



TUGAS AKHIR - TE090362

**KOTAK HITAM SEBAGAI PENCATAT Pengereman,
KECEPATAN, DAN TEMPERATUR MESIN SEPEDA MOTOR
DENGAN MEDIA *SECURE DIGITAL CARD***

Rudy Arif Prasojo
NRP 2211 030 016
Ghosyiyah Aisisar
NRP 2211 030 080

Dosen Pembimbing
Ir. Josaphat Pramudijanto, M.Eng

PROGRAM STUDI D3 TEKNIK ELEKTRO
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya 2014



FINAL PROJECT- TE090362

***THE BLACK BOX AS THE RECORDER
OF MOTORCYCLE'S BREAKING, SPEED AND TEMPERATURE
OF MACHINE THROUGH SECURE DIGITAL CARD***

Rudy Arif Prasojo
NRP 2211 030 016
Ghosyia Aisisar
NRP 2211 030 080

Supervisor
Ir. Josaphat Pramudijanto, M.Eng

*Electrical Engineering D3 Study Program
Industrial Technology Faculty
Institut Teknologi Sepuluh Nopember
Surabaya 2014*

**KOTAK HITAM SEBAGAI PENCATAT Pengereman,
KECEPATAN, DAN TEMPERATUR MESIN SEPEDA
MOTOR DENGAN MEDIA *SECURE DIGITAL CARD***

Nama : Rudy Arif Prasajo
NRP : 2210 030 016
Nama : Ghosyiya Aisisar
NRP : 2210 030 080
Dosen Pembimbing : Ir. Josaphat Pramudijanto, MEng
NIP : 19621005 199003 1 003

ABSTRAK

Penyedia jasa persewaan sepeda motor pada tempat wisata seringkali mengalami kejadian kehilangan sepeda motor akibat dicuri atau terjadi kecelakaan saat sepeda motor tersebut disewa oleh orang yang menyewanya. Tidak adanya sebuah alat yang dapat menyimpan data terakhir kendaraan dan memberikan posisi secara langsung untuk menemukan sepeda motor inilah yang akan menjadi permasalahan ketika sepeda motor yang sedang disewa hilang atau mengalami kecelakaan. Proses identifikasi yang akan dilakukan oleh pihak kepolisian atau pihak pemilik jasa akan persewaan mengalami kesulitan.

Oleh karena itu dibutuhkan suatu inovasi yang dapat mencatat *history* terakhir dari sepeda bermotor dan mampu memberikan posisi sepeda motor, konsep paling sederhana membuat semacam kotak hitam dan *transponder* seperti terpasang di pesawat terbang namun dipasang pada sepeda motor. Kotak hitam ini akan dipasang sensor tekanan pada bagian pengereman, sensor kecepatan agar kecepatan putaran roda dapat diketahui dan sensor temperatur pada bagian mesin selain itu dilengkapi *gps* untuk memberikan posisi, sehingga posisi dari sepeda motor akan diketahui.

Kotak hitam ini akan menyimpan data sensor dan posisi menggunakan *secure digital card* dan akan mengirimkan sms berupa posisi setiap satu menit sekali.

Kata Kunci : Kotak Hitam, *Transponder*, *Data Logger*, *Secure Digital Card*

Halaman ini sengaja dikosongkan

**THE BLACK BOX AS THE RECORDER OF MOTORCYCLE'S
BREAKING, SPEED AND TEMPERATURE OF MACHINE
THROUGH SECURE DIGITAL CARD**

Name of Student 1 : Rudy Arif Prasojo
Number of Registration : 2210 030 016
Name of Student 2 : Ghosyiya Aisisar
Number of Registration : 2210 030 080
Supervisor : Ir. Josaphat Pramudijanto, M.Eng
ID : 19621005 199003 1 003

ABSTRACT

Rental motorcycle company on spot holiday oftentimes have bad tragedy, like missing or accident their motorcycle when it renting to tourist. Until now no one technology which save last history motorcycle and can give the position, make a new problem when the motorcycle still use with other person who renting the motorcycle. Oftentimes, process of identification for victim of traffic accident has been done less accurately by polices

Therefore, it needs some innovations that can record the last history fom vehicle. The simplest concept is how to make a kind of black box that set in vehicle. This equipment is adoption from the black box of airplane. The black box and transpoder is set by cencor of pressure at breaking part, censor of speed in order that speed of wheel's rotation when breaking part, censor of speed in order that speed of wheel's rotation when breaking is ascertainable from the censor of temperatur at engine-fittings. The way of working this tool is save the data from vehicle like pressure when breaking, speed and temperature of vehicle engine that will be keep automatically in secure digital card. In other word, the black box will be the last history from vehicle automatically and this system be equipped with gps for sending the position of motorcycle soon.

This black box will save the censor data dan position with secure digital card, and give the position via short massage service in one minutes in this system on.

Kata Kunci : *Black Box, Transponder, Datalogger, Secure Digital Card*

Halaman ini sengaja dikosongkan

**KOTAK HITAM SEBAGAI PENCATAT Pengereman,
KECEPATAN, DAN TEMPERATUR MESIN SEPEDA MOTOR
DENGAN MEDIA *SECURE DIGITAL CARD***

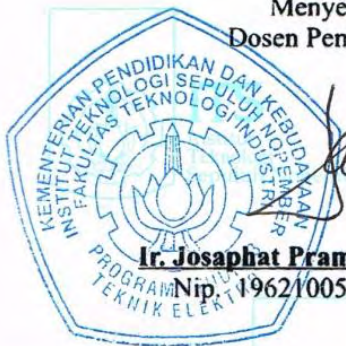
TUGAS AKHIR

**Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Ahli Madya**

Pada

**Bidang Studi Komputer Kontrol
Program Studi D3 Teknik Elektro
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember**

**Menyetujui :
Dosen Pembimbing,**



Ir. Josaphat Pramudijanto M.Eng

Nip. 19621005 199003 1 003

**SURABAYA
JULI, 2014**

KATA PENGANTAR

Dengan mengucapkan puji syukur kehadirat Allah SWT yang telah melimpahkan rahmat, hidayah serta karunia-Nya sehingga penulis dapat menyelesaikan Tugas Akhir dengan judul :

“KOTAK HITAM SEBAGAI PENCATAT Pengereman, KECEPATAN, DAN TEMPERATUR MESIN SEPEDA MOTOR DENGAN MEDIA *SECURE DIGITAL CARD* ”

Dengan selesainya Tugas Akhir ini penulis menyampaikan terima kasih sebesar-besarnya kepada:

1. Kedua Orang Tua atas limpahan doa, kasih sayang, dukungan dan dorongan baik berupa moril atau materil bagi penulis.
2. Bapak Eko Setijadi, ST. MT. Ph.D selaku Ketua Program D3 Teknik Elektro Jurusan Teknik Elektro FTI-ITS Surabaya.
3. Bapak Ir. Josaphat Pramudijanto, M.Eng selaku dosen pembimbing yang telah meluangkan waktu dan bimbingannya dalam pembuatan tugas akhir ini.
4. Seluruh staf pengajar dan administrasi Prodi D3 Teknik Elektro FTI-ITS.
5. Seluruh mahasiswa D3 Teknik Elektro khususnya angkatan 2011 dan semua pihak yang telah membantu.

Kritik dan saran untuk perbaikan Tugas Akhir ini sangat diperlukan untuk perbaikan agar kotak hitam dan *transponder* ini bisa bermanfaat untuk masyarakat dan kemajuan bangsa Indonesia.

Surabaya, 8 Juli 2014

Penulis

Halaman ini sengaja dikosongkan

DAFTAR ISI

HALAMAN JUDUL	i
LEMBAR PENGESAHAN	iii
ABSTRAK	v
ABSTRACT	vii
KATA PENGANTAR	ix
DAFTAR ISI	xi
DAFTAR GAMBAR	xiii
DAFTAR TABEL	xv

BAB I PENDAHULUAN

1.1 Latar Belakang	1
1.2 Permasalahan.....	2
1.3 Batasan Masalah	2
1.4 Tujuan	3
1.5 Manfaat	3
1.6 Sistematika Laporan.....	4
1.7 Relevansi	5

BAB II TEORI PENUNJANG

2.1. Arduino	7
2.1.1. <i>Arduio Mega 2560</i>	8
2.1.2. <i>Arduino Ethernet Shield</i>	9
2.2. Sensor <i>Load Cell</i>	9
2.3. Sensor <i>Rotary Encoder</i>	10
2.4. Sensor <i>Thermocouple</i>	12
2.5. RTC (<i>Real Time Clock</i>).....	14
2.6. <i>Data Logging</i>	17
2.7. <i>Secure Digital Card (SD Card)</i>	18
2.8. GPS <i>U-Blox Neo 6M</i>	20
2.9. <i>Modem GSM Fastrack M1306B</i>	22

BAB. III PERANCANGAN DAN PEMBUATAN ALAT

3.1 Perancangan Perangkat Keras	23
3.1.1 <i>Arduino Mega 2560 dan Arduino Ethernet Shield</i> ...	24
3.1.2 Perancangan Rangkaian Sensor Tekanan	25
3.1.3 Perancangan Rangkaian Sensor Kecepatan	25
3.1.4 Perancangan Rangkaian Sensor Temperatur.....	26

3.1.5	Regulator	27
3.1.6	Perancangan Rangkaian (RTC) <i>Real Time Clock</i> ...	27
3.1.7	Perancangan Rangkaian Driver Max 232	28
3.2	Perancangan Desain Mekanik.....	28
3.3	Perancangan Desain <i>Software</i> Sistem.....	29

BAB. IV PENGUJIAN DAN ANALISA DATA

4.1	Pembuatan Prototipe Kotak Hitam	33
4.2	Pengujian <i>Stainlesssteel</i>	34
4.3	Pengujian <i>Arduino Mega dan Ethernet Shield</i>	34
4.4	Pengujian Sensor Tekanan <i>Loadcell</i>	36
4.5	Pengujian Sensor Temperatur <i>Thermocouple</i>	40
4.6	Pengujian Sensor Kecepatan <i>Rotary Encoder</i>	43
4.7	Pengujian <i>Real Time Clock</i>	47
4.8	Pengujian Regulator	48
4.9	Pengujian <i>Global Positioning System (GPS)</i>	50
4.10	Pengujian <i>Modem Wavecom GSM</i>	54
4.11	Pengujian Tampilan <i>Datalogger SD Card</i>	55
4.12	Uji Coba Kotak Hitam dan <i>Transponder Sepeda Motor</i> ...	56

BAB. V PENUTUP

5.1.	Kesimpulan	59
5.2.	Saran	60

DAFTAR PUSTAKA	61
LAMPIRAN A <i>Listing Program</i>	A-1
LAMPIRAN B <i>Flowchart</i>	B-1
LAMPIRAN C <i>Datasheet</i>	C-1
LAMPIRAN D Bentuk Alat	D-1
DAFTAR RIWAYAT HIDUP	E-1

DAFTAR TABEL

Tabel 3.1	Penggunaan <i>Port</i> Arduino	24
Tabel 4.1	Hasil Pengujian Bahan <i>Stainlessteel</i>	34
Tabel 4.2	Hasil Pengujian <i>Arduino Mega 2560</i> dan <i>Ethernet Shield</i>	35
Tabel 4.3	Data Pengukuran Tekanan <i>Loadcell</i>	38
Tabel 4.4	Data Pengukuran Temperatur <i>Thermocouple</i>	42
Tabel 4.5	Data Pengukuran Kecepatan <i>Rotary Encoder</i>	45
Tabel 4.6	Data Pengukuran Tegangan Regulator.....	50
Tabel 4.7	Perbandingan Hasil <i>GPS</i>	51

Halaman ini sengaja dikosongkan

DAFTAR GAMBAR

Gambar 2.1	<i>Arduino Mega ADK 2560</i>	8
Gambar 2.2	<i>Arduino Ethernet Shield</i>	9
Gambar 2.3	Sensor <i>Load Cell</i>	10
Gambar 2.4	Sensor <i>Rotary Encoder</i>	11
Gambar 2.5	Sensor <i>Thermocouple</i>	12
Gambar 2.6	Rangkaian <i>RTC DS1307</i>	15
Gambar 2.7	Macam-macam <i>Data Logger</i>	17
Gambar 2.8	Macam-macam <i>SD Card</i>	19
Gambar 2.9	<i>GPS U-Blox Neo 6M</i>	20
Gambar 2.10	<i>Modem GSM Fasttrack M1306B</i>	22
Gambar 3.1	Diagram Fungsional Sistem	23
Gambar 3.2	Rangkaian <i>INA 125 Load Cell</i>	25
Gambar 3.3	Rangkaian Pulsa <i>Led-Photodiode</i>	26
Gambar 3.4	Rangkaian <i>LM741 Temperature</i>	26
Gambar 3.5	Rangkaian <i>Regulator 5 Volt dan 9 Volt</i>	27
Gambar 3.6	Rangkaian <i>RTC DS1307</i>	27
Gambar 3.7	Rangkaian <i>MAX232</i>	28
Gambar 3.8	Desain <i>Hardware Sistem</i>	29
Gambar 3.9	<i>Flowchart Sistem</i>	30
Gambar 4.1	Hasil <i>Hardware Sistem</i>	33
Gambar 4.2	<i>Listing Program Mengaktifkan Pin Arduino</i>	35
Gambar 4.3	Proses Pengukuran Sensor <i>Load Cell</i>	37
Gambar 4.4	Linier Pengukuran <i>Load Cell</i>	39
Gambar 4.5	Pengukuran Tegangan (2000 Gram)	39
Gambar 4.6	Proses Pengukuran Sensor <i>Thermocouple</i>	40
Gambar 4.7	Linier Pengukuran <i>Thermocouple</i>	42
Gambar 4.8	Proses Pengukuran Tempetur dengan Bejana.....	43
Gambar 4.9	Proses Pengukuran <i>Rotary Encoder</i>	44
Gambar 4.10	Linier Pengukuran <i>Rotary Encoder</i>	47
Gambar 4.11	Hasil Pengaturan <i>Real Time Clock</i>	48
Gambar 4.12	Hasil Pengujian <i>Regulator</i>	49
Gambar 4.13	Tampilan <i>Longitude dan Latitude</i>	50
Gambar 4.14	Pengujian <i>Modem Sms pada Handphone</i>	54
Gambar 4.15	Pengujian Tampilan <i>Data Logger</i>	55
Gambar 4.16	Tampilan <i>Data Logger Secure Digital Card</i>	56
Gambar 4.17	Tampilan <i>Microsoft Excel</i>	57
Gambar 4.18	Tampilan <i>Latitude dan Longitude Handphone</i>	57
Gambar 4.19	Tampilan <i>Google Map</i>	58

Halaman ini sengaja dikosongkan

BAB I

PENDAHULUAN

1.1 Latar Belakang

Angka pengunjung pariwisata di Indonesia dari tahun ketahun semakin meningkat dengan jumlah objek wisata yang ditawarkan oleh masing masing daerah di Indoneia semakin banyak. Berdasarkan data Kementrian Pariwisata dan Ekonomi Kreatif Republik Indonesia (Parekraf RI) bahwa presentase kenaikan kunjungan wisatawan asing dan domestik tiap bulannya mencapai 2,67% hingga 52,89 % pada bulan Januari hingga Mei 2014. Banyak sekali faktor yang menunjang kenaikan pariwisata di Indonesia. Dengan ini ekonomi dalam negeri, dalam skala kecil hingga besar ikut terdorong naik, salah satunya adalah penyedia jasa persewaan sepeda motor (*Motorcycle Rental*) untuk turis-turis yang ingin mengunjungi tempat-tempat wisata yang ada atau sekedar berkeliling objek wisata dengan menggunakan motor.

Namun dalam proses menyewa sepeda motor tersebut, hanya mengandalkan kepercayaan antara *customer* atau pengunjung dengan pemilik persewaan motor tersebut. Masih banyak tempat persewaan sepeda motor ini tidak memiliki sebuah sistem keamanan untuk menyimpan *history* dari keadaan sepeda motor selama disewa dan dapat mendeteksi posisi dari sepeda motor yang disewa secara mudah. Hal ini dimaksudkan apabila kendaraan yang disewa ini sengaja menghilang atau dicuri atau terjadi sesuatu kejadian yang tidak diinginkan seperti kecelakaan lalu lintas yang seringkali terjadi tiba tiba. Banyak sekali faktor yang menyebabkan kecelakaan, dari kondisi motor saat digunakan tidak baik hingga kondisi pengendara yang tidak fokus dalam berkendara, namun seringkali proses identifikasi korban kecelakaan lalu lintas yang dilakukan oleh pihak kepolisian kurang akurat, karena hanya dengan mengandalkan keterangan saksi, keterangan para korban yang dilihat dari proses wawancara dan keadaan psikis pasca kecelakaan, melihat keadaan tempat kejadian perkara, periksa dokumen, pemeriksaan keadaan kendaraan terakhir tanpa mengetahui data data terakhir kendaraan yang terlibat kecelakaan, seperti kecepatan, panas mesin, dan tekanan saat pengereman terjadi. Dalam kasus kecelakaan jalan raya khususnya sepeda bermotor, biasanya polisi

hanya menyimpulkan dari keterangan mengenai keadaan terakhir kendaraan. Selain itu ketika sepeda motor itu hilang karena dicuri, biasanya memerlukan waktu banyak untuk kembali menemukan sepeda motor tersebut dengan berbagai syarat, seperti batas maksimal melapor adalah setengah jam setelah motor tersebut hilang sebelum tersangka melarikan diri lebih jauh.

Maka dari itu pemilik rental sangat memerlukan sebuah teknologi kotak hitam dan *transponder* yang terpasang pada sepeda motor. Kotak hitam ini akan secara otomatis menyimpan *history* terakhir dari sepeda motor seperti menyimpan data tekanan pengereman, kecepatan dan temperatur mesin sepeda motor dan transponder untuk memberikan posisi secara kontinyu dengan sms melalui *handphone* pemilik persewaan sepeda motor tersebut. Selain sangat bermanfaat untuk pemilik persewaan sepeda motor, kotak hitam dan *transponder* ini juga dapat membantu pihak kepolisian dalam proses identifikasi kecelakaan lalu lintas dan pencurian.

1.2 Permasalahan

Dalam tugas akhir ini permasalahan yang terjadi adalah apakah sudah ada sebuah alat yang bisa memberikan posisi terakhir, mencatat *history* dari tekanan pengereman, kecepatan dan temperatur mesin terakhir dari sebuah sepeda motor secara *real time*? Hal ini yang mendorong agar kotak hitam dapat dipasang pada sebuah sepeda motor. Kotak hitam ini merupakan sebuah inovasi baru dari pengembangan kotak hitam dan transponder yang ada pada pesawat terbang, namun dipasang pada sepeda motor. Ini akan membuat sepeda motor akan selalu mengirim letak posisinya dan memiliki data atau *history* terakhir setelah dipergunakan. Dalam kasus persewaan sepeda motor, motor mengalami kecelakaan dan kehilangan yang disengaja oleh penyewa sepeda motor, namun seringkali proses identifikasi korban kecelakaan lalu lintas yang dilakukan oleh pihak kepolisian kurang akurat, karena hanya dengan mengandalkan keterangan saksi, keterangan para korban yang dilihat dari proses wawancara dan keadaan psikis pasca kecelakaan, melihat keadaan tempat kejadian perkara, periksa dokumen, pemeriksaan keadaan kendaraan terakhir tanpa mengetahui data data terakhir kendaraan yang terlibat kecelakaan, seperti tekanan pengereman, kecepatan dan panas mesin.

Untuk kasus kehilangan sepeda motor, pemilik sepeda motor tidak bisa langsung mengetahui posisi dari sepeda motor yang sedang disewa tersebut. Jadi bagaimana membuat sebuah kotak hitam dan transponder yang mudah terpasang pada sepeda motor dan sensor bisa terhubung pada bagian-bagian penting dari sepeda motor. Selain itu kemampuan untuk memberikan letak posisi sepeda motor sangat diperlukan untuk mendukung kinerja dari kotak hitam. Sensor yang dipasang antara lain sensor tekanan pengereman, sensor kecepatan agar kecepatan putaran roda dapat diketahui dan sensor temperatur pada bagian mesin dengan bantuan media penyimpanan berupa *secure digital card*. Untuk proses pengiriman posisi, menggunakan *GPS* dengan modem *GSM* melalui media *short message service* yang akan mengirim sms setiap satu menit sekali. Kotak hitam atau *black box* ini akan secara otomatis menjadi *history* terakhir dari sebuah sepeda motor.

1.3 Batasan Masalah

Batasan masalah yang akan diselesaikan pada Tugas Akhir ini dibatasi pada pembuatan sebuah prototipe kotak hitam dan *transponder* dengan fungsi *Arduino Mega 2560* dan *Ethernet Shield* menyimpan data dari *Load Cell*, *Rotary Encoder* dan *Thermocouple* dengan media *Secure Digital Card* secara *real time* dengan *RTC*, dan memberikan posisi *GPS* melalui *modem GSM*.

1.4 Tujuan

Adapun beberapa tujuan dari pembuatan alat pada tugas akhir ini antara lain adalah sebagai berikut :

1. Membuat prototipe kotak hitam / *black box* dan *transponder* yang bisa dipasang pada sepeda motor bebek
2. Menyimpan data sensor dengan *Arduino Mega 2560* dan *Ethernet Shield* dengan media *Secure Digital Card* dan menampilkannya pada *microsoft excel* dengan bantuan sebuah laptop
3. Memberikan posisi sepeda motor menggunakan *GPS* dengan *modem GSM* melalui media *short message service*

1.5 Manfaat

Kotak hitam dan *transponder* yang dipasang pada sepeda motor ini dapat memberikan posisi sepeda motor secara kontinyu dan memudahkan dalam melihat data terakhir dari sepeda motor.

Data terakhir ini berupa tekanan rem, kecepatan putaran roda sepeda motor dan temperatur mesin yang nantinya akan disimpan pada *secure digital card* dan dibaca dengan menggunakan laptop, selain itu kemampuan untuk memberikan posisi setiap satu menit sekali melalui sms adalah untuk memantau letak sepeda motor ketika sedang dipergunakan oleh orang lain.

1.6 Sistematika Laporan

Sistematika pembahasan laporan ini ada lima bab, yaitu pendahuluan, teori penunjang, perencanaan dan pembuatan alat, pengujian dan analisa alat, serta penutup. Dalam penyusunan buku Tugas Akhir ini, pembahasan dibagi menjadi lima bab dengan sistematika sebagai berikut :

1. Bab I : PENDAHULUAN

Dalam bab ini mendiskripsikan tentang latar belakang, permasalahan, batasan masalah, tujuan, ruang lingkup, sistematika laporan, serta relevansi penulisan Tugas Akhir ini.

2. BAB II : TEORI PENUNJANG

Dalam bab ini berisi penjelasan dasar teori dalam perancangan Tugas Akhir ini, meliputi pembahasan tentang *Arduino Mega 2560 dan Ethernet Shield, Loadcell, Rotary Encoder, Thermocouple, RTC, Data Logger, Secure Digital Card, GPS U-Blox Neo 6M, Modem GSM Fastrack M1306B*

3. BAB III : PERANCANGAN DAN PEMBUATAN ALAT

Dalam bab ini akan dibahas secara detail tentang diagram fungsional alat. Tahapan perancangan dan pembuatan alat merupakan tahap yang dilakukan setelah melakukan pencarian literatur dari berbagai sumber. Kegiatan ini termasuk kegiatan yang memakan waktu banyak karena pada tahap ini melakukan analisa awal

(perhitungan-perhitungan), pembuatan rangkaian, uji coba pemasangan sensor, memilih komponen yang sesuai, menyimpan data dengan *secure digital card*, setting gps dengan modem gsm dan perencanaan prototipe kotak hitam yang mana akan di implementasikan ke sesuatu yang nyata.

4. **BAB IV : PENGUJIAN DAN ANALISA DATA**

Dalam bab ini berisi data-data pengukuran dan pengujian beserta analisa terhadap prinsip kerja dari alat yang dibuat meliputi pengujian *hardware*, *software* dan pengujian keseluruhan.

5. **BAB V : PENUTUP**

Dalam bab ini berisi kesimpulan yang diperoleh dari pembuatan Tugas Akhir ini dan saran-saran untuk pengembangan lebih lanjut.

1.7 **Relevansi**

Kotak hitam dan *transponder* dari pembuatan alat tugas akhir ini diharapkan dapat memberikan manfaat yaitu membantu pemilik persewaan sepeda motor untuk selalu memberikan letak posisi sepeda motor dan memberikan *history* terakhir sepeda motor jika terjadi kecelakaan saat sepeda motor sedang disewa, sehingga ini juga dapat memudahkan pihak kepolisian dalam proses identifikasi kecelakaan sepeda motor.

Kotak hitam yang dipasang pada sepeda motor ini memudahkan dalam melihat data terakhir / *history* dari sepeda motor. Data terakhir ini berupa tekanan rem, kecepatan putaran roda sepeda motor dan temperatur mesin yang nantinya data akan disimpan pada *secure digital card* dan dibaca dengan menggunakan laptop. Ketika sepeda motor dicuri atau hilang, kita masih bisa mencarinya, karena dilengkapi dengan *GPS* yang akan setiap 1 menit sekali mengirim posisi terakhir dari sepeda motor. Meskipun kotak hitam ini masih berupa prototipe, namun jika dipasang dan diimplementasikan dengan baik, ini bisa menjadi salah satu cara untuk identifikasi proses kecelakaan kendaraan bermotor dengan penyelidikan langsung pada data yang tersimpan pada sepeda motor yang tersimpan pada kotak hitam ini.

Halaman ini sengaja dikosongkan

BAB II TEORI PENUNJANG

Pada bab ini menjelaskan mengenai teori – teori yang berkaitan dengan topik penelitian yang dilakukan. Teori yang mendukung penyelesaian Tugas Akhir ini diantaranya adalah mengenai *Arduino Mega 2560*, *Arduino Ethernet Shield*, *Sensor (Thermocouple, Load Cell, Rotary Encoder)*, *RTC (Real Time Clock)*, *Data Logger*, *Secure Digital Card*, *GPS U-Blox Neo 6M*, *Modem GSM Fastrack M1306B*. Penjelasan ini bertujuan untuk memberikan gambaran secara umum terhadap sistem yang dibuat dan berguna sebagai penunjang dalam pengembangan.

2.1 *Arduino*

Arduino merupakan sebuah mikrokontroler *single-board* yang bersifat *open-source*. *Arduino* dirancang sedemikian rupa sehingga memudahkan para penggunanya di bidang elektronika. *Board Arduino* didesain menggunakan *processor Atmel AVR* dan mendukung *input/output* pada *board*-nya. *Software* untuk *arduino* terdiri dari *compiler* bahasa pemrograman *standart* dan *boot-loader*. Bahasa pemrograman yang digunakan adalah C. Dalam mikrokontroler *Arduino* dapat pula ditanamkan berbagai macam *library* maupun metode selama kapasitas memori dari sebuah mikrokontroler mencukupi.

Arduino juga menggunakan *Integrated Development Environment (IDE)* berbasis *processing*, dimana *processing* adalah bahasa *open-source* untuk menuliskan program ke komputer lainnya. Jika ada sebuah *project* yang memerlukan beberapa komputer untuk berkomunikasi dengan *arduino*, maka *processing* tersebut dapat digunakan, sehingga komputer-komputer tersebut dapat saling berkomunikasi dengan *arduino*. Supaya mikrokontroler *Arduino* dapat berfungsi, *Arduino* juga dapat dipasangkan dengan berbagai macam sensor dan *actuator*. Salah satu contoh sensor yang dapat dipasangkan adalah sensor suhu *Thermocouple*. *Thermocouple* dapat mendeteksi nilai suhu. Sedangkan contoh untuk *actuator* adalah *Ethernet Shield*. *Ethernet Shield* adalah sebuah shield yang berfungsi untuk melakukan koneksi dengan internet menggunakan *LAN* dan itu salah satunya masih banyak lagi.

2.1.1 *Arduino Mega 2560*

Arduino Mega 2560 sangat diperlukan sebagai komponen utama atau otak dari semua proses kerja yang akan dilakukan oleh sensor dan perangkat yang lainnya. *Arduino Mega* adalah *board Arduino* terbaru dan merupakan mikrokontroler yang berdasarkan pada *ATmega 2560 (datasheet)*. Mempunyai 54 *pin digital input/output* (14 diantaranya dapat digunakan sebagai *PWM output*), 16 *pin input analog*, 4 *UART (hardware serial ports)*, sebuah *crystal osilator 16 MHz*, koneksi *USB*, sebuah *dc jack power*, sebuah *header ICSP* dan tombol *reset*. *Arduino Mega ADK* ini dibuat berdasarkan *Arduino Mega 2560*. Gambar dan spesifikasi *Arduino Mega ADK 2560* bisa dilihat pada Gambar 2.1.



Gambar 2.1 *Arduino Mega ADK 2560*

Spesifikasi *Arduino Mega ADK 2560* :

<i>Microcontroller</i>	: <i>Atmega 2560</i>
<i>Operating Voltage</i>	: 5V
<i>Input Voltage (recom)</i>	: 7-12V
<i>Input Voltage (limits)</i>	: 5,5-16V
<i>Digital I/O Pins</i>	: 54 (<i>of which 14 provide PWM output</i>)
<i>Analog Input Pins</i>	: 16
<i>DC Current per I/O Pin</i>	: 40 mA
<i>DC Current for 3.3V Pin</i>	: 50 mA
<i>Flash Memory</i>	: 256 KB <i>of which 8 KB used by bootloader</i>
<i>SRAM</i>	: 8 KB
<i>EEPROM</i>	: 4 KB
<i>Clock Speed</i>	: 16 MHz

2.1.2 *Arduino Ethernet Shield*

Arduino Shield Ethernet W5100 ini adalah modul yang dapat dipasangkan langsung di atas papan *Arduino* untuk menambahkan fungsi *LAN / Ethernet* dalam proyek rangkaian elektronika Anda yang menggunakan papan pengembang *Arduino*. Dengan menggunakan modul komunikasi ini *Arduino* anda dapat dihubungkan dengan perangkat lain yang mendukung protokol *TCP/IP* atau *UDP*. Gambar dan spesifikasi *Arduino Ethernet Shield* bisa dilihat pada Gambar 2.2.



Gambar 2.2 *Arduino Ethernet Shield*

Spesifikasi *Arduino Ethernet Shield* :

1. *Requires an Arduino board (not included)*
2. *Operating voltage 5V (supplied from the Arduino Board)*
3. *Ethernet Controller: W5100 with internal 16K buffer*
4. *Connection speed: 10/100Mb*
5. *Connection with Arduino on SPI port*

2.2. *Sensor Load Cell*

Load Cell adalah transduser yang digunakan untuk mengubah tekanan menjadi sinyal elektrik. Biasanya alat ini terdiri dari empat *strain gauges* dalam *wheat stone bridge configuration* tetapi ada juga yang yang terdiri dari satu atau dua *strain gauges*. *Strain gauges* telah digunakan luas dalam pengukuran besar fisika gaya dalam bidang mekanikal, perkapalan atau pelayaran, penerbangan, teknik

sipil, *automotive*, dunia kedokteran, dan lainnya. Gambar 2.3 dibawah ini fisik dari sensor *Loadcell*.



Gambar 2.3 Sensor *Load Cell*

Load Cell mempunyai 2 *strain gauge* yang dilem pada batang *loadcell*. Pada *strain gauge* dalam operasinya memanfaatkan perubahan resistensi sehingga dapat digunakan untuk mengukur perpindahan yang sangat kecil akibat pembengkokan (*tensile stress*) atau peregangan (*tensile strain*). Definisi elastisitas (ϵ) *strain gauge* adalah perbandingan perubahan panjang (ΔL) terhadap panjang semula (L) yaitu :

$$\epsilon = \frac{\Delta L}{L}$$

Strain atau regangan diukur untuk mengetahui besarnya deformasi pada saat terjadinya tegangan mekanik sehingga didapat besaran gaya yang terjadi seperti beban ataupun tegangan. Selain itu juga digunakan untuk memperoleh nilai keamanan atau kekuatan suatu bahan atau suatu elemen struktural yang mengandung bahan tersebut. Namun yang paling baik digunakan untuk pengukuran tegangan material ialah strain gauges karena karakter dan kemampuan ukurnya yang jauh mengungguli yang lain. Selain itu strain gauges sangat mudah penggunaannya dan memberikan stabilitas dan akurasi yang tinggi.

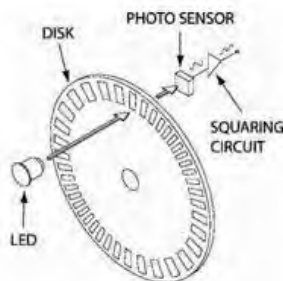
2.3. Rotary Encoder

Rotary Encoder digunakan untuk mengubah gerakan linier atau putaran menjadi sinyal digital, dimana sensor putaran memonitor gerakan putaran dari suatu alat. Sensor ini biasanya terdiri dari dua lapis jenis penyandi yaitu : Pertama, Penyandi

Rotary tambahan (yang menstransmisikan jumlah tertentu dari pulsa untuk masing-masing putaran) yang akan membangkitkan gelombang kotak pada objek yang diputar. Kedua, panyandi absolute yang memperlangkapi kode *binary* tertentu untuk masing-masing posisi sudut). Dua buah sensor *optis* (*chanel A* dan *chanel B*) pendeteksi “Hitam dan Putih” digunakan sebagai acuan untuk menentukan arah gerakan. Searah jarum jam (*clock-wise*) CW atau berlawanan arah jarum jam (*counter clock-wise*) CCW, sedangkan jumlah pulsa (baik A atau B) dapat dihitung (menggunakan prinsip *counter*) sebagai banyak langkah yang ditempuh .

Dengan demikian arah gerakan dan posisi dapat dideteksi dengan baik. Biasanya *encoder* dipasang segaris dengan poros (*shaft*) motor, *gearbox*, sendi atau bagian berputar lainnya. Beberapa tipe *encoder* memiliki poros berlubang (*hollow shaft encoder*) yang didesain untuk *system* sambungan langsung ke poros objek dideteksi. Rangkaian penghasil pulsa yang digunakan umumnya memiliki *output* yang berubah dari +5 volt menjadi +0,5 volt ketika cahaya diblok oleh piringan dan ketika diteruskan ke *photo-transistor*.

Karena *device* ini umumnya bekerja dekat dengan motor DC maka banyak *noise* yang timbul sehingga biasanya *output* akan dimasukan ke *low-pass filter* terlebih dahulu. Apabila *low-pass filter* digunakan frekuensi *cutt-off* yang dipakai umumnya ditentukan oleh jumlah *slot* yang ada pada piringan dan seberapa cepat piringan tersebut berputar. Gambar 2.4 dibawah ini adalah piringan pembaca pulsa *rotary encoder*.



Gambar 2.4 Sensor *Rotary Encoder*

2.4. **Sensor *Thermocouple***

Thermocouple adalah dua logam yang didekatkan yang apabila terpapar oleh kalor dengan suhu tertentu akan menghasilkan beda potensial. *Thermocouple* Suhu didefinisikan sebagai jumlah dari energi panas dari sebuah objek atau sistem. Perubahan suhu dapat memberikan pengaruh yang cukup signifikan terhadap proses ataupun material pada tingkatan molekul. Sensor suhu adalah *device* yang dapat melakukan deteksi pada perubahan suhu berdasarkan pada parameter-parameter fisik seperti hambatan, ataupun perubahan *voltage*. Salah satu jenis sensor suhu yang banyak digunakan sebagai sensor suhu pada suhu tinggi adalah *Thermocouple* seperti pada Gambar 2.5 dibawah ini :



Gambar 2.5 Sensor *Thermocouple* Type K

Thermocouple merupakan jenis logam yang berbeda disatukan salah satu ujungnya dan ujung tersebut dipanaskan maka akan timbul beda potensial pada ujung-ujung yang lain, hal ini diakibatkan oleh kecepatan gerak elektron dari dua material yang berbeda daya hantar panas sehingga mengakibatkan beda potensial. Dalam perancangan serta penggolongan dari *Thermocouple* sendiri sudah diatur oleh *Instrument Society of America (ISA)*. *Thermocouple* dibangun berdasarkan Asas *Seebeck* dimana bila dua jenis logam yang berlainan disambungkan ini akan menjadi rangkaian tertutup sehingga perbedaan *temperature* pada sambungan akan menimbulkan beda potensial listrik pada kedua logam tersebut, selanjutnya akan dibaca oleh alat ukur temperatur.

Thermocouple dapat dihubungkan secara seri satu sama lain untuk membuat termopile, dimana tiap sambungan yang panas diarahkan ke suhu yang lebih tinggi dan semua sambungan dingin ke suhu yang lebih rendah. Dengan begitu, tegangan pada setiap *thermocouple* menjadi naik, yang memungkinkan untuk digunakan pada tegangan yang lebih tinggi.

Dengan adanya suhu tetapan pada sambungan dingin, yang berguna untuk pengukuran di laboratorium, secara sederhana termokopel tidak mudah dipakai untuk kebanyakan indikasi sambungan langsung dan instrumen kontrol. Mereka menambahkan sambungan dingin tiruan ke sirkuit mereka yaitu peralatan lain yang sensitif terhadap suhu (seperti *Thermistor* atau *dioda*) untuk mengukur suhu sambungan input pada peralatan, dengan tujuan untuk mengurangi gradiasi suhu di antara ujung-ujungnya. Di sini, tegangan yang berasal dari hubungan dingin yang diketahui dapat disimulasikan, dan koreksi yang baik dapat diaplikasikan. Tersedia beberapa jenis *Thermocouple*, tergantung aplikasi penggunaannya, yaitu :

1. Tipe K (*Chromel (Ni-Cr alloy) / Alumel (Ni-Al alloy)*)
Termokopel untuk tujuan umum. Lebih murah. Tersedia untuk rentang suhu $-200\text{ }^{\circ}\text{C}$ hingga $+1200\text{ }^{\circ}\text{C}$
2. Tipe E (*Chromel / Constantan (Cu-Ni alloy)*)
Tipe E memiliki *output* yang besar ($68\text{ }\mu\text{V}/^{\circ}\text{C}$) membuatnya cocok digunakan pada temperatur rendah, tipe E adalah tipe non magnetik
3. Tipe J (*Iron / Constantan*)
Rentangnya terbatas (-40 hingga $+750\text{ }^{\circ}\text{C}$) membuatnya kurang populer dibanding tipe K.
4. Tipe J memiliki sensitivitas sekitar $\sim 52\text{ }\mu\text{V}/^{\circ}\text{C}$
5. Tipe N (*Nicrosil (Ni-Cr-Si alloy) / Nisil (Ni-Si alloy)*). Stabil dan tahanan yang tinggi terhadap oksidasi membuat tipe N cocok untuk pengukuran suhu yang tinggi tanpa platinum. Dapat mengukur suhu di atas $1200\text{ }^{\circ}\text{C}$. Sensitifitasnya sekitar $39\text{ }\mu\text{V}/^{\circ}\text{C}$ pada 900°C , sedikit di bawah tipe K. Tipe N merupakan perbaikan tipe K
6. Termokopel tipe B, R, dan S adalah termokopel logam mulia yang memiliki karakteristik yang hampir sama. Mereka adalah termokopel yang paling stabil, tetapi karena sensitifitasnya rendah (sekitar $10\text{ }\mu\text{V}/^{\circ}\text{C}$) mereka biasanya

hanya digunakan untuk mengukur temperatur tinggi (>300 °C).

7. Tipe B (*Platinum-Rhodium/Pt-Rh*) dapat mengukur suhu di atas 1800 °C. Tipe B memberi *output* yang sama pada suhu 0°C hingga 42°C sehingga tidak dapat dipakai di bawah suhu 50°C.
8. Tipe R (*Platinum /Platinum with 7% Rhodium*) dapat mengukur suhu di atas 1600 °C. sensitivitas rendah (10 $\mu\text{V}/^\circ\text{C}$) dan biaya tinggi membuat mereka tidak cocok dipakai untuk tujuan umum.
9. Tipe S (*Platinum /Platinum with 10% Rhodium*) dapat mengukur suhu di atas 1600 °C. Sensitivitas rendah (10 $\mu\text{V}/^\circ\text{C}$) dan biaya tinggi membuat mereka tidak cocok dipakai untuk tujuan umum. Karena stabilitasnya yang tinggi Tipe S digunakan untuk standar pengukuran titik leleh emas (1064.43 °C).
10. Tipe T (*Copper / Constantan*) Cocok untuk pengukuran antara -200 hingga 350 °C. Konduktor *positif* terbuat dari tembaga, dan yang *negatif* terbuat dari konstantan. Sering dipakai sebagai alat pengukur alternatif sejak penelitian kawat tembaga. Tipe T memiliki sensitifitas $\sim 43 \mu\text{V}/^\circ\text{C}$.

2.5. RTC (*Real Time Clock*)

Real Time Clock merupakan suatu *chip* (IC) yang memiliki fungsi sebagai penyimpan waktu dan tanggal. RTC DS1307 merupakan *Real-time clock* (RTC) yang dapat menyimpan data-data detik, menit, jam, tanggal, bulan, hari dalam seminggu, dan tahun *valid* hingga 2100. *56-byte, battery-backed, RAM nonvolatile* (NV) RAM untuk penyimpanan. RTC DS1307 merupakan *Real-time clock* (RTC) dengan jalur data *parallel* yang memiliki Antarmuka serial *Two-wire* (I2C), Sinyal luaran gelombang-kotak terprogram (*Programmable squarewave*), Deteksi otomatis kegagalan-harga (*power-fail*) dan rangkaian *switch*, Konsumsi daya kurang dari 500nA menggunakan mode baterai cadangan dengan operasional *osilator*.

Tersedia fitur industri dengan ketahanan suhu: -40°C hingga $+85^{\circ}\text{C}$. Tersedia dalam kemasan *8-pin DIP* atau *SOIC*. Gambar 2.6 dibawah ini adalah RTC DS1307.



Gambar 2.6 RTC DS1307

Dari penjelasan spesifikasi RTC DS 1307 diatas, berikut penjelasan Pin-Pin Pada IC RTC DS1307 :

1. X1
Merupakan pin yang digunakan untuk dihubungkan dengan kristal sebagai pembangkit clock.
2. X2
Berfungsi sebagai keluaran / *output* dari *crystal* yang digunakan. Terhubung juga dengan X1.
3. VBAT
Merupakan *backup supply* untuk RTC DS1307 dalam menjalankan fungsi waktu dan tanggal. Besarnya adalah 3V dengan menggunakan jenis *Lithium Cell* atau sumber *energy* lain. Jika pin ini tidak di gunakan maka harus terhubung dengan *Ground*. Sumber tegangan dengan 48mAH atau lebih besar dapat digunakan sebagai cadangan *energy* sampai lebih dar 10 tahun, namun dengan persyaratan untuk pengoprasian dalam suhu 25°C .
4. GND
Berfungsi sebagai *Ground*.
5. SDA
Berfungsi sebagai masukan / keluaran (*I/O*) untuk I2C serial interface. Pin ini bersifat *open drain*, oleh sebab itu membutuhkan eksternal *pull up resistor*.

6. SCL
Berfungsi sebagai *clock* untuk input ke I2C dan digunakan untuk mensinkronisasi pergerakan data dalam serial *interface*. bersifat *open drain*, oleh sebab itu membutuhkan eksternal *pull up resistor*.
7. SWQ/OUT
Sebagai *square wave / Output Driver* . jika di aktifkan, maka akan menjadi 4 frekuensi gelombang kotak yaitu 1 Hz, 4kHz, 8kHz, 32kHz sifat dari pin ini sama dengan sifat pin SDA dan SCL sehingga membutuhkan eksternal *pull up resistor*. Dapat dioperasikan dengan VCC maupun dengan VBAT.
8. VCC
Merupakan sumber tegangan utama. Jika sumber tegangan terhubung dengan baik, maka pengaksesan data dan pembacaan data dapat dilakukan dengan baik. Namun jika backup *supply* terhubung juga dengan VCC, namun besar VCC di bawah VTP, maka pengaksesan data tidak dapat dilakukan. Beberapa aplikasi yang dapat digunakan dari IC ini yaitu sebuah jam digital. Berikut contoh pemrograman mikrokontroler atmega8535 untuk membuat jam digital menggunakan IC RTC DS1307 dengan tampilan menggunakan LCD 2×16.

Dengan I2C kita hanya membutuhkan dua jalur untuk berkomunikasi antar perangkat. dua jalur tersebut adalah SDA (*Serial Data*) dan SCL (*Serial Clock*). SCL merupakan jalur yang digunakan untuk mensinkronisasi transfer data pada jalur I2C, sedangkan SDA merupakan jalur untuk data. Beberapa fitur dari DS1307 adalah:

1. *Real-time clock* (RTC) menghitung detik, menit, jam, tanggal, bulan, hari, dan tahun *valid* sampai tahun 2100
2. Ram 56-byte, *nonvolatile* untuk menyimpan data.
3. Dua jalur serial *interface* (I2C).
4. *Output* gelombang kotak yang diprogram.
5. *Automatic power-fail detect and switch*
6. Konsumsi arus hanya 500nA pada *bateri* internal.
7. Mode dengan *oscillator running*.
8. *Temperature range*: -40°C sampai +85°C

2.6 Data Logging

Logging data (data logging) adalah proses otomatis pengumpulan dan perekaman data dari sensor untuk tujuan pengarsipan atau tujuan analisis. Sensor digunakan untuk mengkonversi besaran fisik menjadi sinyal listrik yang dapat diukur secara otomatis dan akhirnya dikirimkan ke komputer atau mikroprosesor untuk pengolahan. Berbagai macam sensor sekarang tersedia. Sebagai contoh, suhu, intensitas cahaya, tingkat suara, sudut rotasi, posisi, kelembaban relatif, pH, oksigen terlarut, pulsa (detak jantung), bernapas, kecepatan angin, dan gerak. Selain itu, banyak peralatan laboratorium dengan output listrik dapat digunakan bersama dengan konektor yang sesuai dengan *data logger*.

Data logger (perekam data) adalah sebuah alat elektronik yang mencatat data dari waktu ke waktu baik yang terintegrasi dengan sensor dan instrumen didalamnya maupun eksternal sensor dan instrumen. Atau secara singkat *data logger* adalah alat untuk melakukan *data logging*. Biasanya ukuran fisiknya kecil, bertenaga baterai, *portabel*, dan dilengkapi dengan mikroprosesor, memori internal untuk menyimpan data dan sensor. Beberapa *data logger* diantarmukakan dengan komputer dan menggunakan *software* untuk mengaktifkan *data logger* dan melihat dan menganalisa data yang terkumpul, sementara yang lain memiliki peralatan antarmuka sendiri (*keypad* dan *LCD*) dan dapat digunakan sebagai perangkat yang berdiri sendiri (*Stand-alone device*). Datalogger menggunakan *secure digital card (SD Card)* bisa dilihat pada Gambar 2.7 dibawah.



Gambar 2.7 Macam Macam *Data Logger*

Salah satu keuntungan menggunakan *data logger* adalah kemampuannya secara otomatis mengumpulkan data setiap 24 jam. Setelah diaktifkan, *data logger* digunakan dan ditinggalkan untuk mengukur dan merekam informasi selama periode pemantauan. Hal ini memungkinkan untuk mendapatkan gambaran yang komprehensif tentang kondisi lingkungan yang dipantau, contohnya seperti suhu udara dan kelembaban relatif.

Data logger biasanya dilengkapi dengan *Real-Time Clock* (RTC) didalamnya sehingga dapat ditampilkan waktu dan tanggal sampling untuk memastikan bahwa setiap data yang dicatat sesuai dengan tanggal dan waktu akuisisi. *Data logger* bervariasi macamnya dari *single-channel input* sederhana hingga instrumen *multi-channel* yang kompleks. Biasanya, perangkat yang lebih sederhana fleksibilitas pemrogramannya juga kurang. Beberapa instrumen yang lebih canggih memungkinkan untuk perhitungan *cross-channel* dan alarm berdasarkan kondisi yang telah ditentukan.

Data logger yang terbaru dapat melayani halaman *web*, yang memungkinkan banyak orang untuk memantau sistem jarak jauh. Sifat tanpa pengawasan dari aplikasi *data logger* banyak menyiratkan kebutuhan dalam beberapa aplikasi untuk beroperasi menggunakan sumber listrik DC, seperti baterai. Tenaga surya dapat digunakan untuk melengkapi sumber-sumber daya. Hal ini membuat produsen harus dapat memastikan bahwa *data logger* yang dipasarkan sangat hemat daya dibandingkan dengan jika menggunakan komputer.

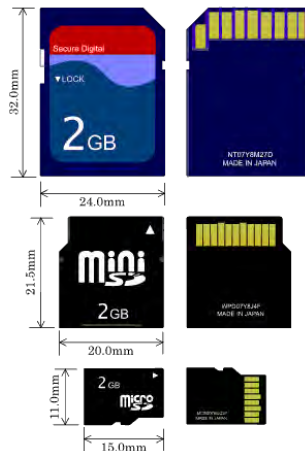
Dalam banyak kasus *data logger* diharuskan untuk bekerja dalam kondisi lingkungan yang ekstrim dimana komputer tidak akan berfungsi sehandal itu. Sifat tanpa pengawasan juga menyatakan bahwa *data logger* harus sangat dapat diandalkan. Karena *data logger* dapat beroperasi untuk waktu yang lama tanpa henti dengan sedikit atau tanpa pengawasan manusia, dan dapat dipasang di lokasi kasar atau terpencil, sangat penting bahwa selama *data logger* memiliki daya, alat tersebut tidak akan gagal untuk mengumpulkan data dalam situasi apapun.

2.7 *Secure Digital Card*

Secure Digital (SD) adalah sebuah format kartu memori flash. Dia digunakan dalam alat portabel, seperti PDA, kamera digital dan telepon genggam. Kartu SD dikembangkan oleh

SanDisk, Toshiba, dan Panasonic berdasarkan Kartu Multi Media (MMC) yang sudah lebih dulu ada. Selain memiliki sistem pengaman yang lebih bagus daripada MMC, SD Card juga bisa dengan mudah dibedakan dari MMC karena memiliki ukuran yang lebih tebal dibanding kartu MMC standar. Kartu SD standar memiliki ukuran 32 mm x 24 mm x 2,1 mm, tetapi ada beberapa kartu SD yg setipis MMC (1.4 mm).

Dalam perkembangannya, kartu SD diproduksi juga dalam dua variasi ukuran yg lebih kecil, kedua varian tersebut dikenal dengan nama *MiniSD* dan *MicroSD* atau *TransFlash (T-Flash)*. Gambar 2.8 adalah macam macam jenis dari *Secure Digital Card*.



Gambar 2.8 Macam macam SD Card

Secara umum, kartu SD dibedakan dari kecepatan transfer data yang tersedia, yaitu kecepatan biasa (150 KB/s) dan kecepatan tinggi. Beberapa kamera digital memerlukan kartu yg berkecepatan-tinggi untuk merekam video secara lancar atau menangkap gambar berturut-turut. Alat yang dilengkapi dengan slot SD dapat menggunakan kartu MMC yang lebih tipis, tetapi kartu SD standar tidak dapat digunakan ke *slot* kartu MMC yang lebih tipis. Kartu SD dapat digunakan dalam *slot Compact Flash* atau kartu PC dengan sebuah adapter.

2.8 GPS *U-Blox Neo 6M*

Modul berukuran ringkas ini (25x35mm untuk modul, 25x25mm untuk antena) berfungsi sebagai *penerima GPS (Global Positioning System Receiver)* yang dapat mendeteksi lokasi dengan menangkap dan memroses sinyal dari satelit navigasi. Aplikasi dari modul ini melingkupi sistem navigasi, sistem keamanan terhadap kemalingan pada kendaraan / perangkat bergerak, akuisisi data pada sistem pemetaan medan, penjejak lokasi / *location tracking*, dsb. Modul ini kompatibel dengan APM2 dan APM2.5 dengan EEPROM terpadu yang dapat digunakan untuk menyimpan data konfigurasi. Gambar 2.9 adalah GPS *U-Blox Neo 6M*.



Gambar 2.9 GPS *U-Blox Neo 6M*

Antarmuka menggunakan serial TTL (RX/TX) yang dapat diakses dari mikrokontroler yang memiliki fungsi UART atau emulasi serial TTL (pada *Arduino* dapat menggunakan pustaka komunikasi serial / *serial communication library* yang sudah tersedia dalam paket *Arduino IDE*). *Baud rate* diset secara *default* di 9600 bps. *GPS Processor* dari modul ini menggunakan u-blox NEO-6 GPS Module dengan mesin penjejak posisi yang berkinerja tinggi dengan versi ROM terbaru (ROM7.03). Modul ini dapat memproses hingga 50 kanal sinyal secara cepat dengan waktu *Cold TTF* (*Cold-Start Time-To-First-Fix*, waktu yang diperlukan untuk menentukan posisi dari kondisi mati total) kurang dari 27 detik (sebagai pembandingan, rata-rata *GPS navigator* yang umum dijual di toko variasi mobil memiliki waktu *Cold TTF* lebih dari 50 detik),

dapat dipercepat dengan fitur pemandu (*aiding*) hingga kurang dari 3 detik. Pada kondisi hot start, waktu TTFF yang dibutuhkan mencapai kurang dari 1 detik. Kinerja tinggi ini dicapai dengan didedikasinya prosesor khusus untuk mengumpulkan data sinyal satelit yang memiliki hingga 2 juta *korelator* yang sanggup memroses data waktu dan *frekuensi* secara masif dengan sangat cepat sehingga mampu menemukan sinyal dari satelit navigasi secara instan. Prosesor ini juga menerapkan teknologi DSP terkini untuk meredam sumber pengacak (*jamming sources*) dan mengurangi secara signifikan efek *interferensi* multi-jalur. Sumber tenaga dapat menggunakan catu daya antara 3 Volt hingga 5 Volt, ideal untuk digunakan pada berbagai development board mulai dari aneka macam *Arduino Board*, *Raspberry Pi*, dan lain sebagainya. Spesifikasi Teknis *U-blox NEO-6M* adalah sebagai berikut :

1. Tipe penerima: 50 kanal, GPS L1 frekuensi, C/A Code. SBAS: WAAS, EGNOS, MSAS
2. Sensitivitas penjejak & navigasi: -161 dBm (reakuisisi dari *blank-spot*: -160 dBm)
3. Sensitivitas saat baru memulai: -147 dBm pada cold-start, -156 dBm pada hot start
4. Kecepatan pembaharuan data / *navigation update rate*: 5 Hz
5. Akurasi penetapan lokasi GPS secara horisontal: 2,5 meter (SBAS = 2m)
6. Rentang frekuensi pulsa waktu yang dapat disetel: 0,25 Hz hingga 1 kHz
7. Akurasi sinyal pulsa waktu: RMS 30 ns (99% dalam kurang dari 60 ns) dengan granularitas 21 ns atau 15 ns saat terkompensasi
8. Akurasi kecepatan: 0,1 meter / detik
9. Akurasi arah (*heading accuracy*): $0,5^\circ$
10. Batasan operasi: daya tarik maksimum 4x gravitasi, ketinggian maksimum 50 Km, kecepatan maksimum 500 meter / detik (1800 km/jam). *red*: dengan limit seperti ini, modul ini bahkan dapat digunakan di pesawat jet super-cepat sekalipun.

2.9 *Modem GSM Fastrack M1306B*

Pada pembuatan kotak hitam dan *transponder* ini, kami menggunakan *Modem GSM Wavecom Fastrack M1306B*. Gambar 2.10 adalah bentuk fisik *Modem GSM Fastrack M1306B*.



Gambar 2.10 *Modem GSM Fastrack M1306B*

Beberapa keunggulan *Modem Wavecom Fastrack* dibandingkan dengan *Modem GSM/HP*:

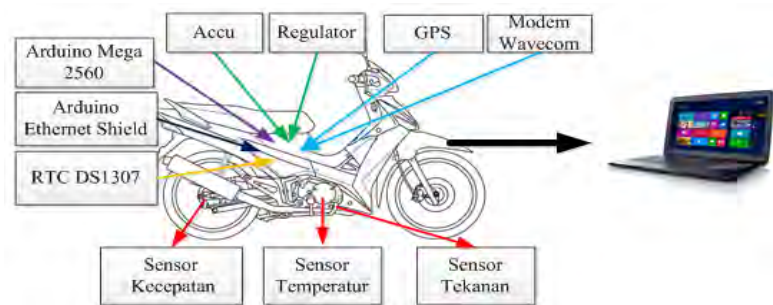
1. *Wavecom* jauh lebih stabil dibanding *Modem GSM/HP*
2. *Wavecom* tidak gampang panas dibanding *Modem GSM/HP*
3. Pengiriman SMS yang lebih cepat dibanding *Modem GSM/HP* (1000 s/d 1200 SMS per jam)
4. *Support AT Command*, bisa digunakan untuk cek sisa pulsa, cek point, cek pemakaian terakhir dll
5. Tidak semua *Modem GSM/HP* support *AT Command*
6. Tidak memakai baterai sehingga lebih praktis digunakan
7. Dapat langsung dikoneksikan dengan mikrokontroler melalui RS232

BAB III PERANCANGAN DAN PEMBUATAN ALAT

Pada bab ini dibahas mengenai perancangan sistem penyimpanan data sensor untuk kecepatan, tekanan dan temperatur pada kendaraan bermotor menggunakan *Arduino*. Perancangan dan pembuatan alat meliputi perancangan *hardware* dan *software*.

3.1 Perancangan Perangkat Keras

Gambaran dari keseluruhan Kotak Hitam dan Transponder sebagai Pengirim Posisi dan pencatat Pengereman, Kecepatan dan Temperatur Mesin Sepeda Motor dengan Media *Secure Digital Card* yang di desain pada tugas akhir ini terlihat seperti Gambar 3.1 dibawah ini :



Gambar 3.1 Diagram Fungsional Sistem

Gambaran kerja dari sistem secara keseluruhan yaitu sebuah kotak hitam yang merekam tekanan, kecepatan dan temperatur pada kendaraan bermotor menggunakan *Arduino*. Sensor yang digunakan adalah sensor *load cell*, *rotary encoder*, dan *thermocouple* yang bekerja secara *real time*. *Input* sensor yang bekerja dan dipasang pada sepeda motor juga diproses oleh *Arduino Mega 2560* yang kemudian di simpan pada *Arduino Ethernet Shield* dengan *secure digital card* atau *SD card* yang kemudian bisa dibaca pada *personal computer* atau perangkat keras dengan format (.txt). Selain itu ditambah dengan *GPS* untuk memberikan posisi dengan sms setiap 1 menit sekali menggunakan *modem GSM Wavecom*.

Dengan seperti ini, posisi dari sepeda motor dapat diketahui setiap 1 menit sekali selain itu data atau *history* akhir dari sepeda motor bisa diketahui pula dengan membaca *secure digital card* nya Perancangan *hardware* dilakukan dengan merancang rangkaian – rangkaian elektronika dan rancangan mekanik. Perancangan *hardware* ini meliputi :

1. *Arduino Mega ADK 2560*
2. *Arduino Ethernet Shield*
3. Sensor Tekanan *Load Cell*
4. Sensor Kecepatan *Rotary Encoder*
5. Sensor Temperatur *Thermocouple type K*
6. *RTC (Real Time Clock)*
7. *Secure Digital Card*
8. *GPS U-Blox Neo 6M*
9. *Modem GSM Fastrack M1306B*

Sedangkan untuk perancangan *software* yang meliputi pembuatan program menggunakan *software* Arduino ERW 1.0.5

3.1.1 *Arduino Mega 2560 dan Arduino Ethernet Shield*

Arduino Mega 2560 sangat diperlukan sebagai komponen utama atau otak dari semua proses kerja yang akan dilakukan oleh sensor dan perangkat yang lainnya. *Arduino Mega 2560* ini merupakan memiliki *pin digital* dan *pin analog* yang digunakan sebagai *input* dari sensor yang nantinya akan tersimpan datanya pada *secure digital card*. Selain itu untuk mendukung penggunaan yaitu menyimpan data menggunakan *secure digital card*, maka *arduino mega* ditambah *arduino ethernet shield*. Penggunaan masing masing *port I/O Arduino 2560* dan *Ethernet Shield* dalam sistem ini pada Tabel 3.1 berikut berikut :

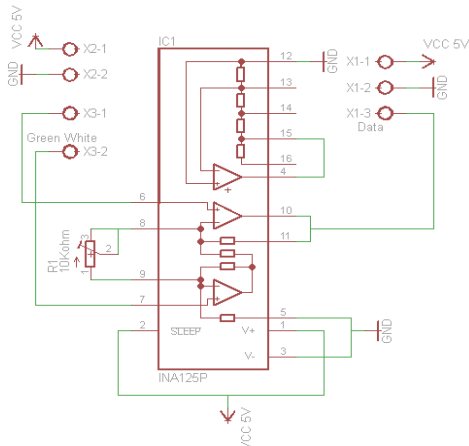
Tabel 3.1 Penggunaan *Port* Arduino

Port	Penggunaan
Analog 0	Input <i>Loadcell</i>
Analog 1	Input <i>Rotary Encoder</i>
Analog 2	Input <i>Themocouple</i>
Rx Tx 1	Komunikasi <i>GPS</i>
Rx Tx 2	Komunikasi <i>Modem GSM</i>

Port	Penggunaan
Digital 3, 5, 6, 7 8, 9	LCD 8x2
Digital 20, 21	Komunikasi SDA dan SCL RTC
Digital 11, 12, 13, 4 50, 51, 52, 53	Konfigurasi <i>Ethernet Shield</i>

3.1.2 Perancangan Rangkaian Sensor Tekanan *Load Cell*

Perancangan sensor tekanan *load cell* menggunakan penguat INA 125 *instrumentation amplifier*. Penggunaan rangkaian ini menggunakan supply 5v yang terhubung pada kabel *load cell* warna merah. Penguatan dari INA 125 ini sendiri adalah dengan mengganti *resistor* sesuai dengan penguatan yang diinginkan. *Datasheet* INA 125 juga memberikan perhitungan manual dan nilai gain masing masing dari *resistor* yang dipilih. Dalam proses pembuatan ini, *resistor variable* juga diperlukan untuk mengatur nilai penguatan yang diinginkan. Gambar 3.2 adalah rangkaian penguat dengan INA 125.

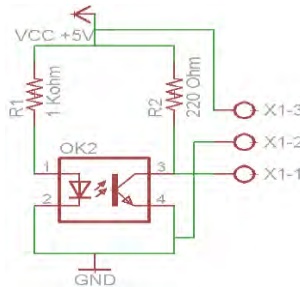


Gambar 3.2 Rangkaian INA 125 *Load Cell*

3.1.3 Perancangan Rangkaian Sensor Kecepatan *Rotary Encoder*

Perancangan sensor menggunakan sebuah *led superbright* dan *fotodiode*. *led superbright* berfungsi sebagai pengirim (*transmitter*) sedangkan *fotodiode* akan menerima cahaya yang dipantulkan oleh cahaya dari sinar *led superbright* yang nantinya akan diperoleh berapa nilai dari cahaya yang dipantulkan oleh *led superbright*.

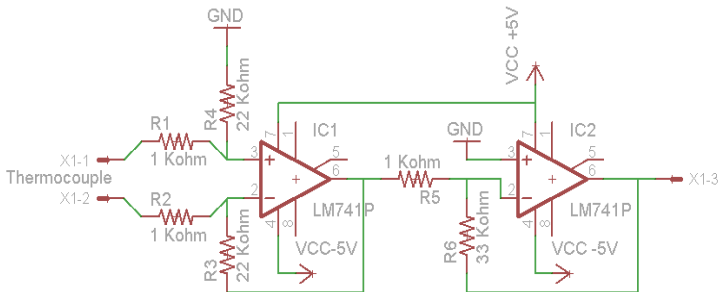
Fotodiode dan *led superbright* sangat berpengaruh terhadap cahaya yang datang dari luar, misalnya dari sinar tampak yaitu matahari serta lampu yang ada diruangan. Oleh sebab itu peletakan posisi sensor perlu diperhatikan agar *error* dari penerimaan oleh *fotodiode* bisa dikurangi. Kemudian dari rangkaian sensor masuk ke arduino port analog (A1). Rangkaian sensor membutuhkan tegangan 5 Volt yang diambil dari arduino. Gambar 3.3 adalah rangkaian pembaca pulsa rotary encoder.



Gambar 3.3 Rangkaian Pulsa LED dan Photodiode

3.1.4 Perancangan Sensor Temperatur *Thermocouple Type K*

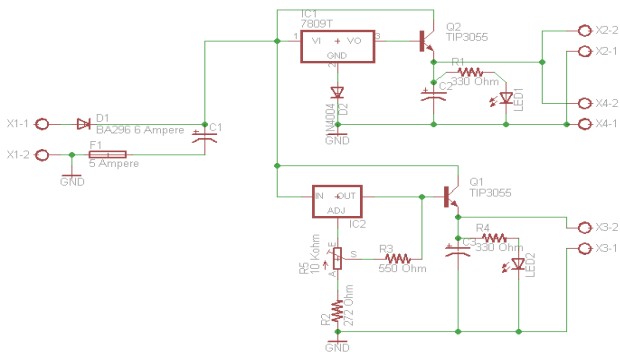
Perancangan sensor temperatur menggunakan penguat LM 741 *instrumentation amplifier*. Penggunaan rangkaian ini menggunakan supply 5v dan -5v. Penguatan dari LM741 ini sendiri adalah dengan mengganti *resistor* sesuai dengan penguatan yang diinginkan. Penguatan sebesar 782 inilah yang akan memberikan tegangan agar terbaca dan bisa dikonversi ke program. Gambar 3.4 adalah rangkaian penguat thermocouple dengan LM 741.



Gambar 3.4 Rangkaian LM 741 Temperature

3.1.5 Regulator

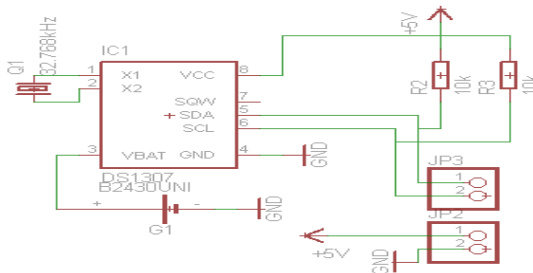
Regulator tegangan ini adalah bagian *power supply* yang berupa accu 12 Volt 3,5 Amp yang berfungsi untuk memberikan stabilitas *output* pada suatu *power supply*. Hasil output tegangan dari regulator dengan IC 7809 dan adjustable adalah berfungsi sebagai catu daya *Arduino Mega 2560* dan *Arduino Ethernet Shield* sebesar +9 Volt, dan catu daya modem GSM sebesar + 7,5Volt. Gambar 3.5 yaitu rangkaian regulator.



Gambar 3.5 Rangkaian Regulator +7,5 Volt dan +9 Volt

3.1.6 Perancangan Rangkaian Real Time Clock (RTC)

Rangkaian RTC diperlukan sebagai acuan bagi sensor yang terpasang pada Arduino untuk bekerja secara *real time*. Rangkaian RTC berfungsi sebagai penyimpan waktu, tanggal, bulan, dan tahun. RTC yang di gunakan adalah RTC DS1307. Gambar 3.6 yaitu rangkaian RTC DS1307.

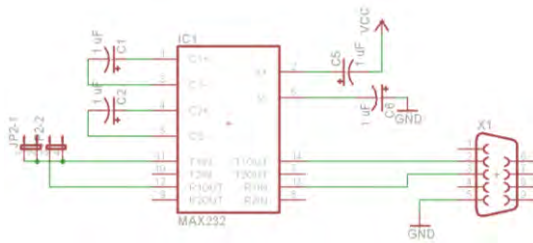


Gambar 3.6 Rangkaian RTC DS1307

Terlihat pada Gambar 3.6, *Port 1* dan *Port 2* dihubungkan ke *crystal*, *Port 3* dipasang ke baterai 3 Volt, *Port 4* adalah *ground*, *Port 8* dihubungkan ke *vcc*, *Port 5* dan *6* berturut – turut dihubungkan ke *Port SDA* dan *SCL* yang terdapat pada pin *Arduino Mega 2560*. Serta di *pull up* ke *vcc*.

3.1.7 Perancangan Rangkaian Max 232

MAX232 merupakan salah satu jenis IC rangkaian antar muka *dual RS-232 transmitter/receiver* yang memenuhi semua spesifikasi standar EIA-232-E. IC MAX232 hanya membutuhkan *power supply 5 Volt (single power supply)* sebagai catu. IC MAX232 di sini berfungsi untuk merubah level tegangan pada COM menjadi level tegangan TTL/CMOS. IC MAX232 terdiri atas tiga bagian yaitu *dual charge-pump voltage converter*, *driver RS232*, dan *receiver RS232*. Gambar 3.7 adalah rangkaian *driver MAX232*.

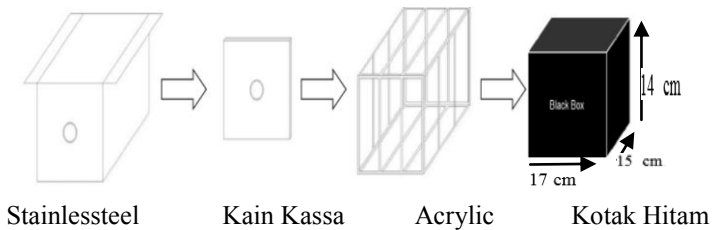


Gambar 3.7 Rangkaian *Driver MAX232*

3.2 Perancangan Desain Mekanik

Pembuatan prototipe kotak hitam yang digunakan untuk sepeda motor ini menggunakan *stainlesssteel* dengan ketebalan 4mm. Ketebalan ini sudah diuji dengan memberikan benturan keras berupa pukulan palu dan dijatuhkan dari ketinggian 2 meter, dan terbukti mampu melindungi komponen dari benturan keras jika ketika dipasang pada sepeda motor terjadi benturan keras sewaktu kecelakaan.

Kemudian dibagian dalam *stainlesssteel* diberi lapisan kain kassa setebal 2mm yang sudah diuji ketahanannya dengan membakar kain yang sudah dipasang selama 15 menit, kain kassa ini dapat melindungi dari panas atau api yang sewaktu waktu bisa terjadi. Selain itu untuk penempatan komponen elektronik utama, diberikan kotak lapisan kedua dengan *acrylic* 3mm. Desain hardware terlihat pada Gambar 3.8.



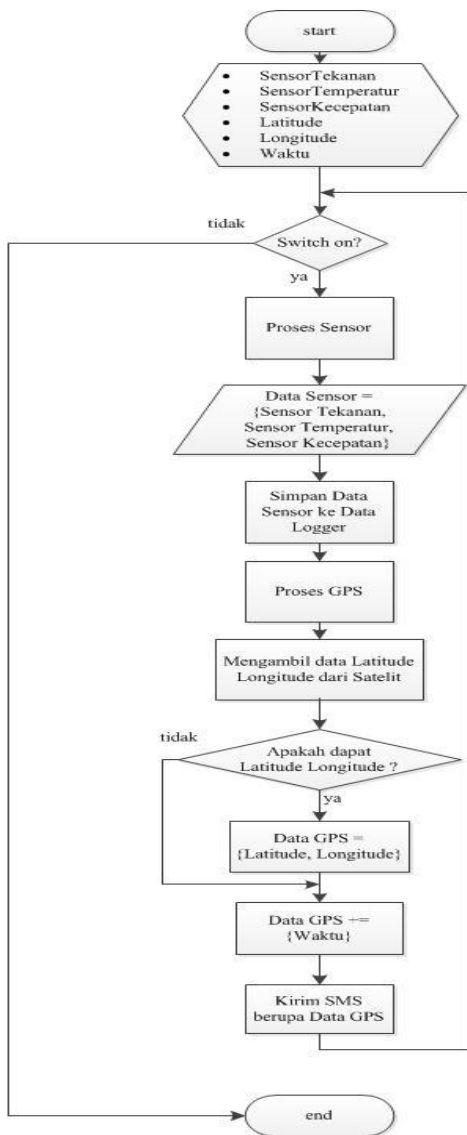
Gambar 3.8 Desain *Hardware* Sistem

Acrylic ini berfungsi sebagai pelindung komponen dari singgungan langsung dengan besi *stainlesssteel* yang bisa menyebabkan *short* atau rusak pada arduino. Fungsi lain *acrylic* ini adalah sebagai penempatan komponen dari kotak hitam yang ditata sedemikian rupa agar memenuhi kerapihan dan keamanan pemasangan komponen elektronik dalam sepeda motor. Kemudian kotak hitam dipasang pada bagian tengah sepeda motor atau bagasi sepeda motor, karena kotak hitam memerlukan *space* tempat yang sesuai agar bisa digunakan dengan baik.

3.3 Perancangan *Software* Sistem

Perangkat lunak yang digunakan adalah *Software Arduino ERW 1.0.5*. Berikut ini akan dibahas mengenai perancangan perangkat lunak (*software*) pada *Arduino* untuk memproses algoritma dengan menggunakan *software arduino*.

Algoritma dari *flowchart* perangkat lunak merupakan alur yang digunakan untuk memprogram sistem kotak hitam dengan *secure digital cardt*. Logika pemrograman menghasilkan suatu program yang nantinya akan diunduh ke dalam *Arduino*. Algoritma perangkat lunak digambarkan pada Gambar 3.9.



Gambar 3.9 Flowchart Sistem Kotak Hitam dan Transponder

Algoritma dari *flowchart* di atas dapat dijelaskan sebagai berikut :

1. Dimulai dengan inisialisasi variabel
2. Dilakukan cek terlebih dahulu terhadap switch on/off, apabila on maka dilakukan proses selanjutnya yaitu proses Sensor dan proses GPS serta pengiriman sms.
3. Jika of maka sistem akan otomatis mati.
4. Untuk Proses Sensor, pengiriman input dari Sensor *Loadcell*, *Rotary Encoder* dan *Thermocouple* yang akan disimpan dalam *Secure Digital Card (Data Logger)*.
5. Data akan tersimpan pada *Data Logger*, jika masukan sensor *Thermocouple* adalah $30^0\text{ C} - 100^0\text{ C}$.
6. Data akan tersimpan pada *Data Logger*, jika masukan sensor *Loadcell* adalah 0 gram – 4898 gram.
7. Data akan tersimpan pada *Data Logger*, jika masukan sensor *Rotary Encoder* adalah 0 km/jam – 120 km/jam.
8. Proses Sensor ini akan disimpan setiap 1 – 2 detik.
9. Untuk Proses GPS, GPS mengambil dan mengirimkan data berupa *Latitude* dan *Longitude*.
10. Apabila GPS mendapatkan data *Latitude* dan *Longitude* maka sistem akan mengirimkan SMS berupa data *Latitude*, *Longitude*, dan Waktu Pengiriman.
11. Sedangkan apabila GPS tidak mendapatkan data *Latitude* dan *Longitude*, maka sistem hanya akan mengirimkan data Waktu pengiriman saja.
12. Proses GPS akan diulangi setiap satu menit sekali. Jadi setiap satu menit sistem akan mengirimkan SMS pada nomor yang dituju.

Halaman ini sengaja dikosongkan

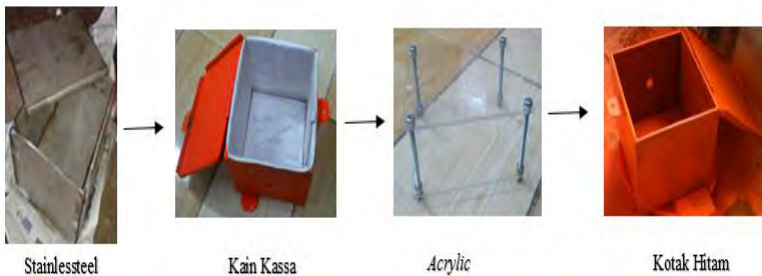
BAB IV

Pengujian Dan Analisa Data

Untuk mengetahui apakah tujuan dari pembuatan alat ini telah terlaksana dengan baik atau tidak, maka perlu dilakukan pengujian dan analisa terhadap alat yang dibuat. Gambar dan data dibawah menunjukkan hasil pembuatan *hardware* dan *software* sistem dan yang nantinya akan diletakkan pada sepeda motor sebagai Kotak Hitam atau *Black Box*.

4.1 Pembuatan Prototipe Kotak Hitam

Dengan menggunakan kotak *stainlesssteel* dengan panjang 17 cm, lebar 15 cm dan tinggi 14 cm dan ketebalan 4mm bisa menjaga komponen elektronik pada prototipe kotak hitam dari benturan yang keras untuk menghindari kerusakan. Selain itu untuk melindungi dari panas yang berlebihan/terbakarnya kotak hitam, pada prototipe diberikan lapisan kain kassa putih yang dilekatkan pada bagian samping dari *stainlesssteel*, kemudian diberikan *acrylic* sebagai tempat instalasi komponen utama dalam kotak hitam yang berfungsi penting untuk menyimpan data sensor. Setelah itu prototipe kotak hitam di beri warna dengan warna *orange*. Gambar 4.1 adalah prototipe kotak hitam.



Gambar 4.1 Hasil *Hardware* Sistem

Dari *hardware* yang telah dibuat tersebut akan dilakukan pengujian dari berbagai komponen *system* seperti sensor tekanan (*loadcell*), sensor kecepatan (*rotary encoder*), sensor temperatur (*thermocouple*), dan rangkaian *real time clock*, selain itu pengujian juga dilakukan pada *Arduino Mega* dengan *Ethernet Shield* sebagai penerima dan penyimpanan data sensor dengan *secure digital card* atau *SD Card*.

4.2 Pengujian Stainlesssteel

Proses pengujian besi *stainlesssteel* 4mm pada prototipe kotak hitam ini seperti memberikan beban berat, diberi pukulan dengan palu, dijatuhkan dari ketinggian 2 meter. Manfaat pengujian terhadap *cover* atau bagian depan dari kotak hitam ini adalah untuk memastikan tingkat kekuatan dari bahan yang digunakan sebagai pelindung utama. Apakah timbul keretakan atau tidak dari kotak hitam tersebut. Untuk melihat data dari hasil pengujian bahan *stainlesssteel* ini tersedia pada Tabel 4.1 berikut :

Tabel 4.1 Hasil Pengujian Bahan *Stainlesssteel*

No	Bahan	Objek Penguji	Kondisi
1	<i>Stainlesssteel</i>	Beban seberat 50 kg	Ok
2	<i>Stainlesssteel</i>	Palu	Ok
3	<i>Stainlesssteel</i>	Posisi jatuh dari ketinggian 2 meter	Ok
4	<i>Stainlesssteel</i>	Lempar dari jarak 3 meter	Ok
5	<i>Stainlesssteel</i>	Dibakar	Ok

4.3 Pengujian Arduino Mega 2560 dan Ethernet Shield

Untuk menguji *Arduino Mega* dan *Arduino Ethernet Shield*, maka perlu diisi dengan program sederhana, dimana manfaatnya adalah untuk menyalakan dan mematikan *led* yang dipasang pada tiap *pin* yang digunakan dan melakukan pengukuran tegangan pada tiap *pin* saat diberi logika 1 (*high*) dan logika 0 (*low*). Berikut langkah – langkah melakukan pengujian *Arduino Mega*.

1. Membuat program dengan *software Arduino ERW 1.0.5*
2. Setelah itu membuat program sederhana pada tiap *pin arduino* yang digunakan dengan logika 1 (*high*) dan logika 0 (*low*).
3. Langkah selanjutnya adalah *download* program tersebut ke *Arduino Mega*, terlebih dulu pastikan bahwa koneksi ke *PC* ke *Arduino* telah terhubung. Setelah itu pilih *opsi File* → *Upload* , jika sudah tidak ada *error* dalam program maka akan tertera *Done Compiling*
4. Pasang *led* pada *pin* yang diuji
5. Setelah itu ukur tegangan tiap *pin* yang digunakan, dengan menghubungkan *output pin* yang digunakan dengan kaki positif *voltmeter digital* dan kaki negatif dihubungkan pada *ground*.

Hasil dari pengukuran *Arduino* dengan logika 1 (*high*) dan logika 0 (*low*) dengan tegangan yang dihasilkan regulator *power supply* sebesar 8.96 *Volt* yang kemudian digunakan untuk mencatu *Arduino mega* dengan *output* tegangan sebesar 4,96 *Volt*, Gambar 4.2 adalah *listing* program membuat *pin arduino* bernilai atau berlogika 1 (*high*) dan 0 (*low*).

```

int vccPin = 0;
int gndPin = 1;
int dataac=0, dataaf=0;
int dataacb=0, dataafb=0;
void setup()
{
  // put your setup code here, to run once:
  pinMode(vccPin, OUTPUT); digitalWrite(vccPin, HIGH);
  pinMode(gndPin, OUTPUT); digitalWrite(gndPin, LOW);
}
void loop()
{
  // put your main code here, to run repeatedly:
}

```

Gambar 4.2 *Listing* Program Mengaktifkan *Pin Arduino*

Hasil pengujian tiap *pin arduino* yang akan digunakan, didapatkan kesimpulan bahwa *pin* yang diberi logika 1 (*high*) dan 0 (*low*) akan dapat menyalakan lampu *led* yang dipasang pada *pin* tersebut. Hasil pengujian tiap *pin arduino* dituliskan dalam Tabel 4.2 sebagai berikut :

Tabel 4.2 Pengujian *Arduino Mega 2560* dan *Ethernet Shield*

No	Pin	Tegangan	Indikator LED
1	A0	4,96 V	Nyala
2	A1	4,96 V	Nyala
3	A2	4,96 V	Nyala
4	D0 (RX)	4,96 V	Nyala
5	D1 (TX)	4,96 V	Nyala
6	D3	4,96 V	Nyala
7	D4	4,96 V	Nyala
8	D5	4,96 V	Nyala
9	D6	4,96 V	Nyala
10	D7	4,96 V	Nyala
11	D8	4,96 V	Nyala
12	D9	4,96 V	Nyala

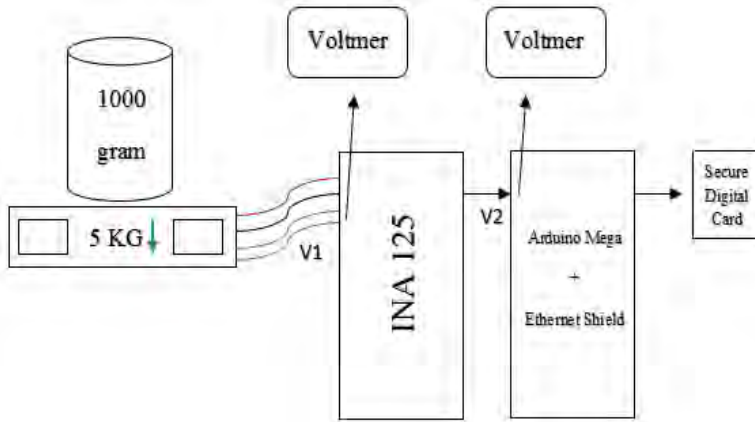
No	Pin	Tegangan	Indikator <i>LED</i>
13	D11	4,96 V	Nyala
14	D12	4,96 V	Nyala
15	D13	4,96 V	Nyala
16	D20 (SDA)	4,96 V	Nyala
17	D21 (SCL)	4,96 V	Nyala
16	D50	4,96 V	Nyala
17	D51	4,96 V	Nyala
18	D52	4,96 V	Nyala
19	D53	4,96 V	Nyala

Jadi berdasarkan Tabel 4.2 diatas bisa diambil kesimpulan bahwa pada pin *analog* dan *digital* yang berhasil diuji dengan menyalakan lampu *led* bisa untuk melakukan instruksi untuk penggunaan *sensor*, dan perangkat lainnya.

4.4 Pengujian Sensor Tekanan *Load Cell*

Loadcell merupakan salah satu sensor utama yang digunakan dalam tugas akhir ini. Sensor pada tugas akhir ini digunakan sebagai pengukur tekanan pengereman pada sepeda motor. *Loadcell* yang di gunakan adalah dengan kapasitas maksimum 5000 gram. Sensor ini menghasilkan *output* tegangan yang sangat kecil, yaitu 00,1 mV jadi memerlukan rangkaian penguat untuk bisa mengukur *output* tegangan dari sensor yang sangat kecil. Pengambilan data sensor dilakukan dengan memberikan timbel berat timbangan dengan rentang nilai 500, 1000 gram, 2000 gram dan 5000 gram yang diujikan pada *loadcell* agar mengetahui tegangan *loadcell* setelah dikuatkan apakah setiap kenaikan berat timbel yang diujikan, tegangan *loadcell* akan secara linier naik atau tidak.

Keluaran dari sensor ini terdiri dari empat kabel yag berwarna merah, hitam, hijau dan putih. Kabel merah merupakan *input* tegangan sensor sebesar 5 Volt kabel hitam adalah *ground*. Kabel berwarna hijau merupakan *output* positif dari sensor dan kabel putih adalah *output* negatif dari sensor. Cara pengambilan data dengan cara mengukur *output* tegangan rangkaian penguat sinyal setiap perubahan 500 gram, pengukuran tersebut pada *output loadcell* pada kabel hijau dan putih (v1) dan *output* INA 125 (v2) dengan menggunakan *volt meter digital*. Cara pengambilan data dapat dilihat pada Gambar 4.3



Gambar 4.3 Proses Pengukuran Sensor *Loadcell*

Langkah-langkah pengukuran :

1. Peralatan yang digunakan :
 - a. *Arduino Mega 2560* dan *Ethernet Shield*
 - b. Sensor *loadcell* 5000 gram
 - c. Rangkaian Penguat Sinyal INA125
 - d. *Acrylic*
 - e. *Voltmeter Digital*
 - f. Timbal timbangan 500, 1000, 2000, dan 5000 gram
2. Cara pengambilan data :
 - a. Mempersiapkan rancangan sensor *loadcell*
 - b. Menghubungkan *loadcell* dengan INA 125 dan sambungkan ke *arduino*
 - c. Mengatur nilai R pada *resistor variable* sebagai nilai *gain*
 - d. Mengukur tegangan *output* tegangan *loadcell* sebelum masuk penguat (v1) dan *output* tegangan setelah penguatan (v2) dengan *voltmeter digital*
 - e. Melakukan pencatatan data tiap perubahan 500 gram
 - f. Mengisi data percobaan ke dalam Tabel 4.3

Dengan menggunakan *resistor variable*, nilai R disetting sebesar 30 Ω . Jika dimasukkan kedalam rumus penguatan INA 125, maka penguatan yang terjadi jika menggunakan rumus adalah sebesar 2400, untuk melihat perhitungan bisa melihat rumus dibawah ini :
Gain / Penguatan =

$$G = 4 + \frac{60k\Omega}{R_g} \quad (4.1)$$

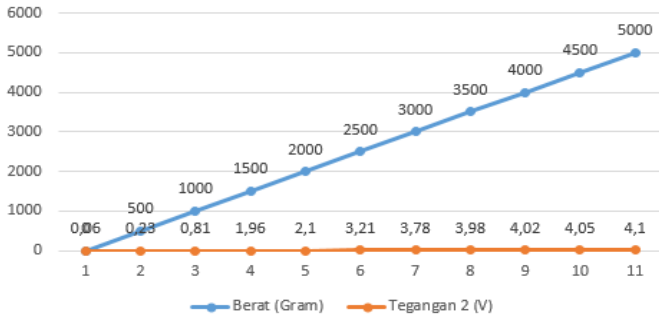
$$G = 4 + \frac{60k\Omega}{30\Omega} = 2400 \text{ kali penguatan} \quad (4.2)$$

Setelah menghitung penguatan, kemudian hasil pengukuran sensor *loadcell* ada pada Tabel 4.3.

Tabel 4.3 Data Pengukuran Tekanan *Loadcell*

No	Berat (Gram)	Tegangan 1 (mV)	Tegangan 2 (V)
1	0	00,0	0,06
2	500	00,1	0,23
3	1000	00,6	0,81
4	1500	01,0	1,96
5	2000	01,5	2,10
6	2500	02,0	3,21
7	3000	02,2	3,78
8	3500	03,0	3,98
9	4000	03,4	4,02
10	4500	03,9	4,05
11	5000	04,4	4,10

Berdasarkan data hasil pengujian sensor *load cell* yang dipasang untuk mengukur tekanan pengereman, saat *loadcell* tidak dibebani nilai tegangan *output* sensor *loadcell* setelah dikuatkan sebesar 2400 kali penguatan bernilai 0,06 V. Ketika *loadcell* dibebani dengan 500 gram maka nilai tegangan *output* sensor *loadcell* sebesar 0,23 V ketika beban yang ditimbang sebesar 5000 gram diperoleh nilai tegangan 4,10 V. Nilai tegangan *output* dari sensor telah dikuatkan oleh rangkaian penguat sinyal INA 125 agar dapat terbaca tegangannya. Grafik dari data diatas ada pada Gambar 4.4 dibawah ini.



Gambar 4.4 Linier Pengukuran *Loadcell*

Dari Gambar 4.4, bisa disimpulkan bahwa perbandingan data antara timbel berat yang di ujikan pada *loadcell* (gram) dengan tegangan *loadcell* setelah diberi penguatan (V) adalah linier, artinya jika berat timbel semakin besar maka tegangan akan ikut naik. Dokumentasi dari pengukuran sensor *loadcell* ini ada pada Gambar 4.5 dengan menggunakan timbel berat.



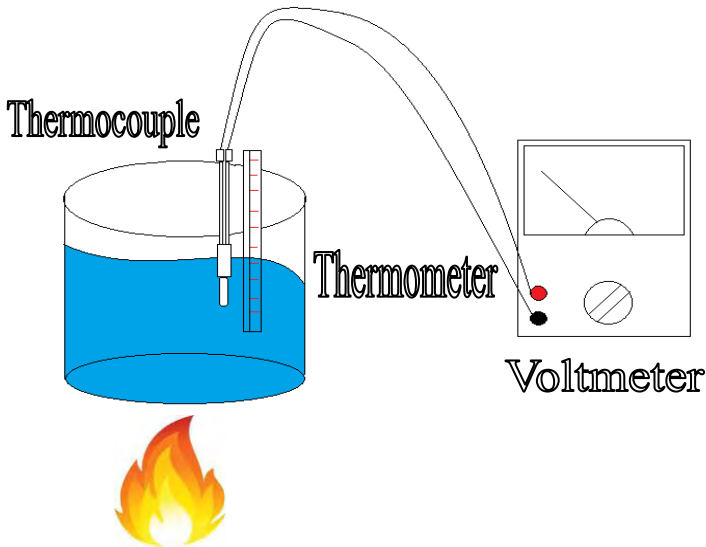
Gambar 4.5 Pengukuran Tegangan (2000 Gram)

4.5 Pengujian Sensor Temperatur *Thermocouple Type K*

*Thermokoupe*l baut tipe K dengan maksimal pembacaan hingga 800 °C. Sensor ini menghasilkan *output* tegangan yang sangat kecil, yaitu 00,1 mv. Maka dari itu perlu diberikan sebuah rangkaian penguat sinyal, rangkaian penguat sinyal ini menggunakan LM741, nantinya tegangan sebelum diberikan penguatan yang sebelumnya akan diukur dan dibandingkan dengan tegangan setelah diberi penguatan sesuai kenaikan suhu yang terjadi pada *thermometer*.

Pengujian dan pengambilan data sensor dilakukan dengan meletakkan sensor pada sebuah bejana yang dipanasi kemudian diukur output perubahan tegangannya tiap kenaikan 10°C. Selain itu suhu dalam bejana juga diukur menggunakan *Thermometer* suhu 300°C. Untuk memberikan batasan dalam pengujian, pengambilan data dilakukan pada rentang temperatur 30 – 100 °C.

Cara dan hasil pemrosesan data, dimulai dari pengambilan hingga pencatatan data dapat dilihat pada Gambar 4.6 dan Tabel 4.4:



Gambar 4.6 Proses Pengukuran Sensor *Thermocouple Type K*

Langkah-langkah pengukuran :

1. Peralatan yang digunakan :
 - a. Sensor *Thermocouple Type K*
 - b. Bejana Air
 - c. Spirtus
 - d. Termometer 300°C
 - e. *Voltmeter Digital*
2. Cara pengambilan data :
 - a. Mempersiapkan rangkaian pengkondisi sinyal *Thermocouple* LM741
 - b. Menghubungkan *Thermocouple* dengan LM741 dan *Arduino*
 - c. Masukan *Thermocouple* yang sudah disambung dengan LM741 kedalam bejana air yang sudah dipanaskan, dan siapkan *Voltmeter* dan ukur *output* tegangan yang terjadi.
 - d. Melakukan pencatatan data tiap 10 °C sampai 100 °C
 - e. Mengisi data percobaan ke dalam Tabel 4.4

Dengan menggunakan *resistor*, nilai R adalah 10 k dan 86 Ω .
Jika dimasukan kedalam rumus penguatan LM324, maka penguatan yang terjadi jika menggunakan rumus adalah sebesar 117 kali penguatan, untuk melihat perhitungan bisa melihat rumus dibawah ini :
Gain / Penguatan =

$$V_{out} = V_{in} \frac{R_f}{R_i} + 1 \quad (4.3)$$

$$V_{out} = V_{in} \frac{22k\Omega}{1k\Omega} + 1 = 23k\Omega \text{ kali penguatan} \quad (4.4)$$

$$V_{out} = V_{in} \frac{33k\Omega}{1k\Omega} + 1 = 34k\Omega \text{ kali penguatan} \quad (4.5)$$

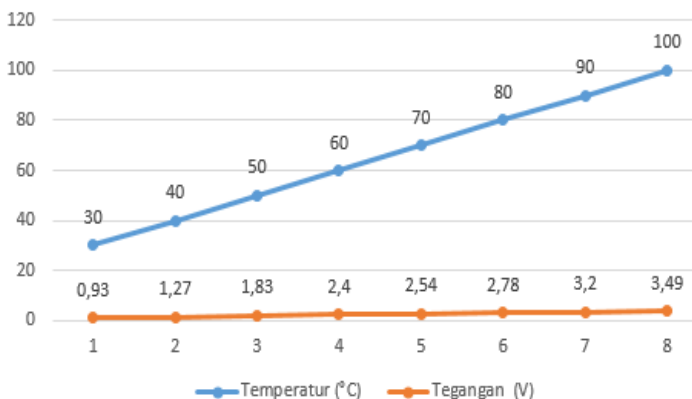
$$Gain = 23k\Omega \times 34k\Omega = 782 \text{ kali penguatan} \quad (4.6)$$

Setelah menghitung penguatan, maka hasil pengukuran ditulis pada Tabel 4.4 dibawah ini:

Tabel 4.4 Data Pengukuran Temperatur *Thermocouple*

No	Temperatur (°C)	Tegangan (mV)	Tegangan (V)
1	30	00,1	0,93
2	40	00,3	1,27
3	50	00,5	1,83
4	60	00,7	2,40
5	70	01,0	2,54
6	80	01,4	2,78
7	90	01,8	3,20
8	100	02.4	3,49

Berdasarkan data hasil pengujian sensor *thermocouple* yang dipasang untuk mengukur temperatur mesin sepeda motor, saat *thermocouple* belum diberi penguatan tegangan saat temperatur 30°C adalah 00,1 mV dan setelah diberi penguatan 0,93 V. Grafik dari data diatas dapat dilihat pada Gambar 4.7.



Gambar 4.7 Linier Pengukuran *Thermocouple*

Dari Gambar 4.7 diatas, bisa disimpulkan bahwa perbandingan data antara temperatur dengan tegangan *thermocouple* (V) adalah linier, karena ketika suhu yang ditunjukkan pada thermometer naik, maka tegangan pada *thermocouple* juga akan mengalami kenaikan.

Dokumentasi dari pengukuran sensor *thermocouple* ini ada pada Gambar 4.8.

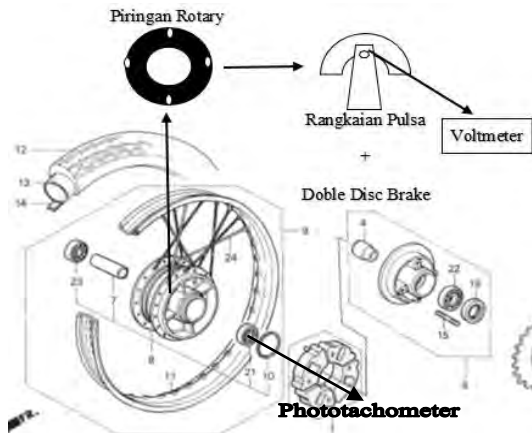


Gambar 4.8 Proses Pengukuran Temperatur Dengan Bejana Air Panas

4.6 Pengujian Sensor Kecepatan *Rotary Encoder*

Rotary encoder pada Tugas Akhir ini digunakan sebagai sensor yang mengukur kecepatan putaran roda sepeda motor. Dalam proses pembuatannya sepeda motor diberi sebuah lempengan *acrylic* hitam dengan tebal 3 mm dan 16 lubang pada bagian *velg* belakang. Pemasangan pada *velg* belakang ini megadopsi pemasangan *back double disc brake* pada sepeda motor bebek pada umumnya, yang menggunakan “dudukan” untuk meletakkan rangkaian pulsa.

Lempengan *acrylic* ini sudah mengalami proses pengujian *acrylic* dengan ketebalan 5mm, 3mm dan 2mm, dan hasil pengamatan, menggunakan *acrylic* 3mm lebih tepat dan pas jika dipasang pada *velg* belakang, selain itu *acrylic* 3mm ini dapat masuk pada bagian rangkaian pulsa. Proses pengukuran *rotary encoder* adalah dengan menggunakan rangkaian pulsa adalah dengan *photo tachometer* untuk mengetahui berapa kecepatan dalam (RPM) putaran roda tiap kenaikan 10 km/jam. Batasan pengukuran kecepatan putaran roda sepeda motor ini adalah 0 – 120 km/jam. Gambar 4.9 adalah gambaran mekanik dari proses pengambilan data putaran roda dengan menggunakan *photo tachometer*.



Gambar 4.9 Proses Pengukuran *Rotary Encoder*

Langkah-langkah pengukuran :

1. Peralatan yang digunakan :
 - a. Lempekan *Acrylic* 3mm, 8 lubang
 - b. Rangkaian pulsa
 - c. *Photo Tachometer*
 - d. *Voltmeter Digital*
 - e. Lempekan besi 2mm
 - f. *Aluminium foil*
2. Cara pengambilan data :
 - a. Memasang lempengan *rotary encoder* 3mm 8 lubang pada roda belakang sepeda motor.
 - b. Memasang mekanik *double disc brake* rangkaian pulsa dengan lempengan plat besi 3mm
 - c. Memasang *aluminium foil* pada salah satu bagian lempengan *acrylic*
 - d. Siapkan *Photo Tachometer* dan mulai pengukuran
 - e. Pengukuran *output* tegangan dengan kecepatan 10-120km/jam
 - f. Mengisi data percobaan ke dalam Tabel 4.6

Kemudian setelah pengukuran kecepatan nyata dan kecepatan perhitungan, data ditulis di Tabel 4.5 berikut.

Tabel 4.5 Data Pengukuran *Rotary Encoder*

No	Kecepatan (Km/Jam)	RPM	Perhitungan Putaran Roda (Km/Jam)	Kesalahan (%)
1	10	92,2	10	0
2	20	162,3	18	10
3	30	260,2	29	3,3
4	40	343,3	38	5
5	50	439,1	48	4
6	60	513,8	56	6,6
7	70	631,8	69	1,4
8	80	703,9	77	3,75
9	90	800,2	88	2,2
10	100	912,1	100	0
11	110	1005	110	0
12	120	1100	120	0

Berdasarkan Tabel 4.5 diatas, pengukuran putaran roda dengan *photo tachometer* diatas, data diperoleh dengan satuan (RPM). Setelah itu perlu perbandingan antara kecepatan nyata (*speedometer*) dengan perhitungan dengan Rumus Keliling Roda (4.7) dan kemudian dikonversi menjadi (Km/Jam) dengan Rumus Kecepatan (4.8) sebagai berikut :

$$\text{Rumus Keliling Roda} = 2 \frac{22}{7} + r \quad (4.7)$$

$$\text{Rumus Kecepatan (Km/Jam)} = \text{KellRoda} \times \text{RPM} \times 0,00001 \times 60 \quad (4.8)$$

Diketahui Keliling Roda Sepeda Motor adalah 182,28 cm dengan jari jari 29 cm, kemudian perhitungan untuk mengkonversi menjadi satuan Km/Jam sebagai berikut :

1. Kecepatan 1 = $182,28 \times 92,2 \times 0,00001 \times 60 = 10 \text{ Km/Jam}$
2. Kecepatan 2 = $182,28 \times 162,3 \times 0,00001 \times 60 = 18 \text{ Km/Jam}$
3. Kecepatan 3 = $182,28 \times 260,2 \times 0,00001 \times 60 = 29 \text{ Km/Jam}$
4. Kecepatan 4 = $182,28 \times 343,3 \times 0,00001 \times 60 = 38 \text{ Km/Jam}$
5. Kecepatan 5 = $182,28 \times 439,1 \times 0,00001 \times 60 = 48 \text{ Km/Jam}$
6. Kecepatan 6 = $182,28 \times 513,8 \times 0,00001 \times 60 = 56 \text{ Km/Jam}$
7. Kecepatan 7 = $182,28 \times 631,8 \times 0,00001 \times 60 = 69 \text{ Km/Jam}$
8. Kecepatan 8 = $182,28 \times 703,9 \times 0,00001 \times 60 = 77 \text{ Km/Jam}$
9. Kecepatan 9 = $182,28 \times 800,2 \times 0,00001 \times 60 = 88 \text{ Km/Jam}$
10. Kecepatan 10 = $182,28 \times 912,1 \times 0,00001 \times 60 = 100 \text{ Km/Jam}$
11. Kecepatan 11 = $182,28 \times 1005 \times 0,00001 \times 60 = 110 \text{ Km/Jam}$
12. Kecepatan 12 = $182,28 \times 1100 \times 0,00001 \times 60 = 120 \text{ Km/Jam}$

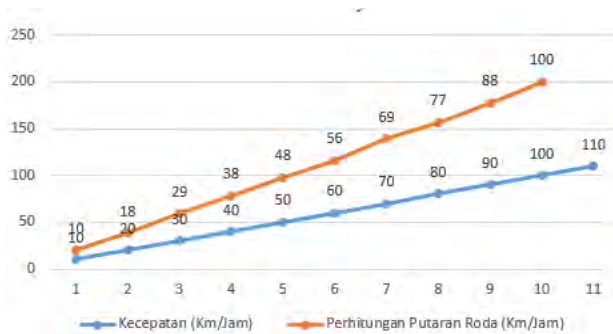
Dari perhitungan diatas, dapat dilihat perbandingan antara kecepatan nyata dengan kecepatan perhitungan, dengan ini dapat dicari kesalahan / *error* dari perbandingan kecepatan nyata dengan kecepatan perhitungan dalam (Km/Jam) dengan Rumus % *Error* (4.9) berikut ini :

$$\%Error = \frac{Kec.Nyata - Kec.Perhitungan}{Kec.Nyata} \times 100\% \quad (4.9)$$

Perhitungan kesalahan (*error*) adalah sebagai berikut :

1. % *Error* kecepatan 10 Km/Jam = $10-10/10 \times 100\% = 0\%$
2. % *Error* kecepatan 20 Km/Jam = $20-18/20 \times 100\% = 10\%$
3. % *Error* kecepatan 30 Km/Jam = $30-29/30 \times 100\% = 3,3\%$
4. % *Error* kecepatan 40 Km/Jam = $40-38/40 \times 100\% = 5\%$
5. % *Error* kecepatan 50 Km/Jam = $50-48/50 \times 100\% = 4\%$
6. % *Error* kecepatan 60 Km/Jam = $60-56/60 \times 100\% = 6,6\%$
7. % *Error* kecepatan 70 Km/Jam = $70-69/70 \times 100\% = 1,4\%$
8. % *Error* kecepatan 80 Km/Jam = $80-77/80 \times 100\% = 3,75\%$
9. % *Error* kecepatan 90 Km/Jam = $90-88/90 \times 100\% = 2,2\%$
10. % *Error* kecepatan 100 Km/Jam = $100-100/100 \times 100\% = 0\%$
11. % *Error* kecepatan 110 Km/Jam = $110-110/110 \times 100\% = 0\%$
12. % *Error* kecepatan 120 Km/Jam = $120-120/120 \times 100\% = 0\%$

Grafik dari data kecepatan nyata kecepatan roda, dengan data kecepatan perhitungan, dapat dilihat pada Gambar 4.10.



Gambar 4.10 Linier Pengukuran *Rotary Encoder*

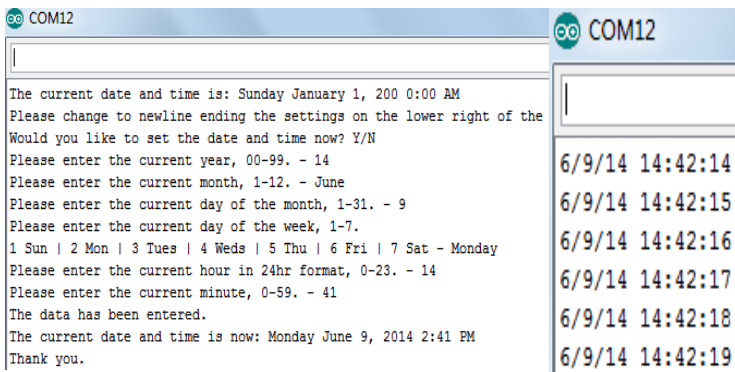
Dari Gambar 4.10 diatas, bisa disimpulkan bahwa perbandingan data antara kecepatan nyata dengan kecepatan perhitungan adalah linier.

4.7 Pengujian *Real Time Clock*

Pengujian rangkaian *real time clock* ini menggunakan *setting* program yang dibuat menggunakan *arduino*. Program ini dibuat untuk mengatur tanggal, bulan, tahun, jam, menit dan detik agar sistem berjalan secara *real time*. Rangkaian RTC ini menggunakan IC DS 1307 dengan komunikasi SDA dan SCL yang masuk pada *pin* SDA dan SCL pada *Arduino Mega*. Langkah-langkah untuk mengatur RTC DS 1307 adalah sebagai berikut :

1. Membuat program dengan *software Arduino ERW 1.0.5 setting RTC* untuk *arduino* (*library arduino* atau forum *arduino*)
2. Langkah selanjutnya adalah *download* program tersebut ke *Arduino Mega*, terlebih dulu pastikan bahwa koneksi ke *PC* ke *Arduino* telah terhubung. Setelah itu pilih *opsi File* → *Upload* , Jika sudah tidak ada *error* dalam program makan akan tertera *Done Compiling*
3. Buka *Serial Monitor* untuk pengaturan Rangkaian *Real Time Clock*
4. Pertama akan keluar pertanyaan, *Would you like to set the date and time now??* Jawab “Y”, pertanyaan selanjutnya jawab dengan konisi yang sesuai akan di atur.
5. Kedua akan keluar pertanyaan, *Please enter the current year*

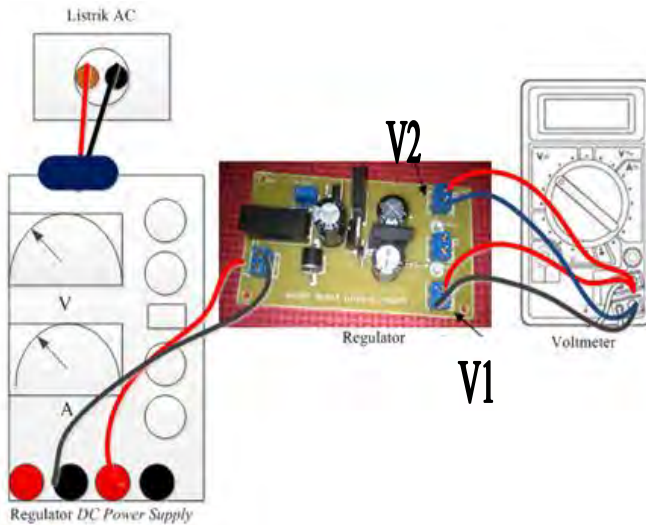
6. Ketiga akan keluar pertanyaan, *Please enter the current month*
7. Keempat akan keluar pertanyaan, *Please enter the current day of the month*
8. Kelima akan keluar pertanyaan, *Please enter the current day of the week*
9. Keenam akan keluar pertanyaan, *Please enter the current hour in 24hr format*
10. Ketujuh akan keluar pertanyaan, *Please enter the current minute*
11. Jika berhasil maka akan terdapat pemberitahuan seperti ini “*The current date and time is now: Thursday June 12, 2014 10:02 AM (sesuai setting), Thank you*”
12. Selanjutnya untuk megecek apakah RTC sudah bekerja secara real time atau belum, maka harus menggunakan program tambahan DS1307 untuk uji coba.
13. Gambar 4.11 ini adalah hasil setting dan uji coba RTC DS 1307.



Gambar 4.11 Hasil Pengaturan *Real Time Clock*

4.8 Pengujian Regulator

Regulator tegangan ini adalah bagian power supply yang berupa accu 12 Volt 3,5 Amp yang berfungsi untuk memberikan stabilitas output pada suatu *power supply*. Pengujian rangkaian regulator ini dengan sebuah *Regulated DC Power Supply model PK-653* dengan mengubah input tegangan dengan range 9 Volt hingga 17 Volt untuk menguji apakah pada regulator stabil ketika tegangan input naik atau turun. Gambar 4.12 adalah proses pengujian rangkaian regulator.



Gambar 4.12 Hasil Pengujian Regulator

Langkah-langkah pengukuran :

- 1.. Peralatan yang digunakan :
 - a. *Regulated DC Power Suplly*
 - b. Rangkaian regulator
 - c. *Voltmeter Digital*
- 2.. Cara pengambilan data :
 - a. Memasang Regulated Power Supply kedalam rangkaian regulator
 - b. Siapkan range pengukuran 9-16 Volt
 - c. Ukur terlebih dahulu *output Regulator DC Power Suplly* nya
 - d. Setelah sesuai dengan tegangan yang diinginkan, maka sambungan on/off pada rangkaian regulator dinyalakan
 - e. Ukur *output* tegangan regulator V1 (9v) dan V2 (7,5v)
 - f. Mengisi data percobaan ke dalam Tabel 4.6

Tabel 4.6 dibawah ini adalah hasil uji coba pengukuran regulator yang akan dipasang pada sepeda motor.

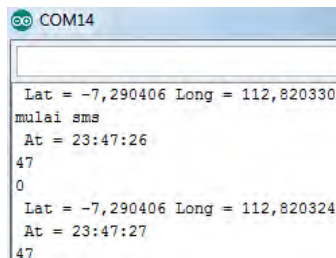
Tabel 4.6 Data Pengukuran Tegangan Regulator

No	Tegangan (Volt)	Output X1 (9 Volt)	Output X2 (7,5 Volt)
1	9	6,87	6,50
2	10	7,80	7,4
3	11	8,74	7,5
4	12,02	9,02	7,5
5	13,02	9,02	7,5
6	14	9,02	7,5
7	14,98	9,02	7,5
8	16	9,02	7,5

Berdasarkan Tabel 4.6 menunjukkan bahwa jika tegangan input kotak hitam secara keseluruhan kurang dari 11 volt atau tegangan turun maka dapat disimpulkan bahwa sistem tidak akan bekerja. Karena tegangan *input arduino* membutuhkan masukan sebesar 7 – 12 volt untuk mengaktifkan sistem dan sensor.

4.9 Pengujian *Global Positioning System* (GPS)

Pengujian dilakukan pada 10 lokasi yang telah ditentukan. Sebagai contoh pengujian dilakukan di daerah kampus ITS dan daerah sekitar, seperti gebang, dan medokan semampir. Selanjutnya dilakukan pengambilan *latitude* dan *longitude* data GPS. Pengujian dibandingkan penerima GPS *U-Blox Neo 6M* dari Garmin *GPSMAP 60CS*. Pada Gambar 4.13 tampilan *latitude* dan *longitude* dengan *display serial monitor arduino*.



Gambar 4.13 Tampilan *Latitude* dan *Longitude*

Pada Tabel 4.7 adalah perbandingan hasil GPS U-Blox Neo 6M dari Garmin GPSMAP 60CS dengan mengambil sampel 10 tempat.

Tabel 4.7 Perbandingan Hasil GPS

No	Lokasi	Koordinat Alat (<i>Display Monitor</i>)	Koordinat Garmin GPSMAP 60CS
1	Teknik Perkapalan ITS	-7,279543, 112,798156	-7,282538, 112,796911
2	Teknik Elektro ITS	-7,281005, 112,798085	-7,285028, 112,796777
3	Gedung Rektorat ITS	-7,279256, 112,796414	-7,285027, 112,79408
4	Gerbang Masuk ITS	-7,277830, 112,794442	-7,279694, 112,790691
5	Gebang Kidul No. 32 (Gozi)	-7,279462, 112,792204	-7,282639, 112,786805
6	Apartemen Arif Rahman Hakim	-7,282769, 112,792546	-7,288052, 112,787666
7	Gerbang Galaxy Bumi Permai	-7,295478, 112,791899	-7,296496, 112,786445
8	Gerbang Araya	-7,281318, 112,791503	-7,285583, 112,758588
9	Medokan Semampir (TNI)	-7,291489, 112,793001	-7,302400, 112,788905
10	Bengkel Jangkar Miring Corvette	-7,290436, 112,7999981	-7,300750, 112,800833

Berdasarkan Tabel 4.7 di atas, nilai pembacaan GPS garmin harus dikonversi terlebih dahulu dari DMS (*Degree Minute Second*) to *Decimal Degrees* dengan rumus dibawah 4.8 ini :

$$Konversi = NilaiDegree + \frac{NilaiMinute}{60} + \frac{NilaiSecond}{3600} \quad (4.11)$$

Setelah itu diperoleh nilai *latitude* yang memiliki nilai (-) dan *longitude* bernilai (+) karena Indonesia terletak pada lintang selatan (LS) dan terletak pada garis bujur barat (BT), dengan ketentuan *latitude north* (+), *latitude south* (-) dan *longitude east* (+), *longitude west* (-),

kemudian dapat kita cari kesalahan relatif antara *GPS U-Blox Neo 6M* dengan *Garmin GPSMAP 60CS* dengan rumus sebagai berikut:

$$e = \frac{|x - k|}{k} \times 100\% \quad (4.10)$$

dimana e = kesalahan relatif

x = koordinat alat

k = koordinat pembanding.

Data kesalahan relatif untuk masing-masing lokasi adalah sebagai berikut:

1. a. *Error latitude* = $\frac{7,279543 - 7,282538}{7,282538} \times 100\% = 0,0411\%$
 b. *Error longitude* = $\frac{112,798156 - 112,796911}{112,796911} \times 100\% = 0,1103\%$
2. a. *Error latitude* = $\frac{7,281005 - 7,285028}{7,285028} \times 100\% = 0,0552\%$
 b. *Error longitude* = $\frac{112,798085 - 112,796911}{112,796911} \times 100\% = 0,1040\%$
3. a. *Error latitude* = $\frac{7,279256 - 7,285027}{7,285027} \times 100\% = 0,0792\%$
 b. *Error longitude* = $\frac{112,796414 - 112,79408}{112,79408} \times 100\% = 0,2069\%$
4. a. *Error latitude* = $\frac{7,277830 - 7,279694}{7,279694} \times 100\% = 0,0256\%$
 b. *Error longitude* = $\frac{112,794442 - 112,790691}{112,790691} \times 100\% = 0,3325\%$
5. a. *Error latitude* = $\frac{7,279462 - 7,282639}{7,282639} \times 100\% = 0,0436\%$
 b. *Error longitude* = $\frac{112,792204 - 112,786805}{112,786805} \times 100\% = 0,4786\%$

6. a. $Error\ latitude = \frac{7,282769 - 7,288052}{7,288052} \times 100\% = 0,0724\%$
 b. $Error\ longitude = \frac{112,792546 - 112,787666}{112,787666} \times 100\% = 0,4326\%$
7. a. $Error\ latitude = \frac{7,285478 - 7,296496}{7,296496} \times 100\% = 0,1510\%$
 b. $Error\ longitude = \frac{112,791899 - 112,786445}{112,786445} \times 100\% = 0,483\%$
8. a. $Error\ latitude = \frac{7,291318 - 7,285583}{7,285583} \times 100\% = 0,0787\%$
 b. $Error\ longitude = \frac{112,791503 - 112,758588}{112,758588} \times 100\% = 0,0291\%$
9. a. $Error\ latitude = \frac{7,291489 - 7,302400}{7,302400} \times 100\% = 0,1494\%$
 b. $Error\ longitude = \frac{112,793001 - 112,788905}{112,788905} \times 100\% = 0,3631\%$
10. a. $Error\ latitude = \frac{7,290436 - 7,300750}{7,300750} \times 100\% = 0,1412\%$
 b. $Error\ longitude = \frac{112,7999981 - 112,800833}{112,800833} \times 100\% = 0,0740\%$

Untuk rata-rata kesalahan relatif *latitude* didapatkan nilai sebesar 0,08374% dan untuk rata-rata kesalahan relatif *longitude* didapatkan nilai sebesar 0,26141%.

Kemudian akan dicari *error* dari selisih jarak posisi antara koordinat alat dan koordinat pembanding, kemudian dikonversi ke satuan meter, dimana 1' = 1,85 Km.

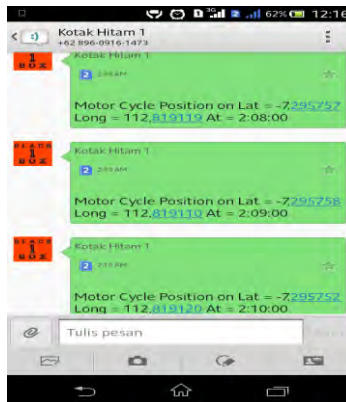
1. Selisih *latitude* = 203 m
 Selisih *longitude* = 92 m
2. Selisih *latitude* = 100 m
 Selisih *longitude* = 92 m
3. Selisih *latitude* = 100 m
 Selisih *longitude* = 166 m
4. Selisih *latitude* = 111 m
 Selisih *longitude* = 259 m
5. Selisih *latitude* = 203 m
 Selisih *longitude* = 351 m

6. Selisih *latitude* = 100 m
Selisih *longitude* = 333 m
7. Selisih *latitude* = 74 m
Selisih *longitude* = 351 m
8. Selisih *latitude* = 30 m
Selisih *longitude* = 300 m
9. Selisih *latitude* = 100 m
Selisih *longitude* = 259 m
10. Selisih *latitude* = 100 m
Selisih *longitude* = 814 m

Rata – rata *error* selisih jarak antara koordinat alat dan koordinat pembandingan untuk *latitude*-nya adalah 112,1 m, sedangkan *longitude*-nya adalah 300,1 m.

4.10 Pengujian Modem Wavecom GSM

Modem GSM yang digunakan berfungsi sebagai perangkat pengirim sms (*short message service*) kepada *handphone* berupa data *latitude* dan *longitude* dari *GPS U-Blox Neo 6M* yang dipasang pada sepeda motor guna mengetahui posisi dari sepeda motor tersebut yang nantinya akan dikonversi secara manual pada *google map* dengan memasukan nilai *latitude* dan *longitude* nya. Gambar 4.14 menunjukkan tampilan sms yang masuk dan berhasil menerima sms yang dikirimkan modem.



Gambar 4.14 Pengujian Modem Kirim SMS pada Handphone

Terlihat pada Gambar 4.14, SMS berhasil terkirim jam 2:08:00, 2:09:00 dan 2:10:00 karena di atur setiap 1 menit dapat mengirimkan sms, dengan *latitude* -7,295757 dan *longitude* 112,819119, *latitude* -7,295758 dan *longitude* 112,819110, *latitude* -7,295752 dan *longitude* 112,819120 dari nomer 089627092964 ke nomer 085642778048.

4.11 Pengujian Tampilan Data Logger Secure Digital Card

Pada kotak hitam yang dijadikan tugas akhir ini untuk menyimpan data sensor menggunakan sistem *datalogger*, data dari semua sensor *loadcell*, *thermocouple*, *rotary encoder* dan posisi berupa *latitude* dan *longitude* disimpan dengan media *secure digital card*. Pengujian tampilan *datalogger* ini dimaksudkan untuk membuat tampilan sederhana penampil sensor dari *secure digital card* ketika dibaca dengan menggunakan *personal computer* atau *laptop*. Dengan menyimpan data berupa (.txt), dengan format yang disimpan tersebut pada *secure digital card* tersebut, memudahkan dalam proses pembacaan dan dapat menampung kapasitas yang besar dikarenakan format tersebut tidak memakan memori yang besar. Dengan program yang dibuat maka Gambar 4.15 ini adalah tampilan dari *secure digital card* jika dibaca pada perangkat keras.

Tanggal;	waktu;	Temperature;	Kecepatan;	Tekanan;	Lat;	Long;
7/7/14;	1:50:47;	Suhu = 100 C;	0 KM/Jam;	5054 Grams;	Lat = -7,295741;	Long = 112,819112;
7/7/14;	1:50:49;	Suhu = 99 C;	0 KM/Jam;	5054 Grams;	Lat = -7,295740;	Long = 112,819112;
7/7/14;	1:50:51;	Suhu = 98 C;	0 KM/Jam;	5054 Grams;	Lat = -7,295739;	Long = 112,819112;
7/7/14;	1:50:52;	Suhu = 100 C;	0 KM/Jam;	5054 Grams;	Lat = -7,295742;	Long = 112,819111;
7/7/14;	1:50:54;	Suhu = 99 C;	0 KM/Jam;	5054 Grams;	Lat = -7,295741;	Long = 112,819112;
7/7/14;	1:50:55;	Suhu = 98 C;	0 KM/Jam;	5054 Grams;	Lat = -7,295739;	Long = 112,819113;
7/7/14;	1:50:57;	Suhu = 99 C;	0 KM/Jam;	5054 Grams;	Lat = -7,295738;	Long = 112,819113;
7/7/14;	1:50:59;	Suhu = 98 C;	0 KM/Jam;	5054 Grams;	Lat = -7,295736;	Long = 112,819114;

Gambar 4.15 Pengujian Tampilan Data Logger Secure Digital Card

Dari tampilan *datalogger* Gambar 4.15 dengan format (.txt), dapat dilihat urutan tampilan pada *notepad* berupa tanggal, waktu, *temperature*, kecepatan, tekanan, dan posisi yang berupa *latitude* dan *longitude* menyimpan data setiap 1-2 detik. Tempat penyimpanan data menggunakan *secure digital card* ini berkapasitas 8 Gb. Dari hasil pengujian, *memory* yang digunakan sekitar satu jam penggunaan ketika sistem berjalan adalah 60 Kb, jika digunakan sehari penuh (24 jam) maka *memory* kapasitas yang dipergunakan sebesar 1,4 Mb. Sehingga

memory 8 Gb ini akan habis kapasitasnya sekitar 10 tahun pemakaian jika tidak ada proses setting ulang atau penghapusan data (*history*) sebelumnya.

4.12 Uji Coba Kotak Hitam dan Transponder Sepeda Motor

Proses ujicoba kotak hitam ini menggunakan sample tempat mengitari perum ITS Blok U. Hal yang diuji disini adalah keseluruhan sistem kotak hitam dan *transponder*, apakah sensor dan gps sudah bekerja dengan baik atau belum dan dapat mengirimkan posisi sepeda motor melalui sms. *Datalogger* berupa *notepad* (.txt) dapt dilihat pada Gambar 4.16 dibawah ini :

Tanggal;	waktu;	Temperature;	Kecepatan;	Tekanan;	Lat;	Long;
10/7/14;	9:23:19;	suhu = 40 C;	10 KM/Jam;	0 Grams;	Lat = -7,290412;	Long = 112,820363;
165/165/165;	1:135:0;	suhu = 40 c;	15 KM/Jam;	0 Grams;	Lat = -7,290412;	Long = 112,820363;
10/7/17;	9:23:22;	suhu = 40 C;	15 KM/Jam;	0 Grams;	Lat = -7,290412;	Long = 112,820363;
10/7/14;	9:23:24;	suhu = 39 C;	15 KM/Jam;	0 Grams;	Lat = -7,290412;	Long = 112,820362;
10/7/14;	9:23:26;	suhu = 41 C;	16 KM/Jam;	0 Grams;	Lat = -7,290412;	Long = 112,820363;
10/7/14;	9:23:27;	suhu = 40 C;	25 KM/Jam;	0 Grams;	Lat = -7,290412;	Long = 112,820363;
10/7/14;	9:23:29;	suhu = 40 C;	24 KM/Jam;	229 Grams;	Lat = -7,290413;	Long = 112,820363;
10/7/14;	9:23:30;	suhu = 40 C;	10 KM/Jam;	4988 Grams;	Lat = -7,290413;	Long = 112,820364;
10/7/14;	9:23:32;	suhu = 40 C;	28 KM/Jam;	4988 Grams;	Lat = -7,290413;	Long = 112,820364;
10/7/14;	9:23:34;	suhu = 40 C;	22 KM/Jam;	4987 Grams;	Lat = -7,290413;	Long = 112,820365;
10/7/14;	9:23:35;	suhu = 39 C;	14 KM/Jam;	3502 Grams;	Lat = -7,290413;	Long = 112,820366;
10/7/14;	9:23:37;	suhu = 39 C;	8 KM/Jam;	29 Grams;	Lat = -7,290413;	Long = 112,820366;
10/7/14;	9:23:38;	suhu = 39 C;	7 KM/Jam;	0 Grams;	Lat = -7,290413;	Long = 112,820367;
10/7/14;	9:23:40;	suhu = 39 C;	7 KM/Jam;	179 Grams;	Lat = -7,290413;	Long = 112,820367;
10/7/14;	9:23:42;	suhu = 41 C;	8 KM/Jam;	4391 Grams;	Lat = -7,290413;	Long = 112,820368;

Gambar 4.16 Tampilan Data Logger Secure Digital Card

Dari Gambar 4.16 diperoleh hasil, tanggal menunjukkan 10/7/14 pukul 09:23:19. Data yang disimpan dari sensor dari perjalanan tersebut dengan *temperature* 39 °C – 44 °C, dengan kecepatan 0 Km/Jam – 28 Km/Jam dan tekanan pengereman 0 gram – 4988 gram. Selain itu posisi yang diberikan oleh GPS adalah berupa *latitude* sebesar -7,290413 dan *longitude* sebesar 112,820367. Untuk memudahkan pembacaan hasil *datalogger* diatas, bisa menggunakan *microsoft excel* sebagai tampilannya.

Langkah langkahnya konversinya adalah sebagai berikut :

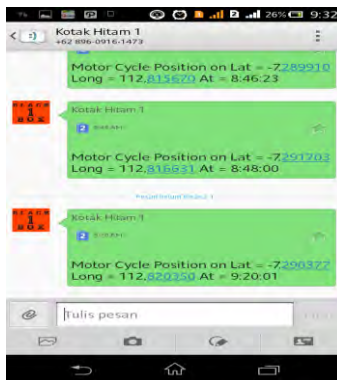
1. Buka *Microsoft Excel* - Klik *Data* - *From Text*
2. Pilih tempat datalog berupa *notepad* (.txt)
3. Klik *document* (*Datalog.txt*)
4. Next – Pilih *Semicolon* – *Text*
5. *Finish* - *New Worksheet*

Dari langkah-langkah diatas, diperoleh tampilan yang lebih memudahkan untuk dibaca dan dianalisa. Terdapat 7 kolom, yaitu tanggal, waktu, *temperature*, kecepatan, tekanan, *latitude* dan *longitude*. Gambar 4.17 adalah tampilan *microsoft excel*.

	A	B	C	D	E	F	G
1	Tanggal	Waktu	Temperature	Kecepatan	Tekanan	Lat	Long
2	10/7/14	9:23:19	Suhu = 40 C	10 KM/Jam	0 Grams	Lat = -7,290412	Long = 112,820363
3	165/165/165	1:135:0	Suhu = 40 C	15 KM/Jam	0 Grams	Lat = -7,290412	Long = 112,820363
4	10/7/17	9:23:22	Suhu = 40 C	15 KM/Jam	0 Grams	Lat = -7,290412	Long = 112,820363
5	10/7/14	9:23:24	Suhu = 39 C	15 KM/Jam	0 Grams	Lat = -7,290412	Long = 112,820362
6	10/7/14	9:23:26	Suhu = 41 C	16 KM/Jam	0 Grams	Lat = -7,290412	Long = 112,820363
7	10/7/14	9:23:27	Suhu = 40 C	25 KM/Jam	0 Grams	Lat = -7,290412	Long = 112,820363
8	10/7/14	9:23:29	Suhu = 40 C	24 KM/Jam	229 Grams	Lat = -7,290413	Long = 112,820363
9	10/7/14	9:23:30	Suhu = 40 C	10 KM/Jam	4988 Grams	Lat = -7,290413	Long = 112,820364
10	10/7/14	9:23:32	Suhu = 40 C	28 KM/Jam	4988 Grams	Lat = -7,290413	Long = 112,820364
11	10/7/14	9:23:34	Suhu = 40 C	22 KM/Jam	4987 Grams	Lat = -7,290413	Long = 112,820365
12	10/7/14	9:23:35	Suhu = 39 C	14 KM/Jam	3502 Grams	Lat = -7,290413	Long = 112,820366
13	10/7/14	9:23:37	Suhu = 39 C	8 KM/Jam	29 Grams	Lat = -7,290413	Long = 112,820366
14	10/7/14	9:23:38	Suhu = 39 C	7 KM/Jam	0 Grams	Lat = -7,290413	Long = 112,820367
15	10/7/14	9:23:40	Suhu = 39 C	7 KM/Jam	179 Grams	Lat = -7,290413	Long = 112,820367
16	10/7/14	9:23:42	Suhu = 41 C	8 KM/Jam	4391 Grams	Lat = -7,290413	Long = 112,820368
17	10/7/14	9:23:43	Suhu = 40 C	6 KM/Jam	0 Grams	Lat = -7,290413	Long = 112,820368
18	10/7/14	9:23:45	Suhu = 40 C	6 KM/Jam	0 Grams	Lat = -7,290413	Long = 112,820368
19	10/7/14	9:23:46	Suhu = 44 C	0 KM/Jam	0 Grams	Lat = -7,290413	Long = 112,820368

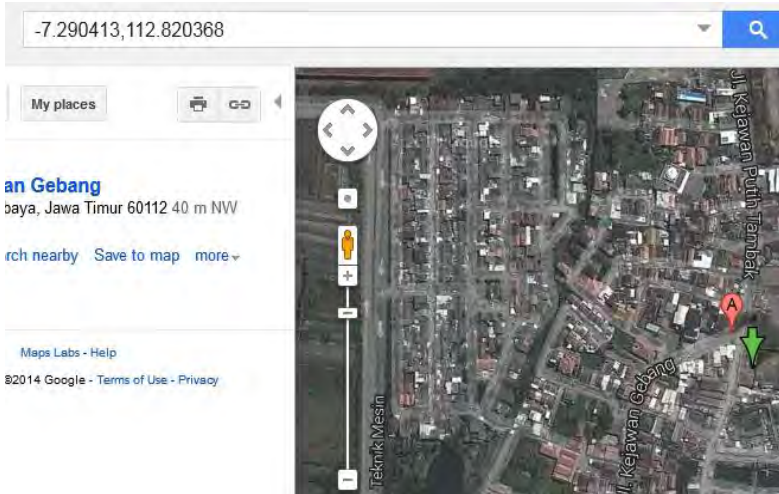
Gambar 4.17 Tampilan *Microsoft Excel*

Setelah data diperoleh, kini tampilan sms posisi pada *handphone* berupa *latitude* dan *longitude* setiap 1 menit. Gambar 4.18 adalah tampilan sms yang diterima pada ujicoba menempuh jarak Perum ITS U 160 sekitar 100 meter.



Gambar 4.18 Tampilan *Latitude* dan *Longitude Handphone*

Setelah menerima sms berupa data *latitude* dan *longitude*, data tersebut bisa kita lihat menjadi sebuah posisi pada *google map* dengan memasukan nilai *latitude* dan *longitudenya*. Gambar 4.19 berikut adalah posisi dimana menunjukan Perum ITS Blok U 160 dengan *error* posisi 100 meter.



Gambar 4.19 Tampilan *Google Map*

Berdasar gambar 4.19 menunjukan bahwa Perum ITS Blok U 160 Keputih Sukolilo ini berada pada posisi Kejawen Gebang yang berjarak 100 meter dari posisi yang ditentukan. Itu ditunjukkan oleh panah posisi berwarna hijau.

BAB V

PENUTUP

Bab Penutup berisi tentang kesimpulan-kesimpulan yang didapatkan selama proses pembuatan Tugas Akhir ini beserta saran-saran untuk perbaikan dan pengembangannya.

5.1 KESIMPULAN

Berdasarkan proses pengujian dan analisis sistem dapat ditarik kesimpulan sebagai berikut :

1. Pada sensor *loadcell*, tekanan pengereman berkisar antara -5 gram hingga 4989 gram, nilai minus ini dikarenakan peletakan sensor yang tidak sejajar.
2. Pada sensor *thermocouple*, panas mesin sepeda motor yang pernah diuji adalah sekitar 30-60 °C.
3. Pada sensor *rotary encoder*, terkadang tidak bisa mendeteksi kecepatan dikarenakan ada getaran dari sepeda motor.
4. Data pada sensor dan posisi dapat disimpan menggunakan *secure digital card* dengan format (.txt)
5. Konversi tampilan pembacaan *datalogger* menggunakan *microsoft excel*.
6. Dalam penggunaan GPS ini masih belum memiliki nilai akurasi yang baik, nilai *error* dari selisih jarak posisi sebenarnya, antara koordinat alat dan koordinat pembanding memiliki rata - rata *error* yang cukup besar yaitu untuk *latitude*-nya adalah 112,1 m, sedangkan *longitude*-nya adalah 300,1 m.
7. Data GPS dapat dikirim menggunakan *short message service* berupa *latitude* dan *longitude* selama 1 menit.
8. Untuk mengetahui posisi sepeda motor, proses input posisi GPS masih manual dengan menginputkan posisi dari sms ke *google map*.

5.2 SARAN

Saran yang dapat diajukan untuk penelitian lebih lanjut mengenai topik ini adalah :

1. Pengujian atau *trial and error* kotak hitam dan *transponder* ini lebih baik dilakukan berulang kali, karena terkadang *error* terdapat pada pemasangan sensor pada kotak hitam dan *transponder* ini, karena kotak hitam dan *transponder* yang berukuran kecil menjadikan pemasangan sensor dan pengkabelan menjadi tidak sempurna.
2. Sistem pengiriman data oleh SMS bisa diganti dengan GPRS *Wireless Network*, sehingga bisa secara otomatis menunjukkan posisi berupa *map* dan tidak manual untuk mengetahui posisi sepeda motor.
3. Untuk konversi tampilan, bisa lebih baik menggunakan *Visual Basic* agar lebih menarik.
4. Pengujian sensor bisa dilakukan saat kondisi motor berjalan dengan jarak tempuh dan waktu yang lama, misal jarak 200 Km dengan waktu 4 jam.
5. GPS yang digunakan bisa lebih bagus, agar nilai *error* dari posisi yang disajikan lebih kecil dan akurat.

DAFTAR PUSTAKA

- [1] Enrique Ramos Melgar, *Arduino and Kinect Project*, Friends of Apress, New York, 2012
- [2], Kalibrasi *Thermocouple*, URL: http://pyromation.com/downloads/data/emfk_c.pdf, 8 Januari 2014.
- [3], *Thermocouple type K*, URL: <http://thermocoupleinfo.com>, 8 Januari 2014.
- [4], *Loadcell 5kg*, URL: <http://robotshop.com/en/micro-load-cell-5-kg.html>, 8 Januari 2014.
- [5], *Arduino loadcell/ scale*, URL: <http://www.instructables.com/id/Arduino-Load-Cell-Scale/?ALLSTEPS>, 8 Januari 2014.
- [6], *Rotary Encoder Pengukuran* URL: <http://digilib.its.ac.id/.../ITS-paper-23036-2210105037-Pap...>, 8 Januari 2014.
- [7], *Prinsip Kerja GPS*, URL: <http://radenseta.blogspot.com/2012/10/prinsip-kerja-gps-global-positioning.html>, 8 Januari 2014.
- [8], *GPS U-Blox Neo 6M Module*, URL: www.u-blox.com/en/gps-modules/pvt-modules/.../neo-6-family.html, 8 Januari 2014.
- [9], *SMS Gateway*, URL: <http://informatika.web.id/category/sms-gateway>, 10 Januari 2014
- [10], *Modem Wavecom Fastrack*, URL: <http://itmbali.blogspot.com/2012/11/modem-wavecom-fastrack-m1306b-pp-usb.html>, 10 Januari 2014.
- [11], *Pengetahuan Dasar AT Command Modem GSM*, URL: <http://pccontrol.wordpress.com/2012/02/13/pengetahuan-dasar-mengirim-menerima-sms-melalui-mikrokontroler-dgn-at-command-modem-gsm/>, 10 Januari 2014.
- [12], *Real Time Clock*, URL: <http://indonetwork.co.id/group+143444/real-time-clock-rtc.htm>, 10 Januari 2014.
- [13], *Flowchart Shapes*, [msdn.microsoft.com/en-us/library/ee264487\(v=bts.10\).aspx](http://msdn.microsoft.com/en-us/library/ee264487(v=bts.10).aspx), 10 Januari 2014.

[Halaman ini sengaja dikosongkan]

DAFTAR RIWAYAT HIDUP



Nama : Rudy Arif Prasajo
TTL : Pekalongan, 19 Februari 1993
Kelamin: Laki-laki
Agama : Islam
Alamat : Binagriya Blok C Jalan Delima
No. 361 Kota Pekalongan
Telp/HP : 085642778048
E-mail : ruddyprasajo@gmail.com

RIWAYAT PENDIDIKAN

1. 1999 – 2005 : SD Negeri Medono 07 Pekalongan
2. 2005 – 2008 : SMP Negeri 2 Pekalongan
3. 2008 – 2011 : SMA Negeri 1 Pekalongan
4. 2011 – 2014 : D3 Teknik Elektro Komputer Kontrol FTI - ITS

PENGALAMAN KERJA

1. Kerja Praktek di PT. Telkom *Civil Mechanical Electrical* (CME)
Kebalen Surabaya Jawa Timur

PENGALAMAN ORGANISASI

1. Staf Propaganda, Aksi dan Pewacanaan Kementerian Sosial Politik (KEMENSOSPOL BEM ITS) 2012 – 2013
2. Staf Departemen Dalam Negeri (DAGRI HIMAD3TEKTRO) FTI - ITS 2012 – 2013
3. Kadept Departemen Dalam Negeri (DAGRI HIMAD3TEKTRO) FTI - ITS 2013 – 2014
4. UKM Lembaga Pers Mahasiswa Satu Kosong (LPM 1.0) ITS 2012 – 2013
5. Ketua Arek Pekalongan di Surabaya (APEL 2012) – 2014
6. Sekertaris Honda Mega-GL Pro Club Pekalongan (HMGC) 2008-2011

Halaman Ini Sengaja Dikosongkan

DAFTAR RIWAYAT HIDUP



Nama : Ghosyiya Aisisar
TTL : Madiun , 18 Desember 1992
Kelamin: Laki-laki
Agama : Islam
Alamat : Desa Bagi Rt 08 Rw 02 Kec.
Kab. Madiun
Telp/HP : 085732277984
E-mail : ghosyiya@gmail.com

RIWAYAT PENDIDIKAN

1. 1999 – 2005 : SD 09 Negeri Endrakila Madiun
2. 2005 – 2008 : SMP Negeri 1 Madiun
3. 2008 – 2011 : SMA Negeri 6 Madiun
4. 2011 – 2014 : D3 Teknik Elektro Komputer Kontrol FTI - ITS

PENGALAMAN KERJA

1. Kerja Praktek di PT. ANGKASA PURA I BANDARA JUANDA Surabaya Jawa Timur

PENGALAMAN ORGANISASI

1. Staff Pendidikan Departemen Kewirausahaan Mahasiswa (KWU HIMAD3TEKTRO) FTI - ITS 2012 – 2013
2. Staff Pendidikan Departemen Kewirausahaan Mahasiswa (KWU HIMAD3TEKTRO) FTI - ITS 2013 – 2014

Halaman Ini Sengaja Dikosongkan

LAMPIRAN A

LISTING PROGRAM

a. Program

```
//include card-web
#include <SPI.h>
#include <Ethernet.h>
#include <SD.h>
const int chipSelect = 53;
#include <LiquidCrystal.h>
#include "RTClib.h"

RTC_DS1307 rtc;

LiquidCrystal lcd(8, 9, 3, 5, 6,7);
//RS --> digital pin 8
//E --> digital pin 9
//DB4 --> digital pin 3
//DB5 --> digital pin 5
//DB6 --> digital pin 6
//DB7 --> digital pin 7
//RS, E, DB4, DB5, DB6 and DB7 are pins of LCD
//RW to ground

//variable loadcell
float berat=0, berattotal=0, beratfix=0;

//rtc
#include "Wire.h"
#define DS1307_ADDRESS 0x68

//suhu
float suhu=0;
```

```

//rotary
int i=0, data[1000];
float derajat=0, counter=0, radb=0;

// Enter a MAC address and IP address for your controller below.
// The IP address will be dependent on your local network:
byte mac[] = {
  0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
IPAddress ip(192,168,1,17);

// Initialize the Ethernet server library
// with the IP address and port you want to use
// (port 80 is default for HTTP):
EthernetServer server(80);

int menit;
int menitsimpan;
int menitcount = 0;
const int sentenceSize = 80;
char sentence[sentenceSize];
;

char field[20];
String lati; String lat;
String longti; String longi;
String bujur;
String lintang;
String penambahanlintang;
int simpantambah;
String penambahanbujur;
int simpantambah2;
String penambahwaktu;
int waktubenar;
String waktu;
String time;
String isisms;

void setup() {

```

```

lcd.begin(8,2);
lcd.clear();
// Open serial communications and wait for port to open:
Serial.begin(9600);

Serial1.begin(9600);// GPS
delay(800);
Serial2.begin(9600);// SMS

Serial2.print("AT");      // tentukan nomor yang dituju
Serial2.write(10);        // kirim karakter tanda kutip (")
delay(5000);

Serial2.print("AT+CMGF=1"); // tentukan nomor yang dituju
Serial2.write(10);        // kirim karakter tanda kutip (")
delay(5000);

Wire.begin();//rtc
while (!Serial) {
  ; // wait for serial port to connect. Needed for Leonardo only
}

//sd card
Serial.print("Initializing SD card...");
// make sure that the default chip select pin is set to
// output, even if you don't use it:
pinMode(53, OUTPUT);

// see if the card is present and can be initialized:
if (!SD.begin(chipSelect)) {
  Serial.println("Card failed, or not present");
  // don't do anything more:
  return;
}
Serial.println("card initialized.");

// start the Ethernet connection and the server:

```

```

Ethernet.begin(mac, ip);
server.begin();
Serial.print("server is at ");
Serial.println(Ethernet.localIP());
Serial.println();

lcd.setCursor(0, 0);
lcd.print(">System<");
lcd.setCursor(0, 1);
lcd.print("...OK...");

File dataFile1 = SD.open("datalog.txt", FILE_WRITE);

dataFile1.print("Tanggal;\tWaktu;\tTemperature;\tKecepatan;\tTekanan;\tLat;\tLong;");
dataFile1.println();
dataFile1.close();

Serial.println("Tanggal\tWaktu\t\tTemperature\tKecepatan\tTekanan\t\tLat\t\tLong");

    delay(500);
}

byte bcdToDec(byte val) {
// Convert binary coded decimal to normal decimal numbers
return ( (val/16*10) + (val%16) );
}

void loop() {

    static int j = 0;
    if (Serial1.available()) {
        char ch = Serial1.read();
        if (ch != '\n' && j < sentenceSize)
            {

```

```

        sentence[j] = ch;
        j++;
    }
    else
    {
        sentence[j] = '\0';
        j = 0;
        displayGPS();
    }
} // read from port 0, send to port 1:

float sensorsuhu = analogRead(A2);
float analogloadcell = analogRead(A1);
int sensorValue = analogRead(A0);
berat=(analogloadcell-16)/849*5000; //mapping analog dan maksimal
loadcell
berattotal=berattotal+berat; //akumulasi counter loadcell dalam 1 detik

i++; //variable untuk delay dan penyimpanan data rotary k array
data[i]=sensorValue; //masukkan data rotary k array
//melakukan counter rotary saat ada leading maupun raising
if((data[i]-data[i-1])>=500||(data[i-1]-data[i])>=500)
{
    counter++;
}
derajat=((counter/2)/8*360)/360*50*3.14/100000*3600; //nilai lubang
ada 20 sedangkan jumlah putaran ada 360 derajat, 2 berarti 1 lubang ada
leading & raising

//semua program penampilan maupun penyimpanan data dilakukan tiap
1-2 detik,
//sehingga tidak mengganggu counter loadcell maupun rotary
if(i==999){

    //termocouple
    //suhu=sensorsuhu/8.784;
    //suhu=sensorsuhu*150/1023;
    //suhu=sensorsuhu;

```

```

suhu=sensorsuhu*30/260;

beratfix=berattotal/1000; //rata-rata berat dalam 1 detik yang
tercounter 1000 kali
berattotal=0; //reset nilai dari sensor

//rtc begin
Wire.beginTransmission(DS1307_ADDRESS);
byte zero = 0x00;
Wire.write(zero); Wire.endTransmission();
Wire.requestFrom(DS1307_ADDRESS, 7);

//penampung variable rtc
int second = bcdToDec(Wire.read());
int minute = bcdToDec(Wire.read());
int hour = bcdToDec(Wire.read() & 0b111111); //24 hour time
int weekday = (Wire.read()); //hari
int day = bcdToDec(Wire.read());
int month = bcdToDec(Wire.read());
int year = bcdToDec(Wire.read());

//data yang akan masuk sdcard harus berbentuk string
String dataString = "";
String dataString1 = "";
String dataString2 = "";
String dataString3 = "";
String dataString4 = "";
String dataString5 = "";

//variable thermocouple

radb= derajat/360*100;

//menampung string yang akan masuk ke sdcard
dataString += String ((int)day)+"/"+ String ((int)month)+"/"+ String
((int)year)+";t"+ String ((int)hour)+":"+ String ((int)minute)+":"+
String ((int)second); //data rtc

```

```

//dataString1 += "C = " +
String((int)thermocouple.readCelsius()+", "+String((int)datacb) + " ;
"+"F = "+String((int)thermocouple.readFahrenheit()) + ", " +
String((int)datafb); //data thermocouple
  dataString1 += "Suhu = " + String((int)suhu)+" C" ; //data
thermocouple
  dataString2 += String ((int)derajat)+" "+"KM/Jam" ; //data rotary
  dataString3 += String ((int)beratfix)+" "+"Grams"; //data loadcell
  dataString4 += String (lati); //Lat
  dataString5 += String (longti); //Long

  File dataFile = SD.open("datalog.txt", FILE_WRITE); //membuka dan
menulis note pada sccard

// if the file is available, write to it:
if (dataFile) {
  dataFile.print(dataString);dataFile.print(";");
  dataFile.print(dataString1);dataFile.print(";");
  dataFile.print(dataString2);dataFile.print(";");
  dataFile.print(dataString3);dataFile.print(";");
  dataFile.print(dataString4);dataFile.print(";");
  dataFile.print(dataString5);dataFile.print(";");
  dataFile.println();
  dataFile.close();

  // print to the serial port too:
  Serial.print(dataString);Serial.print("\n");
  Serial.print(dataString1);Serial.print("\n");
  Serial.print(dataString2);Serial.print("\n");
  Serial.print(dataString3);Serial.print("\n");
  Serial.print(dataString4);Serial.print("\n");
  Serial.print(dataString5);Serial.print("\n");
  Serial.println();
}
// if the file isn't open, pop up an error:
else {
  Serial.println("error opening datalog.txt");
}

```



```

//ethernet begin
EthernetClient client = server.available();
if (client) {
  Serial.println("new client");
  // an http request ends with a blank line
  boolean currentLineIsBlank = true;
  while (client.connected()) {
    if (client.available()) {
      char c = client.read();
      Serial.write(c);
      // if you've gotten to the end of the line (received a newline
      // character) and the line is blank, the http request has ended,
      // so you can send a reply
      if (c == '\n' && currentLineIsBlank) {
        // send a standard http response header
        client.println("HTTP/1.1 200 OK");
        client.println("Content-Type: text/html");
        client.println("Connection: close"); // the connection will be
        closed after completion of the response
        client.println("Refresh: 1"); // refresh the page automatically
        every 5 sec
        client.println();
        client.println("<!DOCTYPE HTML>");
        client.println("<html>");

        //menampilkan data rtc ke web
        client.print(day);client.print("/");
        client.print(month);client.print("/");
        client.print(year);client.print(" ");
        client.print(hour);client.print(":");
        client.print(minute);client.print(":");
        client.println(second);
        client.println("<br />");

        //menampilkan data rotary ke web
        client.print(derajat);
        client.print(" KM/Jam");

```

```

client.print(" ");
client.println("<br />");

// menampilkan data loadcell ke web
client.print(beratfix);
client.print(" Grams");
client.println("<br />");

client.println("</html>");
break;
}
if (c == '\n') {
    // you're starting a new line
    currentLineIsBlank = true;
}
else if (c != '\r') {
    // you've gotten a character on the current line
    currentLineIsBlank = false;
}
}
}
// give the web browser time to receive the data
delay(1);
// close the connection:
client.stop();
Serial.println("client disconnected");

}
counter=0; //reset nilai rotary
i=0; //reset array rotary
}
delay(1);
}

void displayGPS(){

```

```

getField(field, 0);
if (strcmp(field, "$GPGGA") == 0)
{
    getField(field, 1);
    waktu = field;
    penambahwaktu = waktu.substring(0,2);
    waktubenar = penambahwaktu.toInt();
    waktubenar = waktubenar + 7;
    if (waktubenar > 24)
    {
        waktubenar = waktubenar - 24;
    }
    String stringku = String(waktubenar);
    waktu = " At = " + stringku + ":" + waktu.substring(2,4) + ":" +
waktu.substring(4,6);
    //Serial.println(waktu);
    time = field;
    time = time.substring(2,4);
    menit = time.toInt();
    //Serial.println(menit);
    //Serial.println(menitcount);

    getField(field, 2); // Koordinat lintang
    lati = field;
    getField(field, 3); // lintang selatan atau utara
    lintang = field;
    if (lintang == "S")
    {
        penambahanlintang = lati.substring(2,4);
        simpantambah = penambahanlintang.toInt();
        simpantambah = simpantambah + 12; //2900
        String lintangbenar = String(simpantambah);
        lati = " Lat = -" + lati.substring(1,2) + "," + simpantambah +
lati.substring(5,9);
        // + lati.substring(3,4)
        //Serial.print(lati);
    }
}

```

```

else if (lintang == "U")
{
    penambahanlintang = lati.substring(2,4);
    simpantambah = penambahanlintang.toInt();
    simpantambah = simpantambah + 12; //2900
    String lintangbenar = String(simpantambah);
    lati = " Lat = " + lati.substring(1,2) + "," + simpantambah +
lati.substring(5,9);
    //Serial.print(lati);
}

getField(field, 4); // koordinat
longti = field;
getField(field, 5); // bujur timur atau barat
bujur = field;
if (bujur == "E")
{
    penambahanbujur = longti.substring(3,5);
    simpantambah2 = penambahanbujur.toInt();
    simpantambah2 = simpantambah2 + 34; //8700
    String bujurbenar = String(simpantambah2);
    longti = " Long = " + longti.substring(0,3) + "," +
simpantambah2+longti.substring(6,10);
    //+ longti.substring(3,5)
    //Serial.println(longti);
}
else if (bujur == "W")
{
    penambahanbujur = longti.substring(3,5);
    simpantambah2 = penambahanbujur.toInt();
    simpantambah2 = simpantambah2 + 34; //8700
    String bujurbenar = String(simpantambah2);
    longti = " Long = -" + longti.substring(0,3) + "," + simpantambah2 +
longti.substring(6,10);
    //longti.substring(3,5)
    //Serial.println(longti);
}
if (menitsimpan != menit)

```

```

{
    menitsimpan = menit;
    menitcount = menitcount + 1;
}

if (menitcount == 1)
{

    Serial.println("mulai sms");
    String nope="+6285642778048"; //+6285642778048
+6283857105060
    isisms = "Motor Cycle Position on" + lati + longti + waktu;

    Serial2.print("AT+CMGS="); // tentukan nomor yang dituju
    Serial2.write(34); // kirim karakter tanda kutip (")
    Serial2.print(nope); // lengkapi *** dengan nomor telepon yg
dituju
    Serial2.write(34); // kirim karakter tanda kutip (")
    Serial2.write(13); // kirim karakter tanda kutip (")
    Serial2.write(10); // kirim karakter tanda kutip (")
    delay(500);

    Serial2.print(isisms); // tentukan nomor yang dituju
    Serial2.write(26); // kirim karakter tanda kutip (")
    delay(500);

    menitcount = 0;

}

}
}

void getField(char* buffer, int index)
{
    int sentencePos = 0;
    int fieldPos = 0;

```

```
int commaCount = 0;
while (sentencePos < sentenceSize)
{
    if (sentence[sentencePos] == ',')
    {
        commaCount ++;
        sentencePos ++;
    }
    if (commaCount == index)
    {
        buffer[fieldPos] = sentence[sentencePos];
        fieldPos ++;
    }
    sentencePos ++;
}
buffer[fieldPos] = '\0';
}
```

[Halaman ini sengaja dikosongkan]


LAMPIRAN B FLOWCHART

Flowchart Shapes


BizTalk Orchestration Designer provides eight **Flowchart** shapes that are available on the Flowchart stencil. These shapes are used to describe your business process. The eight shapes on the **Flowchart** stencil can be used to describe the structure and meaning of XLANG schedules. XLANG is an XML-based language that describes business-process interactions.

The following table lists and describes the **Flowchart** shapes.

Shape name	Description
------------	-------------

Begin 	The Begin shape is not available on the Flowchart stencil, it cannot be deleted, and you cannot create additional Begin shapes. The Begin shape represents the start of an XLANG schedule drawing. The Begin shape is created automatically on each Business Process , Compensation for Transaction , and On Failure of Transaction page. The Begin shape that appears on the Business Process page has configurable properties. The Begin shapes that appear on the Compensation for Transaction page and the On Failure of Transaction page do not have configurable properties. You cannot enclose a Begin shape within a transaction. The business process sequence must flow from the Begin shape to the first Flowchart shape in your drawing.
---------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Action 	The Action shape represents a process that receives a message from a port or sends a message to a port. The send or receive action can be synchronous or asynchronous, depending on the component or implementation to which the port is bound.
------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Decision 	The Decision shape represents a process that evaluates one or more rules sequentially. This shape has one inbound flow and one or more outbound flows. Each outbound flow is associated with a rule that evaluates to TRUE or FALSE. The first rule that evaluates to TRUE determines which outbound flow is followed in the
--------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

business process. The sequence of the business process follows the flow from the first rule that evaluates to TRUE. If no rules evaluate to TRUE, the Else flow is followed. The **Decision** shape must contain at least one rule. Each rule must contain a script expression.

The **While** shape contains one rule and represents a process that can be repeated. If the rule evaluates to TRUE, the flow from the rule is followed to completion and then it repeats. If the rule evaluates to FALSE, the Continue flow is followed. When the business process sequence flows from a rule in a **While** shape, the sequence must conclude in a single **End** shape.

You can also configure the preservation of state for a while loop. By right-clicking a **While** shape and clicking **Properties**, you can display the **While Properties** dialog box. In the **State persistence** area, you can choose **Yes** to save the messages used in each loop iteration as XLANG schedule state. If the while loop is part of a transaction that fails, an **On Failure of Transaction** or **Compensation for Transaction** page will be called for each completed loop iteration. If you choose **No**, only messages used in the latest loop iteration will be saved as XLANG schedule state. If the while loop is part of a transaction that fails, an **On Failure of Transaction** or **Compensation for Transaction** page will only be called once.

When a business process sequence flows to a **While** shape that is within a nested transaction that fails, the messages and ports that have been created on the **On Failure of Transaction** or **Compensation for Transaction** page for the nested transaction will not be available to the **On Failure of Transaction** page for the outer transaction. Design the On Failure or Compensation code to force the collected messages to flow normally out of the nested transaction.





The **Fork** shape introduces concurrency into a business process. One flow can enter a fork, and as many as 64 flows can leave a fork. Each flow that leaves a fork is executed concurrently. All of the business process sequences that flow from a single **Fork** shape must connect to a single **Join** shape or terminate in an **End** shape.



The **Join** shape synchronizes concurrent flows in a business process. As many as 64 flows can enter a **Join** shape, but only one flow leaves a **Join** shape. The logical operators **AND** and **OR** are used to determine how to synchronize the flows. You can set the following **Join** properties:

- **OR.** Enables the first flow that arrives to continue. The other flows will continue to execute.
- **AND.** Synchronizes all incoming flows before the outbound flow can continue.

The **Transaction** shape represents a collection of actions that are either all executed, or else none are. There are three types of transactions:

- Short-lived transactions
- Long-running transactions
- Timed transactions that are long-running

Transactions are used to make an application more reliable and to simplify error handling in large applications.



The **Transaction** shape is limited to a single path in, and a single path out. The **Transaction** shape cannot contain **End** shapes. Long-running transactions can contain nested transactions; however, short-lived transactions cannot contain nested transactions. Transaction retry attempts are only permitted for short-lived transactions. If you enclose part of your business process within a short-lived **Transaction** shape, you can configure the transaction **Retry count** property. If the short-lived transaction fails, it is retried for the number of times that you have specified.

By defining transaction properties, you can make available either the **Compensation for Transaction** page (for nested transactions) or the **On Failure of Transaction** page. On either page, you can model the error-handling processes that are specific to the transaction.

Note

- If the borders of an inner **Transaction** shape overlap with any of the borders of the outer **Transaction** shape, the inner transaction will not be nested. Do not allow the borders of an inner transaction to overlap with the borders of an outer transaction.

End

End



The **End** shape represents the completion of one process flow. One drawing can use multiple **End** shapes if the drawing includes **Decision**, **While**, or **Fork** shapes.

Abort

Abort



The **Abort** shape terminates execution within a transaction group. This enables either an **On Failure of Transaction** or **Compensation for Transaction** error-handling page, or else it retries the transaction.

Notes

- For the **Compensation for Transaction** and **On Failure of Transaction** pages, the parameter *Transaction* is replaced with the name that you give to the associated transaction on the **Business Process** page.
- Shape names must meet certain naming conventions. The following conventions apply to transaction shapes, port shapes, messages, rules, and fields:
 - The name must be a valid XML token name. For more information about XML tokens, go to the W3C Web site (www.w3c.org).
 - The name cannot begin with underscores (___).
 - The name cannot include colons (:).

- The name length must be less than or equal to 32 characters.
- Constant names and message names cannot begin with a numeric character.
- Transaction names in single-byte character sets must be less than or equal to 16 characters in length. Transaction names in double-byte character sets must be less than or equal to 8 characters in length.
- Actions are exempt from all naming conventions except the 32-character size limit.
- You cannot name any of the following shapes: **Abort**, **Begin**, **Decision**, **End**, **Fork**, **Join**, and **While**.

[Halaman ini sengaja dikosongkan]

LAMPIRAN C

DATASHEET



DS1307

64 x 8 Serial Real-Time Clock

www.maxim-ic.com

FEATURES

- Real-time clock (RTC) counts seconds, minutes, hours, date of the month, month, day of the week, and year with leap-year compensation valid up to 2100
- 56-byte, battery-backed, nonvolatile (NV) RAM for data storage
- Two-wire serial interface
- Programmable squarewave output signal
- Automatic power-fail detect and switch circuitry
- Consumes less than 500nA in battery backup mode with oscillator running
- Optional industrial temperature range: -40°C to +85°C
- Available in 8-pin DIP or SOIC
- Underwriters Laboratory (UL) recognized

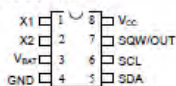
ORDERING INFORMATION

DS1307	8-Pin DIP (300-mil)
DS1307Z	8-Pin SOIC (150-mil)
DS1307N	8-Pin DIP (Industrial)
DS1307ZN	8-Pin SOIC (Industrial)

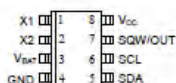
DESCRIPTION

The DS1307 Serial Real-Time Clock is a low-power, full binary-coded decimal (BCD) clock/calendar plus 56 bytes of NV SRAM. Address and data are transferred serially via a 2-wire, bi-directional bus. The clock/calendar provides seconds, minutes, hours, day, date, month, and year information. The end of the month date is automatically adjusted for months with fewer than 31 days, including corrections for leap year. The clock operates in either the 24-hour or 12-hour format with AM/PM indicator. The DS1307 has a built-in power sense circuit that detects power failures and automatically switches to the battery supply.

PIN ASSIGNMENT



DS1307 8-Pin DIP (300-mil)

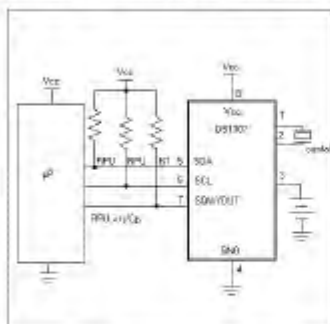


DS1307 8-Pin SOIC (150-mil)

PIN DESCRIPTION

V _{CC}	- Primary Power Supply
X1, X2	- 32.768kHz Crystal Connection
V _{BAT}	- +3V Battery Input
GND	- Ground
SDA	- Serial Data
SCL	- Serial Clock
SQW/OUT	- Square Wave/Output Driver

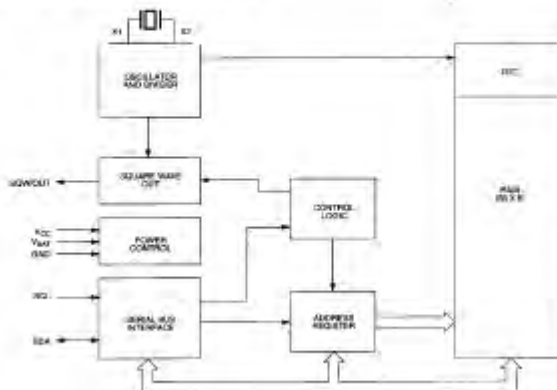
TYPICAL OPERATING CIRCUIT



OPERATION

The DS1307 operates as a slave device on the serial bus. Access is obtained by implementing a START condition and providing a device identification code followed by a register address. Subsequent registers can be accessed sequentially until a STOP condition is executed. When V_{CC} falls below $1.25 \times V_{BAT}$ the device terminates an access in progress and resets the device address counter. Inputs to the device will not be recognized at this time to prevent erroneous data from being written to the device from an out of tolerance system. When V_{CC} falls below V_{BAT} the device switches into a low-current battery backup mode. Upon power-up, the device switches from battery to V_{CC} when V_{CC} is greater than $V_{BAT} + 0.2V$ and recognizes inputs when V_{CC} is greater than $1.25 \times V_{BAT}$. The block diagram in Figure 1 shows the main elements of the serial RTC.

DS1307 BLOCK DIAGRAM Figure 1



2 of 12



INSTRUMENTATION AMPLIFIER With Precision Voltage Reference

FEATURES

- LOW QUIESCENT CURRENT: 480µA
- PRECISION VOLTAGE REFERENCE: 1.24V, 2.5V, 5V or 10V
- SLEEP MODE
- LOW OFFSET VOLTAGE: 250µV max
- LOW OFFSET DRIFT: 2µV/°C max
- LOW INPUT BIAS CURRENT: 20nA max
- HIGH CMR: 100dB min
- LOW NOISE: 38nV/√Hz at f = 1kHz
- INPUT PROTECTION TO ±40V
- WIDE SUPPLY RANGE
Single Supply: 2.7V to 36V
Dual Supply: ±1.35V to ±18V
- 18-PIN DIP AND SO-16 SOIC PACKAGES

DESCRIPTION

The INA125 is a low power, high accuracy instrumentation amplifier with a precision voltage reference. It provides complete bridge excitation and precision differential-input amplification on a single integrated circuit.

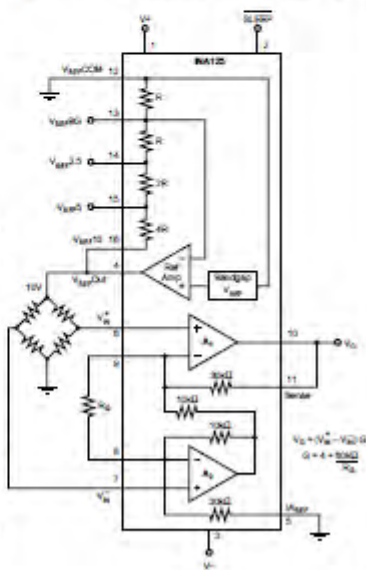
A single external resistor sets any gain from 4 to 10,000. The INA125 is laser-trimmed for low offset voltage (250µV), low offset drift (2µV/°C), and high common-mode rejection (100dB at G = 100). It operates on single (+2.7V to +36V) or dual (±1.35V to ±18V) supplies.

The voltage reference is externally adjustable with pin-selectable voltages of 2.5V, 5V, or 10V, allowing use with a variety of transducers. The reference voltage is accurate to ±0.5% (max) with ±25ppm/°C drift (max). Sleep mode allows shutdown and duty cycle operation to save power.

The INA125 is available in 16-pin plastic DIP and SO-16 surface-mount packages and is specified for the -40°C to +85°C industrial temperature range.

APPLICATIONS

- PRESSURE AND TEMPERATURE BRIDGE AMPLIFIERS
- INDUSTRIAL PROCESS CONTROL
- FACTORY AUTOMATION
- MULTI-CHANNEL DATA ACQUISITION
- BATTERY OPERATED SYSTEMS
- GENERAL PURPOSE INSTRUMENTATION



SPECIFICATIONS: $V_S = \pm 15V$

At $T_A = +25^\circ C$, $V_S = \pm 15V$, I_A common = $0V$, V_{REF} common = $0V$, and $R_L = 10k\Omega$, unless otherwise noted.

PARAMETER	CONDITIONS	INA125P, U			INA125PA, UA			UNITS
		MIN	TYP	MAX	MIN	TYP	MAX	
INPUT								
Offset Voltage, RTI								
Initial	$V_S = \pm 15V$ to $\pm 15V$, $G = 4$		150	1250		*	1500	μV
vs Temperature			± 0.25	12		*	15	$\mu V/^\circ C$
vs Power Supply			15	120		*	150	$\mu V/V$
Long-Term Stability				10.2		*		$\mu V/mo$
Impedance, Differential				$10^{11} 2$		*		ΩpF
Common-Mode			$10^{11} 2$		*		ΩpF	
Safe Input Voltage			See Text	140		*	V	
Input Voltage Range						*		
Common-Mode Rejection	$V_{CM} = -10.7V$ to $+10.2V$							
	$G = 4$	78	84		72	*	dB	
	$G = 10$	88	94		80	*	dB	
	$G = 100$	100	114		90	*	dB	
	$G = 500$	100	114		90	*	dB	
BIAS CURRENT	$V_{CM} = 0V$		10	25		*	50	nA
vs Temperature			160			*		$\mu A/^\circ C$
Offset Current			10.5	12.5		*	15	nA
vs Temperature			10.5			*		$\mu A/^\circ C$
NOISE, RTI	$R_N = 0\Omega$							
Voltage Noise, $f = 10Hz$			40			*		nV/\sqrt{Hz}
$f = 100Hz$			36			*		nV/\sqrt{Hz}
$f = 1kHz$			33			*		nV/\sqrt{Hz}
$f = 0.1Hz$ to $10Hz$			0.8			*		$\mu V/p$
Current Noise, $f = 10Hz$			170			*		nA/\sqrt{Hz}
$f = 1kHz$			56			*		nA/\sqrt{Hz}
$f = 0.1Hz$ to $10Hz$			5			*		pA/p
GAIN								
Gain Equation		4	$4 + 53k/R_L$	10,000	*	*		V/V
Range of Gain	$V_C = -14V$ to $+13.3V$				*	*		V/V
Gain Error	$G = 4$		± 0.01	± 0.075		*	± 0.1	%
	$G = 10$		± 0.03	± 0.3		*	± 0.5	%
	$G = 100$		± 0.06	± 0.5		*	± 1	%
	$G = 500$		± 0.1			*		%
Gain vs Temperature	$G = 4$		± 1	115		*	*	ppm/°C
	$G = 40$		125	1100		*	*	ppm/°C
Nonlinearity	$V_C = -14V$ to $+13.3V$					*		% of FS
	$G = 4$		± 0.0004	± 0.002		*	± 0.004	% of FS
	$G = 10$		± 0.0004	± 0.002		*	± 0.004	% of FS
	$G = 100$		± 0.001	± 0.01		*	*	% of FS
	$G = 500$		± 0.002			*	*	% of FS
OUTPUT								
Voltage: Positive		$(V^+) - 1.7$	$(V^+) - 0.9$		*	*		V
Negative		$(V^-) + 1$	$(V^-) + 0.4$		*	*		V
Load Capacitance Stability			1000		*	*		pF
Short-Circuit Current			-8 ± 12		*	*		mA
VOLTAGE REFERENCE								
Accuracy	$V_{REF} = +2.5V, +5V, +10V$		± 0.15	± 0.5		*	± 1	%
vs Temperature	$I_L = 0$		110	136		*	± 100	ppm/°C
vs Power Supply, V_S	$V_S = (V_{REF} + 1.25V)$ to $+36V$		120	150		*	± 100	ppm/V
vs Load	$I_L = 0$ to $5mA$		3	75		*	*	ppm/mA
Dropout Voltage, $(V^+) - V_{REF}$	Ref Load = $2k\Omega$	1.25	1		*	*		V
Bandgap Voltage Reference			1.24			*		V
Accuracy	$I_L = 0$		± 0.5			*		%
vs Temperature	$I_L = 0$		118			*		ppm/°C

The information provided herein is believed to be reliable, however, BURR-BROWN assumes no responsibility for inaccuracies or omissions. BURR-BROWN assumes no responsibility for the use of this information, and all use of such information shall be entirely at the user's own risk. Prices and specifications are subject to change without notice. No patent rights or licenses to any of the circuits described herein are implied or granted to any third party. BURR-BROWN does not authorize or warrant any BURR-BROWN product for use in life support devices and/or systems.

APPLICATION INFORMATION

Figure 1 shows the basic connections required for operation of the INA125. Applications with noisy or high impedance power supplies may require decoupling capacitors close to the device pins as shown.

The output is referred to the instrumentation amplifier reference ($I_{A_{REF}}$) terminal which is normally grounded. This must be a low impedance connection to assure good common-mode rejection. A resistance of 12Ω in series with the $I_{A_{REF}}$ pin will cause a typical device to degrade to approximately 80dB CMR ($G = 4$).

Connecting V_{REF_OUT} (pin 4) to one of the four available reference voltage pins (V_{REF_BG} , $V_{REF_2.5}$, V_{REF_5} , or V_{REF_10}) provides an accurate voltage source for bridge applications.

For example, in Figure 1 V_{REF_OUT} is connected to V_{REF_10} thus supplying 10V to the bridge. It is recommended that V_{REF_OUT} be connected to one of the reference voltage pins even when the reference is not being utilized to avoid saturating the reference amplifier. Driving the SLEEP pin LOW puts the INA125 in a shutdown mode.

SETTING THE GAIN

Gain of the INA125 is set by connecting a single external resistor, R_G , between pins 8 and 9:

$$G = 4 + \frac{60k\Omega}{R_G} \quad (1)$$

Commonly used gains and R_G resistor values are shown in Figure 1.

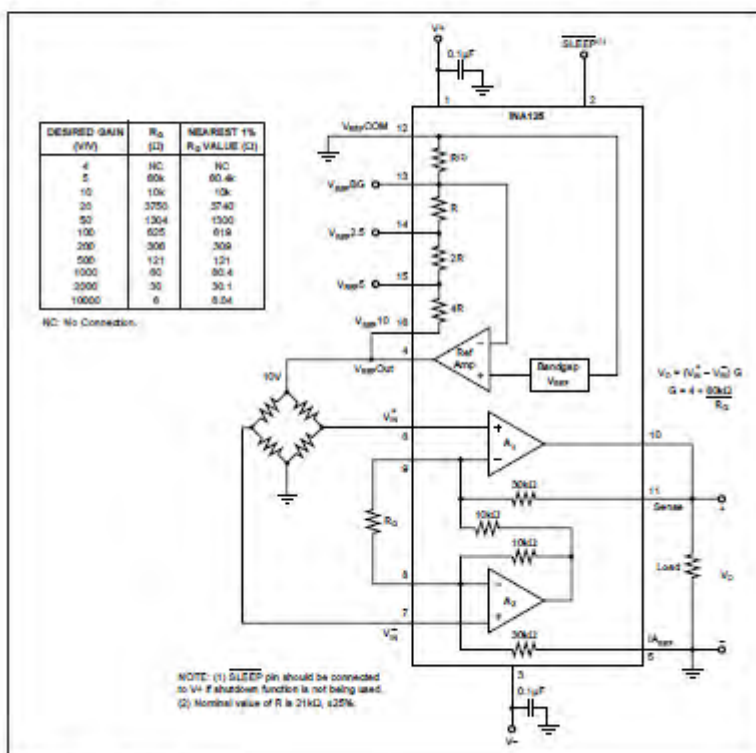


FIGURE 1. Basic Connections.



The $60k\Omega$ term in equation 1 comes from the internal metal film resistors which are laser trimmed to accurate absolute values. The accuracy and temperature coefficient of these resistors are included in the gain accuracy and drift specifications of the INA125.

The stability and temperature drift of the external gain setting resistor, R_G , also affects gain. R_G 's contribution to gain accuracy and drift can be directly inferred from the gain equation (1). Low resistor values required for high gain can make wiring resistance important. Sockets add to the wiring resistance, which will contribute additional gain error (possibly an unstable gain error) in gains of approximately 100 or greater.

OFFSET TRIMMING

The INA125 is laser trimmed for low offset voltage and offset voltage drift. Most applications require no external offset adjustment. Figure 2 shows an optional circuit for trimming the output offset voltage. The voltage applied to the IA_{REF} terminal is added to the output signal. The op amp buffer is used to provide low impedance at the IA_{REF} terminal to preserve good common-mode rejection.

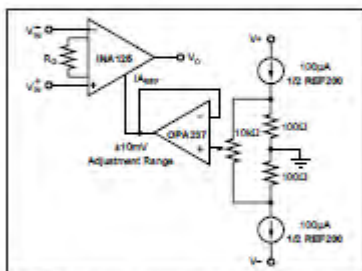


FIGURE 2. Optional Trimming of Output Offset Voltage.

INPUT BIAS CURRENT RETURN

The input impedance of the INA125 is extremely high—approximately $10^{11}\Omega$. However, a path must be provided for the input bias current of both inputs. This input bias current flows out of the device and is approximately 10nA. High input impedance means that this input bias current changes very little with varying input voltage.

Input circuitry must provide a path for this input bias current for proper operation. Figure 3 shows various provisions for an input bias current path. Without a bias current path, the inputs will float to a potential which exceeds the common-mode range, and the input amplifiers will saturate.

If the differential source resistance is low, the bias current return path can be connected to one input (see the thermocouple example in Figure 3). With higher source impedance, using two equal resistors provides a balanced input with possible advantages of lower input offset voltage due to bias current and better high frequency common-mode rejection.

INPUT COMMON-MODE RANGE

The input common-mode range of the INA125 is shown in the typical performance curves. The common-mode range is limited on the negative side by the output voltage swing of A_2 , an internal circuit node that cannot be measured on an external pin. The output voltage of A_2 can be expressed as:

$$V_{A2} = 1.3V_{IC} - (V_{IN}^+ - V_{IN}^-) (10k\Omega/R_G)$$

(voltages referred to IA_{REF} terminal, pin 5)

The internal op amp A_2 is identical to A_1 , its output swing is limited to approximately 0.8V from the positive supply and 0.25V from the negative supply. When the input common-mode range is exceeded (A_2 's output is saturated), A_1 can still be in linear operation, responding to changes in the non-inverting input voltage. The output voltage, however, will be invalid.

PRECISION VOLTAGE REFERENCE

The on-board precision voltage reference provides an accurate voltage source for bridge and other transducer applications or ratiometric conversion with analog-to-digital converters. A reference output of 2.5V, 5V or 10V is available by connecting V_{REF_OUT} (pin 4) to one of the V_{REF} pins ($V_{REF2.5}$, V_{REF5} , or V_{REF10}). Reference voltages are laser-trimmed for low initial error and low temperature drift. Connecting V_{REF_OUT} to V_{REF_BG} (pin 13) produces the bandgap reference voltage ($1.24V \pm 0.5\%$) at the reference output.

Positive supply voltage must be 1.25V above the desired reference voltage. For example, with $V^+ = 2.7V$, only the 1.24V reference (V_{REF_BG}) can be used. If using dual supplies V_{REF_COM} can be connected to V^- , increasing the

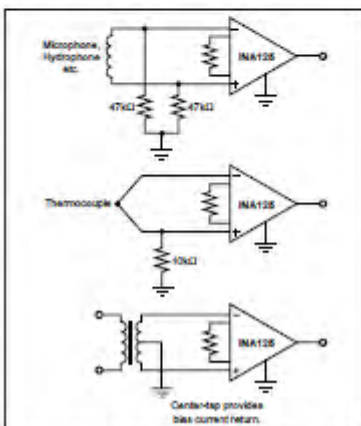


FIGURE 3. Providing an Input Common-Mode Current Path.

NEO-6

u-blox 6 GPS Modules

Data Sheet

Abstract

Technical data sheet describing the cost effective, high-performance u-blox 6 based NEO-6 series of GPS modules, that brings the high performance of the u-blox 6 positioning engine to the miniature NEO form factor.

These receivers combine a high level of integration capability with flexible connectivity options in a small package. This makes them perfectly suited for mass-market end products with strict size and cost requirements.



16.0 x 12.2 x 2.4 mm

www.u-blox.com

Document Information

Title	NEO-6
Subtitle	u-blox 6 GPS Modules
Document type	Data Sheet
Document number	GPS.G6-HW-09005-E

Document status
Document status information

Objective Specification	This document contains target values. Revised and supplementary data will be published later.
Advance Information	This document contains data based on early testing. Revised and supplementary data will be published later.
Preliminary	This document contains data from product verification. Revised and supplementary data may be published later.
Released	This document contains the final product specification.

This document applies to the following products:

Name	Type number	ROM/FLASH version	PCN reference
NEO-6G	NEO-6G-0-001	ROM7.03	LIBX-TN-11047-1
NEO-6Q	NEO-6Q-0-001	ROM7.03	LIBX-TN-11047-1
NEO-6M	NEO-6M-0-001	ROM7.03	LIBX-TN-11047-1
NEO-6P	NEO-6P-0-000	ROM6.02	N/A
NEO-6V	NEO-6V-0-000	ROM7.03	N/A
NEO-6T	NEO-6T-0-000	ROM7.03	N/A

This document and the use of any information contained therein, is subject to the acceptance of the u-blox terms and conditions. They can be downloaded from www.u-blox.com.

u-blox makes no warranties based on the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and product descriptions at any time without notice. Reproduction, use or disclosure to third parties without express permission is strictly prohibited. Copyright © 2011, u-blox AG.

u-blox® is a registered trademark of u-blox Holding AG in the EU and other countries. ARM® is the registered trademark of ARM Limited in the EU and other countries.



1 Functional description

1.1 Overview

The NEO-6 module series is a family of stand-alone GPS receivers featuring the high performance u-blox 6 positioning engine. These flexible and cost effective receivers offer numerous connectivity options in a miniature 16 x 12.2 x 2.4 mm package. Their compact architecture and power and memory options make NEO-6 modules ideal for battery operated mobile devices with very strict cost and space constraints.

The 50-channel u-blox 6 positioning engine boasts a Time-To-First-Fix (TTFF) of under 1 second. The dedicated acquisition engine, with 2 million correlators, is capable of massive parallel time/frequency space searches, enabling it to find satellites instantly. Innovative design and technology suppresses jamming sources and mitigates multipath effects, giving NEO-6 GPS receivers excellent navigation performance even in the most challenging environments.

1.2 Product features

Model	Type					Supply		Interfaces				Features						
	GPS	PPS	Time Sync	Binary Data	Dead Reckoning	1.75 V _{CE} ± 20 V	3.7 V _{CE} ± 6 V _{CE}	UART	USB	SPI	IOCC (IC Compliance)	Programmable (UART) for update	TTFF	RTT (msec)	Autonomous setup and suspension	Configuration pins	Temperature	External interrupt (GPIO)
NEO-9G	•					•		•	•	•	•	•	•	•	•	•	•	•
NEO-9Q	•						•	•	•	•	•		•	•	•	•	•	•
NEO-9M	•						•	•	•	•	•		•	•	•	•	•	•
NEO-6P	•	•		•			•	•	•	•	•		•	•	•	•	•	•
NEO-6V	•				•		•	•	•	•	•		•	•	•	•	•	•
NEO-6T	•		•	•			•	•	•	•	•		•	•	•	•	•	•

• = Requires external components and integration on application processor

Table 1: Features of the NEO-6 Series



All NEO-6 modules are based on GPS chips qualified according to AEC-Q100. See Chapter 5.1 for further information.

2 Pin Definition

2.1 Pin assignment

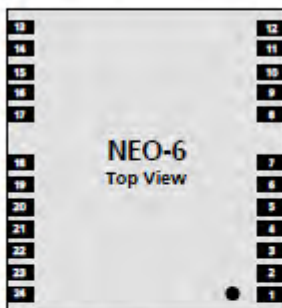


Figure 2 Pin Assignment

No	Module	Name	I/O	Description
1	All	Reserved	I	Reserved
2	All	SS_N	I	SP Slave Select
3	All	TIMEPULSE	O	Timepulse (1PPS)
4	All	EXTINT0	I	External Interrupt Pin
5	All	USB_DM	I/O	USB Data
6	All	USB_DP	I/O	USB Data
7	All	VDDUSB	I	USB Supply
8	All	Reserved		See Hardware Integration Manual Pin 8 and 9 must be connected together.
9	All	VCC_RF	O	Output Voltage (RF section) Pin 8 and 9 must be connected together.
10	All	GND	I	Ground
11	All	RF_IN	I	GPS signal input
12	All	GND	I	Ground
13	All	GND	I	Ground
14	All	MISO_CFG_COM0	O/I	SP MOSI / Configuration Pin. Leave open if not used.
15	All	MISO_CFG_COM1	I	SP MISO / Configuration Pin. Leave open if not used.
16	All	CFG_GPSDSCK	I	Power Mode Configuration Pin / SP Clock. Leave open if not used.
17	All	Reserved	I	Reserved
18	All	SDA2	I/O	I2C Data
19	All	SCL2	I/O	I2C Clock
20	All	TxD1	O	Serial Port 1
21	All	RxD1	I	Serial Port 1

1.3 GPS performance

Parameter	Specification	NEO-6M ¹	NEO-6M ²	NEO-6P
Receiver type	50 Channels GPS L1 frequency, C/A Code SBAS: WAAS, EGNOS, MSAS			
Time-To-First-Fix ³	Cold Start ⁴	26 s	27 s	32 s
	Warm Start ⁴	26 s	27 s	32 s
	Hot Start ⁴	1 s	1 s	1 s
	Aided Starts ⁴	1 s	<3 s	<3 s
Sensitivity ⁵		NEO-6M ²	NEO-6M ²	NEO-6P
	Tracking & Navigation	-162 dBm	-161 dBm	-160 dBm
	Reacquisition ⁶	-150 dBm	-160 dBm	-160 dBm
	Cold Start (without aiding)	-148 dBm	-147 dBm	-146 dBm
	Hot Start	-157 dBm	-156 dBm	-155 dBm
Maximum Navigation update rate		NEO-6M ²	NEO-6M ²	NEO-6P
		5 Hz	1 Hz	
Horizontal position accuracy ⁷	GPS	2.5 m		
	SBAS	2.0 m		
	SBAS + PPP ⁸	< 1 m (20, R50) ⁹		
	SBAS + PPP ⁸	< 2 m (30, R50) ⁹		
Configurable Timepulse frequency range		NEO-6M ²	NEO-6P	
		0.25 Hz to 1 kHz	0.25 Hz to 10 MHz	
Accuracy for Timepulse signal	RMS	30 ns		
	99%	<60 ns		
	Granularity	21 ns		
	Compensated ¹⁰	15 ns		
Velocity accuracy ¹¹		0.1 m/s		
Heading accuracy ¹²		0.5 degrees		
Operational Limits	Dynamics	± 4 g		
	Altitude ¹³	50,000 m		
	Velocity ¹⁴	500 m/s		

Table 2: NEO-6 GPS performance
¹ All satellites at -130 dBm

² Without aiding

³ Dependent on aiding data connection speed and latency

⁴ Demonstrated with a good active antenna

⁵ For an outage duration ≤ 10s

⁶ CIP, 50%, 24 hours static, -130dBm, SEP, <3.5m

⁷ NEO-6P only

⁸ Demonstrated under following conditions: 24 hours, stationary, first 600 seconds of data discarded, HDOP < 1.5 during measurement period, strong signals. Continuous availability of valid SBAS correction data during full test period.

⁹ Quantization error information can be used with NEO-6T to compensate the granularity related error of the timepulse signal

¹⁰ Assuming Airborne-cfg platform

11

12

M2M modem

FASTRACK GSM/GPRS MODEM

M1306B



DEPEND ON A WIRELESS WORKHORSE

Wavocom's rugged, discrete Fastrack GSM/GPRS modem has proven itself for stable, reliable performance on wireless networks worldwide for more than five years. Updated with new features, the now legendary Fastrack continues to deliver rapid time to market and painless integration for machine-to-machine applications.

Housed in a rugged metallic casing, 25 mm shorter than the previous version, the Fastrack M1306B now offers two general purpose input/output access points, which can be multiplexed with an I2C bus to connect peripherals. In addition, the new serial auto shut down feature enables a programmable energy conservation mode especially valuable for battery-powered applications.

Fully certified, the dual-band 900/1800 MHz Fastrack M1306B offers GPRS Class 10 capability, supports Open AT and IT protocols such as IP connectivity.

RUGGED MACHINE-TO-MACHINE RELIABILITY

Add wireless to existing applications

Such as remote control and monitoring, vending, fleet management and others.

Reduce extra components

By embedding your application right on the Wavocom platform with Open AT.

Save time

Thanks to a fully integrated, fully certified wireless solution.

Get connected

And benefit from wireless services: GSM/GPRS data, SMS and voice via a simple serial connection.

wavocom®

WIRELESS SOLUTIONS FOR EVERYONE

M2M modem SPECIFICATIONS

PRODUCT FEATURES

Dual Band GSM modem (ECSM900/1800 MHz) designed for data, fax, SMS and voice applications

Fully Type Approved

Fully compliant with ETSI GSM Phase 2 + Small MS

Output power:

Class 4 (2W @ 900 MHz)

Class 1 (1W @ 1800 MHz)

Power supply:

Input voltage: 5V-32V

- 5mA in idle mode, 140mA in communication GSM 900 @ 12V

- 5mA in idle mode, 100mA in communication GSM 1800 @ 12V

- Peak 1.7A @ 5.5V

Overall dimensions: 73 x 54 x 25mm

Weight: 82g

VOICE, DATA/FAX, SHORT MESSAGE SERVICES

Voice features:

- Telephony
- Emergency calls
- Full Rate, Enhanced Full Rate and Half Rate (FR/FRHR)
- Dual Tone Multi Frequency Function (DTMF)

GSM Data/Fax features:

- Data circuit asynchronous, transparent and non transparent up to 14,400 bits/s
- Automatic fax group 3 (Class 1 and Class 2)

- MNP2, V.42bis

GPRS packet Data features:

- GPRS Class 10, PBCCH support

Coding schemes: CS1 to CS4

Compliant with SMG31bis

- Embedded TCP/IP stack

Short Messages Services features:

- Text and PDU
- Point to point (MT/MO)
- Call Broadcast

GSM SUPPLEMENTARY SERVICES

- Call Forwarding
- Call Barring
- Multiparty
- Call Waiting and Call Hold
- Calling Line Identity
- Advice of Charge
- USSD

- Closed User Group
- Explicit Call Transfer

OTHER FEATURES

- Advanced Open Software Platform: MUSE Platform Open AT
- Fixed Dialling Number
- SIM Toolkit Class 2
- SIM, network and service provider locks
- Real Time Clock
- Alarm management
- Software upgrade through Xmodem protocol
- UCS2 character set management

INTERFACES

- RS-232 and audio through mini sub-D 15-pin connector supporting:
 - Serial link autoshtutdown controlled by software (AT)
 - Baud rate
 - Autobauding

AT commands interface:

- GSM 07.06 and 07.07 AT commands
- comprehensive set of enhanced AT commands

Open AT APIs:

- numerous interfaces for embedded applications:
 - 2x GPIOs/V2C (multiplexed) + power supply through micro-FIT 4-pin connector
 - SMA antenna connector
 - Sliding SIM holder (3V SIM interface)

APPROVALS

The M1306B is approved worldwide under test standards including: Radio and Telecommunication Terminal Equipment (R&TTE), Global Certification Forum – Certification Criteria (GCF-CC), EMC, Safety and Chinese approvals

DELIVERABLES

- User guide
- Power supply cable
- Y-cable for data and audio connection (optional)
- By DDS www.D-D-S.nl

WAVECOM S.A. may, at any time and without notice, make changes or improvements to its products and services without prior notice, providing or communicating them.

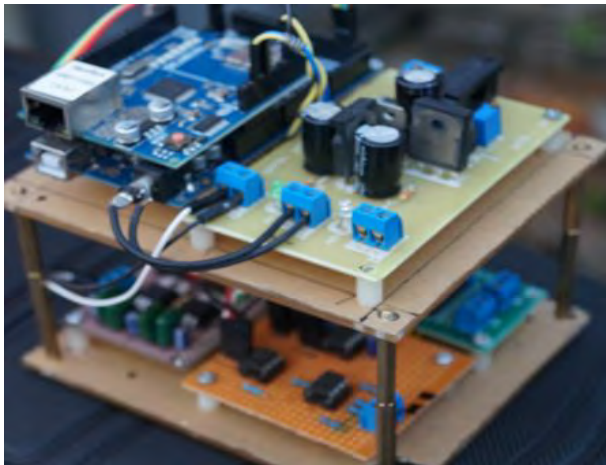
wavecom 

LAMPIRAN D BENTUK ALAT

a. Kotak Hitam Sepeda Motor



b. Gambar Rangkaian Kotak Hitam Sepeda Motor



c. Gambar Transponder Sepeda Motor



d. Sensor Thermocouple



e. Sensor Loadcell



f. Rotary Encoder



g. Tampilan Kotak Hitam dan Transponder Sepeda Motor

