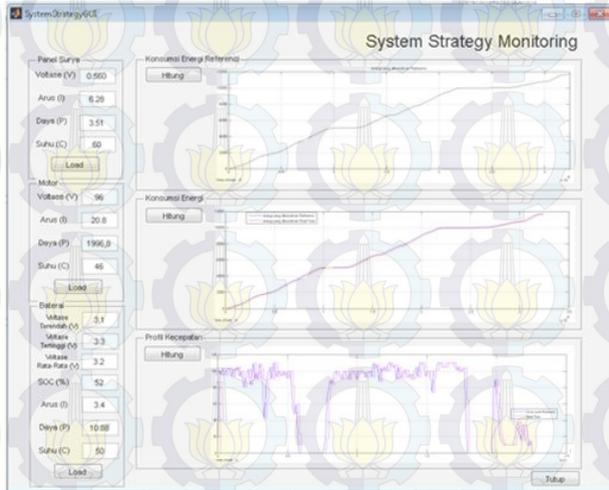
Dapat dilihat bahwa terdapat perbedaan antara grafik referensi dengan grafik aktual. Karena adanya *delay time*, maka grafik aktual memiliki jarak sejak kondisi *starternya*. Hal tersebut dikarenakan mobil membutuhkan *time to accelerate* seperti yang sudah dijelaskan pada subbab sebelumnya.

Pada grafik aktual dapat dilihat bahwa pada sekitar detik ke 200 hingga 600 mobil langsung mengalami percepatan yang signifikan hingga 100 km/jam, hal itu dapat disebabkan karena pemutaran potensiometer yang terlalu besar dalam waktu yang singkat. Lalu pada waktu sisanya, grafik aktual akan terus mengikuti profil pada grafik ideal hingga detik ke 5000. Antara detik ke 600 hingga detik ke 5000, perubahan kecepatannya sangat bervariasi, sehingga dapat ditarik strategi *race* nya untuk menjaga kecepatan sebesar 100 km/jam pada detik ke 600 hingga 5000.

4.3.4 Grafik Kecepatan Fungsi Waktu Hari Keempat

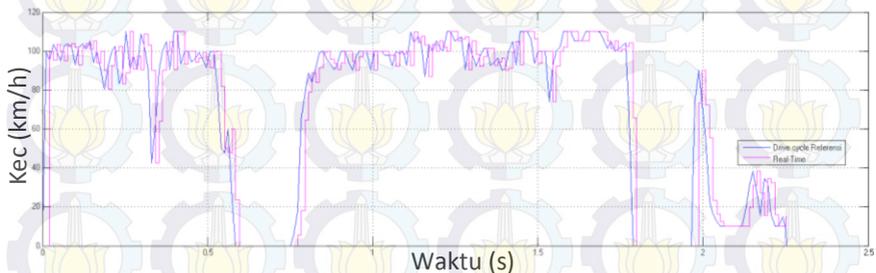
Berikut merupakan grafik kecepatan fungsi waktu pada hari keempat perlombaan. Sumbu x menunjukkan kecepatan (dalam satuan km/jam) dan sumbu y menunjukkan waktu (dalam satuan detik). Grafik yang berwarna biru menunjukkan profil kecepatan ideal/referensi yang didapatkan dari *driving cycle*

sedangkan grafik yang berwarna merah muda menunjukkan profil kecepatan secara *real time* yang didapat dari pengambilan data.



Gambar 4.29 Tampilan Grafik Kecepatan Hari Keempat pada *System Strategy Monitoring*

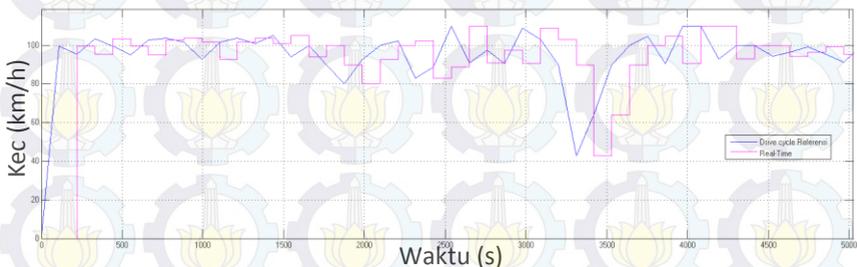
Berikut merupakan tampilan dari grafik kecepatan fungsi waktu untuk hari keempat:



Gambar 4.30 Grafik Kecepatan Fungsi Waktu Hari Keempat

Grafik berwarna biru menunjukkan kecepatan dari Mobil SapuAngin Surya Generasi Kedua selama perlombaan pada hari keempat. Sedangkan grafik berwarna merah muda menunjukkan kecepatan dari Mobil SapuAngin Surya Generasi Kedua selama perlombaan pada hari keempat yang didapat dari pengambilan data dengan mengatur bukaan *throttle* yang diwakilkan oleh putaran potensiometer. Pada hari keempat waktu yang dibutuhkan selama 22500 detik dan kecepatan didapatkan berdasarkan tabel rute dari World Solar Challenge. Sekaligus sudah disesuaikan dengan peraturan-peraturan pada World Solar Challenge bahwa setiap berhenti di control stop harus berhenti terlebih dahulu selama 30 menit. Kecepatan maksimum nya adalah 110 km/jam dan kecepatan rata-ratanya 86 km/jam.

Strategi *race* yang ditentukan untuk hari keempat ditinjau pada waktu antara 0-5000 detik



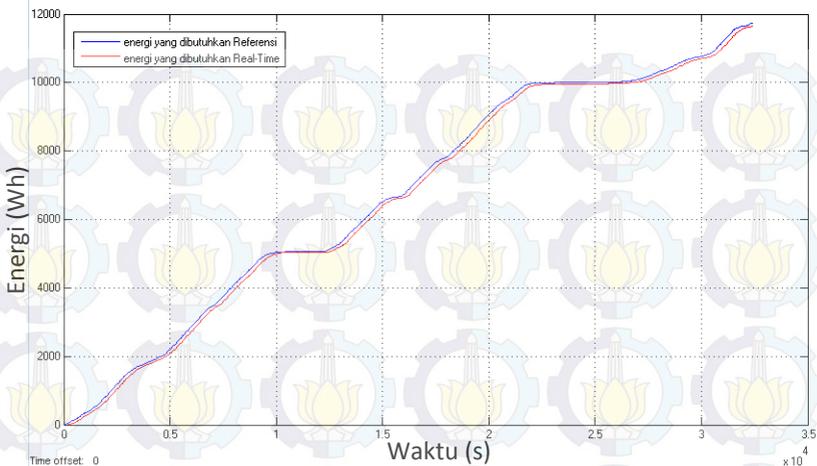
Gambar 4.31 Grafik Kecepatan Fungsi Waktu Hari Keempat Pada Detik 0-5000

Dapat dilihat bahwa terdapat perbedaan antara grafik referensi dengan grafik aktual. Karena adanya *delay time*, maka grafik aktual memiliki jarak sejak kondisi *starternya*. Hal tersebut dikarenakan mobil membutuhkan *time to accelerate* seperti yang sudah dijelaskan pada subbab sebelumnya.

Pada grafik aktual dapat dilihat bahwa pada sekitar detik ke 200 mobil langsung mengalami percepatan yang signifikan hingga 100 km/jam, hal itu dapat disebabkan karena pemutaran potensiometer yang terlalu besar dalam waktu yang singkat. Lalu pada waktu sisa nya, grafik aktual akan terus mengikuti profil pada grafik ideal hingga detik ke 5000. Antara detik ke 200 hingga detik ke 3200, perubahan kecepatannya sangat bervariasi, sehingga dapat ditarik strategi *race* nya untuk menjaga kecepatan sebesar 90 km/jam pada detik ke 200 hingga 3200. Lalu pada detik ke 3300 akan terjadi penurunan kecepatan yang signifikan. Namun, karena penurunan tersebut bukan disebabkan karena adanya pemberhentian pada *control stop*, maka ditentukan strategi *race* nya untuk terus menjaga kecepatan pada 90 km/jam hingga detik ke 5000.

4.3.5 Grafik Konsumsi Energi Hari Pertama

Berikut merupakan grafik perbandingan konsumsi energi referensi dengan konsumsi energi real-time yang didapat dari pengambilan data. Grafik yang berwarna biru menunjukkan konsumsi energi berdasarkan referensi, sedangkan grafik yang berwarna merah muda merupakan konsumsi energi aktual berdasarkan data yang telah diambil sebelumnya.



Gambar 4.32 Grafik Perbandingan Konsumsi Energi Referensi dan Aktual Hari Pertama

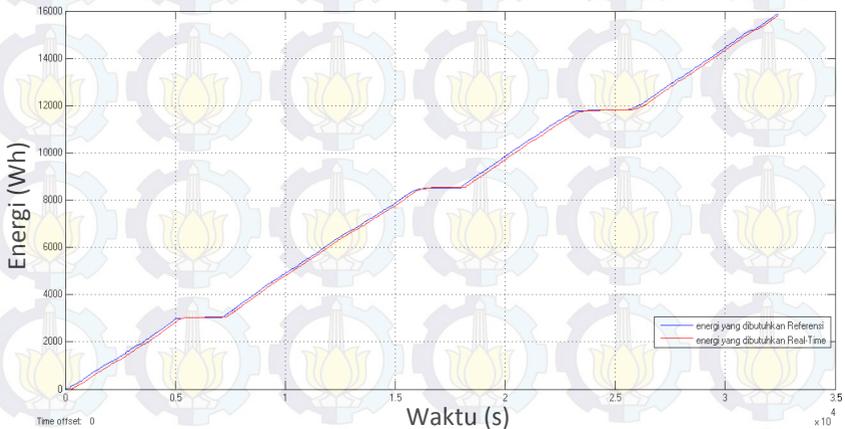
Dari grafik diatas, dapat dilihat bahwa konsumsi energi aktual sudah mendekati konsumsi energi referensinya. Kebutuhan energi berdasarkan referensi selama pukul 08:00 sampai pukul 17:00 sebesar 11,7 KWH. Sedangkan kebutuhan energi baerdasarkan data aktual selama pukul 08:00 sampai pukul 17:00 sebesar 11,6 KWH. Terdapat perbedaan energi sebesar 100 WH.

Karena konsumsi energi aktual nilainya lebih kecil daripada konsumsi energi referensinya, maka strategi yang telah direncanakan dapat diterima. Strategi dapat diterima karena konsumsi energi aktual tidak melebihi konsumsi energi referensi, sehingga tidak terjadi kekurangan energi selama hari pertama *race* berlangsung.

4.3.6 Grafik Konsumsi Energi Hari Kedua

Berikut merupakan grafik perbandingan konsumsi energi referensi dengan konsumsi energi real-time yang didapat dari

pengambilan data. Grafik yang berwarna biru menunjukkan konsumsi energi berdasarkan referensi, sedangkan grafik yang berwarna merah muda merupakan konsumsi energi aktual berdasarkan data yang telah diambil sebelumnya.



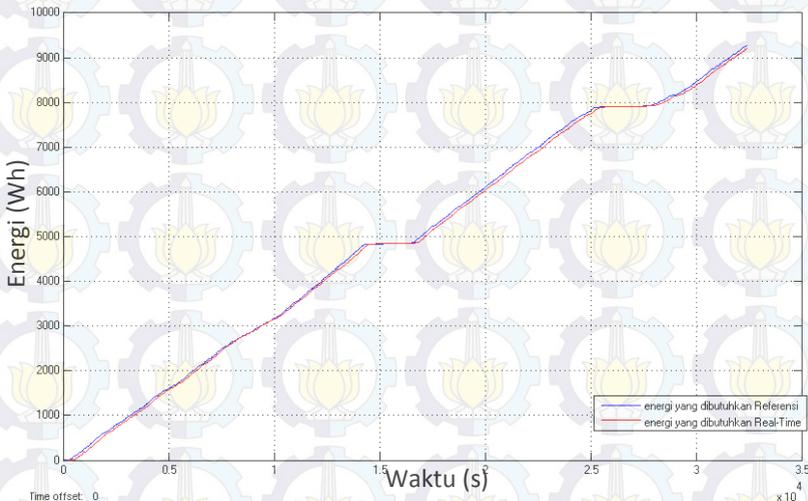
Gambar 4.33 Grafik Perbandingan Konsumsi Energi Referensi dan Aktual Hari Kedua

Dari grafik diatas, dapat dilihat bahwa konsumsi energi aktual sudah mendekati konsumsi energi referensinya. Kebutuhan energi berdasarkan referensi selama pukul 08:00 sampai pukul 17:00 sebesar 15,9 KWH. Sedangkan kebutuhan energi baerdasarkan data aktual selama pukul 08:00 sampai pukul 17:00 sebesar 15,8 KWH. Terdapat perbedaan energi sebesar 100 WH.

Karena konsumsi energi aktual nilainya lebih kecil daripada konsumsi energi referensinya, maka strategi yang telah direncanakan dapat diterima. Strategi dapat diterima karena konsumsi energi aktual tidak melebihi konsumsi energi referensi, sehingga tidak terjadi kekurangan energi selama hari pertama *race* berlangsung.

4.3.7 Grafik Konsumsi Energi Hari Ketiga

Berikut merupakan grafik perbandingan konsumsi energi referensi dengan konsumsi energi real-time yang didapat dari pengambilan data. Grafik yang berwarna biru menunjukkan konsumsi energi berdasarkan referensi, sedangkan grafik yang berwarna merah muda merupakan konsumsi energi aktual berdasarkan data yang telah diambil sebelumnya.



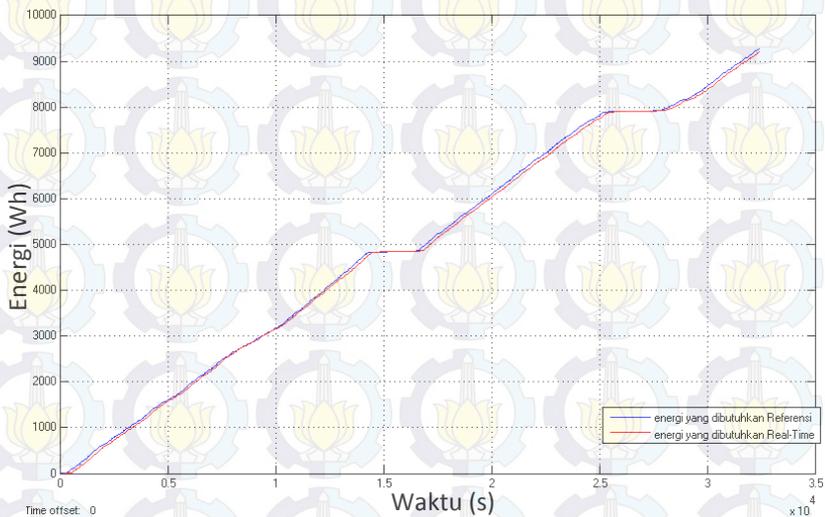
Gambar 4.34 Grafik Perbandingan Konsumsi Energi Referensi dan Aktual Hari Ketiga

Dari grafik diatas, dapat dilihat bahwa konsumsi energi aktual sudah mendekati konsumsi energi referensinya. Kebutuhan energi berdasarkan referensi selama pukul 08:00 sampai pukul 17:00 sebesar 9,27 KWH. Sedangkan kebutuhan energi berdasarkan data aktual selama pukul 08:00 sampai pukul 17:00 sebesar 9,2 KWH. Terdapat perbedaan energi sebesar 70 WH.

Karena konsumsi energi aktual nilainya lebih kecil daripada konsumsi energi referensinya, maka strategi yang telah direncanakan dapat diterima. Strategi dapat diterima karena konsumsi energi aktual tidak melebihi konsumsi energi referensi, sehingga tidak terjadi kekurangan energi selama hari pertama *race* berlangsung.

4.3.8 Grafik Konsumsi Energi Hari Keempat

Berikut merupakan grafik perbandingan konsumsi energi referensi dengan konsumsi energi real-time yang didapat dari pengambilan data. Grafik yang berwarna biru menunjukkan konsumsi energi berdasarkan referensi, sedangkan grafik yang berwarna merah muda merupakan konsumsi energi aktual berdasarkan data yang telah diambil sebelumnya.



Gambar 4.35 Grafik Perbandingan Konsumsi Energi Referensi dan Aktual Hari Keempat

Dari grafik diatas, dapat dilihat bahwa konsumsi energi aktual sudah mendekati konsumsi energi referensinya. Kebutuhan energi selama pukul 08:00 sampai pukul 14:30 sebesar 4,96 KWH. Sedangkan kebutuhan energi baerdasarkan data aktual selama pukul 08:00 sampai pukul 14:30 sebesar 4,95 KWH. Terdapat perbedaan energi sebesar 10 WH.

Karena konsumsi energi aktual nilainya lebih kecil daripada konsumsi energi referensinya, maka strategi yang telah direncanakan dapat diterima. Strategi dapat diterima karena konsumsi energi aktual tidak melebihi konsumsi energi referensi, sehingga tidak terjadi kekurangan energi selama hari pertama *race* berlangsung.



BAB V KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dari tugas akhir yang telah dilakukan ini, dapat ditarik beberapa kesimpulan antara lain:

1. Pada *System Strategy Monitoring* yang telah dibuat, digunakan *hardware* yang berfungsi untuk membaca data serial secara *real-time* dari mobil surya ke Matlab menggunakan koneksi *wireless*.

Hardware yang digunakan adalah sebagai berikut:

- Arduino UNO = Sebagai *microcontroller* yang akan mengumpulkan data-data yang terhubung pada sensor-sensor dari motor listrik, panel surya dan BMS pada mobil surya.
- *Transmitter Xbee Pro S3B* = Merupakan modul *radio frequency* yang fungsinya adalah untuk mentransfer data dari mobil surya menuju ke komputer (Matlab).
- *Receiver Xbee Pro S3B* = Merupakan modul *radio frequency* yang fungsinya adalah untuk menerima data dari mobil surya untuk dibaca oleh komputer (Matlab).
- FTDI sebagai *Serial to USB converter (Xbee Adapter)* = Sebagai *adapter* untuk menyambungkan *receiver Xbee* ke komputer dan untuk mengconvert data dari serial menjadi USB.

Hardware yang telah terangkai ini dapat berjalan dengan baik. Karena belum terselesaikannya mobil SapuAngin Surya II, maka digunakan potensiometer yang dimodelkan sebagai *throttle* pada mobil surya. Namun jika nantinya potensiometer tersebut digantikan dengan

sensor-sensor yang terhubung langsung dengan mobil surya yang sudah terselesaikan, maka *hardware* ini dapat membaca dan melakukan komunikasi secara *wireless* untuk mentransfer data dari mobil surya tersebut.

2. *System Strategy Monitoring* yang telah dibuat dapat memonitor kondisi mobil surya secara *real-time*, dimana output dari *System Strategy Monitoring* itu sendiri adalah:

- Data dari panel surya
- Data dari motor listrik
- Data dari BMS
- Grafik konsumsi energi *real-time*
- Grafik profil kecepatan *real-time*

Software yang telah dibuat dapat membaca data secara *real-time* menggunakan *hardware* yang telah dirangkai. Namun karena belum terselesaikannya mobil SapuAngin Surya II itu sendiri, maka dalam penelitian ini, data *real-time* yang akan diambil diwakilkan oleh potensiometer. Potensiometer tersebut akan dimodelkan sebagai *throttle* dimana tiap putaran dari potensiometer itu sendiri akan mewakili bukaan *throttle* pada mobil surya. Bukaan *throttle* sendiri akan mempengaruhi putaran motor listrik sehingga juga akan mempengaruhi kecepatan pada mobil. Namun jika nantinya potensiometer tersebut digantikan dengan sensor-sensor yang terhubung langsung dengan mobil surya yang sudah terselesaikan, maka *software* ini dapat membaca data-data tersebut.

5.2 Saran

Penelitian ini masih memiliki banyak kekurangan, karenanya penulis mempunyai beberapa saran untuk kelanjutan pengembangan dari *System Strategy Monitoring* agar didapatkan pengembangan yang lebih baik kedepannya, antara lain:

1. Pembuatan software *System Strategy Monitoring* perlu perencanaan dan analisa yang lebih lanjut untuk mendapatkan manajemen energi dan strategi *race* yang lebih baik.
2. Perlunya didapatkan datasheet mengenai hubungan *throttle* dengan putaran motor yang akurat agar didapat perbandingan yang lebih baik.
3. Perlunya ketelitian lebih dalam pengambilan data agar didapatkan data yang lebih baik.



BAB V KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dari tugas akhir yang telah dilakukan ini, dapat ditarik beberapa kesimpulan antara lain:

1. Pada *System Strategy Monitoring* yang telah dibuat, digunakan *hardware* yang berfungsi untuk membaca data serial secara *real-time* dari mobil surya ke Matlab menggunakan koneksi *wireless*.

Hardware yang digunakan adalah sebagai berikut:

- Arduino UNO = Sebagai *microcontroller* yang akan mengumpulkan data-data yang terhubung pada sensor-sensor dari motor listrik, panel surya dan BMS pada mobil surya.
- *Transmitter Xbee Pro S3B* = Merupakan modul *radio frequency* yang fungsinya adalah untuk mentransfer data dari mobil surya menuju ke komputer (Matlab).
- *Receiver Xbee Pro S3B* = Merupakan modul *radio frequency* yang fungsinya adalah untuk menerima data dari mobil surya untuk dibaca oleh komputer (Matlab).
- FTDI sebagai *Serial to USB converter (Xbee Adapter)* = Sebagai *adapter* untuk menyambungkan *receiver Xbee* ke komputer dan untuk mengconvert data dari serial menjadi USB.

Hardware yang telah terangkai ini dapat berjalan dengan baik. Karena belum terselesaikannya mobil SapuAngin Surya II, maka digunakan potensiometer yang dimodelkan sebagai *throttle* pada mobil surya. Namun jika nantinya potensiometer tersebut digantikan dengan

sensor-sensor yang terhubung langsung dengan mobil surya yang sudah terselesaikan, maka *hardware* ini dapat membaca dan melakukan komunikasi secara *wireless* untuk mentransfer data dari mobil surya tersebut.

2. *System Strategy Monitoring* yang telah dibuat dapat memonitor kondisi mobil surya secara *real-time*, dimana output dari *System Strategy Monitoring* itu sendiri adalah:

- Data dari panel surya
- Data dari motor listrik
- Data dari BMS
- Grafik konsumsi energi *real-time*
- Grafik profil kecepatan *real-time*

Software yang telah dibuat dapat membaca data secara *real-time* menggunakan *hardware* yang telah dirangkai. Namun karena belum terselesaikannya mobil SapuAngin Surya II itu sendiri, maka dalam penelitian ini, data *real-time* yang akan diambil diwakilkan oleh potensiometer. Potensiometer tersebut akan dimodelkan sebagai *throttle* dimana tiap putaran dari potensiometer itu sendiri akan mewakili bukaan *throttle* pada mobil surya. Bukaan *throttle* sendiri akan mempengaruhi putaran motor listrik sehingga juga akan mempengaruhi kecepatan pada mobil. Namun jika nantinya potensiometer tersebut digantikan dengan sensor-sensor yang terhubung langsung dengan mobil surya yang sudah terselesaikan, maka *software* ini dapat membaca data-data tersebut.

5.2 Saran

Penelitian ini masih memiliki banyak kekurangan, karenanya penulis mempunyai beberapa saran untuk kelanjutan pengembangan dari *System Strategy Monitoring* agar didapatkan pengembangan yang lebih baik kedepannya, antara lain:

1. Pembuatan software *System Strategy Monitoring* perlu perencanaan dan analisa yang lebih lanjut untuk mendapatkan manajemen energi dan strategi *race* yang lebih baik.
2. Perlunya didapatkan datasheet mengenai hubungan *throttle* dengan putaran motor yang akurat agar didapat perbandingan yang lebih baik.
3. Perlunya ketelitian lebih dalam pengambilan data agar didapatkan data yang lebih baik.



DAFTAR PUSTAKA

Boulgakov, Alexandra. 2012. *Sunswift IV Strategy for the 2011 World Solar Challenge*. Australia: The University of New South Wales School of Electrical Engineering and Telecommunication.

Elshafei, Moustafa. 2013. *Solar Car Optimization For the World Solar Challenge*. Saudi Arabia: King Fahd University of Petroleum and Minerals.

Kaloko, Bambang Sri. 2011. *Design and Development of Small Electric Vehicle Using MATLAB/Simulink*. Indonesia: Dept. of Electrical Engineering Institute of Technology Sepuluh Nopember, Surabaya.

Stanford Solar Car Team. 2013. *Stanford Solar Car Project*. From <http://solarcar.stanford.edu/> , 10 Februari 2014. Amerika: Stanford University.

Sukarman. 2006. *Komunikasi Perangkat Keras Menggunakan Perangkat Lunak Matlab*. Indonesia: Sekolah Tinggi Teknologi Nuklir – Batan.

Wong, K. dan Rizos, C. 1996. *The Australian Surveyor, Part 1- Surveying the Stuart Highway with GPS for the 1996 World Solar Challenge*. Australia.

World Solar Challenge 2013. Bridgestone World Solar Challenge 2013. From <http://www.worldsolarchallenge.org/> , 10 Februari 2014. Australia



LAMPIRAN

1.) Code pada GUI Matlab

```
function varargout = SystemStrategyGUI(varargin)
% SYSTEMSTRATEGYGUI MATLAB code for
% SystemStrategyGUI.fig
%     SYSTEMSTRATEGYGUI, by itself, creates a new
% SYSTEMSTRATEGYGUI or raises the existing
% singleton*.
%
%     H = SYSTEMSTRATEGYGUI returns the handle to a
% new SYSTEMSTRATEGYGUI or the handle to
% the existing singleton*.
%
% SYSTEMSTRATEGYGUI('CALLBACK',hObject,eventData,handles
%,...) calls the local
% function named CALLBACK in SYSTEMSTRATEGYGUI.M
% with the given input arguments.
%
%     SYSTEMSTRATEGYGUI('Property','Value',...)
% creates a new SYSTEMSTRATEGYGUI or raises the
% existing singleton*. Starting from the left,
% property value pairs are
% applied to the GUI before
% SystemStrategyGUI_OpeningFcn gets called. An
% unrecognized property name or invalid value
% makes property application
% stop. All inputs are passed to
% SystemStrategyGUI_OpeningFcn via varargin.
%
% *See GUI Options on GUIDE's Tools menu. Choose
% "GUI allows only one
% instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES
%
% Edit the above text to modify the response to help
% SystemStrategyGUI
% Last Modified by GUIDE v2.5 20-Jun-2014 07:34:17
```

```

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton,
                  ...
                  'gui_OpeningFcn', @SystemStrategyGUI_OpeningFcn, ...
                  'gui_OutputFcn',  @SystemStrategyGUI_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end
if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State,
varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before SystemStrategyGUI is made
% visible.
function SystemStrategyGUI_OpeningFcn(hObject, ~,
handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future
% version of MATLAB
% handles    structure with handles and user data (see
GUIDATA)
% varargin   command line arguments to
SystemStrategyGUI (see VARARGIN)

% Choose default command line output for
SystemStrategyGUI
handles.output = hObject;
% Update handles structure
guidata(hObject, handles);

```

```

% UIWAIT makes SystemStrategyGUI wait for user
response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the
command line.
function varargout = SystemStrategyGUI_OutputFcn(~, ~,
handles)
% varargout cell array for returning output args (see
VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future
version of MATLAB
% handles structure with handles and user data (see
GUIDATA)

% Get default command line output from handles
structure
varargout{1} = handles.output;

function edit1_Callback(~, ~, ~)
% hObject handle to edit1 (see GCBO)
% eventdata reserved - to be defined in a future
version of MATLAB
% handles structure with handles and user data (see
GUIDATA)

% Hints: get(hObject,'String') returns contents of
edit1 as text
% str2double(get(hObject,'String')) returns
contents of edit1 as a double

% --- Executes during object creation, after setting
all properties.
function edit1_CreateFcn(hObject, ~, ~)
% hObject handle to edit1 (see GCBO)
% eventdata reserved - to be defined in a future
version of MATLAB

```

```

% handles    empty - handles not created until after
all CreateFcns called

% Hint: edit controls usually have a white background
on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit2_Callback(~, ~, ~)
% hObject    handle to edit2 (see GCBO)
% eventdata reserved - to be defined in a future
version of MATLAB
% handles    structure with handles and user data (see
GUIDATA)
% Hints: get(hObject,'String') returns contents of
edit2 as text
% str2double(get(hObject,'String')) returns
contents of edit2 as a double

% --- Executes during object creation, after setting
all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata reserved - to be defined in a future
version of MATLAB
% handles    empty - handles not created until after
all CreateFcns called

% Hint: edit controls usually have a white background
on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```
function edit3_Callback(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    structure with handles and user data (see
GUIDATA)

% Hints: get(hObject,'String') returns contents of
edit3 as text
%          str2double(get(hObject,'String')) returns
contents of edit3 as a double

% --- Executes during object creation, after setting
all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    empty - handles not created until after
all CreateFcns called

% Hint: edit controls usually have a white background
on Windows.
%          See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit4_Callback(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    structure with handles and user data (see
GUIDATA)
```

```

% Hints: get(hObject,'String') returns contents of
edit4 as text
%      str2double(get(hObject,'String')) returns
contents of edit4 as a double

% --- Executes during object creation, after setting
all properties.
function edit4_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    empty - handles not created until after
all CreateFcns called

% Hint: edit controls usually have a white background
on Windows.
%      See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit9_Callback(hObject, eventdata, handles)
% hObject    handle to edit9 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    structure with handles and user data (see
GUIDATA)

% Hints: get(hObject,'String') returns contents of
edit9 as text
%      str2double(get(hObject,'String')) returns
contents of edit9 as a double

% --- Executes during object creation, after setting
all properties.
function edit9_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit9 (see GCBO)

```

```

% eventdata reserved - to be defined in a future
version of MATLAB
% handles empty - handles not created until after
all CreateFcns called

% Hint: edit controls usually have a white background
on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit10_Callback(~, eventdata, handles)
% hObject handle to edit10 (see GCBO)
% eventdata reserved - to be defined in a future
version of MATLAB
% handles structure with handles and user data (see
GUIDATA)

% Hints: get(hObject,'String') returns contents of
edit10 as text
% str2double(get(hObject,'String')) returns
contents of edit10 as a double

% --- Executes during object creation, after setting
all properties.
function edit10_CreateFcn(hObject, eventdata, handles)
% hObject handle to edit10 (see GCBO)
% eventdata reserved - to be defined in a future
version of MATLAB
% handles empty - handles not created until after
all CreateFcns called

% Hint: edit controls usually have a white background
on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))

```

```

    set(hObject,'BackgroundColor','white');
end

```

```

function edit11_Callback(hObject, eventdata, handles)
% hObject    handle to edit11 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    structure with handles and user data (see
GUIDATA)

```

```

% Hints: get(hObject,'String') returns contents of
edit11 as text
%        str2double(get(hObject,'String')) returns
contents of edit11 as a double

```

```

% --- Executes during object creation, after setting
all properties.

```

```

function edit11_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit11 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    empty - handles not created until after
all CreateFcns called

```

```

% Hint: edit controls usually have a white background
on Windows.

```

```

%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

function edit12_Callback(hObject, eventdata, handles)
% hObject    handle to edit12 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB

```

```
% handles      structure with handles and user data (see
GUIDATA)
```

```
% Hints: get(hObject,'String') returns contents of
edit12 as text
%         str2double(get(hObject,'String')) returns
contents of edit12 as a double
```

```
% --- Executes during object creation, after setting
all properties.
```

```
function edit12_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit12 (see GCBO)
% eventdata    reserved - to be defined in a future
version of MATLAB
% handles      empty - handles not created until after
all CreateFcns called
```

```
% Hint: edit controls usually have a white background
on Windows.
```

```
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
% --- Executes on button press in
HitungKonsumsiEnergi.
```

```
function HitungKonsumsiEnergi_Callback(hObject,
eventdata, handles)
% hObject      handle to HitungKonsumsiEnergi (see GCBO)
% eventdata    reserved - to be defined in a future
version of MATLAB
% handles      structure with handles and user data (see
GUIDATA)
```

```
global input;
global h;
global x;
global im;
```

```

h = open('EnergyRequirement.fig');
x = getframe(h);
im = x.cdata;

```

```

close(h);
handles.im = im;
axes(handles.axes4);
imshow(im);

```

```

function edit13_Callback(hObject, eventdata, handles)
% hObject    handle to edit13 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    structure with handles and user data (see
GUIDATA)

% Hints: get(hObject,'String') returns contents of
edit13 as text
%         str2double(get(hObject,'String')) returns
contents of edit13 as a double

% --- Executes during object creation, after setting
all properties.
function edit13_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit13 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    empty - handles not created until after
all CreateFcns called

% Hint: edit controls usually have a white background
on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```
function edit14_Callback(hObject, eventdata, handles)
% hObject    handle to edit14 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    structure with handles and user data (see
GUIDATA)

% Hints: get(hObject,'String') returns contents of
edit14 as text
%         str2double(get(hObject,'String')) returns
contents of edit14 as a double

% --- Executes during object creation, after setting
all properties.
function edit14_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit14 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    empty - handles not created until after
all CreateFcns called

% Hint: edit controls usually have a white background
on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit15_Callback(hObject, eventdata, handles)
% hObject    handle to edit15 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    structure with handles and user data (see
GUIDATA)
```

```
% Hints: get(hObject,'String') returns contents of
edit15 as text
%      str2double(get(hObject,'String')) returns
contents of edit15 as a double

% --- Executes during object creation, after setting
all properties.
function edit15_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit15 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    empty - handles not created until after
all CreateFcns called

% Hint: edit controls usually have a white background
on Windows.
%      See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit16_Callback(hObject, eventdata, handles)
% hObject    handle to edit16 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    structure with handles and user data (see
GUIDATA)

% Hints: get(hObject,'String') returns contents of
edit16 as text
%      str2double(get(hObject,'String')) returns
contents of edit16 as a double

% --- Executes during object creation, after setting
all properties.
function edit16_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit16 (see GCBO)
```

```
% eventdata reserved - to be defined in a future
version of MATLAB
% handles empty - handles not created until after
all CreateFcns called

% Hint: edit controls usually have a white background
on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit17_Callback(hObject, eventdata, handles)
% hObject handle to edit17 (see GCBO)
% eventdata reserved - to be defined in a future
version of MATLAB
% handles structure with handles and user data (see
GUIDATA)

% Hints: get(hObject,'String') returns contents of
edit17 as text
% str2double(get(hObject,'String')) returns
contents of edit17 as a double

% --- Executes during object creation, after setting
all properties.
function edit17_CreateFcn(hObject, eventdata, handles)
% hObject handle to edit17 (see GCBO)
% eventdata reserved - to be defined in a future
version of MATLAB
% handles empty - handles not created until after
all CreateFcns called

% Hint: edit controls usually have a white background
on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
```

```

    set(hObject,'BackgroundColor','white');
end

function edit18_Callback(hObject, eventdata, handles)
% hObject    handle to edit18 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    structure with handles and user data (see
GUIDATA)

% Hints: get(hObject,'String') returns contents of
edit18 as text
%         str2double(get(hObject,'String')) returns
contents of edit18 as a double

% --- Executes during object creation, after setting
all properties.
function edit18_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit18 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    empty - handles not created until after
all CreateFcns called

% Hint: edit controls usually have a white background
on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit19_Callback(hObject, eventdata, handles)
% hObject    handle to edit19 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB

```

```

% handles      structure with handles and user data (see
GUIDATA)

% Hints: get(hObject,'String') returns contents of
edit19 as text
%          str2double(get(hObject,'String')) returns
contents of edit19 as a double

% --- Executes during object creation, after setting
all properties.
function edit19_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit19 (see GCBO)
% eventdata    reserved - to be defined in a future
version of MATLAB
% handles      empty - handles not created until after
all CreateFcns called

% Hint: edit controls usually have a white background
on Windows.
%          See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata,
handles)
% hObject      handle to pushbutton3 (see GCBO)
% eventdata    reserved - to be defined in a future
version of MATLAB
% handles      structure with handles and user data (see
GUIDATA)
global input;
global h;
global x;
global im;

h = open('EnergyRequirement.fig');
x = getframe(h);

```

```
im = x.cdata;
```

```
close(h);
handles.im = im;
axes(handles.axes4);
imshow(im);
```

```
% --- Executes on button press in ResetKonsumsiEnergi.
function ResetKonsumsiEnergi_Callback(hObject,
eventdata, handles)
```

```
% hObject    handle to ResetKonsumsiEnergi (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles     structure with handles and user data (see
GUIDATA)
```

```
cla(gca, 'reset');
set(handles.ResetKonsumsiEnergi, 'enable', 'Off');
set(handles.HitungKonsumsiEnergi, 'enable', 'On');
```

```
% --- Executes on button press in btnLoadBaterai.
```

```
function btnLoadBaterai_Callback(~, eventdata,
handles)
```

```
% hObject    handle to btnLoadBaterai (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles     structure with handles and user data (see
GUIDATA)
```

```
SerialPort = 'COM3';
```

```
s = serial(SerialPort);
get(s);
```

```
fopen(s);
```

```
tic;
i=0;
```

```
    data = fscanff(s, '%f');
```

```
    i=i+1;
```

```
    time(i)=toc;
```

```
v(i)=data;
disp(v);

% --- Executes on button press in btnLoadMotor.
function btnLoadMotor_Callback(hObject, eventdata, handles)
% hObject    handle to btnLoadMotor (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    structure with handles and user data (see
GUIDATA)

% --- Executes on button press in btnLoadPanel.
function btnLoadPanel_Callback(hObject, eventdata, handles)
% hObject    handle to btnLoadPanel (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    structure with handles and user data (see
GUIDATA)

% --- Executes on button press in btnTutup.
function btnTutup_Callback(hObject, eventdata, handles)
% hObject    handle to btnTutup (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    structure with handles and user data (see
GUIDATA)
close;

% --- Executes during object creation, after setting
all properties.
function GrafikKecepatan_CreateFcn(~, ~, ~)
% hObject    handle to GrafikKecepatan (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
```

```
% handles    empty - handles not created until after
all CreateFcns called
```

```
% Hint: place code in OpeningFcn to populate
GrafikKecepatan
```

```
% --- Executes on button press in pushbutton10.
```

```
function pushbutton10_Callback(hObject, eventdata,
handles)
```

```
cla(gca, 'reset');
```

```
set(handles.pushbutton10, 'enable', 'Off');
```

```
set(handles.pushbutton12, 'enable', 'On');
```

```
% hObject    handle to pushbutton10 (see GCBO)
```

```
% eventdata  reserved - to be defined in a future
version of MATLAB
```

```
% handles    structure with handles and user data (see
GUIDATA)
```

```
% --- Executes on button press in pushbutton11.
```

```
function pushbutton11_Callback(hObject, eventdata,
handles) global input;
```

```
global h;
```

```
global x;
```

```
global im;
```

```
h = open('potPlot.m');
```

```
x = getframe(h);
```

```
im = x.cdata;
```

```
close(h);
```

```
handles.im = im;
```

```
axes(handles.GrafikEnergi);
```

```
imshow(im);
```

```
SerialPort = 'COM3';
```

```
s = serial(SerialPort);
```

```
get(s);
```

```
fopen(s);
```

```

tic;
i=0;
while
    data = fscanf(s,'%f');
    i=i+1;
    time(i)=toc;
    v(i)=data;
    figure(1);
    plot(time,v,'r');
    disp(v);
    drawnow;
    pause(0.1);
end
fclose(s);
clear s;

% hObject    handle to pushbutton11 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    structure with handles and user data (see
GUIDATA)

% --- Executes during object creation, after setting
all properties.
function axes4_CreateFcn(hObject, eventdata, handles)
% hObject    handle to axes4 (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    empty - handles not created until after
all CreateFcns called
plot(ScopeData.time,ScopeData.signals.values);
plot(handles.axes4,handles.x,handles.y);
% Hint: place code in OpeningFcn to populate axes4

% --- Executes on button press in pushbutton12.
function pushbutton12_Callback(hObject, eventdata,
handles)
global input;

```

```

global h;
global x;
global im;

h = open('potPlot.m');
x = getframe(h);
im = x.cdata;

close(h);
handles.im = im;
axes(handles.GrafikEnergi);
imshow(im);

SerialPort = 'COM3';

s = serial(SerialPort);
get(s);
fopen(s);

tic;
i=0;
while
    data = fscanf(s,'%f');
    i=i+1;
    time(i)=toc;
    v(i)=data;
    figure(1);
    plot(time,v,'r');
    disp(v);
    drawnow;
    pause(0.1);
end
fclose(s);
clear s;
% hObject handle to pushbutton12 (see GCBO)
% eventdata reserved - to be defined in a future
version of MATLAB
% handles structure with handles and user data (see
GUIDATA)

```

```
% --- Executes during object creation, after setting
all properties.
function GrafikEnergi_CreateFcn(hObject, eventdata,
handles)
% hObject    handle to GrafikEnergi (see GCBO)
% eventdata  reserved - to be defined in a future
version of MATLAB
% handles    empty - handles not created until after
all CreateFcns called

% Hint: place code in OpeningFcn to populate
GrafikEnergi
```

2.) Code pada Arduino

```
// Requires remote XBee in loopback mode (DIN connected  
to DOUT)
```

```
#include <SoftwareSerial.h>
```

```
#define Rx 3 // DOUT to pin 3
```

```
#define Tx 4 // DIN to pin 4
```

```
SoftwareSerial Xbee (Rx, Tx);
```

```
void setup() {
```

```
  Serial.begin(9600); // Set to No line ending;
```

```
  Xbee.begin(9600); // type a char, then hit enter
```

```
  delay(500);
```

```
}
```

```
void loop() {
```

```
  int sensorValue = analogRead(A0);
```

```
char outgoing = Serial.read(); // Read character, send to  
XBee
```

```
  Xbee.println(sensorValue);
```

```
  Serial.println(sensorValue);
```

```
  if(Xbee.available()) { // Is data available from XBee?
```

```
    char incoming = Xbee.read(); // Read character,
```

```
    Serial.println(incoming); // send to Serial Monitor
```

```
  }
```

```
  delay(500);
```

```
}
```

3.) **Setting** pada Xbee Pro S3B

```

<?xml version="1.0" encoding="UTF-8" ?>
<data>
<profile>
<description_file>XBP9B_8067.xml</description_file>
<settings>
<setting command="CM">FFFFFFFFF7FFFF</setting>
<setting command="HP">0</setting>
<setting command="ID">7FFF</setting>
<setting command="MT">3</setting>
<setting command="PL">4</setting>
<setting command="RR">A</setting>
<setting command="CE">0</setting>
<setting command="BH">0</setting>
<setting command="NH">7</setting>
<setting command="MR">1</setting>
<setting command="NN">3</setting>
<setting command="DH">0</setting>
<setting command="DL">FFFF</setting>
<setting command="TO">C0</setting>
<setting command="NI">0x20</setting>
<setting command="NT">82</setting>
<setting command="NO">0</setting>
<setting command="CI">11</setting>
<setting command="EE">0</setting>
<setting command="KY" />
<setting command="BD">3</setting>
<setting command="NB">0</setting>
<setting command="SB">0</setting>
<setting command="RO">3</setting>
<setting command="FT">13F</setting>
<setting command="AP">0</setting>
<setting command="AO">0</setting>
<setting command="DO">1</setting>
<setting command="D1">0</setting>
<setting command="D2">0</setting>
<setting command="D3">0</setting>

```

```
<setting command="D4">0</setting>
<setting command="D5">1</setting>
<setting command="D6">0</setting>
<setting command="D7">1</setting>
<setting command="D8">1</setting>
<setting command="D9">1</setting>
<setting command="P0">1</setting>
<setting command="P1">0</setting>
<setting command="P2">0</setting>
<setting command="P3">1</setting>
<setting command="P4">1</setting>
<setting command="PD">7FFF</setting>
<setting command="PR">7FFF</setting>
<setting command="M0">0</setting>
<setting command="M1">0</setting>
<setting command="LT">0</setting>
<setting command="RP">28</setting>
<setting command="AV">1</setting>
<setting command="IC">0</setting>
<setting command="IF">1</setting>
<setting command="IR">0</setting>
<setting command="SM">0</setting>
<setting command="SO">2</setting>
<setting command="SN">1</setting>
<setting command="SP">12C</setting>
<setting command="ST">BB8</setting>
<setting command="WH">0</setting>
<setting command="CC">2B</setting>
<setting command="CT">64</setting>
<setting command="GT">3E8</setting>
<setting command="DD">B000</setting>
</settings>
</profile>
</data>
```



BIOGRAFI PENULIS



Penulis dilahirkan di Surabaya, 28 Februari 1993. Merupakan anak pertama dari tiga bersaudara. Penulis selama hidupnya telah menempuh pendidikan formal di SD Negeri Pemuda Bangsa Depok II selama tiga tahun kemudian pindah pada tahun ketiga di SD Islam Terpadu Darul Abidin Depok I dan setelah itu pindah ke kota Malang di tahun keenam di MI Jendral Sudirman Malang. Penulis kemudian melanjutkan pendidikan formal di SMP Negeri 3 Malang dan SMA Negeri 3 Malang. Setelah tamat pendidikan SMA tahun 2010, penulis melanjutkan pendidikan tingkat sarjana di jurusan Teknik Mesin Institut Teknologi Sepuluh Nopember (ITS), Surabaya.

Di jurusan Teknik Mesin ITS, penulis aktif dalam berbagai kegiatan kemahasiswaan dan juga kepanitiaan di berbagai acara. Penulis tercatat sebagai pengurus aktif Lembaga Bengkel Mahasiswa Mesin ITS sebagai Staff Divisi Hubungan Luar tahun 2011-2012 dan Kabiro Promosi tahun 2012-2013. Penulis juga tergabung dalam ITS *Solar Car Racing Team* yang telah mengikuti kompetisi tingkat internasional *World Solar Challenge* 2013. Penulis dapat dihubungi melalui email dengan alamat nadhira.raafianti@gmail.com.