



UNIVERSITAS INDONESIA

**SISTEM PENJEJAKAN DAN KEAMANAN PADA
KENDARAAN BERMOTOR MENGGUNAKAN SMS
BERBASIS MIKROKONTROLER ATMEGA162**

SKRIPSI

**SYUKRON ZAHRI
0606074400**

**FAKULTAS TEKNIK
PROGRAM STUDI TEKNIK ELEKTRO
DEPOK
DESEMBER 2011**



UNIVERSITAS INDONESIA

**SISTEM PENJEJAKAN DAN KEAMANAN PADA
KENDARAAN BERMOTOR MENGGUNAKAN SMS
BERBASIS MIKROKONTROLER ATMEGA162**

SKRIPSI

Diajukan sebagai salah satu syarat untuk memperoleh gelar sarjana teknik

**SYUKRON ZAHRI
0606074400**

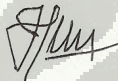
**FAKULTAS TEKNIK
PROGRAM STUDI TEKNIK ELEKTRO
DEPOK
DESEMBER 2011**

HALAMAN PERNYATAAN ORISINALITAS

**Skripsi ini adalah hasil karya saya sendiri,
dan semua sumber baik yang dikutip maupun dirujuk
telah saya nyatakan dengan benar**

Nama : Syukron Zahri

NPM : 0606074400

Tanda Tangan : 

Tanggal : 26 Desember 2011

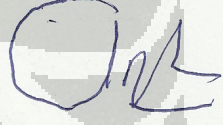


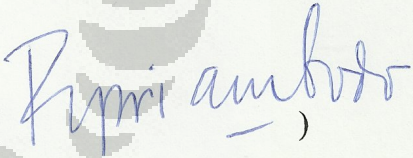
HALAMAN PENGESAHAN

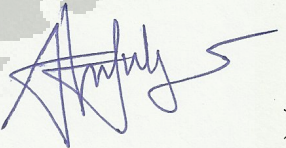
Skripsi ini diajukan oleh :
Nama : Syukron Zahri
NPM : 0606074400
Program Studi : Teknik Elektro
Judul Skripsi : Sistem Penjejakan Dan Keamanan Pada Kendaraan
Bermotor Menggunakan SMS Berbasis
Mikrokontroler ATmega162

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Teknik pada Program Studi Teknik Elektro, Fakultas Teknik, Universitas Indonesia.

DEWAN PENGUJI

Pembimbing : Dr. Ir. Arman D. Diponegoro ()

Penguji I : Dr. Ir. Purnomo S. Priambodo, MSEE ()

Penguji II : Prof. Dr. Ir. Harry Sudibyo, DEA. ()

Ditetapkan di : Depok

Tanggal : 16 Januari 2012

UCAPAN TERIMA KASIH

Segala puji milik Allah *Subhanahu Wa Ta'ala* atas rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi ini. Penulis ingin menyampaikan rasa terima kasih kepada:

1. Dr. Ir. Arman D. Diponegoro sebagai dosen pembimbing, yang telah memberikan waktu, petunjuk, saran dan bimbingannya kepada penulis.
2. Ibu, kakak dan adik yang telah memberi semangat kepada penulis.
3. Rekan-rekan dari Teknik Elektro yang telah turut serta membantu penulis.
4. Dan pihak lain yang tidak disebutkan, yang juga turut membantu penulis dalam menyelesaikan laporan ini.

Penulis meminta maaf yang sebesar-besarnya bila kesalahan atau kekurangan di dalam laporan ini. Penulis menyadari bahwa dalam skripsi ini jauh dari kesempurnaan, sehingga saran dan kritik yang membangun sangat dibutuhkan. Penulis juga mengharapkan laporan ini dapat berguna untuk semua pembaca. Sebagai penutup, penulis berharap agar Allah *Subhanahu Wa Ta'ala* berkenan membalas atas segala kebaikan semua pihak yang telah membantu.

Jakarta, 26 Desember 2011

Penulis

**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI
TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS**

Sebagai sivitas akademik Universitas Indonesia, saya yang bertanda tangan di bawah ini:

Nama : Syukron Zahri
NPM : 0606074400
Program Studi : Teknik Elektro
Departemen : Teknik Elektro
Fakultas : Teknik
Jenis Karya : Skripsi

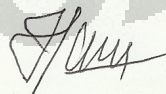
demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Indonesia **Hak Bebas Royalti Noneklusif** (*Non-exclusive Royalti-Free Right*) atas karya ilmiah saya yang berjudul:

Sistem Penjejakan Dan Keamanan Pada Kendaraan Bermotor Menggunakan SMS
Berbasis Mikrokontroler Atmega162

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (database), merawat, dan memublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Jakarta
Pada tanggal : 19 Januari 2012
Yang menyatakan



(Syukron Zahri)

ABSTRAK

Nama : Syukron Zahri
Program Studi : Teknik Elektro
Judul : Sistem Penjejakan Dan Keamanan Pada Kendaraan Bermotor
Menggunakan SMS Berbasis Mikrokontroler ATmega162

Pencurian kendaraan sudah banyak terjadi terutama pada kendaraan beroda dua. Beberapa cara dilakukan untuk mencegah pencurian kendaraan tersebut salah satunya adalah dengan menggunakan alat penjejak yang dikenal sebagai *GPS tracker*. Pada skripsi ini dibuat sebuah sistem penjejak dan keamanan kendaraan bermotor yang dapat menampilkan lokasi kendaraan, mengendalikan mesin, klakson, dan mengaktifkan sistem alarm ketika kendaraan dihidupkan oleh orang yang tidak dikehendaki. Sistem ini memanfaatkan GPS dan SMS dalam operasinya. Dalam sistem ini, digunakan ATmega162 sebagai pusat pemrosesan dan modem Wavecom M1306B untuk komunikasi via SMS. Dari pengujian waktu respon yang dilakukan dengan mengambil data posisi, didapatkan waktu respon rata-rata 13,22 detik, mulai dari SMS dikirim hingga SMS balasan diterima.

Kata kunci: GPS, SMS, penjejakan, alarm, remote control saklar, mikrokontroler, ATmega162, AT *command*, komunikasi serial, *handphone*, modem, Wavecom Fastrack M1306B, BASCOM AVR, Visual Basic .NET

ABSTRACT

Name : Syukron Zahri
Study program : Electrical Engineering
Judul : Tracking And Security System For Vehicles Using SMS Based
On ATmega162 Microcontroller

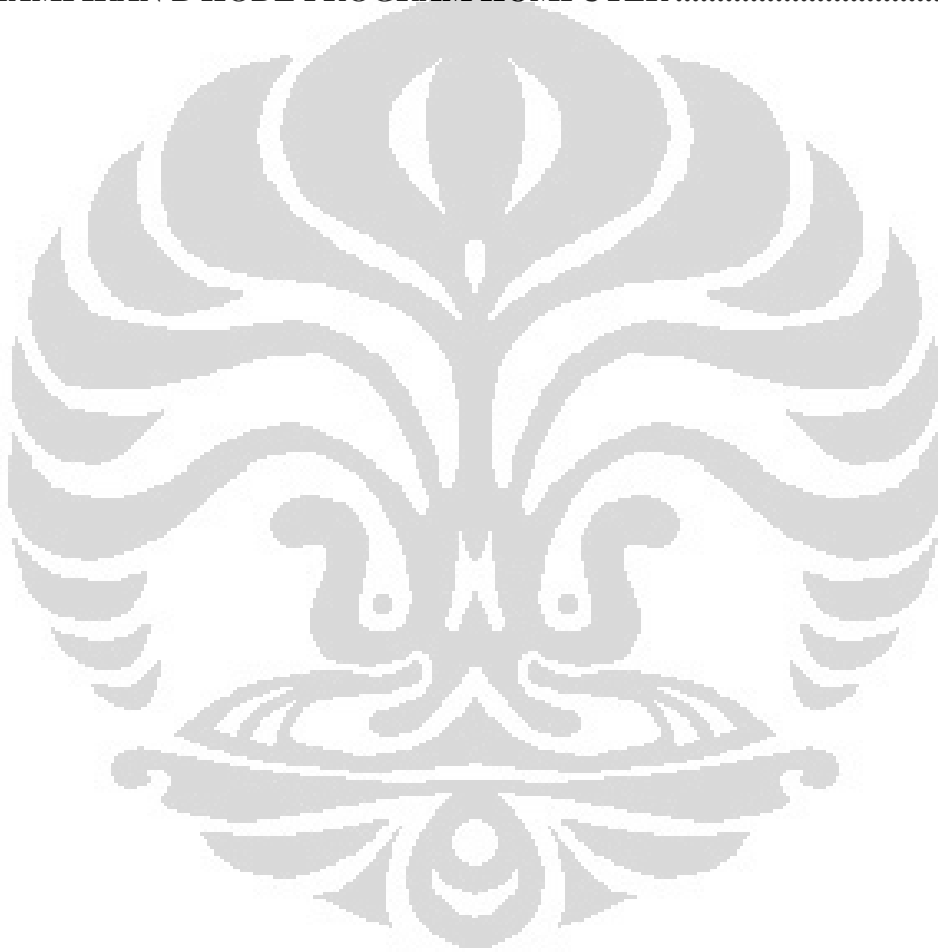
Vehicle theft have occurred many times especially on two-wheeled vehicles. Some efforts have been made in order to prevent that vehicle theft, one of them is by installing a tracker system or known as GPS tracker. In this final project, a tracker and security system has been created which is able to display vehicle position, control engine, horn, and activate its alarm system when the vehicle is switched on by someone who is not desired. This system utilizes GPS and SMS in its operation. In this system, Atmega162 is used as the central processing unit and Wavecom M1306B modem is used as communication unit through SMS. From the testing of response time that performed by taking the position data, this system gives an average of response time in 13,22 seconds, started from sending SMS until replied SMS is received.

Keywords: GPS, SMS, tracking, alarm, remote controlled switching, microcontroller, ATmega162, AT command, serial communication, handphone, modem, Wavecom Fastrack, M1306B, BASCOM AVR, Visual Basic .NET

DAFTAR ISI

HALAMAN PERNYATAAN ORISINALITAS	i
HALAMAN PENGESAHAN	ii
UCAPAN TERIMA KASIH.....	iii
HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI.....	iv
ABSTRAK.....	v
ABSTRACT	vi
DAFTAR ISI.....	vii
DAFTAR GAMBAR.....	ix
DAFTAR TABEL	xi
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Tujuan	2
1.3 Ruang Lingkup	2
1.4 Metodologi	2
1.5 Sistematika Penulisan	2
BAB 2 LANDASAN TEORI.....	4
2.1 GPS (Global Positioning System)	4
2.1.1 Segmen GPS	4
2.1.2 Sinyal GPS	5
2.1.3 Layanan Posisi GPS.....	6
2.1.4 Prinsip Penentuan Posisi Dengan GPS	6
2.1.5 Ketelitian posisi absolut.....	9
2.1.6 Format Data Output GPS Receiver	9
2.2 Mikrokontroler	11
2.2.1 Mikrokontroler AVR ATmega162	13
2.2.2 Konfigurasi Pin ATmega162	13
2.3 Komunikasi Serial	15
2.4 AT Command	16
2.5 Short Message Service (SMS).....	17
BAB 3 PERANCANGAN SISTEM.....	20
3.1 Desain Sistem	20
3.2 Prinsip Kerja Sistem	22
3.3 Format SMS	23
3.4 Perancangan Perangkat Keras	24
3.4.1 Mikrokontroler ATmega162	26
3.4.2 GPS Receiver	27
3.4.3 Relay	28
3.4.4 Modem.....	29
3.5 Perancangan Perangkat Lunak	30
3.5.1 Perancangan Perangkat Lunak pada Mikrokontroler	31
3.5.2 Perancangan Perangkat Lunak pada Komputer	34
BAB 4 PENGUJIAN SISTEM DAN ANALISIS	37
4.1 Implementasi Sistem.....	37
4.2 Pengujian Sistem	37
4.2.1 Pengujian Menampilkan Posisi	39
4.2.2 Mengendalikan Mesin dan Klakson	42

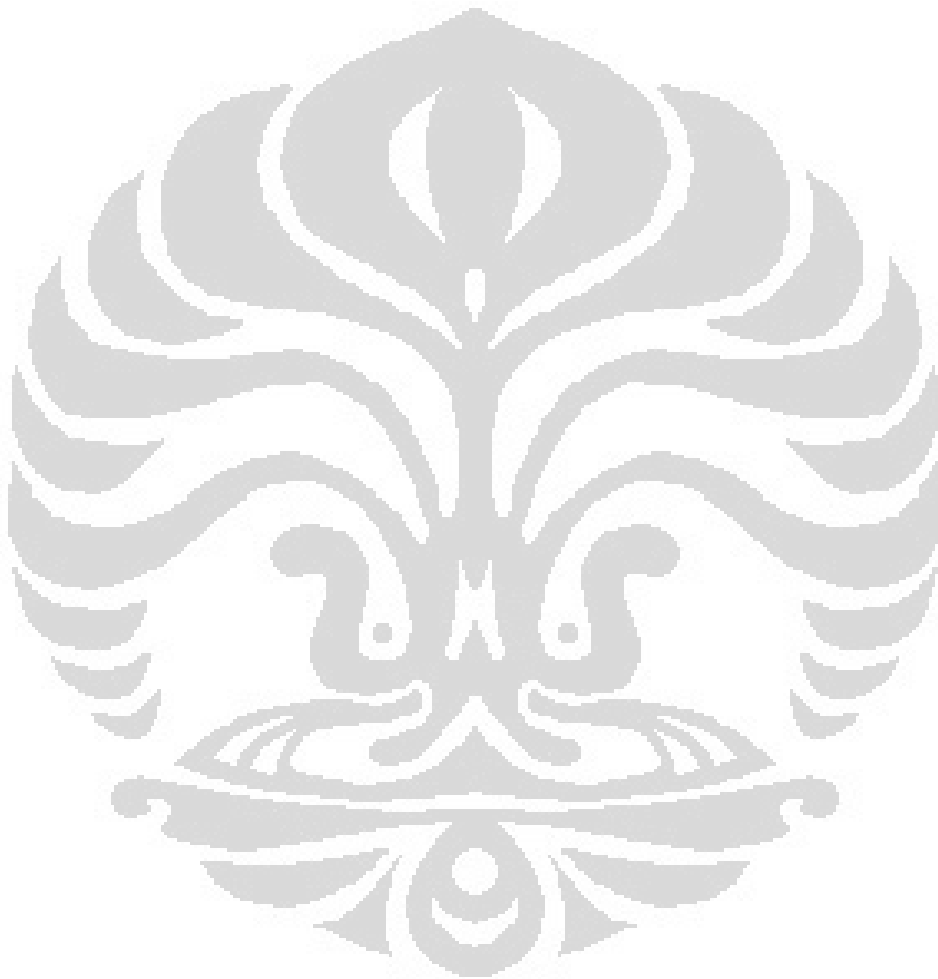
4.2.2.1	Mengendalikan Mesin.....	42
4.2.2.2	Mengendalikan Klakson	43
4.2.3	Pengujian Sistem Alarm	45
4.2.4	Cek Status	47
4.2.5	Reset Sistem.....	47
4.3	Pengujian Waktu Respon.....	48
4.4	Pengujian Objek Dalam Keadaan Bergerak.....	52
4.5	Analisis	54
BAB 5 KESIMPULAN		56
DAFTAR REFERENSI		57
LAMPIRAN A KODE PROGRAM MIKROKONTROLER.....		58
LAMPIRAN B KODE PROGRAM KOMPUTER		63



DAFTAR GAMBAR

Gambar 2.1	Segmen GPS	4
Gambar 2.2	Prinsip penentuan posisi dalam koordinat 3 dimensi	7
Gambar 2.3	Data keluaran GPS receiver pada program HyperTerminal	11
Gambar 2.4	Pin-pin pada ATmega162	14
Gambar 2.5	Format frame pada komunikasi serial.....	16
Gambar 2.6	AT Command	16
Gambar 2.7	Alur pengiriman SMS	18
Gambar 3.1	Diagram sistem secara keseluruhan.....	20
Gambar 3.2	Blok diagram sistem objek.....	21
Gambar 3.3	Blok diagram sistem pengamat yang menggunakan komputer.....	22
Gambar 3.4	Skematik sistem minimum	25
Gambar 3.5	Layer bawah PCB sistem minimum	26
Gambar 3.6	Tata letak komponen pada PCB	26
Gambar 3.7	GPS Parallax RXM-SG	27
Gambar 3.8	DI-Relay2	29
Gambar 3.9	Modem Wavecom Fastrack M1306B	29
Gambar 3.10	Algoritma program pada mikrokontroler	31
Gambar 3.11	Algoritma pembacaan isi SMS	32
Gambar 3.12	Tampilan perangkat lunak di komputer	34
Gambar 3.13	Algoritma program pada komputer.....	36
Gambar 4.1	Implementasi sistem objek	37
Gambar 4.2	Tampilan halaman Connection pada perangkat lunak.....	38
Gambar 4.3	Tampilan halaman Status pada perangkat lunak	39
Gambar 4.4	Format SMS untuk meminta posisi	39
Gambar 4.5	Hasil dari permintaan data posisi di handphone.....	40
Gambar 4.6	SMS balasan ketika data GPS tidak valid.....	40
Gambar 4.7	Tampilan posisi objek pada perangkat lunak	41
Gambar 4.8	Data koordinat ditampilkan di halaman Status.....	41
Gambar 4.9	Format SMS untuk mengendalikan mesin. (a) mesin dapat digunakan dan (b) mesin tidak dapat digunakan	42
Gambar 4.10	Kondisi relay ketika mendapatkan SMS yang berisi “mesin on” ..	43
Gambar 4.11	Kondisi relay ketika mendapatkan SMS yang berisi “mesin off” .	43
Gambar 4.12	Format SMS untuk mengendalikan klakson. (a) untuk membunyikan klakson dan (b) untuk mematikan klakson.....	44
Gambar 4.13	Kondisi relay ketika mendapatkan SMS yang berisi “klakson on”	44
Gambar 4.14	Kondisi relay ketika mendapatkan SMS yang berisi “klakson off”	45
Gambar 4.15	Format SMS untuk mengaktifkan alarm	45
Gambar 4.16	Format SMS untuk menetapkan nomor darurat.....	46
Gambar 4.17	SMS yang diterima yang didapat ketika kendaraan dihidupkan dan alarm dalam keadaan aktif.....	46
Gambar 4.18	Mesin tidak dapat digunakan dan klakson berbunyi.....	46
Gambar 4.19	Format SMS untuk meminta status dari sistem objek	47
Gambar 4.20	SMS yang diterima setelah mengirimkan perintah ‘status’	47
Gambar 4.21	Format SMS untuk mereset sistem objek.....	48

Gambar 4.22 Mesin bisa digunakan dan klakson tidak berbunyi setelah mendapatkan SMS yang berisi “reset” 48



DAFTAR TABEL

Tabel 2.1 Kalimat NMEA 0183 pada GPS	10
Tabel 2.2 Perbedaan arsitektur CISC dan RISC	12
Tabel 2.3 Jenis-jenis perintah AT Command.....	17
Tabel 3.1 Penggunaan port mikrokontroler.....	27
Tabel 3.2 Konfigurasi modem pada mikrokontroler	30
Tabel 3.2 Konfigurasi Port Serial.....	33
Tabel 4.1 Waktu respon pada sistem objek yang menggunakan Three dan pengamat yang menggunakan berbagai macam operator (dalam detik)	49
Tabel 4.2 Waktu respon pada sistem objek yang menggunakan XL dan pengamat yang menggunakan berbagai macam operator (dalam detik).....	50
Tabel 4.3 Waktu respon pada sistem objek yang menggunakan IM3 dan pengamat yang menggunakan berbagai macam operator (dalam detik).....	50
Tabel 4.4 Waktu respon pada sistem objek yang menggunakan As dan pengamat yang menggunakan berbagai macam operator (dalam detik).....	51
Tabel 4.5 Waktu respon pada sistem objek yang menggunakan Axis dan pengamat yang menggunakan berbagai macam operator (dalam detik).....	51
Tabel 4.6 Waktu respon minimum dari berbagai macam operator (dalam detik).....	51
Tabel 4.7 Waktu respon rata-rata dari berbagai macam operator (dalam detik)...	52
Tabel 4.8 Persentase keberhasilan dari berbagai macam operator (dalam persen).....	52
Tabel 4.9 Hasil percobaan ketika sistem objek dalam keadaan bergerak.....	53



BAB 1 PENDAHULUAN

1.1 Latar Belakang

Kasus pencurian kendaraan bermotor mungkin sudah tidak asing bagi kita terutama kasus pencurian kendaraan beroda dua. Ada beberapa usaha untuk mencegah kendaraan bermotor diantaranya adalah dengan memarkir kendaraan di tempat yang aman dan memasang alat pengaman tambahan. Ada berbagai macam alat pengaman tambahan pada kendaraan seperti kunci roda, kunci setang, gembok, dan sebagainya. Namun alat pengaman tersebut dirasa masih kurang mencukupi karena alat tersebut hanya berfungsi ketika kendaraan masih di jangkauan pemiliknya.

GPS adalah sistem navigasi berbasis satelit yang dikembangkan oleh Departemen Pertahanan Amerika Serikat. GPS mempunyai cakupan area yang sangat luas yaitu seluruh permukaan bumi dan siapa pun dapat memanfaatkan teknologi ini tanpa perlu membayar. GPS beroperasi selama 24 jam dalam segala kondisi cuaca. Setelah kebijakan *selective availability* dihapus pada tahun 2000, kini GPS mempunyai akurasi yang lebih baik.

SMS adalah teknologi pengiriman pesan singkat pada jaringan GSM yang kini juga didukung oleh jaringan CDMA dan telepon rumah. Saat ini SMS mungkin sudah tidak asing lagi bagi masyarakat dan sudah menjadi bagian dari kehidupan sehari-hari. Kini tarif pengiriman SMS semakin murah karena teknologi seluler semakin berkembang, jumlah pelanggan operator semakin banyak dan persaingan bisnis telekomunikasi mengharuskan operator memberikan tarif yang kompetitif.

Berawal dari fakta di atas, penulis ingin membuat sebuah sistem penjejak (*tracker*) yang dilengkapi dengan fungsi alarm dan pengendalian *switch* sebagai sistem keamanan untuk kendaraan bermotor. Sistem ini memanfaatkan GPS dan SMS dalam aplikasinya. Sistem ini diharapkan dapat membantu pemilik kendaraan dalam memantau keberadaan kendaraan dan mencegah tindakan pencurian.

1.2 Tujuan

Tujuan dari penulisan skripsi ini adalah untuk merancang dan membuat sistem penjejakan dan keamanan pada kendaraan bermotor dengan memanfaatkan teknologi GPS dan SMS.

1.3 Ruang Lingkup

Pada skripsi ini permasalahan dibatasi menjadi:

1. Sistem yang dibuat mampu menampilkan posisi kendaraan jika bagian pengamat menggunakan komputer yang telah terhubung dengan internet.
2. GPS yang digunakan adalah GPS berbasis satelit dengan layanan pemosisian dasar dan tanpa menggunakan augmentasi.
3. Keamanan dalam sistem ini hanya sebatas tindakan preventif dari pencurian yaitu mematikan mesin, menghidupkan klakson dan mengirimkan SMS ketika *switch* kendaraan diputar ke posisi on oleh seseorang.
4. Format SMS yang digunakan adalah teks, format PDU tidak digunakan dalam sistem ini.

1.4 Metodologi

Metode yang digunakan dalam skripsi ini adalah studi literatur, melakukan penelitian pada komponen penyusun sistem, perancangan sistem, pembuatan sistem dan pengujian sistem.

1.5 Sistematika Penulisan

Penulisan skripsi ini dibagi menjadi beberapa bagian yaitu:

1. BAB 1: PENDAHULUAN
Menjelaskan tentang latar belakang, tujuan, rumusan masalah, batasan masalah, metodologi penulisan dan sistematika penulisan
2. BAB 2: LANDASAN TEORI
Menjelaskan landasan teori yang berhubungan dalam skripsi ini diantaranya *Global Positioning System*, mikrokontroler AVR ATmega162, komunikasi serial, *AT Command*, dan *Short Message Service*

3. **BAB 3: PERANCANGAN SISTEM**

Menjelaskan tentang desain sistem, prinsip kerja sistem, perancangan perangkat keras, dan perancangan perangkat lunak.

4. **BAB 4: PENGUJIAN SISTEM DAN ANALISIS**

Menjelaskan tentang implementasi dari perancangan sistem, pengujian dan analisis dari hasil pengujian sistem

5. **BAB 5: KESIMPULAN DAN SARAN**

Menjelaskan tentang kesimpulan yang didapat dari hasil pengujian dan analisis sistem secara keseluruhan dan saran untuk pengembangan skripsi



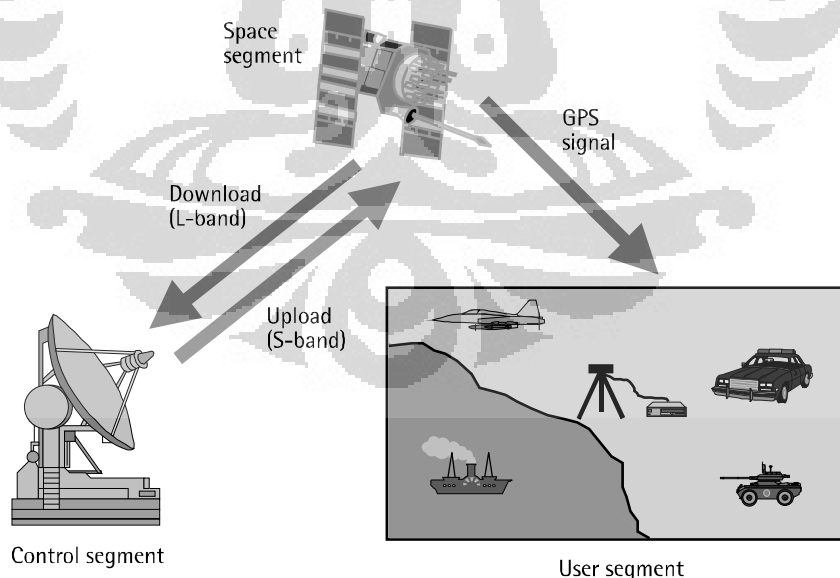
BAB 2 LANDASAN TEORI

2.1 GPS (Global Positioning System) [1]

GPS (*Global Positioning System*) adalah sistem navigasi berbasis satelit yang dikembangkan oleh Departemen Pertahanan Amerika Serikat (*U.S. Department of Defence*) dan mempunyai nama resmi NAVSTAR (*Navigation Satellite Timing and Ranging*). Proyek GPS dimulai pada tahun 1973 dan satelit GPS pertama diluncurkan pada tanggal 21 Februari 1978. Awalnya GPS dikembangkan untuk memenuhi kebutuhan militer Amerika Serikat. Namun kemudian GPS dibuat gratis untuk warga sipil. GPS menyediakan informasi posisi dan waktu secara kontinu dimanapun di bumi dalam kondisi cuaca apapun. GPS dapat melayani pengguna dengan jumlah yang tidak terbatas dan GPS merupakan sistem penjejakan satu arah.

2.1.1 Segmen GPS

GPS terdiri dari tiga segmen yaitu segmen ruang angkasa, segmen pengendali, dan segmen pengguna seperti pada Gambar 2.1.



Gambar 2.1 Segmen GPS [1]

Segmen ruang angkasa terdiri dari 24 satelit GPS. GPS mempunyai enam orbit dimana tiap orbitnya ditempati oleh empat satelit. Dengan konfigurasi ini, empat hingga sepuluh satelit GPS dapat terlihat dimanapun di permukaan bumi jika sudut elevasinya adalah 10° . Orbit satelit GPS berbentuk ellips dengan nilai eksentrisitas maksimum 0,01, dan mempunyai sudut inklinasi sebesar 55° terhadap ekuator. Orbit satelit GPS terletak di ketinggian sekitar 20.200 km di atas permukaan bumi dan mempunyai radius orbit terjauh sebesar 25.600 km. Tiap satelit GPS mempunyai periode orbit selama 11 jam 58 menit.

Segmen pengendali adalah bagian yang memantau keberadaan dan aktivitas satelit GPS seperti memperkirakan dan menentukan posisi satelit, perilaku jam atom, keutuhan sistem, data atmosfer, dan perhitungan lainnya. Segmen pengendali GPS terdiri dari *Master Control Station* (MCS), jaringan stasiun monitor di seluruh dunia, dan *Ground Control Station*. MCS merupakan stasiun pengendali utama, pusat pemrosesan dari segmen pengendali, berlokasi di Colorado Springs, Colorado, Amerika Serikat.

Segmen pengguna adalah siapa saja termasuk pihak militer dan penduduk sipil. Dengan GPS *receiver* yang terhubung dengan antena GPS, pengguna dapat menerima sinyal GPS sehingga dapat menentukan posisinya di bumi.

2.1.2 Sinyal GPS

Tiap satelit GPS memancarkan sinyal yang tersusun dari dua frekuensi *carrier* yang dimodulasikan oleh dua kode digital dan pesan navigasi. Kedua frekuensi *carrier* dihasilkan pada 1,575.42 MHz (L1) dan 1,227.60 MHz (L2). Adanya dua frekuensi *carrier* ini mampu mengoreksi error pada GPS, yang dikenal dengan istilah delay ionosfer. Semua satelit GPS memancarkan frekuensi *carrier* L1 dan L2 yang sama. Namun, kode modulasi untuk tiap satelit berbeda.

Kode GPS ada dua, yaitu *coarse acquisition* (C/A-code) dan *precision* (P-code). C/A-code dimodulasikan hanya pada *carrier* L1 saja, sedangkan P(Y)-code dimodulasikan pada *carrier* L1 dan L2. Modulasi yang digunakan adalah modulasi bifase, yaitu fase *carrier* digeser sejauh 180° ketika nilai kode berubah dari nol ke satu atau dari satu ke nol. Tiap satelit mempunyai C/A-code yang unik sehingga perangkat penerima GPS dapat mengenali satelit mana yang mengirimkan kode tersebut.

2.1.3 Layanan Posisi GPS

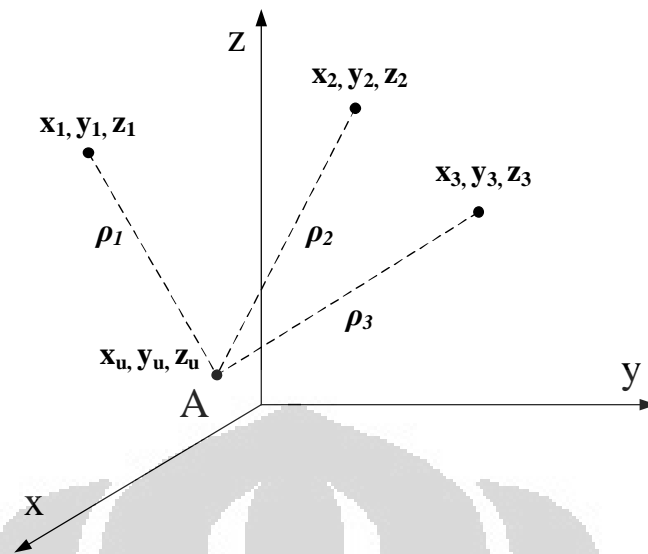
GPS awalnya dikembangkan sebagai sistem militer, tapi kemudian dibuat gratis untuk semua orang. Namun untuk menjaga keuntungan pada pihak militer, Departemen Pertahanan Amerika Serikat menyediakan dua tingkat pelayanan posisi dan pewaktuan GPS, yaitu PPS (*Precise Positioning Service*) dan SPS (*Standard Positioning Service*).

PPS adalah layanan posisi dan pewaktuan yang paling presisi. PPS menggunakan sebuah kode transmisi P(Y)-code yang hanya dapat diakses oleh pihak yang berwenang seperti militer Amerika Serikat. Tingkat akurasi yang diberikan oleh PPS adalah 16 meter untuk komponen horizontal dan 23 meter untuk komponen vertikal.

SPS kurang presisi jika dibandingkan dengan PPS. SPS menggunakan kode C/A-code, yang tidak dikenakan biaya dan untuk siapa saja di seluruh dunia baik pengguna yang berwenang maupun yang tidak mempunyai wewenang. Awalnya SPS menyediakan akurasi posisi dalam orde 100 meter untuk komponen horizontal dan 156 meter untuk komponen vertikal. Kondisi ini tercipta adanya kebijakan SA (*selective availability*). Dengan keputusan presiden Amerika Serikat untuk menghentikan SA, akurasi posisi dari SPS kini dapat menyaingi akurasi dari PPS.

2.1.4 Prinsip Penentuan Posisi Dengan GPS [9]

Prinsip penentuan posisi GPS adalah prinsip reseksi yaitu menentukan sebuah titik dalam sistem koordinat 3 dimensi. Jika ada tiga buah titik yang diketahui posisinya, dan ada sebuah titik A yang tidak diketahui posisinya, titik A tersebut dapat ditentukan posisinya jika jarak antara titik A dengan ketiga titik tersebut diketahui, seperti pada Gambar 2.2.



Gambar 2.2 Prinsip penentuan posisi dalam koordinat 3 dimensi

Pada Gambar 2.2, jika jarak antara ketiga titik yang diketahui posisinya dengan titik A yang tidak diketahui posisinya dinyatakan dalam ρ_1, ρ_2, ρ_3 , maka:

$$\begin{aligned}\rho_1 &= \sqrt{(x_1 - x_u)^2 + (y_1 - y_u)^2 + (z_1 - z_u)^2} \\ \rho_2 &= \sqrt{(x_2 - x_u)^2 + (y_2 - y_u)^2 + (z_2 - z_u)^2} \\ \rho_3 &= \sqrt{(x_3 - x_u)^2 + (y_3 - y_u)^2 + (z_3 - z_u)^2}\end{aligned}\quad (2.1)$$

Dari persamaan 2.1, nilai x_u, y_u, z_u bisa diketahui.

Pada GPS, jarak objek dengan satelit didapatkan dengan mengalikan kecepatan cahaya dengan waktu tempuh sinyal GPS dari satelit ke pengamat. Tiap satelit memancarkan sinyal pada waktu t_{si} . Receiver akan menerima sinyal pada waktu t_u . Dengan demikian jarak antara *receiver* dengan satelit adalah

$$\rho_{iT} = c(t_u - t_{si}) \quad (2.2)$$

Dimana:

ρ_{iT} = jarak yang disebut sebagai *pseudorange* dari satelit i

c = kecepatan cahaya

t_u = waktu penerimaan pada receiver

t_{si} = waktu pengiriman pada satelit

Pada kenyataannya, waktu yang sebenarnya sulit didapatkan, baik pada satelit maupun pada *receiver*. Dengan demikian, hubungan antara jam satelit yang

sebenarnya (t'_{si}) dan jam receiver yang sebenarnya (t'_u) dengan waktu yang sebenarnya adalah:

$$\begin{aligned}t'_{si} &= t_{si} + \Delta b_i \\t'_u &= t_u + b_{ut}\end{aligned}\quad (2.3)$$

Dimana:

Δb_i = kesalahan jam satelit

b_{ut} = kesalahan bias jam receiver

Selain kesalahan jam, ada faktor lain yang mempengaruhi pengukuran *pseudorange* sehingga nilai *pseudorange* dirumuskan menjadi:

$$\rho_i = \rho_{iT} + \Delta D_i + -c(\Delta b_i - b_{ut}) + c(\Delta T_i + \Delta I_i + v_i + \Delta v_i) \quad (2.4)$$

Dimana:

ΔD_i = kesalahan posisi satelit

ΔT_i = kesalahan delay trofosfer

ΔI_i = kesalahan delay ionosfer

v_i = kesalahan pengukuran noise pada receiver

Δv_i = koreksi waktu relativitas

Beberapa kesalahan dapat dikoreksi, misalnya delay trofosfer dapat dimodelkan, kesalahan delay ionosfer dapat dikoreksi dalam receiver 2 frekuensi. Namun, kesalahan jam receiver tidak dapat dikoreksi akan selamanya tidak diketahui. Dengan demikian, persamaan 2.1 dimodifikasi menjadi:

$$\begin{aligned}\rho_1 &= \sqrt{(x_1 - x_u)^2 + (y_1 - y_u)^2 + (z_1 - z_u)^2} + b_u \\ \rho_2 &= \sqrt{(x_2 - x_u)^2 + (y_2 - y_u)^2 + (z_2 - z_u)^2} + b_u \\ \rho_3 &= \sqrt{(x_3 - x_u)^2 + (y_3 - y_u)^2 + (z_3 - z_u)^2} + b_u \\ \rho_4 &= \sqrt{(x_3 - x_u)^2 + (y_3 - y_u)^2 + (z_3 - z_u)^2} + b_u\end{aligned}\quad (2.5)$$

Dimana:

b_u = kesalahan bias jam receiver yang dinyatakan dalam jarak

b_u dinyatakan dalam:

$$b_u = c b_{ut} \quad (2.6)$$

Pada persamaan 2.5, dibutuhkan empat persamaan untuk menemukan nilai dari variabel x_u , y_u , z_u , dan b_u . Oleh karena itu dibutuhkan empat satelit untuk menemukan posisi receiver.

Posisi penerima GPS dinyatakan dalam sistem koordinat kartesian (X, Y, Z) dalam datum WGS 84 (*World Geodetic System 1984*). Untuk keperluan tertentu, koordinat kartesian tersebut dapat dikonversi menjadi koordinat geodetik dalam bentuk lintang (ϕ), bujur (λ), dan ketinggian (h). Titik yang ditentukan posisinya dapat diam (*static positioning*) maupun bergerak (*kinematic positioning*). Penentuan posisi absolut merupakan metode penentuan posisi yang paling mendasar dan paling banyak digunakan untuk aplikasi-aplikasi yang tidak memerlukan tingkat ketelitian posisi yang tinggi dan tersedia secara instan (*realtime*) seperti pada aplikasi navigasi wahana bergerak (darat, laut dan udara) [2].

2.1.5 Ketelitian posisi absolut

Ketelitian posisi absolut GPS sangat bergantung pada tingkat ketelitian data *pseudorange* serta geometri dari satelit pada saat pengukuran. Faktor-faktor yang mempengaruhi ketelitian penentuan posisi dengan GPS adalah sebagai berikut:

- a) Satelit, seperti kesalahan orbit (*ephemeris*) dan jam satelit
- b) Medium propagasi, seperti bias ionosfer dan bias troposfer yang mempengaruhi kecepatan dan arah perambatan sinyal GPS.
- c) GPS *receiver*, seperti kesalahan jam penerima, kesalahan yang terkait dengan antena dan derau (*noise*). Kesalahan-kesalahan ini bergantung pada kualitas dari GPS *receiver*.
- d) Lingkungan sekitar GPS *receiver*, seperti multipath yaitu fenomena dimana sinyal GPS yang tiba di antena GPS *receiver* adalah resultan dari sinyal yang langsung dari satelit GPS dan sinyal yang dipantulkan oleh benda-benda di sekeliling GPS *receiver*.

2.1.6 Format Data Output GPS Receiver

Data output GPS *receiver* dapat dinyatakan dalam bermacam-macam format, biasanya adalah NMEA (*National Marine Electronics Association*) 0183. Protokol NMEA 0183 adalah antarmuka standar untuk perangkat elektronik

marinir seperti *echo sounder*, *anemometer*, *gyrocompass*, *autopilot* dan perangkat lainnya termasuk GPS *receiver* melalui *link* serial.

Data keluaran format NMEA 0183 adalah kalimat (*string*) yang tersusun dari karakter ASCII 8-bit. Setiap kalimat diawali dengan satu karakter '\$', dua karakter *Talker ID*, tiga karakter *Sentence ID*, kemudian diikuti oleh data field yang masing-masing dipisahkan oleh ',' (koma), dan diakhiri dengan *optional checksum* dan karakter CR/LF (*carriage return/line feed*) [2].

Format data NMEA 0183 adalah sebagai berikut [2]:

\$aacc,c---c*hh<CR><LF>

dimana:

aa : *talker ID*, menandakan jenis atau peralatan navigasi yang digunakan. *Talker ID* untuk perangkat GPS adalah 'GP'

ccc : *sentence ID*, menandakan jenis informasi yang terkandung dalam kalimat

c---c : *data fields*, berisi data-data navigasi

*hh : *optional checksum*, untuk pengecekan kesalahan (*error*) kalimat

<CR><LF> : *carriage return/line feed*, menandakan akhir dari kalimat

Ada banyak jenis kalimat pada protokol NMEA 0183 pada perangkat GPS *receiver*. Masing-masing jenis kalimat tersebut memiliki data-data yang berbeda. oleh karena itu pengguna dapat menggunakan data tertentu yang sesuai dengan kebutuhan atau fungsinya. Beberapa jenis kalimat NMEA 0183 yang umum pada perangkat GPS ditunjukkan pada Tabel 2.1.

Tabel 2.1 Kalimat NMEA 0183 pada GPS [2]

Kalimat	Deskripsi
\$GPGGA	<i>Global positioning system fixed data</i>
\$GPGLL	<i>Geographic position - latitude / longitude</i>
\$GPGSA	<i>GNSS DOP and active satellites</i>
\$GPGSV	<i>GNSS satellites in view</i>
\$GPRMC	<i>Recommended minimum specific GNSS data</i>
\$GPVTG	<i>Course over ground and ground speed</i>

Output GPS *receiver* dapat dilihat di program **HyperTerminal** di komputer seperti pada Gambar 2.3.

```

$GPGGA,184429.08,1928.336,S,02410.720,E,1.04,1.0,100.00,M,-33.9,M,,0000*6D
$GPGLL,1928.316,S,02410.725,E,184430.08,A,A*77
$GPVTG,16.78,T,M,74.00,N,137.05,K,A*36
$GPRMC,184432.08,A,1928.277,S,02410.737,E,74.00,16.78,210410.0.0,E,A*21
$GPGGA,184433.08,1928.257,S,02410.743,E,1.04,1.9,100.00,M,-33.9,M,,0000*6C
$GPGLL,1928.238,S,02410.749,E,184434.08,A,A*74
$GPVTG,16.78,T,M,74.00,N,137.05,K,A*36
$GPRMC,184436.08,A,1928.198,S,02410.761,E,74.00,16.78,210410.0.0,E,A*24
$GPGGA,184437.08,1928.178,S,02410.767,E,1.04,2.4,100.00,M,-33.9,M,,0000*6E
$GPGLL,1928.159,S,02410.773,E,184438.08,A,A*75
$GPVTG,16.78,T,M,74.00,N,137.05,K,A*36
$GPRMC,184440.08,A,1928.119,S,02410.785,E,74.00,16.78,210410.0.0,E,A*26
$GPGGA,184441.08,1928.100,S,02410.791,E,1.04,2.4,100.00,M,-33.9,M,,0000*69
$GPGLL,1928.080,S,02410.797,E,184442.08,A,A*77
$GPVTG,16.78,T,M,74.00,N,137.05,K,A*36
$GPRMC,184444.08,A,1928.041,S,02410.809,E,74.00,16.78,210410.0.0,E,A*25
$GPGGA,184445.08,1928.021,S,02410.814,E,1.04,2.7,100.00,M,-33.9,M,,0000*6E
$GPGLL,1928.001,S,02410.820,E,184446.08,A,A*79
$GPVTG,16.78,T,M,74.00,N,137.05,K,A*36
$GPRMC,184448.08,A,1927.962,S,02410.832,E,74.00,16.78,210410.0.0,E,A*26
$GPGGA,184449.08,1927.942,S,02410.838,E,1.04,1.7,100.00,M,-33.9,M,,0000*6C
$GPGLL,1927.922,S,02410.844,E,184450.08,A,A*7B
$GPVTG,16.78,T,M,74.00,N,137.05,K,A*36

```

Gambar 2.3 Data keluaran GPS receiver pada program HyperTerminal

2.2 Mikrokontroler

Mikrokontroler adalah sebuah *central processing unit* (CPU) yang telah dilengkapi dengan berbagai periferal seperti RAM, EEPROM, *analog-to-digital converter* (ADC), osilator, *timer* dan lain-lain. Oleh karena itu mikrokontroler sering disebut *computer on a chip*. Meskipun demikian, terdapat perbedaan antara komputer (CPU) dengan mikrokontroler. Perbedaan tersebut diantaranya:

- CPU pada komputer harus dihubungkan dengan periferal lain agar dapat bekerja sedangkan mikrokontroler tidak harus.
- Kecepatan *clock* CPU komputer mencapai *gigahertz* sedangkan mikrokontroler hanya mencapai beberapa *megahertz*.
- CPU dibuat untuk menjalankan beberapa macam aplikasi sedangkan mikrokontroler dibuat untuk satu macam aplikasi.
- Ukuran memory pada komputer mencapai *gigabyte* atau *terabyte* sedangkan ukuran memory pada mikrokontroler hanya beberapa *byte* atau *kilobyte*.
- Program pada komputer disimpan di dalam ROM dan *hard disk* sedangkan pada mikrokontroler hanya disimpan di dalam ROM

Berdasarkan jumlah instruksinya mikrokontroler dibedakan menjadi dua jenis yaitu CISC (*Complete Instruction Set Computing*) dan RISC (*Reduced Instruction Set Computing*).

Tujuan utama dari arsitektur CISC adalah meminimalkan jumlah instruksi per program sehingga ukuran kode menjadi lebih kecil. Arsitektur CISC mempunyai jumlah instruksi yang banyak tetapi tiap instruksinya dikerjakan dalam beberapa siklus *clock*. Berbeda dengan CISC, tujuan utama dari arsitektur RISC adalah untuk membuat sebuah instruksi yang dapat dikerjakan dalam satu siklus *clock*. Arsitektur RISC mempunyai jumlah instruksi yang sedikit bila dibandingkan dengan CISC sehingga untuk melakukan suatu pekerjaan dibutuhkan beberapa instruksi. Dengan demikian ukuran kode yang dihasilkan menjadi lebih besar.

Mikrokontroler tipe CISC memiliki jumlah instruksi yang banyak tetapi fasilitas internal secukupnya. Kebalikannya, mikrokontroler tipe RISC memiliki jumlah instruksi yang sedikit tetapi kaya fasilitas internal. Perbedaan-perbedaan antara mikrokontroler CISC dengan mikrokontroler RISC dirangkum pada Tabel 2.2

Tabel 2.2 Perbedaan arsitektur CISC dan RISC [3]

CISC	RISC
Penekanan pada perangkat keras	Penekanan pada perangkat lunak
Termasuk instruksi kompleks multi-clock	Single clock, hanya sejumlah kecil instruksi yang multi clock
Memory-ke-memory: “LOAD” dan “STORE” saling bekerjasama	Register-ke-register: “LOAD” dan “STORE” adalah instruksi-instruksi yang terpisah
Ukuran kode kecil	Ukuran kode besar
Kecepatan instruksi rendah	Kecepatan instruksi tinggi
Transistor digunakan untuk menyimpan instruksi-instruksi kompleks	Transistor banyak digunakan untuk register memori

2.2.1 Mikrokontroler AVR ATmega162

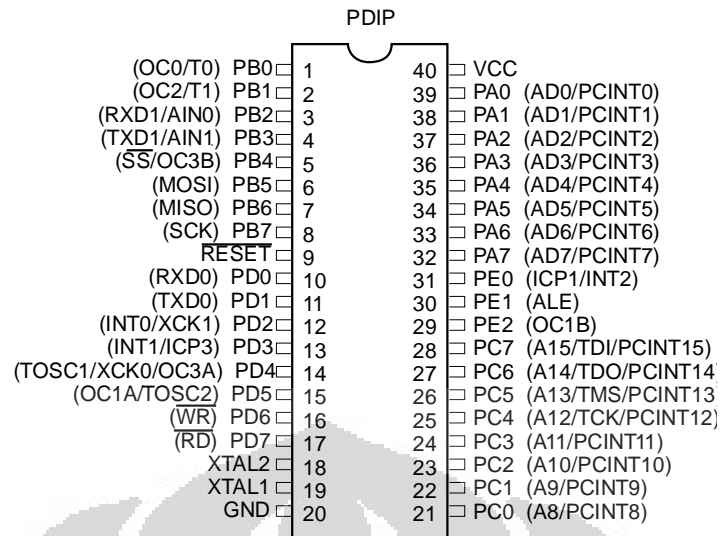
Mikrokontroler ATmega162 adalah mikrokontroler CMOS 8-bit yang diproduksi oleh ATMEL. Atmega162 memiliki arsitektur AVR dan tergolong RISC. Mikrokontroler AVR itu sendiri dikelompokkan menjadi beberapa kelas yaitu TinyAVR, MegaAVR, AVR XMEGA, AVR32 UC3 dan AVR32 AP7 [3]. Perbedaan dari jenis-jenis dari mikrokontroler tersebut terletak dari fasilitas atau fitur-fiturnya.

ATMega162 memiliki 131 buah set instruksi dimana hampir semuanya dieksekusi dalam satu clock. Selain itu mikrokontroler ini memiliki 32×8 general purpose register dan dapat mengeksekusi instruksi hingga 16 MIPS pada clock 16 MHz. Beberapa fitur yang dimiliki ATMega162 adalah [4]:

- a) Saluran I/O sebanyak 35 buah, yaitu port A, B, C, D dan E.
- b) Empat buah timer/counter dengan kemampuan perbandingan
- c) *real time counter* dengan osilator terpisah
- d) SRAM sebesar 1 Kbyte
- e) enam kanal PWM
- f) dua USART
- g) antarmuka SPI
- h) komparator analog
- i) *watchdog timer* dengan osilator terpisah

2.2.2 Konfigurasi Pin ATMega162

Mikrokontroler AVR ATMega162 mempunyai pin-pin yang mempunyai fungsi tertentu seperti yang ditunjukkan pada Gambar 2.4.



Gambar 2.4 Pin-pin pada ATMega162 [4]

Adapun fungsi dari pin-pin yang terdapat pada Mikrokontroler ATMega162 dijelaskan sebagai berikut [4]:

- a. VCC, suplai tegangan digital
- b. GND, pin ground
- c. PORT A (PA0 – PA7). PORT A merupakan port I/O 8-bit *bidirectional* yang dilengkapi dengan resistor *pull-up* internal (dapat dipilih untuk tiap bit). Selain sebagai port I/O, PORT A juga mempunyai fungsi lain seperti antarmuka memori eksternal dan pin change interrupt.
- d. PORT B (PB0 – PB7). PORT B merupakan port I/O 8-bit *bidirectional* dengan resistor *pull-up* internal (dapat dipilih untuk tiap bit). Selain itu, PORT B juga mempunyai fungsi lain yaitu *Serial Peripheral Interface (SPI)*, *Analog Comparator*, input/output *Timer/Counter*.
- e. PORT C (PC0 – PC7). PORT C merupakan port I/O 8-bit *bidirectional* dengan resistor *pull-up* internal (dapat dipilih untuk tiap bit). Selain itu PORT C juga mempunyai fungsi lain yaitu JTAG, antarmuka memori eksternal, dan *pin change interrupt*.
- f. PORT D (PD0 – PD7). PORT D merupakan port I/O 8-bit *bidirectional* dengan resistor *pull-up* internal (dapat dipilih untuk tiap

bit). Selain itu PORT D juga mempunyai fungsi lain yaitu USART, interupsi eksternal, *strobe* memori eksternal, *timer/counter*.

- g. PORT E (PE0 – PE2) PORT E merupakan port I/O 3-bit *bidirectional* dengan resistor *pull-up* internal (dapat dipilih untuk tiap bit). Selain itu PORT E juga mempunyai fungsi lain yaitu timer/counter, *latch enable* memori eksternal, interupsi eksternal.
- h. RESET, berfungsi untuk mereset mikrokontroler jika diberikan sinyal active low dalam selang waktu tertentu
- i. XTAL1, input ke *inverting oscillator amplifier* dan ke rangkaian detak internal
- j. XTAL2, output dari *inverting oscillator amplifier*.

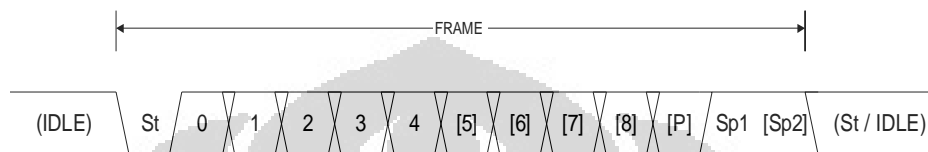
2.3 Komunikasi Serial

Komunikasi serial adalah komunikasi yang dalam prosesnya mengirimkan data 1 bit pada satu waktu secara berurutan di dalam sebuah kanal. Pada mikrokontroler AVR, komunikasi serial menggunakan fasilitas USART (*Universal Synchronous and Asynchronous Receiver Transmitter*). Fasilitas USART sangat berguna dalam berbagai macam aplikasi yang berhubungan dengan antarmuka komunikasi serial dengan PC atau dengan mikrokontroler lain. Ada dua macam cara pengiriman data secara *serial* yaitu *synchronous* dan *asynchronous*. Kedua cara tersebut dibedakan oleh penggunaan *clock* (detak) yang dipakai dalam pengiriman data. Pada *synchronous* detak dikirim bersama-sama dengan data serial. Pada *asynchronous* detak dihasilkan oleh masing-masing perangkat pengirim dan penerima [2].

Data serial mempunyai format frame yang terdiri dari 1 karakter data bit, bit sinkronisasi yang terdiri dari start dan stop, dan bit parity untuk pengecekan error yang dapat dipilih atau tidak. Frame data serial data mempunyai format dan kesemua 30 kombinasinya dapat diterima yaitu [4]:

- a) 1 bit start
- b) 5, 6, 7, 8, atau 9 bit data
- c) *Parity* genap, ganjil, atau tanpa *parity*
- d) 1 atau 2 bit stop

Sebuah frame dimulai dengan bit start diikuti oleh data atau karakter yang dimulai dari *least significant bit* hingga ke *most significant bit*, kemudian diikuti oleh parity bit (opsional), kemudian bit stop. Setiap kali satu frame berhasil dikirimkan, komunikasi serial dapat langsung mengirimkan data berikutnya atau diset ke dalam keadaan *idle (high) state*. Gambar 2.5 menunjukkan format frame pada komunikasi serial dimana bit yang ditandai dengan kurung siku adalah opsional.



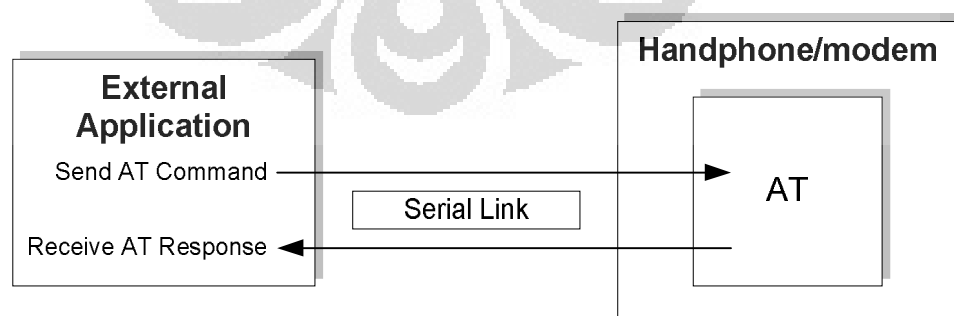
Gambar 2.5 Format frame pada komunikasi serial [4]

Keterangan [4]:

St	bit start, selalu low
(n)	data bit (0 ke 8)
P	bit parity
Sp	bit stop, selalu high
IDLE	tidak ada transfer pada jalur komunikasi (Rx dan Tx). Idle selalu high

2.4 AT Command

AT command adalah sebuah bahasa atau instruksi yang digunakan untuk komunikasi antara telepon seluler atau modem dengan komputer, mikrokontroler dan perangkat lainnya melalui melalui link serial seperti pada Gambar 2.6.



Gambar 2.6 AT Command

Komputer atau mikrokontroler dapat memberikan *AT command* kepada telepon seluler melalui kabel atau *bluetooth*. Dengan penggunaan *AT command*,

komputer atau mikrokontroler dapat melakukan suatu perintah seperti mengirim pesan, membaca pesan, menambah item pada daftar telepon dan sebagainya. *AT command* merupakan pengembangan dari perintah yang dapat diberikan kepada modem Hayes. Dinamakan *AT command* karena semua perintah diawali dengan karakter A dan T yang merupakan singkatan dari *Attention*.

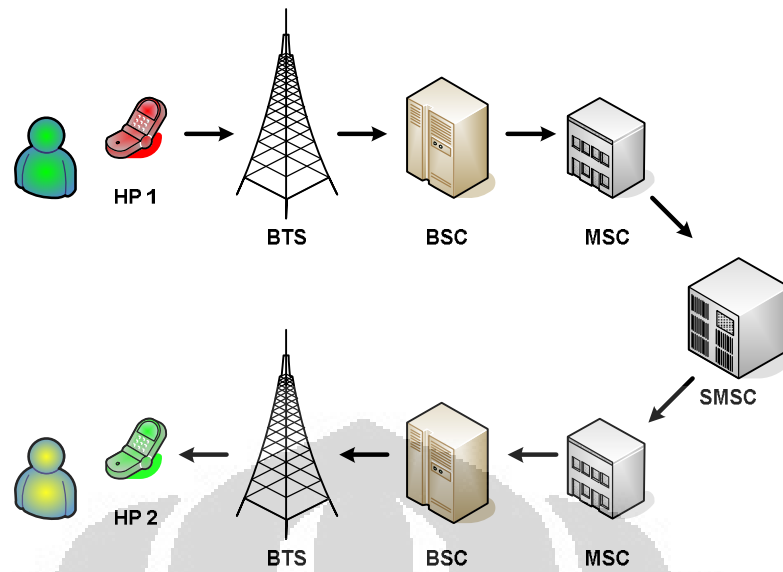
Tiap telepon seluler atau modem mempunyai AT Command yang sama. Namun ada beberapa instruksi AT Command yang ditambahkan sendiri pada ponsel atau modem oleh vendor pembuatnya. Instruksi AT Command dikelompokkan menjadi empat jenis seperti yang ditunjukkan pada Tabel 2.3.

Tabel 2.3 Jenis-jenis perintah AT Command [2]

Perintah	AT Command	Keterangan
Read Command	AT+Cxxx?	Perintah ini digunakan untuk mengetahui nilai parameter yang sedang aktif pada terminal.
Test Command	AT+Cxxx=?	Perintah ini digunakan untuk mengetahui parameter-parameter dan jangkauan nilainya berdasarkan write command atau proses internal.
Write Command	AT+Cxxx=<...>	Perintah ini digunakan untuk mengubah nilai parameter.
Execute Command	AT+Cxxx	Perintah ini untuk mengetahui nilai parameter dari terminal yang tidak dapat diubah nilainya.

2.5 Short Message Service (SMS)

Short Message Service (SMS) adalah protokol layanan pertukaran pesan text singkat (sebanyak 160 karakter per pesan) antar telepon. SMS ini pada awalnya adalah bagian dari standar teknologi seluler GSM, yang kemudian juga tersedia di teknologi CDMA, telepon rumah PSTN, dan lainnya. Gambar 2.7 menunjukkan alur pengiriman SMS.



Gambar 2.7 Alur pengiriman SMS

Keterangan:

BTS: *Base Transceiver Station*

BSC: *Base Station Controller*

MSC: *Mobile Switching Center*

SMSC: *Short Message Service Center*

Alur pengiriman SMS adalah sebagai berikut [6]:

Ketika pengguna mengirim SMS, maka pesan dikirim ke MSC melalui jaringan seluler yang tersedia yang meliputi tower BTS yang sedang menangani komunikasi pengguna, lalu ke BSC, kemudian sampai ke MSC. MSC kemudian meneruskan lagi SMS ke SMSC untuk disimpan. SMSC kemudian mengecek melalui HLR (*Home Location Register*) untuk mengetahui apakah *handphone* tujuan sedang aktif dan dimana lokasi *handphone* tujuan tersebut.

Jika *handphone* sedang tidak aktif maka pesan tetap disimpan di SMSC itu sendiri, menunggu MSC memberitahukan bahwa *handphone* sudah aktif kembali untuk kemudian SMS dikirim dengan batas maksimum waktu tunggu yaitu *validity period* dari pesan SMS itu sendiri. Jika *handphone* tujuan aktif maka pesan disampaikan MSC lewat jaringan yang sedang menangani penerima (BSC dan BTS).

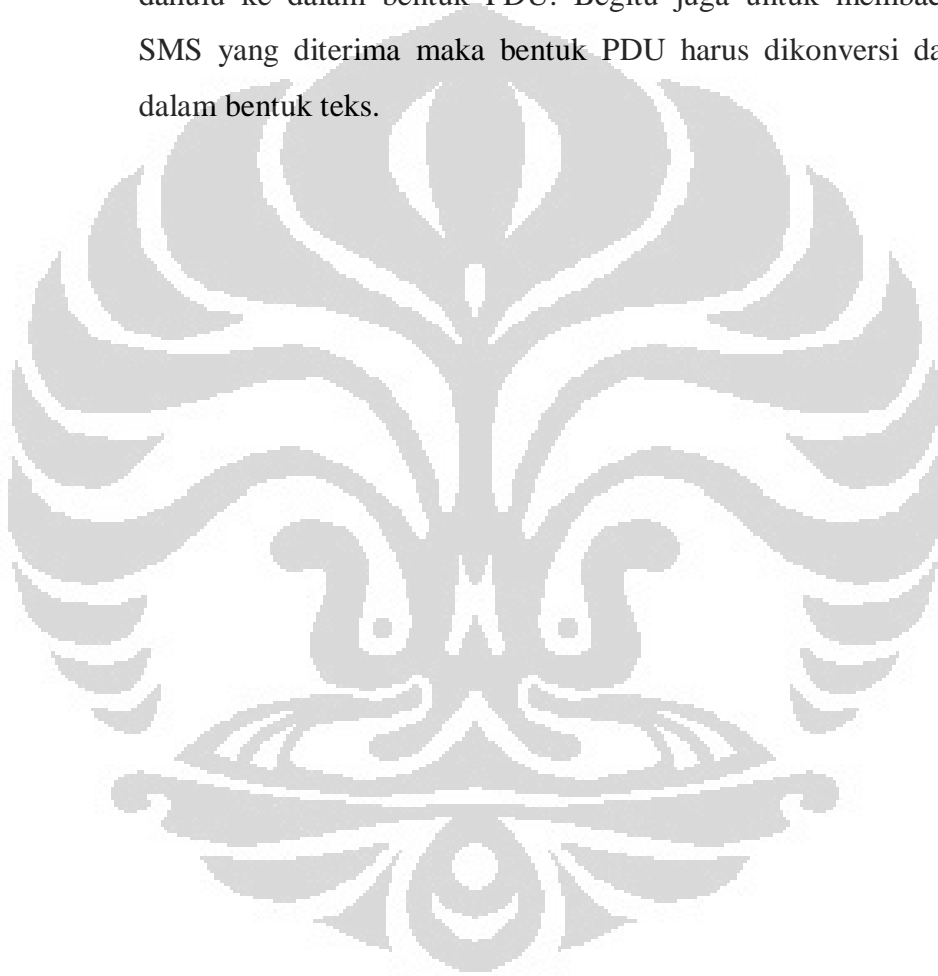
Pada AT Command, ada dua mode pengiriman SMS yaitu:

- a) Mode Teks

Mode teks adalah metode pengiriman pesan yang termudah. Pada mode ini, pesan yang dikirim tidak dilakukan konversi. Teks yang dikirim tetap dalam bentuk aslinya.

b) Mode Protocol Data Unit (PDU)

Mode PDU adalah format pesan dalam heksadesimal oktet dan semi-oktet dengan panjang mencapai 160 (7-bit) atau 140 (8-bit) karakter. Untuk mengirimkan pesan SMS maka teks yang harus dikonversikan dahulu ke dalam bentuk PDU. Begitu juga untuk membaca pesan SMS yang diterima maka bentuk PDU harus dikonversi dahulu ke dalam bentuk teks.



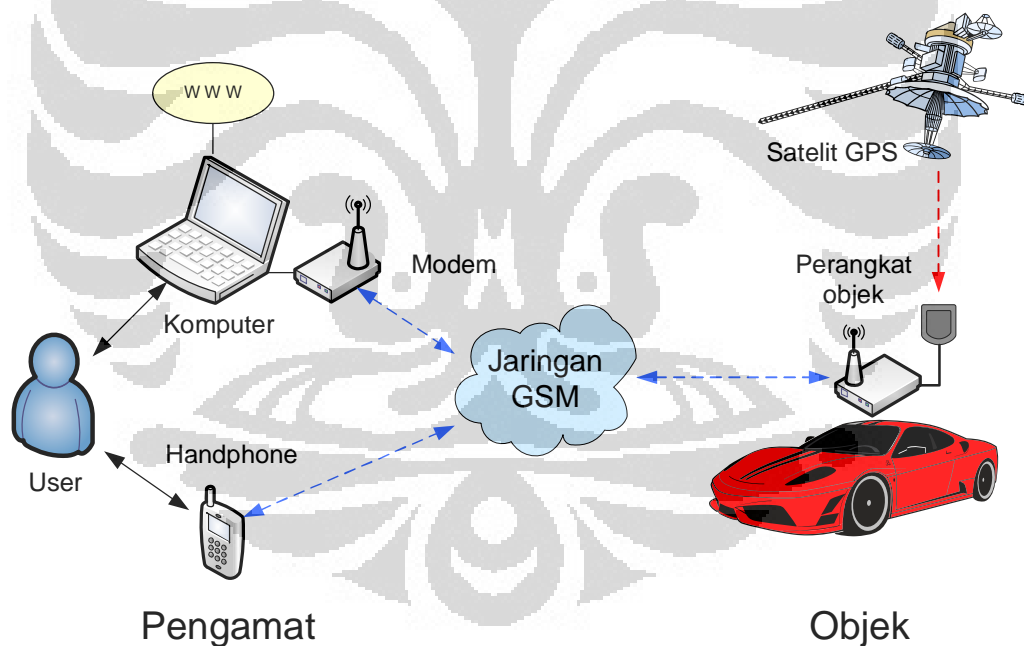
BAB 3 PERANCANGAN SISTEM

3.1 Desain Sistem

Skripsi ini bertujuan untuk merancang sebuah sistem penjejak dan pengaman pada kendaraan bermotor dengan memanfaatkan SMS dan GPS. Fungsi utama dari sistem penjejakan dan pengaman ini adalah:

1. Dapat memberikan posisi kendaraan kepada user melalui SMS
2. Dapat mengendalikan mesin dan klakson melalui SMS
3. Mengaktifkan sistem alarm dan memberi SMS peringatan kepada pemilik kendaraan ketika terjadi pencurian.

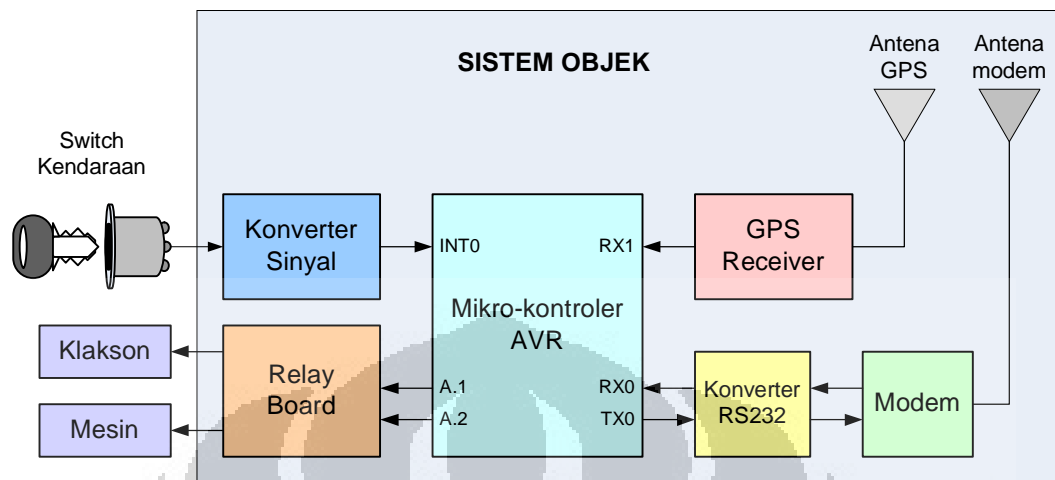
Secara garis besar, sistem penjejak dan keamanan ini terbagi menjadi dua bagian yaitu objek dan pengamat, seperti yang ditunjukkan pada Gambar 3.1.



Gambar 3.1 Diagram sistem secara keseluruhan

Sistem objek adalah sistem yang ditempatkan pada kendaraan dan berperan untuk menerima dan melaksanakan perintah dari sistem pengamat seperti memberikan data posisi, mengendalikan mesin dan klakson, mengaktifkan sistem alarm dan mengirimkan pesan peringatan kepada pemilik kendaraan jika terjadi

pencurian. Sub sistem ini terdiri dari GPS *receiver*, mikrokontroler, relay, konverter sinyal, dan modem seperti pada Gambar 3.2.



Gambar 3.2 Blok diagram sistem objek

Penjelasan dari komponen-komponen penyusun sistem objek adalah sebagai berikut:

1. GPS *receiver*

GPS *receiver* berfungsi untuk mendapatkan posisi dari satelit GPS dan mengolahnya menjadi data output GPS yang kemudian diproses oleh mikrokontroler

2. Modem

Modem berfungsi untuk komunikasi antara sistem objek dengan sistem pengamat melalui SMS.

3. Konverter RS232

Konverter RS232 berfungsi untuk mengubah level logika *link serial* TTL menjadi RS232 dan sebaliknya antara modem dengan mikrokontroler.

4. Mikrokontroler AVR

Mikrokontroler berfungsi sebagai pusat pemrosesan dari sistem objek.

5. Konverter sinyal

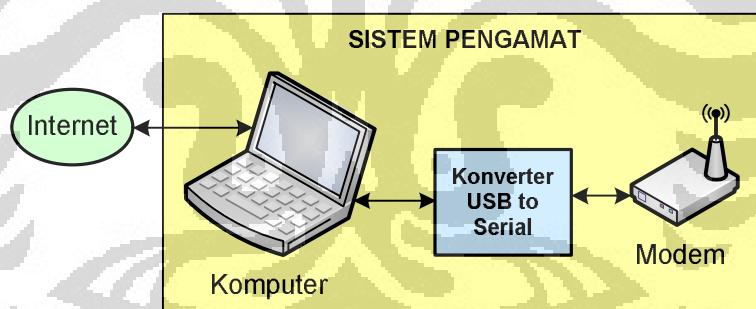
Mikrokontroler mendeteksi perubahan tegangan pada bagian switch untuk sistem alarmnya. Karena tegangan pada switch adalah 12 V, *switch* tidak dapat langsung dihubungkan ke mikrokontroler. Bagian

ini berfungsi untuk mengubah tegangan pada switch kendaraan, yaitu dari 12 V menjadi 5 V agar dapat bekerja dengan mikrokontroler.

6. *Relay board*

Relay berperan sebagai switch untuk mengendalikan komponen kendaraan. *Relay board* ini terdiri dari dua buah relay yang salah satunya terhubung ke bagian mesin, dan yang satunya lagi terhubung ke bagian klakson.

Sistem pengamat adalah sistem yang digunakan oleh *user*. Sistem ini berfungsi untuk memonitor dan mengendalikan sistem objek. Komunikasi antara sistem pengamat dan sistem pengendali dilakukan dengan menggunakan SMS melalui jaringan GSM. Pada bagian pengamat, *user* diberi kebebasan untuk memilih apakah menggunakan *handphone* atau komputer untuk berinteraksi dengan sistem objek. Namun jika menggunakan *handphone*, *user* tidak dapat melihat posisi kendaraan dalam bentuk peta. Jika sistem pengamat menggunakan komputer, maka komponen penyusunnya adalah komputer dengan akses internet, konverter USB to serial (opsional) dan modem, seperti pada Gambar 3.3.



Gambar 3.3 Blok diagram sistem pengamat yang menggunakan komputer

3.2 Prinsip Kerja Sistem

Prinsip kerja sistem ini terbagi menjadi dua kondisi, yaitu kondisi ketika pengamat mengirimkan SMS ke sistem objek dan kondisi ketika terjadi pencurian.

Kondisi yang pertama adalah ketika pengamat mengirimkan SMS ke sistem objek melalui *handphone* atau perangkat lunak di komputer. Mikrokontroler pada sistem objek menerima adanya SMS masuk melalui *link* serial USART0. Jika ada SMS masuk, mikrokontroler akan menyeleksi isi SMS tersebut, apakah sesuai dengan format atau tidak. Jika isi SMS tidak sesuai dengan format maka

mikrokontroler akan mengabaikannya. Jika isi SMS sesuai dengan format, maka mikrokontroler akan menjalankan perintah SMS tersebut.

GPS *receiver* menerima sinyal dari satelit GPS. GPS receiver mengeluarkan output berupa kalimat NMEA 0183. Output GPS tersebut kemudian masuk ke mikrokontroler melalui *link* serial USART1. Jika isi SMS berupa permintaan posisi, mikrokontroler akan menyeleksi data keluaran GPS *receiver* hingga menemukan data yang dibutuhkan. Selanjutnya, data GPS tersebut dikirim kembali ke pengirim melalui SMS. Jika isi SMS berupa pengendalian relay, mikrokontroler akan mengendalikan relay yang bersangkutan. Begitu pula dengan SMS yang lain akan diproses oleh mikrokontroler sesuai dengan format SMS.

Kondisi kedua adalah ketika terjadi pencurian, yaitu ketika *switch* kendaraan diputar ke posisi *on*. Saat *switch* kendaraan diputar ke posisi *on*, aki memberikan tegangan ke komponen-komponen kendaraan berupa lampu, klakson, dan lainnya termasuk *regulator* pada mikrokontroler. Regulator akan mengubah tegangan 12 V dari aki menjadi 5 V sehingga dapat bekerja dengan mikrokontroler. Ketika mikrokontroler mendeteksi adanya perubahan tegangan dari 0 ke 5 V yang artinya kendaraan telah dihidupkan, mikrokontroler akan menjalankan interupsi yaitu mengirimkan SMS peringatan kepada pemilik kendaraan, memutuskan aliran listrik ke mesin dan menghidupkan klakson (alarm).

3.3 Format SMS

Agar mikrokontroler mengerti keinginan *user* melalui SMS, maka mikrokontroler perlu menetapkan format tertentu agar SMS yang dapat diterjemahkan. Tiap format SMS mempunyai fungsi masing-masing. Format SMS tersebut adalah sebagai berikut:

1. **posisi**
Perintah ini berfungsi untuk meminta data GPS dari sistem objek.
2. **mesin on** dan **mesin off**
Perintah ini berfungsi untuk mengendalikan mesin agar dapat atau tidak dapat digunakan.
3. **klakson on** dan **klakson off**
Perintah ini berfungsi untuk menghidupkan atau mematikan klakson.
4. **alarm on** dan **alarm off**

Perintah ini berfungsi untuk mengaktifkan atau menonaktifkan alarm.

5. **status**

Perintah ini berfungsi untuk melihat status dari sistem objek berupa kondisi mesin, klakson, alarm dan lainnya.

6. **reset**

Perintah ini berfungsi untuk mengembalikan kondisi mesin, klakson dan status kecurian ke kondisi normal yaitu mesin dapat digunakan, klakson mati dan status kecurian adalah *false*.

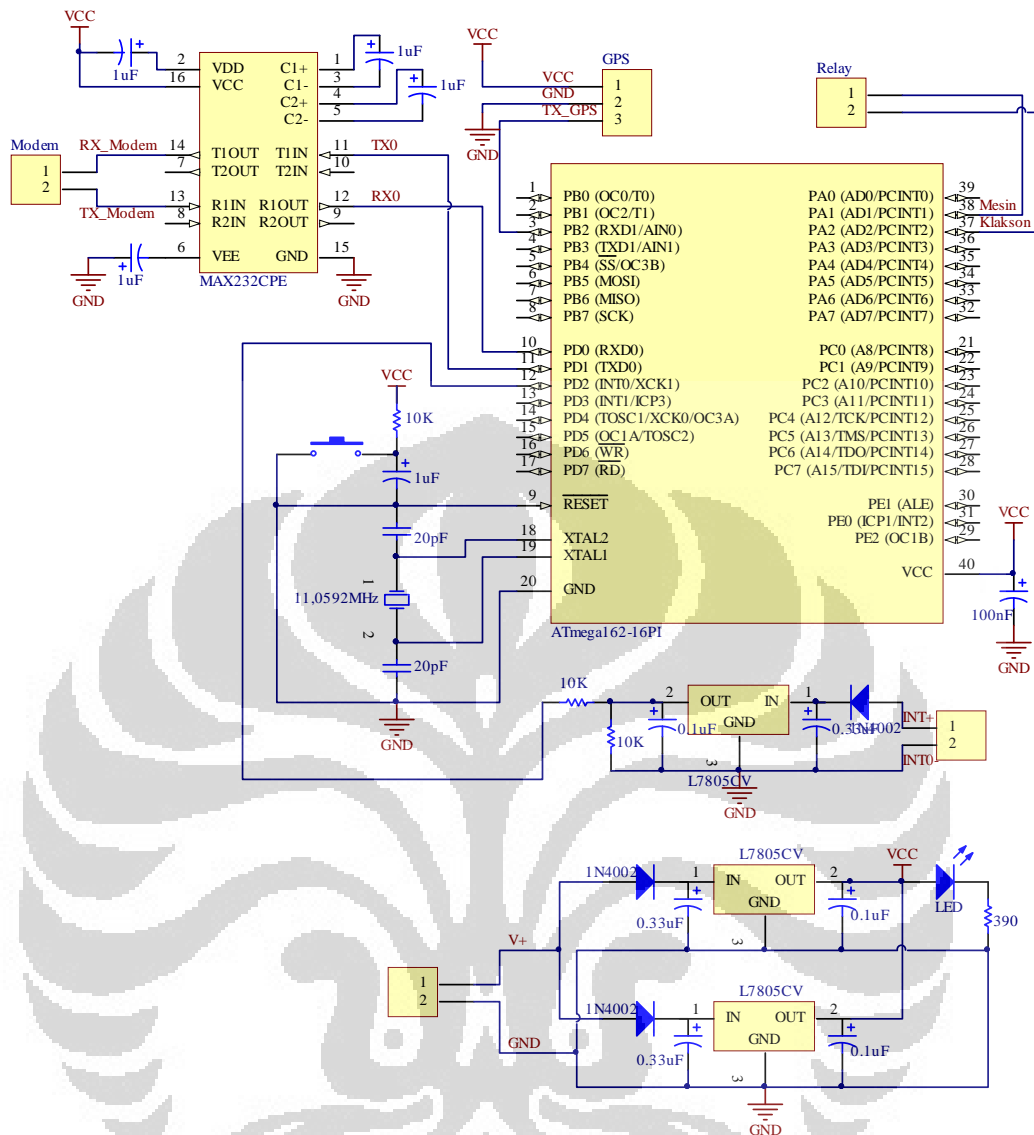
7. **no “xxxxxxxx”**

Perintah ini berfungsi untuk menetapkan nomor *handphone* yang akan dikirimkan SMS ketika pencurian terjadi. “xxxxxxxx” adalah nomor *handphone* pemilik kendaraan yang dapat ditulis dengan menggunakan penomoran internasional seperti “+628956xxxx” atau menggunakan penomoran lokal seperti “0856xxxx”.

3.4 Perancangan Perangkat Keras

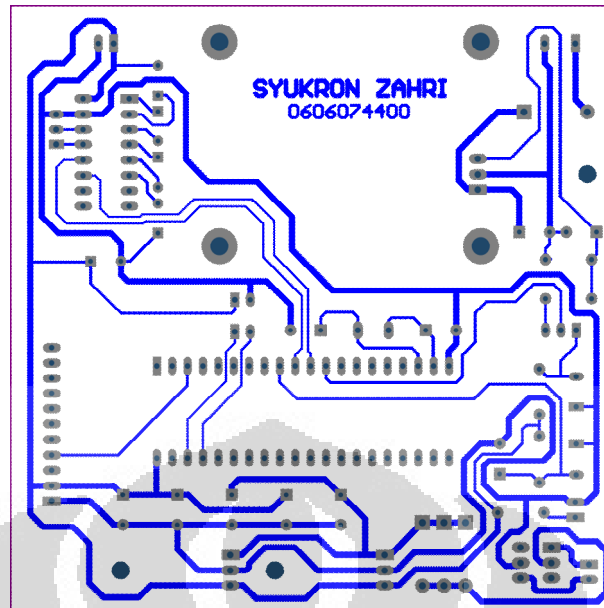
Untuk sistem pengamat, perangkat keras yang digunakan terdiri dari modem dan komputer atau. Sistem pengamat bisa menggunakan *handphone* saja, tetapi dengan kondisi seperti ini *user* tidak dapat melihat posisi kendaraan dalam bentuk peta.

Untuk sistem objek, perangkat keras pada sistem objek secara keseluruhan dibuat berdasarkan komponen-komponen dari Gambar 3.2. Skematik dari perangkat keras penyusun sistem objek ditunjukkan pada Gambar 3.4.

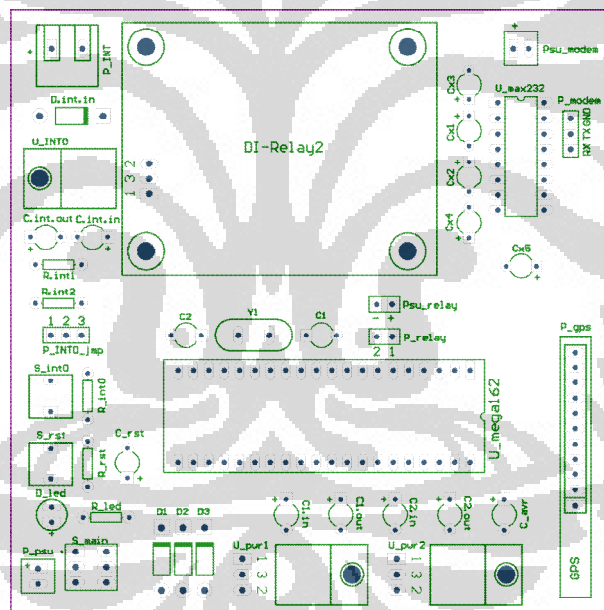


Gambar 3.4 Skematik sistem minimum

Layout PCB dan tata letak komponen pada PCB sistem minimum ditunjukkan pada Gambar 3.5 dan 3.6.



Gambar 3.5 Layer bawah PCB sistem minimum



Gambar 3.6 Tata letak komponen pada PCB

3.4.1 Mikrokontroler ATmega162

Mikrokontroler berfungsi sebagai pusat pemrosesan pada sistem objek. Mikrokontroler yang digunakan adalah mikrokontroler AVR tipe ATmega162. Untuk pendetak mikrokontroler digunakan osilator eksternal berupa osilator kristal yang mempunyai frekuensi 11,0592 MHz. Port serial modem beroperasi pada tegangan RS232, sedangkan port serial mikrokontroler bekerja pada

tegangan TTL. Oleh karena itu diperlukan konverter tegangan pada *link* serial dari TTL ke RS232 dan sebaliknya yaitu dengan menggunakan IC MAX232.

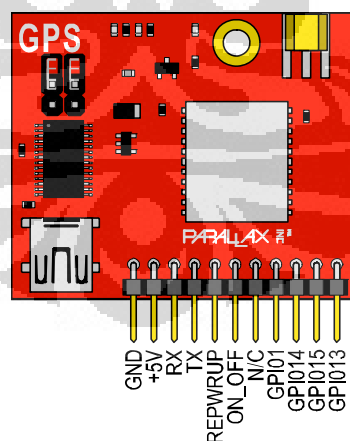
Penggunaan port pada mikrokontroler ditunjukkan pada Tabel 3.1.

Tabel 3.1 Penggunaan port mikrokontroler

No.	Nama port	Dihubungkan dengan
1	A.1	Relay mesin
2	A.2	Relay klakson
3	D.0 (RX0)	TX modem
4	D.1 (TX0)	RX modem
5	B.2 (RX1)	TX GPS Receiver
6	D.2 (INT0)	Regulator atau <i>push button</i>

3.4.2 GPS Receiver

GPS receiver yang digunakan pada sistem adalah GPS RXM-SG yang diproduksi oleh Parallax Inc. GPS receiver ini menggunakan chip SiRFStar III sebagai prosesornya dan memerlukan antena eksternal untuk penerima sinyal GPS. Selain itu GPS receiver ini juga dilengkapi dengan baterai *backup* CR2032 catu daya pada SRAM dan RTC. Bentuk fisik dari GPS Parallax RXM-SG dapat dilihat pada Gambar 3.7.



Gambar 3.7 GPS Parallax RXM-SG [7]

Output GPS receiver ini berupa data NMEA 0183 dan menggunakan antarmuka serial yang beroperasi pada tegangan CMOS. Output GPS receiver mempunyai berbagai pilihan kecepatan yaitu 4800, 9600, 19200, 19200, 38400,

dan 57600 bps. Pada perancangan sistem objek ini kecepatan yang digunakan adalah 9600 bps.

Seperti yang telah dijelaskan dalam Bab 2, kalimat NMEA 0183 output dari GPS receiver terdiri dari berbagai macam bentuk. Pada perancangan sistem ini, kalimat NMEA 0183 yang adalah tipe GPRMC (*Recommended minimum specific GNSS data*). Berikut ini adalah format dan penjelasan data dari tipe kalimat GPRMC:

\$GPRMC,hhmmss,A,ddmm.mmmm,S,dddmm.mmmm,E,x.xx,y.yy,ddmmyy,*hh

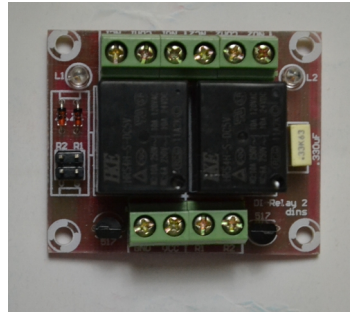
Keterangan:

\$GPRMC : *Recommended Minimum Specific GPS/TRANSIT Data*
hhmmss : waktu UTC (hh = jam, mm = menit, ss = detik)
A : Status (A = valid, V = invalid)
ddmm.mmmm : lintang (dd = derajat, mm.mmmm = menit)
S : N atas S (N = utara, S = selatan)
dddmm.mmmm : bujur (ddd = derajat, mm.mmmm = menit)
E : E atau W (E = timur, W = barat)
x.xx : kecepatan bergerak (dalam knots)
y.yy : arah pergerakan
ddmmyy : tanggal UTC (dd = tanggal, mm = bulan, yy = tahun)
***hh** : *checksum*

LED hijau menunjukkan status penguncian sinyal satelit. LED hijau yang berkedip dengan cepat menandakan GPS *receiver* belum mendapatkan posisi yang valid. LED yang berkedip lambat menandakan GPS *receiver* sudah mendapatkan posisi yang valid. Pin yang digunakan pada GPS *receiver* ini hanya tiga buah, yaitu pin GND, +5V, dan TX. Pin TX dihubungkan dengan pin RX1 pada mikrokontroler.

3.4.3 Relay

Relay yang digunakan pada sistem ini adalah *relay board* DI-Relay2 yang diproduksi oleh Depok Instruments. *Relay board* ini tersusun dari dua buah relay HKE tipe HRS4H-S-DC5V yang berjenis SPDT (*Single Pole Double Throw*) dan bekerja pada tegangan 5 V. Bentuk fisik dari *relay board* ini ditunjukkan oleh Gambar 3.8.



Gambar 3.8 DI-Relay2

Relay board dihubungkan ke port A, dimana pin A.1 mengatur relay mesin sedangkan pin A.2 mengatur relay bagian klakson.

3.4.4 Modem

Modem berfungsi untuk mengirimkan dan menerima SMS melalui jaringan GSM. Untuk mengetahui suatu modem atau *handphone* mendukung AT *Command* atau tidak, dapat diuji dengan menggunakan program HyperTerminal di komputer dengan mengetik karakter AT dan kemudian menekan *Enter*. Apabila mendapat respon “OK”, *handphone* atau modem tersebut sudah mendukung AT *Command*.

Modem yang digunakan pada sistem ini adalah modem Wavecom M1306B. Bentuk fisik modem Wavecom Fastrack M1306B ditunjukkan pada Gambar 3.9.



Gambar 3.9 Modem Wavecom Fastrack M1306B [8]

Beberapa AT *command* atau AT *response* penting untuk mengoperasikan modem ini adalah [5]:

1. AT+CMGR, untuk membaca SMS

2. AT+CMGS, untuk mengirim SMS
3. AT+CMGD, untuk menghapus SMS
4. +CMTI, indikator ada SMS masuk

Sebelum digunakan oleh mikrokontroler, modem harus disetting konfigurasi seperti pada Tabel 3.2.

Tabel 3.2 Konfigurasi modem pada mikrokontroler

Parameter	Nilai	AT Command	Penjelasan
<i>Baud rate</i>	9600	AT+IPR=9600	Mengubah kecepatan link serial modem menjadi 9600 bps. Kecepatan link serial bawaan dari produsen adalah 115200 bps
<i>Flow control</i>	None	AT+IFC=0,0	Tidak menggunakan <i>flow control</i> atau sinyal untuk tunggu
<i>Echo</i>	Off	ATE0	Menonaktifkan <i>echo</i> atau karakter berulang.
<i>Message format</i>	Text	AT+CMGF=1	SMS menggunakan mode teks. Mode 0 adalah mode PDU sedangkan mode 1 adalah mode teks.

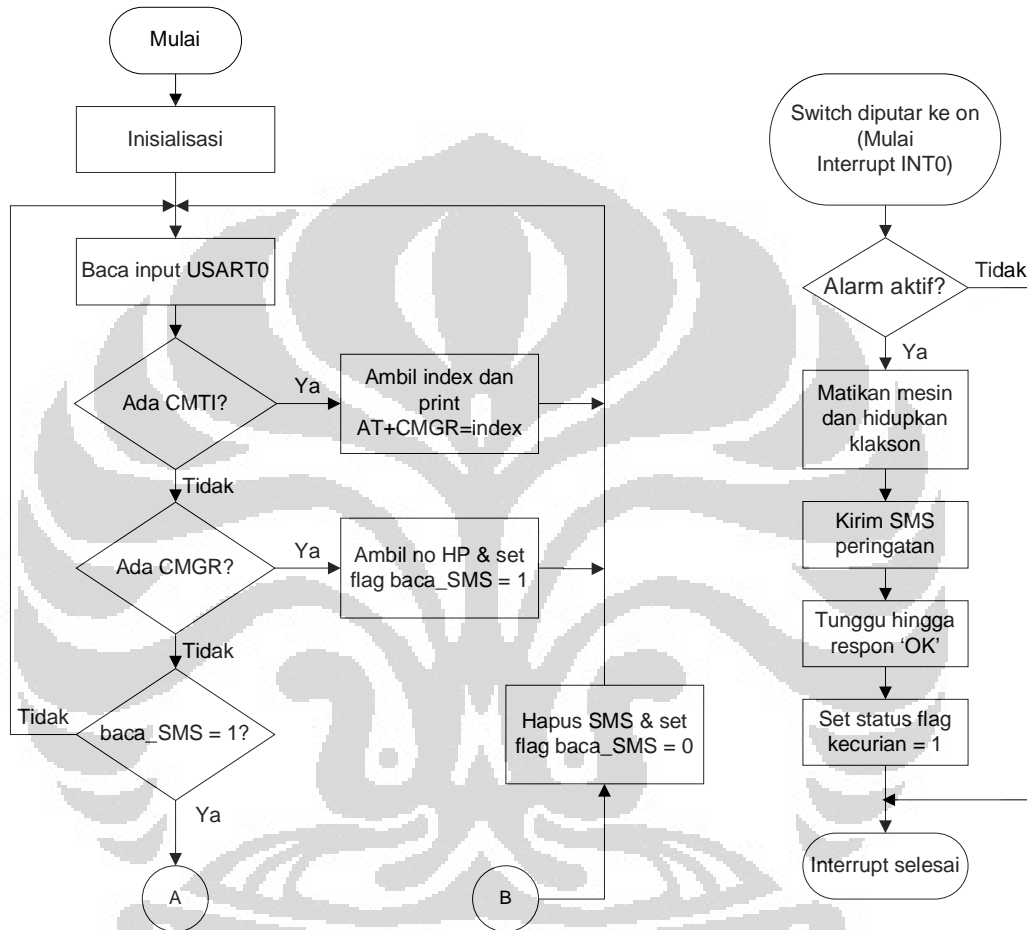
Setelah dilakukan setting seperti pada Tabel 3.2, konfigurasi disimpan dengan menggunakan perintah **AT&W**.

3.5 Perancangan Perangkat Lunak

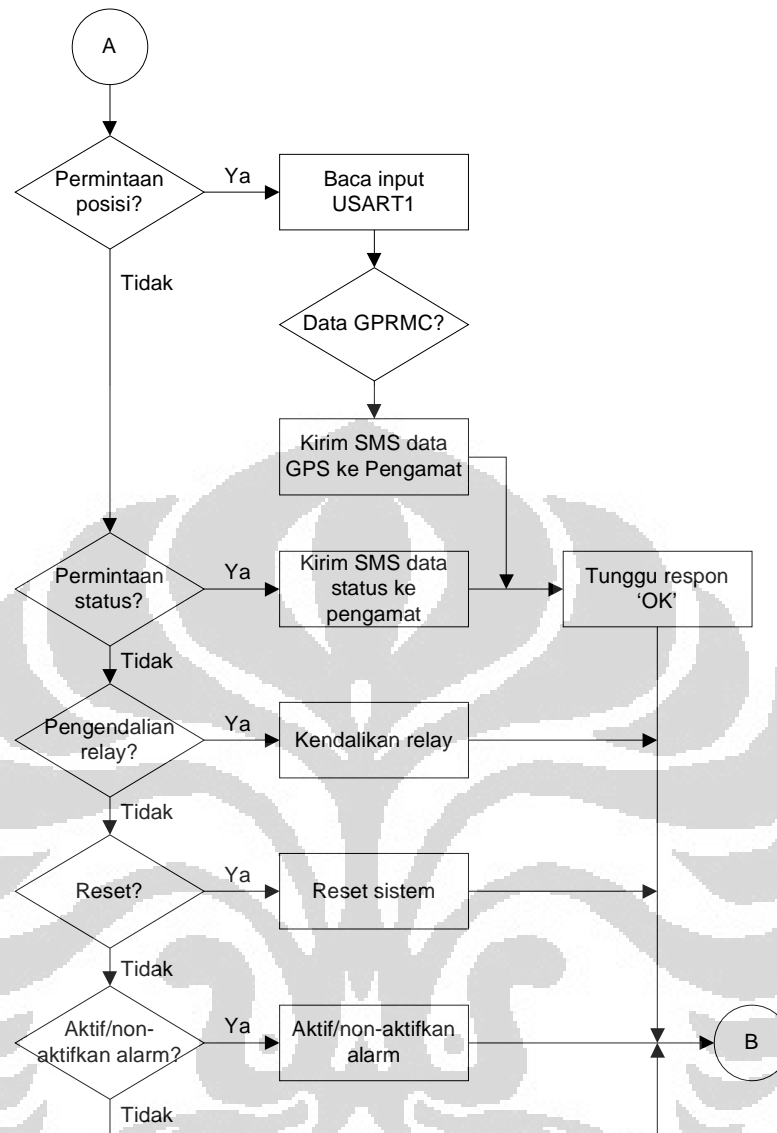
Pemrograman perangkat lunak dilakukan pada kedua sub sistem. Pada sistem objek, perangkat lunak dibuat untuk mikrokontroler ATmega162, sedangkan pada sistem pengamat, perangkat lunak dibuat untuk komputer.

3.5.1 Perancangan Perangkat Lunak pada Mikrokontroler

Perancangan perangkat lunak pada mikrokontroler menggunakan *compiler* BASCOM AVR. Algoritma program pada mikrokontroler ditunjukkan oleh Gambar 3.10 dan 3.11.



Gambar 3.10 Algoritma program pada mikrokontroler



Gambar 3.11 Algoritma pembacaan isi SMS

Tahap awal dari perangkat lunak mikrokontroler ini adalah inisialisasi mikrokontroler. Inisialisasi yang dilakukan adalah mengatur port serial yang digunakan. Mikrokontroler ATmega162 mempunyai dua port serial yang terhubung dengan GPS *receiver* dan dengan modem. Konfigurasi port serial untuk masing-masing perangkat adalah seperti pada Tabel 3.2:

Tabel 3.2 Konfigurasi Port Serial

Port	USART0 (modem)	USART1 (GPS Receiver)
Baud rate	9600	9600
Data bit	8	8
Bit stop	1	1
Parity	Tidak	Tidak
Receive	Ya	Ya
Transmit	Ya	Tidak
Rx Interrupt	Tidak	Tidak
Tx Interrupt	Tidak	Tidak

Selanjutnya, mikrokontroler akan menunggu input dari USART0. Ketika mikrokontroler mendapatkan input pada USART0, mikrokontroler akan melakukan seleksi terhadap input tersebut. Jika input tersebut mengandung kata +CMTI, itu tandanya ada SMS yang masuk. Mikrokontroler kemudian mengambil indeks dari kalimat +CMTI untuk mengetahui di indeks ke berapa pesan tersebut disimpan pada SIM card.

Setelah didapatkan indeks, mikrokontroler kemudian mengirimkan *command* AT+CMGR=indeks untuk membaca pesan. Ketika *command* AT+CMGR diberikan, modem akan merespon mikrokontroler dengan data berupa header dan isi SMS. *Header* SMS ditandai dengan kata +CMGR dan isi SMS diterima selanjutnya setelah karakter <CR> dan <LF> pada baris berikutnya.

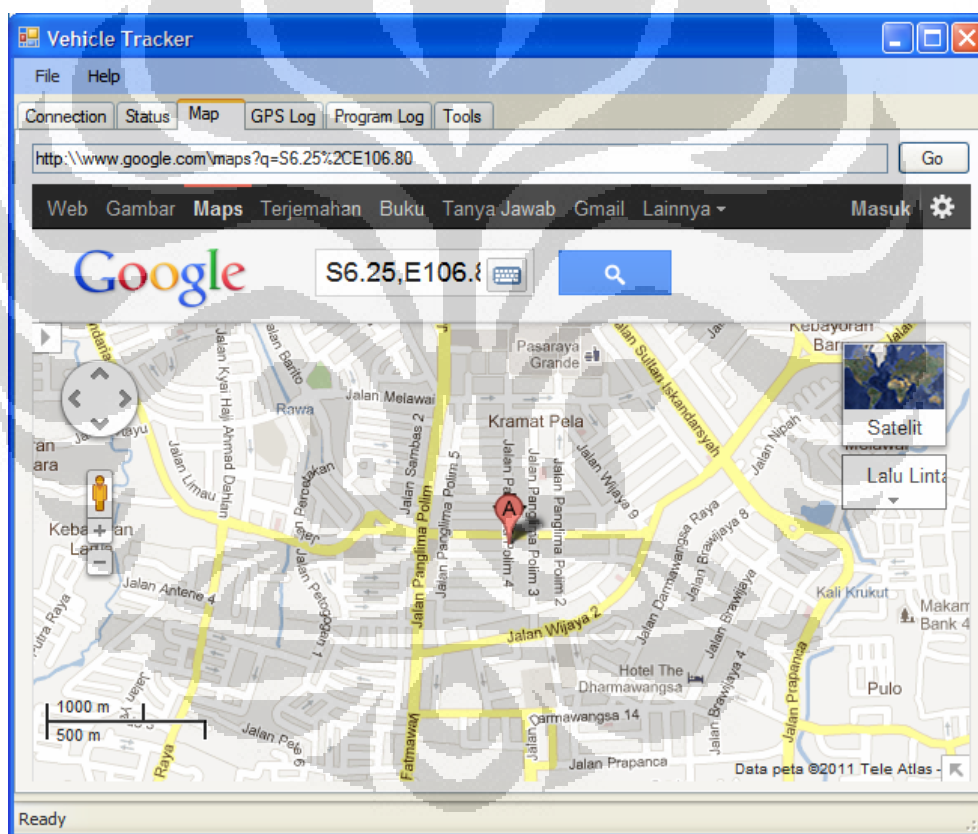
Nomor pengirim diambil dari *header* SMS dan digunakan untuk membalas si pengirim SMS. Selanjutnya Isi SMS diperiksa dan diseleksi apakah sesuai dengan format yang telah ditetapkan. Jika isi SMS sesuai dengan format yang ditetapkan maka mikrokontroler akan melaksanakan isi SMS tersebut seperti pada Gambar 3.11. Namun jika pesan yang diterima tidak sesuai dengan format yang ditetapkan maka mikrokontroler akan mengabaikannya. Setelah membaca isi SMS, mikrokontroler menghapus SMS tersebut.

Ketika *switch* diputar ke posisi on, maka listrik mengalir ke regulator. Kemudian regulator mengubah tegangan aki menjadi tegangan kerja mikrokontroler yaitu sebesar 5 V dan masuk ke pin INT0. Ketika mikrokontroler

mendeteksi adanya perubahan tegangan dari 0 ke 5 V, mikrokontroler akan menghentikan aktivitasnya dan menjalankan interupsi INTO. Interupsi yang dilakukan berupa mengirimkan SMS peringatan kepada pemilik kendaraan, yaitu nomor darurat yang telah ditetapkan sebelumnya.

3.5.2 Perancangan Perangkat Lunak pada Komputer

Perangkat lunak pada komputer dibuat dengan menggunakan *compiler* Visual Basic 2005 .NET. Perangkat lunak ini menggunakan Google Maps untuk menampilkan peta dan posisi sehingga perangkat lunak ini memerlukan koneksi internet. Untuk menjalankan perangkat lunak ini dibutuhkan .NET Framework versi 2. Tampilan perangkat lunak pada komputer ditunjukkan oleh Gambar 3.12.



Gambar 3.12 Tampilan perangkat lunak di komputer

Untuk pengaksesan port serial pada Visual Basic 2005 digunakan *class* SerialPort. *Class* SerialPort merupakan komponen dari .NET yang berfungsi untuk komunikasi serial pada komputer. Salah satu *method* yang dimiliki oleh *class* SerialPort ini adalah ReadLine, yaitu membaca *buffer* dari port serial hingga karakter CR dan LF (*Carriage Return* dan *Line Feed*). *Method* ReadLine ini harus

dijalankan secara terpisah dari alur program utama karena *method* ini dalam prosesnya membaca *buffer* terus menerus hingga menemukan karakter CR dan LF sehingga dapat membuat program berhenti (*hang*). Oleh karena itu *method* *ReadLine* perlu dijalankan dalam sebuah *thread*.

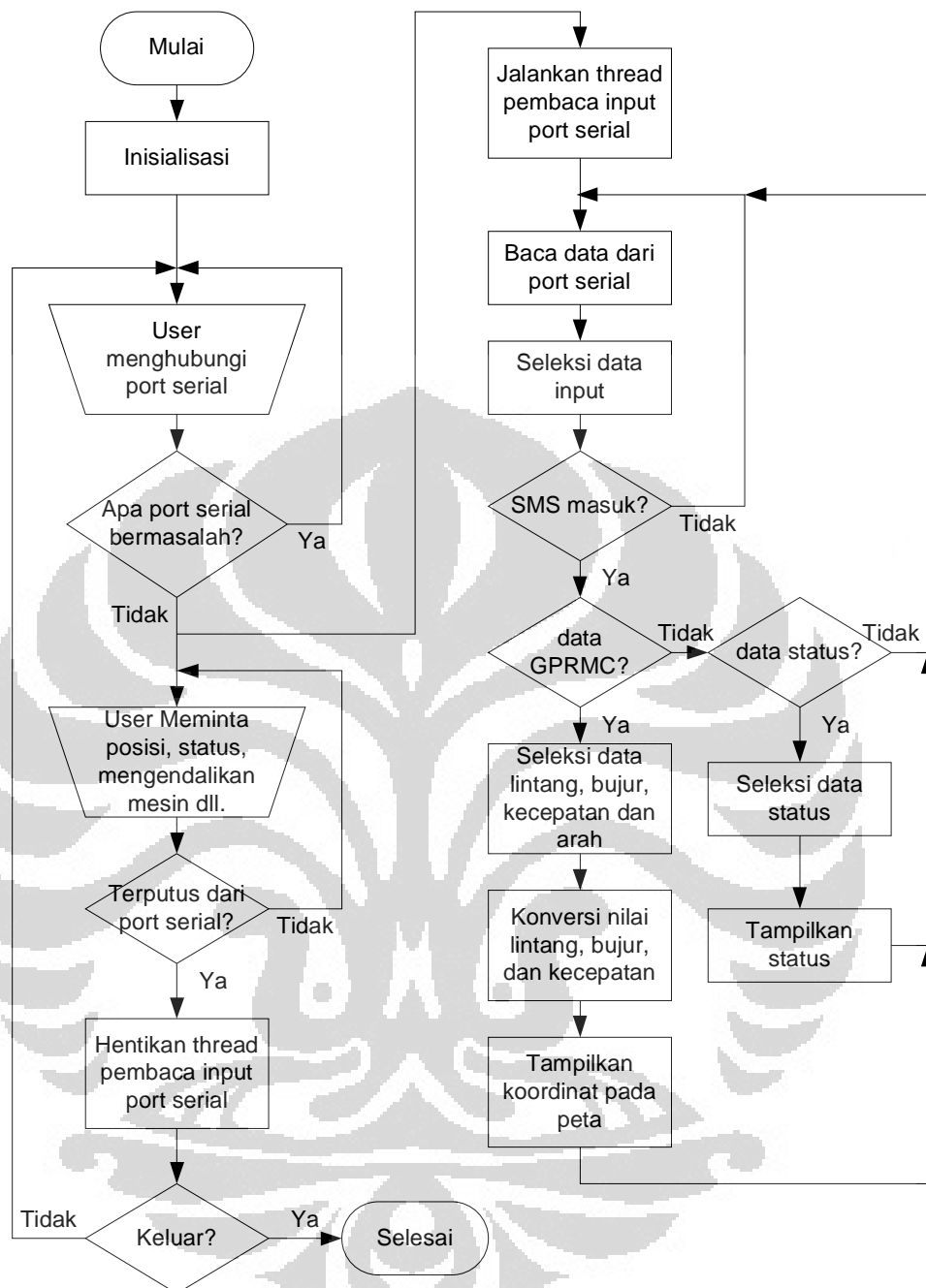
Pemetaan posisi dilakukan dengan menggunakan Google Maps dengan memanfaatkan *class* *WebBrowser*. Untuk menempatkan suatu titik pada Google Maps, maka cara yang termudah adalah dengan menggunakan URL dengan format sebagai berikut:

`http://maps.google.com/maps?q=[LONG]%2C[LATT]`

dimana [LONG] adalah nilai *longitude* atau lintang dan [LATT] adalah nilai *latitude* atau bujur.

Data posisi yang dikirimkan dari mikrokontroler ke komputer adalah data mentah kalimat NMEA 0183 yang mengandung GPRMC. Lintang dan bujur pada kalimat NMEA 0183 mempunyai bentuk *datum* WGS 84 yaitu dinyatakan dalam derajat menit. Karena Google Maps tidak mendukung pemetaan dalam bentuk derajat menit, maka data lintang dan bujur harus dikonversi dahulu ke menjadi bentuk desimal.

Algoritma program pada komputer ditunjukkan oleh Gambar 3.13.

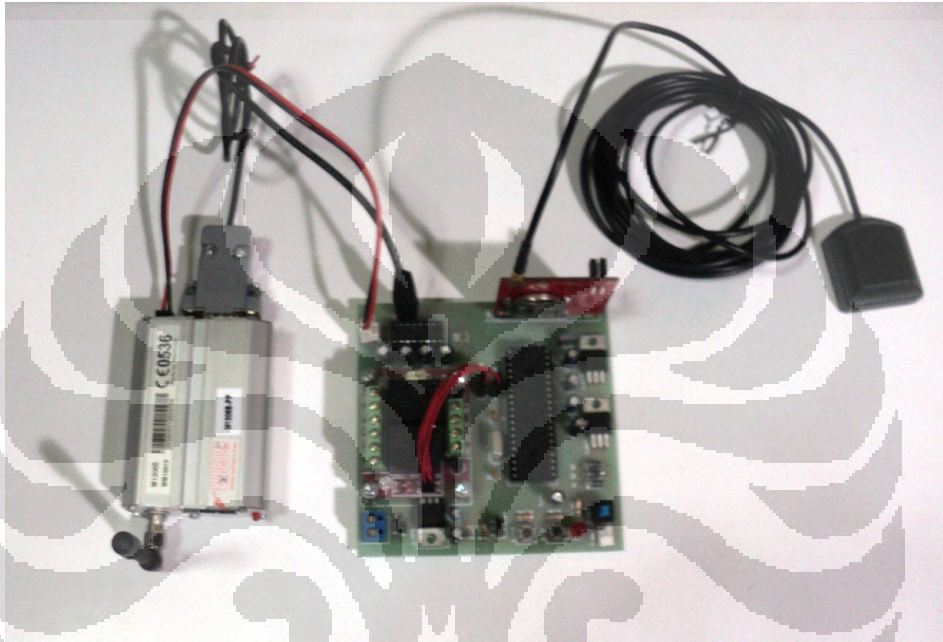


Gambar 3.13 Algoritma program pada komputer

BAB 4 PENGUJIAN SISTEM DAN ANALISIS

4.1 Implementasi Sistem

Setelah dilakukan perancangan sistem pada Bab 3 selanjutnya dilakukan implementasi sistem objek seperti yang diperlihatkan pada Gambar 4.1.



Gambar 4.1 Implementasi sistem objek

4.2 Pengujian Sistem

Untuk mengetahui apakah sistem yang telah dirancang dapat bekerja sesuai dengan harapan, maka dilakukan beberapa pengujian sebagai berikut:

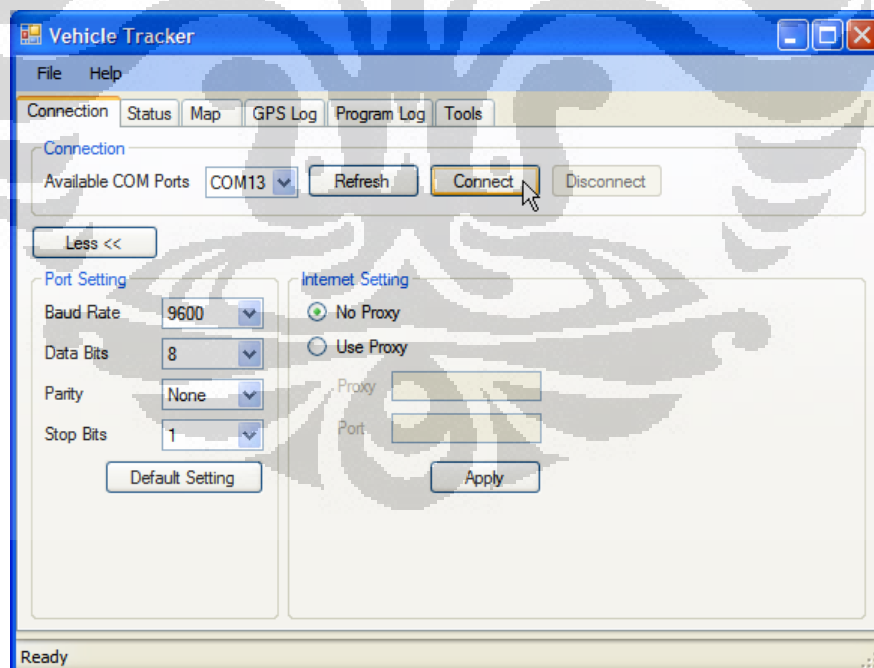
- a. Menampilkan posisi
- b. Mengendalikan mesin dan klakson
- c. Menguji sistem alarm
- d. Menampilkan status
- e. Mereset sistem

Pengujian dilakukan dengan dua cara yaitu mengirimkan SMS melalui *handphone* atau menggunakan perangkat lunak pada komputer yang telah terhubung dengan modem dan internet.

Pengujian berikutnya adalah mengukur waktu respon untuk mengetahui berapa lama waktu rata-rata yang dibutuhkan sistem untuk menampilkan posisi kendaraan di dalam perangkat lunak. Pengujian yang terakhir adalah mengambil data-data koordinat dengan cara menempatkan sistem objek pada kendaraan bergerak.

Untuk menggunakan perangkat lunak, langkah-langkah yang harus dilakukan adalah sebagai berikut:

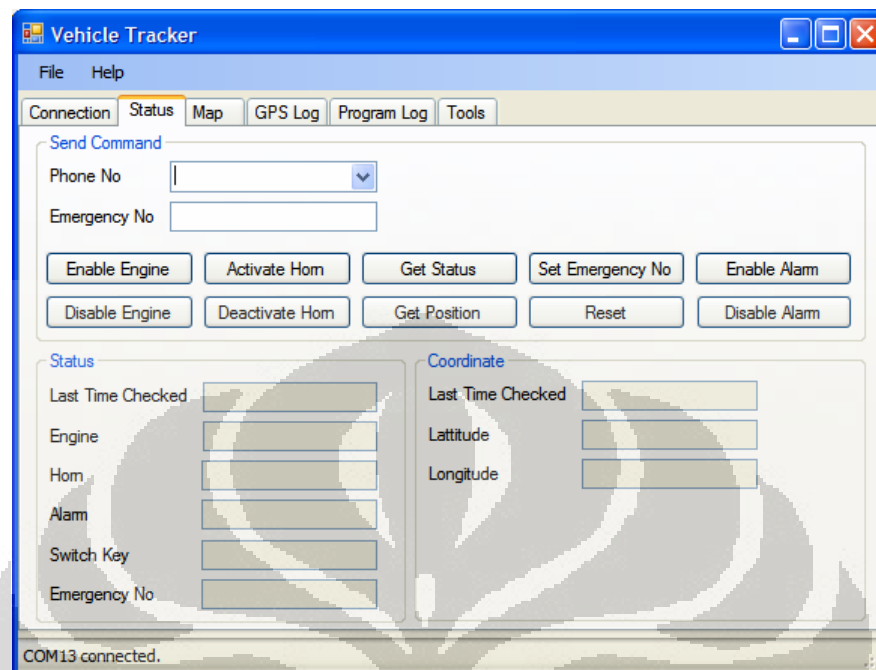
1. Hubungkan modem ke port serial pada komputer.
2. Jalankan aplikasi pemetaan (**VehicleTracker.exe**)
3. Pada tab **Connection** (Gambar 4.2), pilih port serial (COMx) dimana modem berada. Jika belum muncul, klik tombol **Refresh** untuk menampilkan port yang tersedia.
4. Untuk mengatur konfigurasi port serial dan proxy internet, klik tombol **More>>**, kemudian atur konfigurasi sesuai dengan keinginan.
5. Setelah port serial terpilih dan konfigurasi diatur, klik **Connect** untuk terhubung dengan modem.



Gambar 4.2 Tampilan halaman Connection pada perangkat lunak

Setelah perangkat lunak terhubung dengan modem, user dapat mengendalikan sistem objek. Untuk dapat mengendalikan sistem objek, user perlu

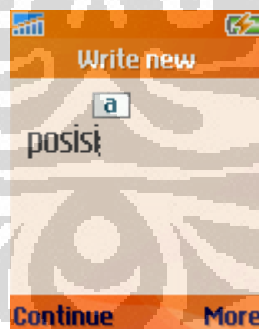
mengisi nomor tujuan dan menekan tombol-tombol perintah yang tersedia pada tab **Status**, seperti pada Gambar 4.3.



Gambar 4.3 Tampilan halaman Status pada perangkat lunak

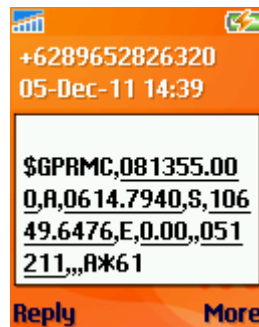
4.2.1 Pengujian Menampilkan Posisi

Permintaan posisi dilakukan dengan cara mengirimkan SMS ke sub sistem A dengan format **posisi**, seperti pada Gambar 4.4.



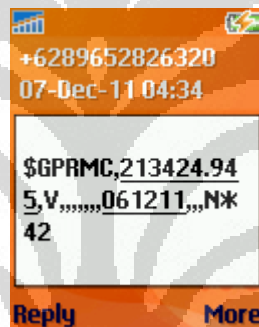
Gambar 4.4 Format SMS untuk meminta posisi

Setelah SMS terkirim, *user* hanya perlu menunggu beberapa detik hingga mendapat SMS balasan. SMS balasan yang didapat adalah data mentah NMEA 0183 tipe GPRMC. Hasil pengujian permintaan posisi ditunjukkan pada Gambar 4.5.



Gambar 4.5 Hasil dari permintaan data posisi di handphone

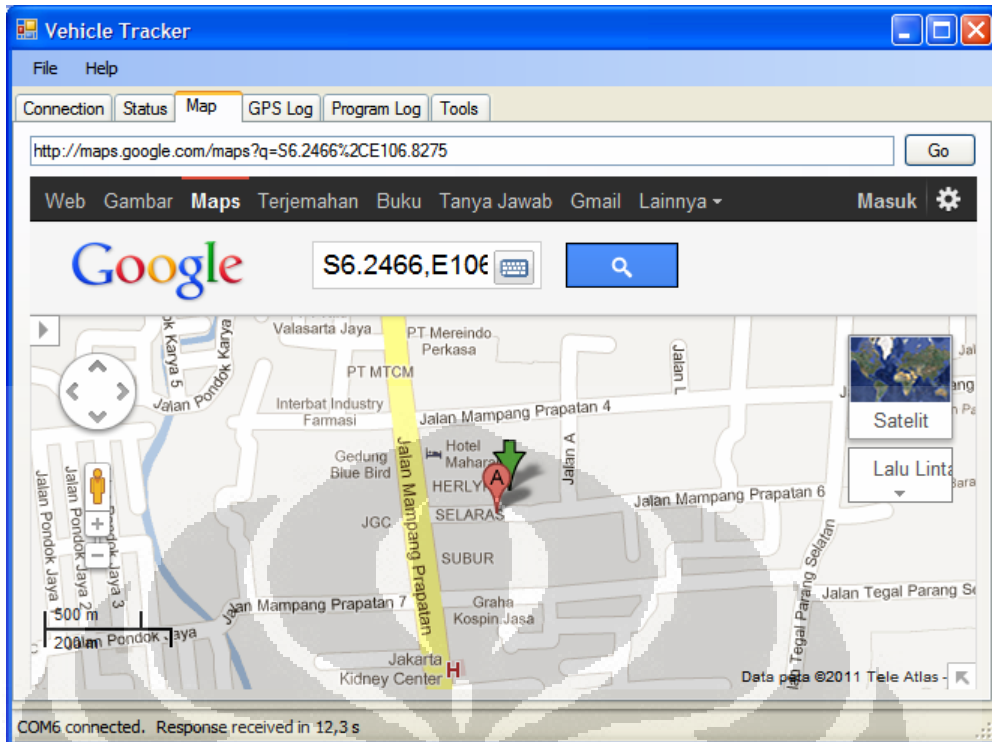
Gambar 4.5 menunjukkan SMS balasan dari sistem objek ketika GPS *receiver* sudah mendapatkan posisi yang valid. Namun, jika GPS receiver tidak mendapatkan posisi yang valid, maka akan mendapatkan SMS balasan seperti pada Gambar 4.6.



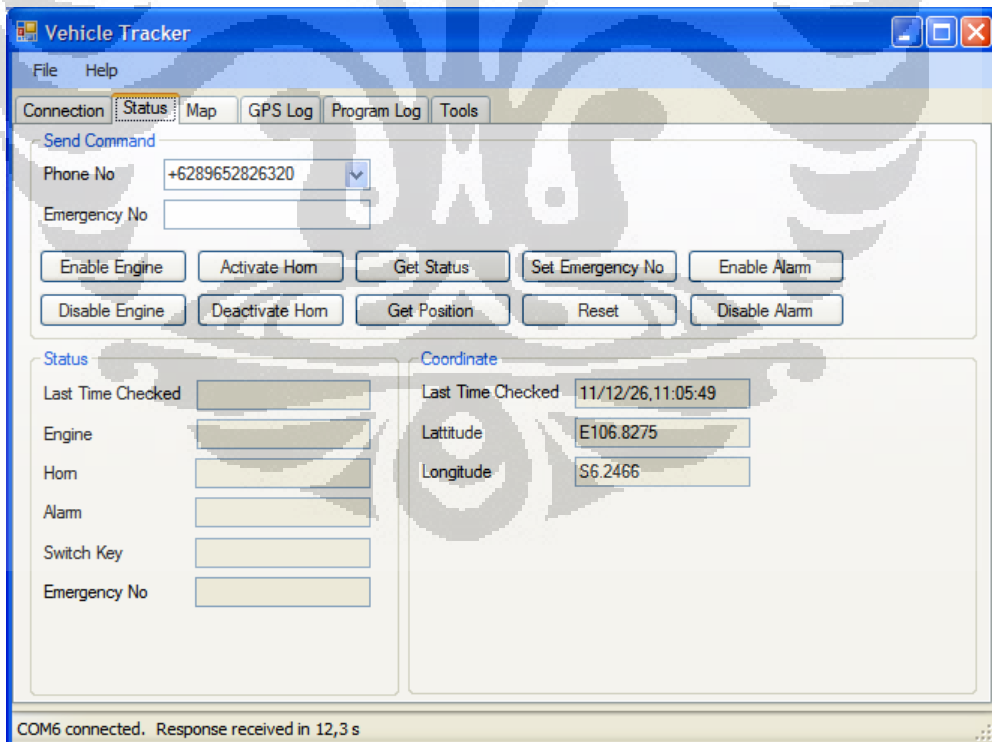
Gambar 4.6 SMS balasan ketika data GPS tidak valid

Untuk menampilkan posisi sistem objek di perangkat lunak, tombol yang harus ditekan di halaman **Status** adalah **Get Position**.

Pada pengujian ini, sistem objek diletakkan pada Jalan Mampang Prapatan 6. Hasil dari pengujian menampilkan posisi pada perangkat lunak ini ditunjukkan pada Gambar 4.7. Sedangkan data koordinat sistem objek ditampilkan pada halaman **Status** pada bagian bagian **Coordinate** seperti yang ditunjukkan pada Gambar 4.8.



Gambar 4.7 Tampilan posisi objek pada perangkat lunak



Gambar 4.8 Data koordinat ditampilkan di halaman Status

Dari Gambar 4.5 dan 4.6, setelah pengguna mengirimkan SMS untuk meminta posisi, *user* mendapat balasan berupa data keluaran GPS, baik valid

maupun tidak. Pada Gambar 4.7 dan 4.8, setelah user menekan tombol Get Position, posisi dan koordinat sistem objek dapat ditampilkan di perangkat lunak. Dengan demikian, fungsi permintaan data posisi dan menampilkan posisi di perangkat lunak telah berjalan dengan baik.

4.2.2 Mengendalikan Mesin dan Klakson

4.2.2.1 Mengendalikan Mesin

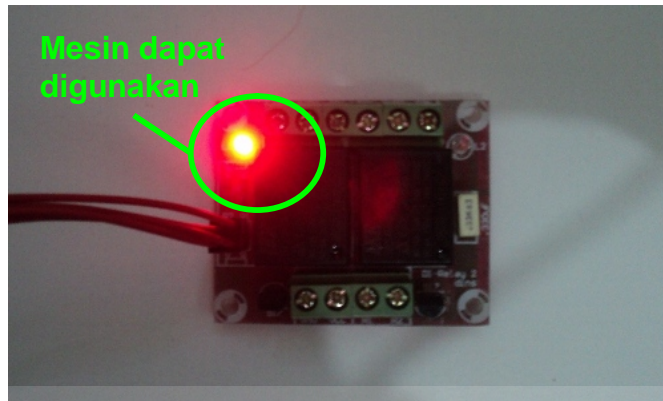
Untuk mengendalikan mesin, format SMS yang harus dikirimkan adalah **mesin on** atau **mesin off**. Untuk mengatur relay agar mesin dapat digunakan, maka SMS yang dikirimkan adalah **mesin on**, seperti pada Gambar 4.9a, sedangkan untuk mengatur relay agar mesin tidak dapat digunakan maka SMS yang dikirimkan adalah **mesin off**, seperti pada Gambar 4.9b.



Gambar 4.9 Format SMS untuk mengendalikan mesin. (a) mesin dapat digunakan dan (b) mesin tidak dapat digunakan

Pada perangkat lunak, untuk mengendalikan relay agar mesin dapat digunakan adalah dengan menekan tombol **Enable Engine**, sedangkan untuk mengendalikan relay agar mesin tidak dapat digunakan adalah dengan menekan tombol **Disable Engine**.

Hasil percobaan pengendalian mesin ditunjukkan pada Gambar 4.10 dan Gambar 4.11.



Gambar 4.10 Kondisi relay ketika mendapatkan SMS yang berisi “mesin on”



Gambar 4.11 Kondisi relay ketika mendapatkan SMS yang berisi “mesin off”

4.2.2.2 Mengendalikan Klakson

Untuk mengendalikan klakson, format SMS yang harus dikirimkan adalah **klakson on** atau **klakson off**. Untuk membunyikan klakson, SMS yang dikirimkan adalah **klakson on** seperti pada Gambar 4.12a, sedangkan untuk mematikan klakson, SMS yang dikirimkan adalah **klakson off** seperti pada Gambar 4.12b.



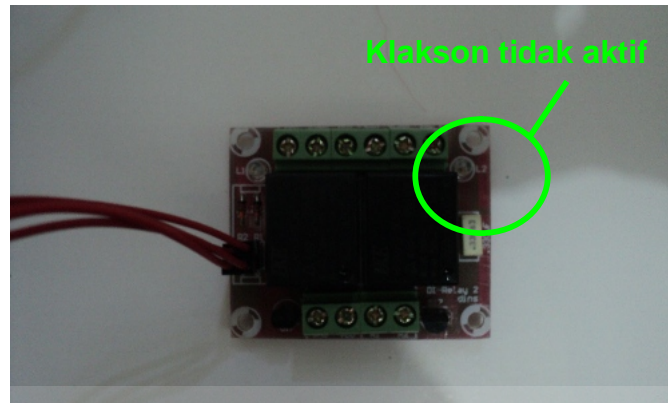
Gambar 4.12 Format SMS untuk mengendalikan klakson. (a) untuk membunyikan klakson dan (b) untuk mematikan klakson

Pada perangkat lunak, untuk mengendalikan relay agar klakson berbunyi adalah dengan menekan tombol **Activate Horn**, sedangkan untuk mengendalikan relay agar klakson tidak berbunyi adalah dengan menekan tombol **Deactivate Horn**.

Hasil pengujian pengendalian ditunjukkan pada Gambar 4.13 dan Gambar 4.14.



Gambar 4.13 Kondisi relay ketika mendapatkan SMS yang berisi “klakson on”



Gambar 4.14 Kondisi relay ketika mendapatkan SMS yang berisi “klakson off”

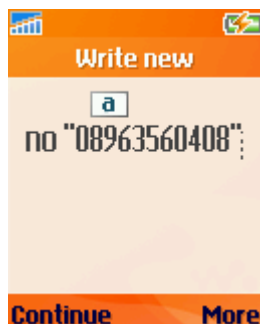
4.2.3 Pengujian Sistem Alarm

Pengujian ini bertujuan untuk mengetahui apakah fungsi alarm dapat bekerja ketika pencuri memutar *switch* kendaraan ke posisi *on*. Untuk menguji fungsi ini, hal yang harus dilakukan adalah mengaktifkan alarm dan menetapkan nomor darurat. Untuk mengaktifkan alarm, format SMS yang dikirimkan adalah **alarm on** seperti pada Gambar 4.15.



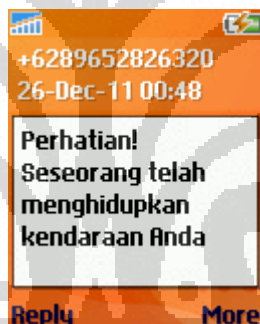
Gambar 4.15 Format SMS untuk mengaktifkan alarm

Selanjutnya, user perlu menetapkan nomor darurat dengan cara mengirimkan SMS dengan format no “xxxxxxx” seperti pada Gambar 4.16. Nomor darurat inilah yang akan dikirim SMS oleh sistem objek ketika *switch* kendaraan diputar ke posisi *on* oleh seseorang.

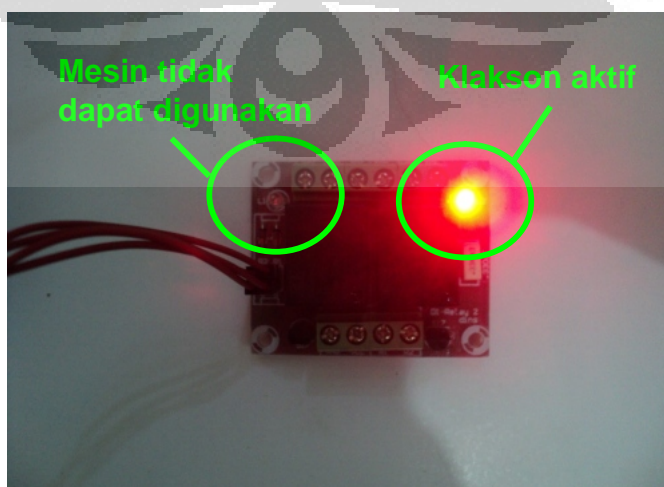


Gambar 4.16 Format SMS untuk menetapkan nomor darurat

Pada pengujian ini, sistem objek tidak dihubungkan dengan *switch* kendaraan yang sesungguhnya, tetapi disimulasikan dengan cara menekan *push button*. Ketika *push button* ditekan, terjadi perubahan logika dari 0 ke 5 V (*rising edge*), dimana dalam kondisi riilnya, mikrokontroler akan mendeteksi adanya perubahan tegangan dari 0 ke 12 V pada *switch* melalui *regulator*. Hasil pengujian ini ditunjukkan pada Gambar 4.17 dan 4.18.



Gambar 4.17 SMS yang diterima yang didapat ketika kendaraan dihidupkan dan alarm dalam keadaan aktif

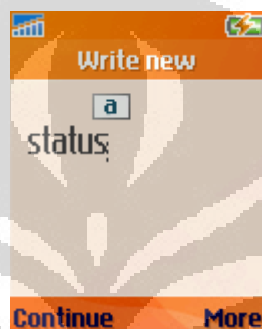


Gambar 4.18 Mesin tidak dapat digunakan dan klakson berbunyi

Ketika *switch* diputar ke posisi *on* (yang disimulasikan dengan menekan *push button*), pemilik nomor darurat dikirim SMS oleh sistem objek seperti pada Gambar 4.17, mesin tidak dapat digunakan dan klakson berbunyi seperti pada Gambar 4.18. Dengan demikian fungsi alarm telah berjalan dengan baik.

4.2.4 Cek Status

Untuk mengecek status dari sistem objek dilakukan dengan cara mengirimkan pesan SMS dengan format **status** seperti pada Gambar 4.19 atau dengan menekan tombol **Get Status** pada perangkat lunak.



Gambar 4.19 Format SMS untuk meminta status dari sistem objek

Setelah beberapa detik maka akan ada SMS balasan dari sistem objek. Hasil pengujian ini ditunjukkan pada Gambar 4.20.

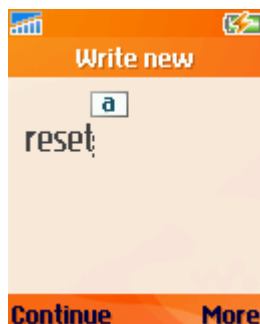


Gambar 4.20 SMS yang diterima setelah mengirimkan perintah 'status'

Ketika user mengirimkan pesan yang berisi "status", *user* mendapat SMS balasan seperti Gambar 4.20. Dengan demikian, percobaan permintaan status telah berjalan dengan baik.

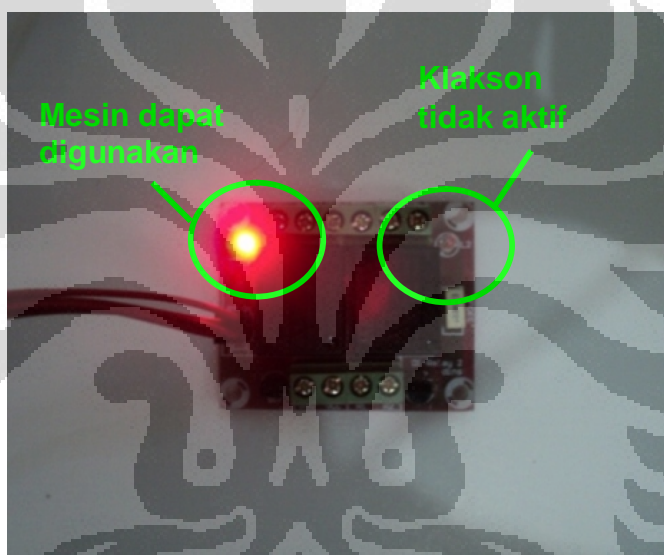
4.2.5 Reset Sistem

Untuk mereset mikrokontroler ke keadaan normal, maka pesan SMS yang harus dikirimkan adalah **reset** seperti pada Gambar 4.21.



Gambar 4.21 Format SMS untuk mereset sistem objek

Keadaan normal adalah keadaan dimana mesin dapat digunakan (relay mesin on), klakson mati (relay klakson off), dan status kecurian adalah false (*flag* kecurian bernilai 0). Hasil percobaan reset ini diperlihatkan pada Gambar 4.22.



Gambar 4.22 Mesin bisa digunakan dan klakson tidak berbunyi setelah mendapatkan SMS yang berisi “reset”

Saat user mengirimkan pesan “reset”, sistem objek langsung berada pada kondisi normal. Dengan demikian fungsi reset sistem telah berjalan dengan baik.

4.3 Pengujian Waktu Respon

Percobaan ini bertujuan untuk mengukur seberapa cepat pengamat mendapatkan SMS balasan dari sistem objek dan untuk menguji kehandalan jaringan dari berbagai macam operator seluler yang berbeda-beda. Pada percobaan ini, kedua sistem (objek dan pengamat) dimasukkan SIM *card* dari berbagai operator secara bergantian hingga membentuk kombinasi, selanjutnya pengamat

meminta posisi kepada sistem objek dan dihitung lamanya waktu hingga mendapatkan SMS balasan.

Lama waktu dihitung sejak menekan tombol “kirim” atau “send” pada *handphone* hingga mendapatkan notifikasi SMS masuk. Pada perangkat lunak, waktu respon dihitung sejak menekan tombol **Get Position** hingga mendapatkan SMS masuk yang berupa data posisi.

Percobaan ini dilakukan di perangkat lunak. Delay dihitung dengan menggunakan *built-in timer* yang terintegrasi pada perangkat lunak sehingga hasilnya lebih akurat dibandingkan dengan cara manual menggunakan stop watch, karena refleks manusia tidaklah secepat komputer. Percobaan ini dilakukan dengan menempatkan objek dan pengamat di Jalan Mampang Prapatan 6, Jakarta Selatan. Oleh karena itu, hasil percobaan ini belum tentu sama hasilnya jika dicoba di tempat lain.

Operator yang digunakan dalam percobaan ini hanya lima, yaitu Three, XL, IM3, As dan Axis. Dalam percobaan ini, jaringan operator dianggap tidak bermasalah (tidak dalam kondisi *maintenance* atau perbaikan jaringan) karena pengiriman SMS langsung sampai ke tujuan. SMS dianggap tidak sampai jika delaynya melebihi 60 detik atau jika SMS tidak diterima sama sekali.

Hasil dari percobaan delay dari berbagai macam operator pada kedua sub sistem ditunjukkan pada Tabel 4.1 – 4.5.

Tabel 4.1 Waktu respon pada sistem objek yang menggunakan Three dan pengamat yang menggunakan berbagai macam operator (dalam detik)

Objek	Pengamat					
	No	Three	XL	IM3	As	Axis
Three	1	13	12,6	12,8	13	15,9
	2	12,7	12,5	13	12,6	14,4
	3	12,8	12,9	11,8	12,5	13,3
	4	13,3	12,5	12,5	11,6	14,2
	5	12,3	12,4	12,3	12,4	14
	6	13,6	12,2	12,6	11,9	14,1
	7	13	13,2	13,5	12,9	17,2
	8	13,3	11,8	12,8	12	-
	9	12,8	12,9	12,9	12,5	15,3
	10	13,2	12,3	11,7	12,3	12
Rata-rata		13,00	12,53	12,59	12,37	14,49

Tabel 4.2 Waktu respon pada sistem objek yang menggunakan XL dan pengamat yang menggunakan berbagai macam operator (dalam detik)

Objek	Pengamat					
	No	Three	XL	IM3	As	Axis
XL	1	15,1	14,7	19,7	13	15,1
	2	13,4	21	13,2	15,2	13,5
	3	14,6	16,5	13,5	-	13,9
	4	12,2	15	14,7	12,5	13,7
	5	14,8	14,7	-	15,7	13,8
	6	14,5	14,4	13,3	13,6	14,9
	7	15,3	15,6	13,2	13,8	14,3
	8	13,7	14,1	16,1	13,1	14,6
	9	15,2	14,5	11,8	13,5	13,7
	10	15,1	20,2	12,8	13,4	-
Rata-rata		14,39	16,07	14,26	13,76	14,17

Tabel 4.3 Waktu respon pada sistem objek yang menggunakan IM3 dan pengamat yang menggunakan berbagai macam operator (dalam detik)

Objek	Pengamat					
	No	Three	XL	IM3	As	Axis
IM3	1	12	11,9	11,2	12,1	13,9
	2	13,5	12,4	11,3	11,8	13,6
	3	12,7	12,3	11,8	11,1	13,4
	4	12	11,5	11,3	11,3	13
	5	12	12,6	11,5	11,1	14,3
	6	12,4	11,8	11,6	10,9	-
	7	12,7	12,7	12,4	11,4	12,5
	8	12	12,4	11,7	11	13,1
	9	12,3	11,9	11,5	11	12,7
	10	11,9	11,6	11,6	11,3	12,9
Rata-rata		12,35	12,11	11,59	11,30	13,27

Tabel 4.4 Waktu respon pada sistem objek yang menggunakan As dan pengamat yang menggunakan berbagai macam operator (dalam detik)

Objek	Pengamat					
	No	Three	XL	IM3	As	Axis
As	1	13,2	12,7	12,9	10,9	15,2
	2	-	12,6	12,6	11,2	14,1
	3	14,1	12,7	13	11	13
	4	16	13,3	12,6	10,9	16,6
	5	13	16,5	10,9	11,2	14,7
	6	12,1	12	11,1	11,2	12,9
	7	-	14,7	13	10,8	12,7
	8	12,7	12,2	13,4	11	13,4
	9	12,8	12,3	13,1	11,1	14,3
	10	18	-	12,4	11	14,7
Rata-rata		13,99	13,22	12,50	11,03	14,16

Tabel 4.5 Waktu respon pada sistem objek yang menggunakan Axis dan pengamat yang menggunakan berbagai macam operator (dalam detik)

Objek	Pengamat					
	No	Three	XL	IM3	As	Axis
Axis	1	16,1	13,6	13,6	13,6	14,6
	2	14,1	12,5	12	12,2	13,3
	3	13,8	13,3	12	12,9	14,1
	4	14,4	13,4	12,3	12,9	13,9
	5	13,9	13,6	12,3	13	14,1
	6	14,5	14	12,5	12	14,5
	7	14,1	13,6	12,7	12,4	13,3
	8	13,5	13,9	13,4	-	13,4
	9	14,2	14	12,3	12,1	14,5
	10	14,2	13,4	13,3	13,8	14,6
Rata-rata		14,28	13,53	12,64	12,77	14,03

Delay minimum dari kedua sub sistem dari berbagai macam operator dirangkum dalam Tabel 4.6.

Tabel 4.6 Waktu respon minimum dari berbagai macam operator (dalam detik)

Objek	Pengamat				
	Three	XL	IM3	As	Axis
Three	12,3	11,8	11,7	11,6	12
XL	12,2	14,1	11,8	13	13,7
IM3	11,9	11,5	11,2	10,9	12,5
As	12,1	12	10,9	10,8	12,7
Axis	13,5	12,5	12	12	13,3

Delay rata-rata dari kedua sub sistem dari berbagai macam operator dirangkum pada Tabel 4.7.

Tabel 4.7 Waktu respon rata-rata dari berbagai macam operator (dalam detik)

Objek	Pengamat				
	Three	XL	IM3	As	Axis
Three	13,00	12,53	12,59	12,37	14,49
XL	14,39	16,07	14,26	13,76	14,17
IM3	12,35	12,11	11,59	11,30	13,27
As	13,99	13,22	12,50	11,03	14,16
Axis	14,28	13,53	12,64	12,77	14,03
Rata-rata	13,22				

Persentase keberhasilan pengambilan data posisi dari berbagai macam operator dirangkum pada Tabel 4.8

Tabel 4.8 Persentase keberhasilan dari berbagai macam operator (dalam persen)

Objek	Pengamat				
	Three	XL	IM3	As	Axis
Three	100	100	100	100	90
XL	100	100	90	90	90
IM3	100	100	100	100	90
As	80	90	100	100	100
Axis	100	100	100	90	100

4.4 Pengujian Objek Dalam Keadaan Bergerak

Pengujian kali ini, objek ditempatkan pada kendaraan yang bergerak untuk mengetahui kinerja sistem dalam kondisi bergerak. Pada pengujian ini, kendaraan bergerak di sepanjang jalan Bogor Raya. Hasil pengujian ditunjukkan pada tabel 4.9.

Tabel 4.9 Hasil percobaan ketika sistem objek dalam keadaan bergerak

No.	Lintang	Bujur	Kecepatan (km/jam)	Arah (°)	Waktu respon (detik)
1	S6.3181	E106.8645	37,50	344.73	13,7
2	S6.3127	E106.8626	20,15	350.44	13,7
3	S6.3090	E106.864	43,19	32.79	13,4
4	S6.3016	E106.8682	33,60	46.07	14,2
5	S6.2933	E106.8721	20,87	359.00	12,3
6	S6.2899	E106.8712	36,35	340.83	12,9
7	S6.2876	E106.8708	0	-	13,7
8	S6.2831	E106.8717	34,09	17.31	12,9
9	S6.2760	E106.8681	22,67	328.10	15,3
10	S6.2740	E106.8672	10,46	2.65	13,1
11	S6.277	E106.8692	0	-	13,5
12	S6.2793	E106.8704	36,19	145.77	13
13	S6.2845	E106.8713	24,67	191.96	22,8
14	S6.2869	E106.8709	15,22	172.68	11,9
15	S6.2902	E106.8714	28,22	160.72	14,3
16	S6.2962	E106.8717	9,37	192.96	13,3
17	S6.3007	E106.8691	36,46	218.45	17,4
18	S6.3065	E106.8650	55,93	208.86	13,1
19	S6.3105	E106.8627	13,28	192.75	12,6
20	S6.3159	E106.8637	37,60	157.71	13,1
21	S6.3185	E106.8647	8,42	163.87	12,9
22	S6.3221	E106.8650	39,76	194.76	12,9
23	S6.3257	E106.8616	18,69	285.74	13,4
24	S6.3267	E106.8593	17,63	187.93	14,1
25	S6.3286	E106.8583	11,98	289.24	12,7
Waktu respon rata-rata					13,85

4.5 Analisis

Pada percobaan permintaan posisi, terkadang pengamat bisa mendapatkan data GPS yang tidak valid seperti pada Gambar 4.5. Data GPS yang tidak valid tersebut disebabkan oleh ketiadaan sinyal GPS yang diterima oleh GPS *receiver*. Ketiadaan sinyal GPS tersebut bisa disebabkan oleh terhalang atap rumah atau gedung. Namun pada beberapa percobaan dengan menempatkan objek di dalam rumah, objek masih bisa memberikan data posisi yang valid meskipun sinyal GPS terhalang oleh atap rumah. Oleh karena itu antenna yang bagus sangat diperlukan dalam sistem ini.

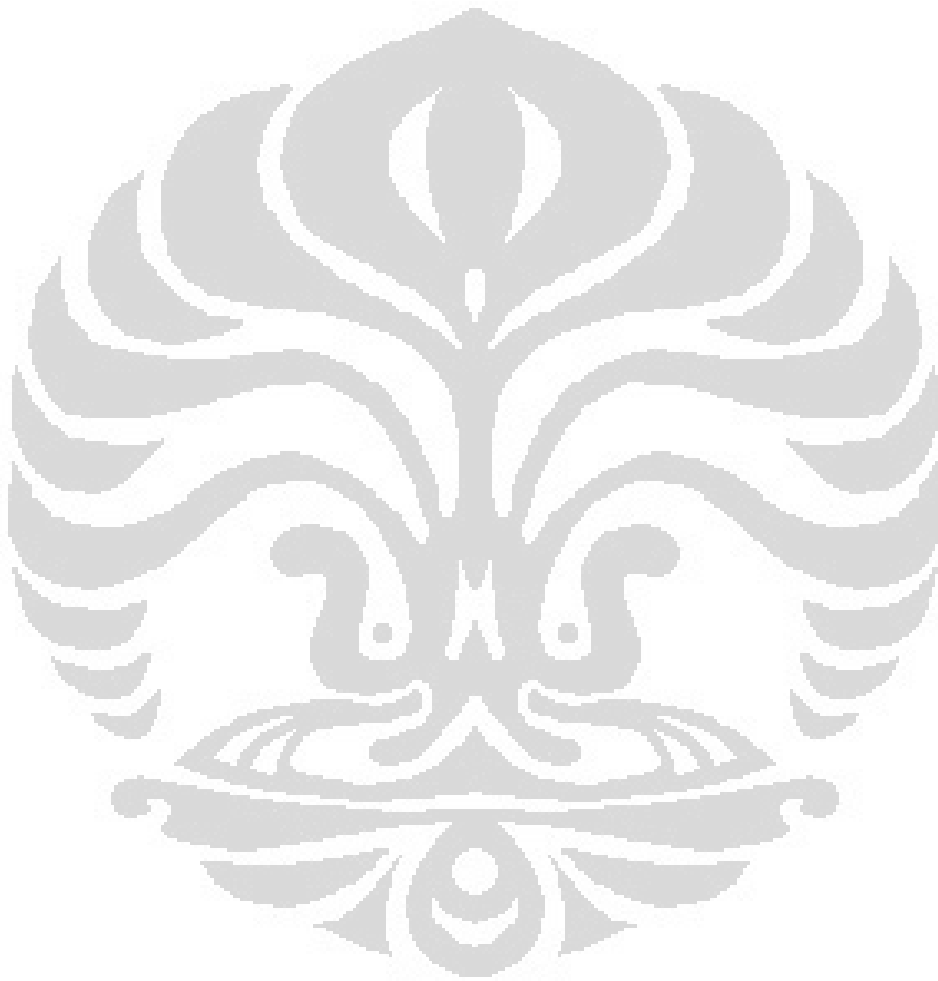
Pada percobaan waktu respon untuk pengambilan data posisi, didapatkan waktu respon yang bervariasi dari berbagai macam operator. Dari tabel 4.7, penggunaan operator As – As tercatat mempunyai waktu respon rata-rata paling kecil yaitu sebesar 11,03 detik sedangkan penggunaan operator XL – XL mempunyai waktu respon rata-rata paling besar yaitu sebesar 16,07 detik. Selisih perbedaan waktu rata-ratanya adalah sebesar 5,04 detik. Selisih waktu respon dari berbagai operator tidak begitu besar dan masih tergolong wajar.

Dari Tabel. 4.6, penggunaan operator As – As tercatat mempunyai waktu respon minimum 10,8 detik, kemudian disusul oleh operator IM3 – As mempunyai waktu respon minimum 10,9 detik dan IM3 – IM3 dengan waktu respon minimum 11,2 detik.

Dari Tabel 4.8, penggunaan operator yang sama mempunyai tingkat keberhasilan pengiriman SMS sebesar 100%. Namun, dengan menggunakan operator yang berbeda, tidak semua SMS dapat dikirimkan dengan berhasil atau SMS diterima melebihi 60 detik. Tercatat, penggunaan operator XL – IM3, XL – As, XL – Axis, As – XL, Axis – As mempunyai tingkat keberhasilan 90%. Penggunaan operator As – Three tercatat mempunyai tingkat keberhasilan 80%. Penyebabnya adalah karena alur pengiriman SMS operator yang berbeda lebih panjang dan jaringannya mengalami kesibukan.

Pada tabel 4.9, waktu respon rata-rata ketika objek bergerak adalah sebesar 13,85. Waktu tersebut tidaklah berbeda jauh dengan ketika objek ketika diam, yaitu sebesar 13,22 detik. Ini menunjukkan bahwa objek yang bergerak tidak terlalu mempengaruhi waktu respon ketika dimintai data posisi. Dari tabel,

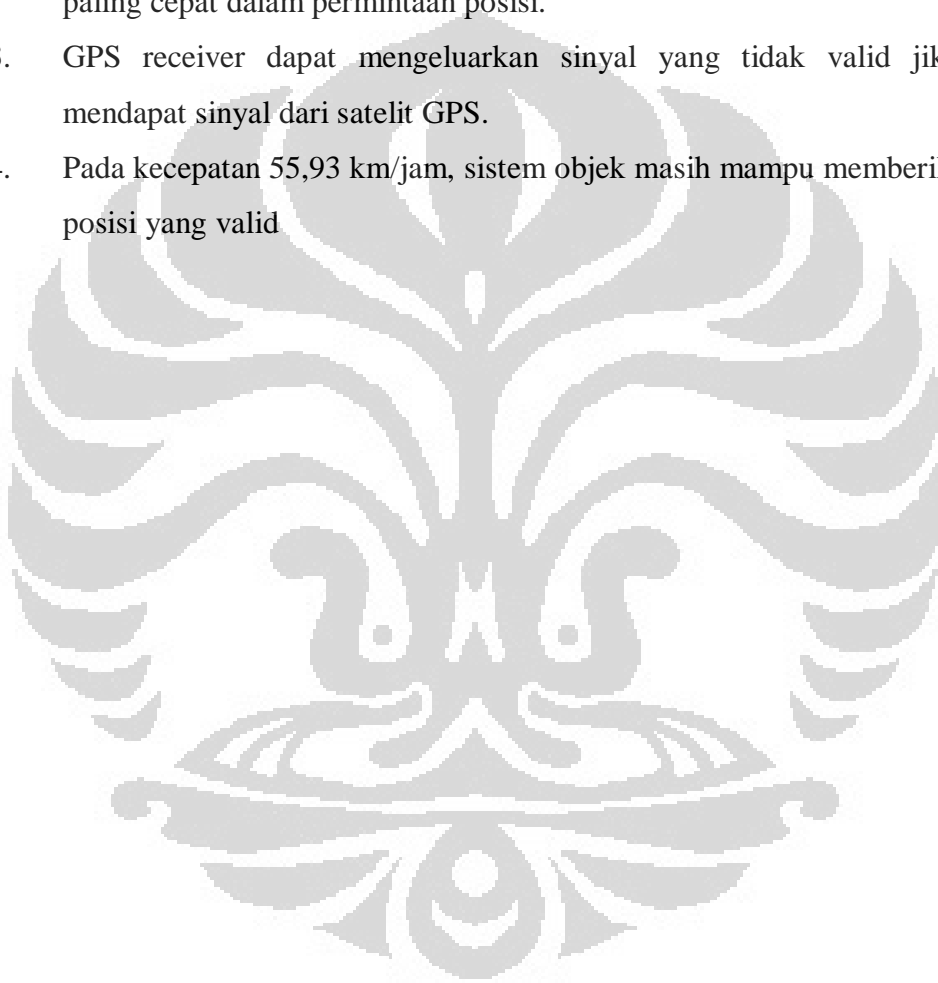
kecepatan maksimum yang terekam adalah 55,93 km/jam. Ini menunjukkan bahwa pada kecepatan tersebut sistem objek masih bisa memberikan data posisi yang valid.



BAB 5 KESIMPULAN

Berdasarkan hasil pengujian dan analisis dapat disimpulkan:

1. Sistem ini dapat diterapkan sebagai alat penjejak dan keamanan pada kendaraan bermotor.
2. Penggunaan operator As – As memberikan waktu respon rata-rata yang paling cepat dalam permintaan posisi.
3. GPS receiver dapat mengeluarkan sinyal yang tidak valid jika tidak mendapat sinyal dari satelit GPS.
4. Pada kecepatan 55,93 km/jam, sistem objek masih mampu memberikan data posisi yang valid



DAFTAR REFERENSI

- [1] El-Rabbany, Ahmad. (2002). *Introduction to GPS: The Global Positioning System*. Norwood: Artech House.
- [2] Taryudi. (2009). *Implementasi dan Uji Kerja Sistem Pemantau Posisi dan Tingkat Pencemaran Udara Bergerak*, Tesis. Depok: Universitas Indonesia
- [3] Putra, Agfianto Eko, Nugraha, Dhani. (2010). *Tutorial Pemrograman Mikrokontroler AVR dengan WinAVR GCC (ATMega16/32/8535)*. [http://www.4shared.com/document/RrDea-VA/Tutorial Pemrograman Mikrokont.html](http://www.4shared.com/document/RrDea-VA/Tutorial%20Pemrograman%20Mikrokont.html).
- [4] *8-bit AVR Microcontroller with 16K Bytes In-System Programmable Flash: Atmega162 ATMega162V*, datasheet. www.atmel.com/atmel/acrobat/doc2513.pdf. ATMEL.
- [5] *AT Commands Interface Guide for 6.57 Release*, datasheet. (2006). Wavecom
- [6] *Teori Dasar SMS*. <http://www.mikron123.com/index.php/Aplikasi-SMS/Teori-Dasar-SMS.html>. 30 Maret 2011
- [7] *RXM-SG GPS Module w/Ext Antenna (#28505)*, datasheet. Parallax Inc.
- [8] *Fastrack Modem M1306B User Guide*, datasheet. (2004). Wavecom
- [9] Bao, James, Tsui, Yen. *Fundamentals of Global Positioning System: A Software Approach*. (2000). John Wiley and Sons

LAMPIRAN A KODE PROGRAM MIKROKONTROLER

```
$regfile = "M162def.dat"
$crystal = 11059200
$baud = 9600      'Modem
$baud1 = 9600     'GPS

***** inisialisasi uart2 *****
Open "COM2:" For Binary As #1

***** inisialisasi port I/O *****
Config Porta = Output

***** inisialisasi interrupt INTO *****
Config Pind.2 = Input
Config Int0 = Rising
Config Debounce = 100

***** variabel SMS *****
Dim No_hp As String * 17
Dim Data_masuk As String * 100
Dim Panjang_data As Byte
Dim Index_pos As Byte
Dim Panjang_sms_index As Byte
Dim Sms_index As String * 3
Dim Kunci As Bit
Dim Alarm As Bit
Dim Baca_sms As Bit
Dim Panjang_no_hp As Byte
Dim Str_array(4) As String * 24
Dim Bcount As Byte
Dim Str_cmgr As String * 64
Dim Stat_mesin As String * 3
Dim Stat_klakson As String * 3
Dim Stat_kunci As String * 3
Dim Stat_alarm As String * 3
Dim No_darurat As String * 17
Dim Str_tmp As String * 8
Dim Panjang_no_darurat As Byte
Dim Data_respon As String * 6
Dim Respon_ok_error As Bit

***** variabel data GPS *****
Dim Data_gps As String * 80

***** deklarasi subrutin *****
```

```

Declare Sub Reset_status
Declare Sub Ekstrak_no_hp
Declare Sub Tunggu_respon

***** enable interrupt *****
Enable Interrupts
Enable Int0

***** jika terjadi interrupt lompat ke label kirim_peringatan *****
On Int0 Kirim_peringatan

***** inialisasi program *****
Porta = &B00000010
Waitms 1000
Alarm = 1
Call Reset_status
Baca_sms = 0
No_hp = Chr(34) + "+6289635604086" + Chr(34)
No_darurat = No_hp
Print "ATE0"

***** program utama *****
Do
  Input Data_masuk
  Panjang_data = Len(data_masuk)

  'periksa apa ada kata CMTI?
  If Instr(data_masuk , "CMTI") <> 0 Then
    Index_pos = Instr(data_masuk , ",")
    Incr Index_pos
    Panjang_sms_index = Panjang_data - Index_pos
    Incr Panjang_sms_index
    Sms_index = Mid(data_masuk , Index_pos , Panjang_sms_index)
    Print "AT+CMGR=" ; Sms_index

  'periksa apa ada kata CMGR?
  ElseIf Instr(data_masuk , "+CMGR") <> 0 Then
    Str_cmgr = Data_masuk
    Baca_sms = 1

  'apakah sedang membaca SMS?
  ElseIf Baca_sms = 1 Then
    Data_masuk = Lcase(data_masuk)           'ubah ke huruf kecil semua

  'apakah perintah untuk mengendalikan relay mesin agar bisa
  'mendapat arus?
  If Instr(data_masuk , "mesin on") <> 0 Then
    Porta.1 = 1

```


'apakah perintah untuk mengendalikan relay mesin agar tidak
'bisa mendapat arus listrik?

```
Elseif Instr(data_masuk , "mesin off") <> 0 Then
  Porta.1 = 0
```

'apakah perintah untuk menghidupkan klakson?

```
Elseif Instr(data_masuk , "klakson on") <> 0 Then
  Porta.2 = 1
```

'apakah perintah untuk mematikan klakson?

```
Elseif Instr(data_masuk , "klakson off") <> 0 Then
  Porta.2 = 0
```

'apakah perintah untuk meminta posisi?

```
Elseif Instr(data_masuk , "posisi") <> 0 Then
```

```
  'filter data GPRMC
```

```
  Do
```

```
    Input #1 , Data_gps
```

```
  Loop Until Instr(data_gps , "GPRMC") <> 0
```

```
  Ekstrak_no_hp
```

```
  'kirim data posisi dengan SMS
```

```
  Print "AT+CMGS=" ; No_hp
```

```
  Print Data_gps ; Chr(26)
```

```
  Data_gps = ""
```

```
  Tunggu_respon
```

'apakah perintah untuk mereset?

```
Elseif Instr(data_masuk , "reset") <> 0 Then
  Call Reset_status
```

'apakah perintah untuk meminta status?

```
Elseif Instr(data_masuk , "status") <> 0 Then
```

```
  Ekstrak_no_hp
```

```
  If Porta.1 = 1 Then
```

```
    Stat_mesin = "ON"
```

```
  Else
```

```
    Stat_mesin = "OFF"
```

```
  End If
```

```
  If Porta.2 = 1 Then
```

```
    Stat_klakson = "ON"
```

```
  Else
```

```
    Stat_klakson = "OFF"
```

```
  End If
```

```
  If Kunci = 1 Then
```

```
    Stat_kunci = "ON"
```

```

Else
  Stat_kunci = "OFF"
End If
If Alarm = 1 Then
  Stat_alarm = "ON"
Else
  Stat_alarm = "OFF"
End If
Print "AT+CMGS=" ; No_hp
Print "Mesin=" ; Stat_mesin ; ", Klakson=" ; Stat_klakson ; _
", Alarm=" ; Stat_alarm ; ", Kunci=" ; Stat_kunci ; ", No darurat=" ; _
No_darurat ; Chr(26)
Tunggu_respon

'Apakah perintah untuk mengaktifkan alarm ketika kunci diputar ke
'posisi on?
Elseif Instr(data_masuk , "alarm on") <> 0 Then
  Alarm = 1

'Apakah perintah untuk menonaktifkan alarm etika kunci diputar ke posisi
off?
Elseif Instr(data_masuk , "alarm off") <> 0 Then
  Alarm = 0

'perintah set no hp untuk alarm
Elseif Instr(data_masuk , "no") <> 0 Then
  Panjang_no_darurat = Panjang_data - 4
  No_darurat = Mid(data_masuk , 5 , Panjang_no_darurat)
End If

Waitms 1000
Print "AT+CMGD=" ; Sms_index           'hapus SMS
Baca_sms = 0                          'set flag baca SMS menjadi 0
End If
Loop

***** subrutin reset status & inialisasi program *****
Sub Reset_status
  Porta.1 = 1
  Porta.2 = 0
  Kunci = 0
  Data_gps = ""
End Sub

***** subrutin ekstrak no hp *****
Sub Ekstrak_no_hp()
  If Instr(str_cmgr , "CMGR") <> 0 Then
    Bcount = Split(str_cmgr , Str_array(1) , ",")

```

```

    No_hp = Str_array(2)
End If
End Sub

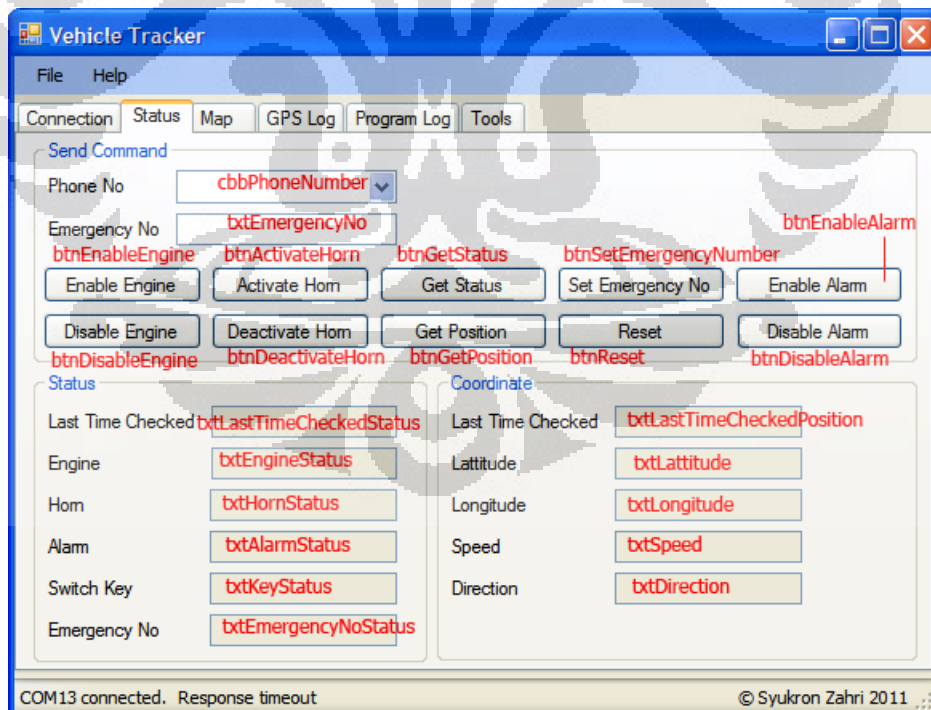
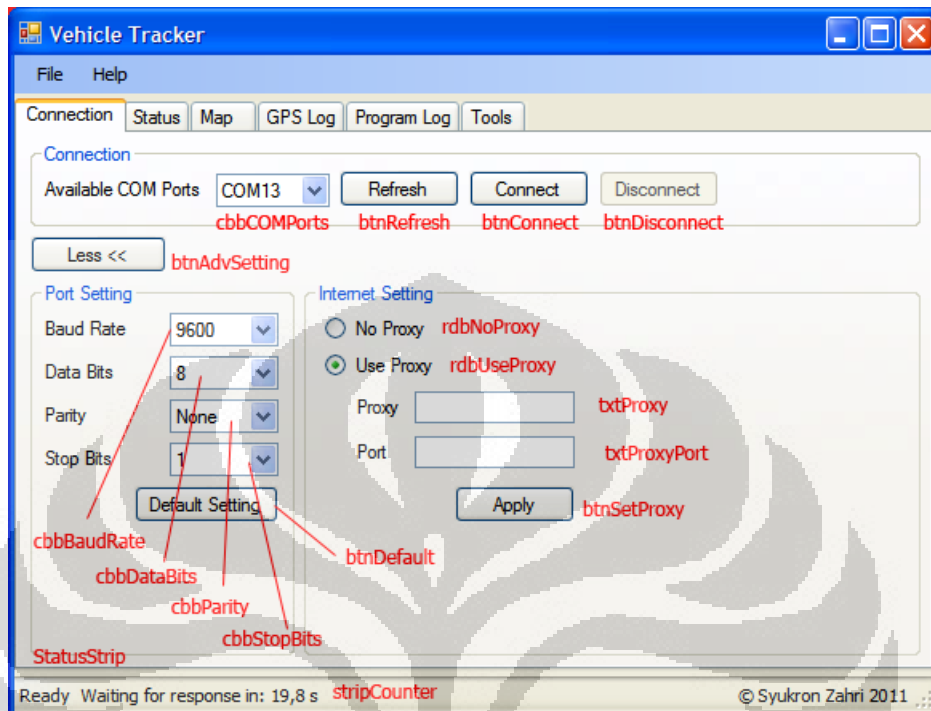
***** subrutin tunggu respon 'OK' atau 'ERROR' dari modem *****
Sub Tunggu_respon()
Do
    Input Data_respon
    If Instr(data_respon , "OK") <> 0 Then
        Respon_ok_error = 1
    ElseIf Instr(data_respon , "ERROR") <> 0 Then
        Respon_ok_error = 1
    End If
    Loop Until Respon_ok_error = 1
    Respon_ok_error = 0
    Data_respon = ""
End Sub

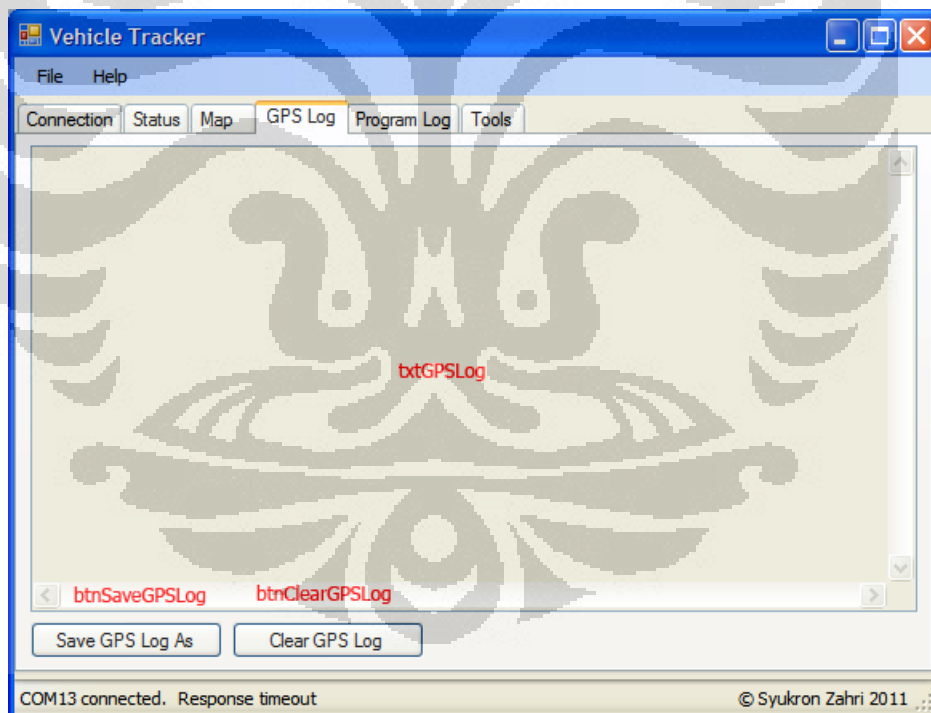
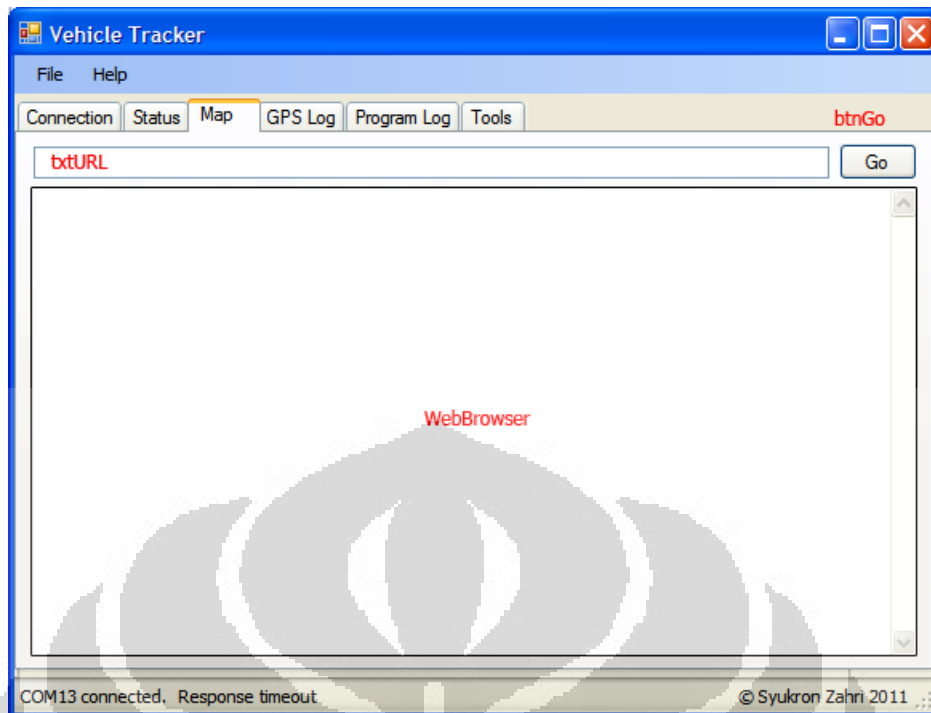
***** mengirim SMS peringatan ketika mendeteksi kunci kontak dihidupkan
seseorang *****
Kirim_peringatan:
If No_darurat = "" Then
    No_darurat = No_hp
End If
If Alarm = 1 Then
    Porta.1 = 0
    Print "AT+CMGS=" ; No_darurat
    Print "Perhatian! Seseorang telah menghidupkan kendaraan Anda" ; Chr(26)
    Tunggu_respon
    Waitms 1000
    Gifr = 64
    Kunci = 1
    Porta.2 = 1
End If
Return

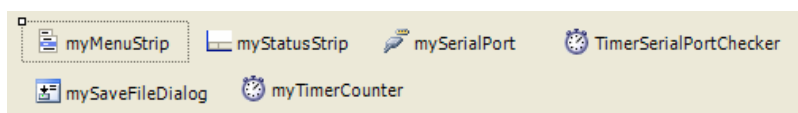
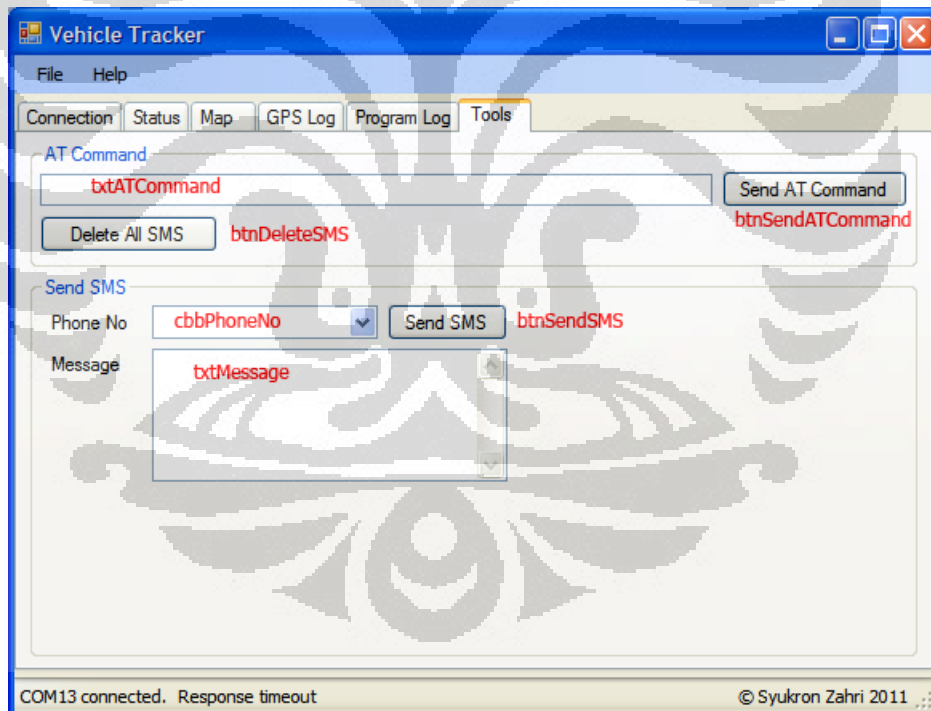
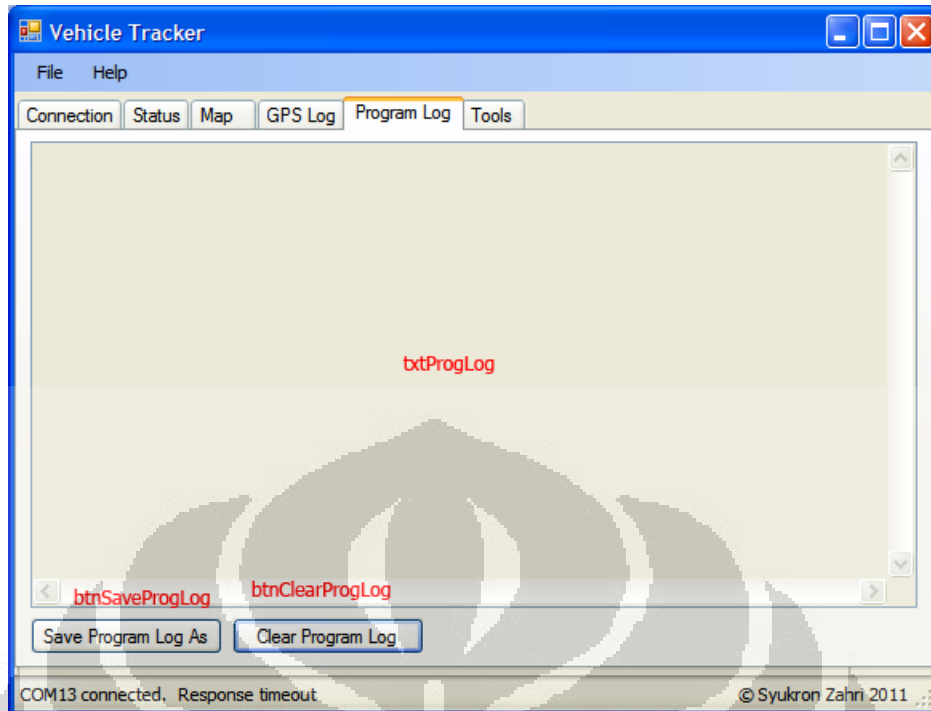
End

```

LAMPIRAN B KODE PROGRAM KOMPUTER







Imports System.Text
 Imports System.Drawing
 Imports System.Windows.Forms
 Imports System.Threading

Imports System.Runtime.InteropServices

Public Class frmMain

#Region "Using Proxy"

<Runtime.InteropServices.DllImport("wininet.dll", SetLastError:=True)> _

Private Shared Function InternetSetOption(ByVal hInternet As IntPtr, ByVal dwOption As Integer, ByVal lpBuffer As IntPtr, ByVal lpdwBufferLength As Integer) As Boolean

End Function

Public Structure Struct_INTERNET_PROXY_INFO

Public dwAccessType As Integer

Public proxy As IntPtr

Public proxyBypass As IntPtr

End Structure

Private Sub UseProxy(ByVal strProxy As String)

Const INTERNET_OPTION_PROXY As Integer = 38

Const INTERNET_OPEN_TYPE_PROXY As Integer = 3

Dim struct_IPI As Struct_INTERNET_PROXY_INFO

struct_IPI.dwAccessType = INTERNET_OPEN_TYPE_PROXY

struct_IPI.proxy = Marshal.StringToHGlobalAnsi(strProxy)

struct_IPI.proxyBypass = Marshal.StringToHGlobalAnsi("local")

Dim intptrStruct As IntPtr =

Marshal.AllocCoTaskMem(Marshal.SizeOf(struct_IPI))

Marshal.StructureToPtr(struct_IPI, intptrStruct, True)

Dim iReturn As Boolean = InternetSetOption(IntPtr.Zero, INTERNET_OPTION_PROXY, intptrStruct, System.Runtime.InteropServices.Marshal.SizeOf(struct_IPI))

End Sub

#End Region

Private ReadLineThread As Thread

Private MapCoordinateThread As Thread

Private Delegate Sub UpdateMainProgramDelegate()

Private Delegate Sub UpdateWebBrowser(ByVal [myURL] As String)

Private Delegate Sub UpdateFormControl(ByVal [Condition] As Boolean)

Private Delegate Sub UpdateStatus()

Private Delegate Sub UpdateCoordinate()

Private Delegate Sub UpdateGPSLog()

Dim SerialPortInputMessage As String

Dim SMSIndex As String

Dim Latitude As String

```

Dim Longitude As String
Dim Speed As String
Dim Direction As String
Dim StatEngine As Boolean
Dim StatHorn As Boolean
Dim StatAlarm As Boolean
Dim StatEmergencyNo As String
Dim StatKey As Boolean
Dim hasResponseOkError As Boolean = False
Dim TimeCounter As Single
Dim isSendingATCommand As Boolean = False
Dim StringCMGR As String
Dim StatLastTimeStatus As String
Dim StatLastTimeCoordinate As String
Dim WaitingTimeOut As Single

Private Function ConvertNMEALatToDecimal(ByVal LattDirection As String,
ByVal NMEALattitude As String) As String
    Dim ResultLat As String
    Try
        NMEALattitude = NMEALattitude.Replace(".", ",")
        Dim tempLat As Double = Convert.ToDouble(NMEALattitude)
        tempLat = tempLat / 100
        Dim tempArray() As String = tempLat.ToString.Split(",")
        ResultLat = LattDirection & tempArray(0).ToString & "." &
(Convert.ToDouble(tempArray(1)) / 60).ToString("#####")
    Catch ex As Exception
        ResultLat = String.Empty
    End Try
    Return ResultLat
End Function

Private Function ConvertNMEALongToDecimal(ByVal LongDirection As
String, ByVal NMEALongitude As String) As String
    Dim ResultLong As String
    Try
        NMEALongitude = NMEALongitude.Replace(".", ",")
        Dim tempLong As Double = Convert.ToDouble(NMEALongitude)
        tempLong = tempLong / 100
        Dim tempArray() As String = tempLong.ToString.Split(",")
        ResultLong = LongDirection & tempArray(0).ToString & "." &
(Convert.ToDouble(tempArray(1)) / 60).ToString("#####")
    Catch ex As Exception
        ResultLong = String.Empty
    End Try
    Return ResultLong
End Function

```



```
Private Function ConvertSpeedKnotsToKPH(ByVal Speed As String) As String
```

```
    Dim ResultSpeed As String
    Try
        Speed = Speed.Replace(".", ",")
        Dim tempSpeed As Double = Convert.ToDouble(Speed)
        tempSpeed = tempSpeed * 1.852
        ResultSpeed = tempSpeed.ToString
    Catch ex As Exception
        ResultSpeed = String.Empty
    End Try
    Return ResultSpeed
End Function
```

```
Private Sub ReadingGPRMC(ByVal GPRMCDData As String)
```

```
    Dim LineArray() As String = GPRMCDData.Split(",")
    If LineArray.Length = 13 Then
        Try
            If (LineArray(2) = "A" Or LineArray(2) = "a") Then
                Latitude = ConvertNMEAALatToDecimal(LineArray(4),
LineArray(3))
                Longitude = ConvertNMEALongToDecimal(LineArray(6),
LineArray(5))
                Speed = ConvertSpeedKnotsToKPH(LineArray(7))
                Direction = LineArray(8)
            Else
                Latitude = String.Empty
                Longitude = String.Empty
                Speed = String.Empty
                Direction = String.Empty
            End If
        Catch ex As Exception
            MessageBox.Show(ex.ToString & vbCrLf & ex.Message, "Error at
ReadGPRMC", MessageBoxButtons.OK, MessageBoxIcon.Error)
        End Try
    Else
        Latitude = "<Not Valid>"
        Longitude = "<Not Valid>"
        Speed = "<Not Valid>"
        Direction = "<Not Valid>"
    End If
End Sub
```

```
Private Sub MapCoordinate()
```

```
    Try
        Dim QueryAddress As New StringBuilder
        QueryAddress.Append("http://maps.google.com/maps?q=")
        If Latitude <> String.Empty Then
```

```

        QueryAddress.Append(Lattitude & "%2C")
    End If
    If Longitude <> String.Empty Then
        QueryAddress.Append(Longitude)
    End If
    NavigateWebBrowser(QueryAddress.ToString)
Catch ex As Exception
    MessageBox.Show(ex.ToString, "Can't map the coordinate",
    MessageBoxButtons.OK, MessageBoxIcon.Error)
End Try
End Sub

Private Sub NavigateWebBrowser(ByVal [myURL] As String)
    If WebBrowser.InvokeRequired Then
        Dim d As New UpdateWebBrowser(AddressOf NavigateWebBrowser)
        Me.Invoke(d, New Object() {[myURL]})
    Else
        txtURL.Text = [myURL]
        WebBrowser.Navigate([myURL])
    End If
End Sub

Private Sub ReadingSerialPortInput()
    Try
        Do
            SerialPortInputMessage = mySerialPort.ReadLine()
            If SerialPortInputMessage.Length <> 0 Then
                SerialPortInputMessage = SerialPortInputMessage.Substring(0,
                SerialPortInputMessage.Length - 1)
            End If
            txtProgLog.Invoke(New UpdateMainProgramDelegate(AddressOf
            UpdateTxtProgLog))
            If InStr(SerialPortInputMessage, "CMTI") Then
                SMSIndex =
                SerialPortInputMessage.Substring(SerialPortInputMessage.IndexOf(",") + 1)
                txtProgLog.Invoke(New UpdateMainProgramDelegate(AddressOf
                ReceiveNewSMS))
                mySerialPort.Write("AT+CMGR=" & SMSIndex & vbCr)
            End If
            If InStr(SerialPortInputMessage, "CMGR") Then
                StringCMGR = SerialPortInputMessage
            End If
            If (InStr(SerialPortInputMessage, "$GPRMC") Or
            InStr(SerialPortInputMessage, "$gprmc")) Then
                If myTimerCounter.Enabled = True Then
                    myTimerCounter.Enabled = False
                    stripCounter.Text = "Response received in " &
                    TimeCounter.ToString("#.#") & " s"
                End If
            End If
        Loop
    Catch ex As Exception
    End Try
End Sub

```

```

End If
ReadingGPRMC(SerialPortInputMessage)
StatLastTimeCoordinate = GetCMGRTime(StringCMGR)
UpdateGrpCoordinate()
WriteGPSLog()
txtProgLog.Invoke(New UpdateMainProgramDelegate(AddressOf
UpdateProgLogCoordinate))
MapCoordinateThread = New Thread(AddressOf MapCoordinate)
MapCoordinateThread.Start()
End If
If InStr(SerialPortInputMessage, "OK") Or
InStr(SerialPortInputMessage, "ERROR") Then
If isSendingATCommand = True Then
ControlGrpATCommandSender(True)
isSendingATCommand = False
End If
End If
If InStr(SerialPortInputMessage, "Mesin=") Then
If myTimerCounter.Enabled = True Then
myTimerCounter.Enabled = False
stripCounter.Text = "Response received in " &
TimeCounter.ToString("#.#") & " s"
End If
ReadingStatus(SerialPortInputMessage)
StatLastTimeStatus = GetCMGRTime(StringCMGR)
UpdateGrpStatus()
End If
Loop
Catch ex As Exception
Exit Try
End Try
End Sub
Private Function GetCMGRTime(ByVal strCMGR As String) As String
Dim StatLastTime As String
Dim SplitCMGRArray() As String = strCMGR.Split(Chr(34))
Dim SplitTimeArray() As String
SplitTimeArray = SplitCMGRArray(5).Split("+")
StatLastTime = SplitTimeArray(0)
Return StatLastTime
End Function
Private Sub ReadingStatus(ByVal StatusMessage As String)
Dim LineArray() As String = StatusMessage.Split(",")
Dim StatNoArray() As String
If InStr(LineArray(0), "ON") Then
StatEngine = True
ElseIf InStr(LineArray(0), "OFF") Then
StatEngine = False
End If

```

```

If InStr(LineArray(1), "ON") Then
    StatHorn = True
ElseIf InStr(LineArray(1), "OFF") Then
    StatHorn = False
End If
If InStr(LineArray(2), "ON") Then
    StatAlarm = True
ElseIf InStr(LineArray(2), "OFF") Then
    StatAlarm = False
End If
If InStr(LineArray(3), "ON") Then
    StatKey = True
ElseIf InStr(LineArray(3), "OFF") Then
    StatKey = False
End If
StatNoArray = LineArray(4).Split("=")
StatEmergencyNo = StatNoArray(1)
End Sub
Private Sub UpdateGrpCoordinate()
    If txtLatitude.InvokeRequired Then
        txtLatitude.Invoke(New UpdateCoordinate(AddressOf
UpdateGrpCoordinate))
    Else
        txtLatitude.Text = Latitude
        txtLongitude.Text = Longitude
        txtSpeed.Text = Speed & " kph"
        txtDirection.Text = Direction & Chr(176)
        txtLastTimeCheckedPosition.Text = StatLastTimeCoordinate
    End If
End Sub
Private Sub UpdateGrpStatus()
    If txtEngineStatus.InvokeRequired Then
        txtEngineStatus.Invoke(New UpdateStatus(AddressOf UpdateGrpStatus))
    Else
        If StatEngine = True Then
            txtEngineStatus.Text = "ON (Enabled)"
        Else
            txtEngineStatus.Text = "OFF (Disabled)"
        End If
        If StatHorn = True Then
            txtHornStatus.Text = "ON (Activated)"
        Else
            txtHornStatus.Text = "OFF (Deactivated)"
        End If
        If StatAlarm = True Then
            txtAlarmStatus.Text = "ON (Enabled)"
        Else
            txtAlarmStatus.Text = "OFF (Disabled)"
        End If
    End Sub

```

```

End If
If StatKey = True Then
    txtKeyStatus.ForeColor = Color.Red
    txtKeyStatus.Text = "ON (Triggered)"
Else
    txtKeyStatus.ForeColor = Color.Black
    txtKeyStatus.Text = "OFF (Not Triggered)"
End If
txtEmergencyNoStatus.Text = StatEmergencyNo
txtLastTimeCheckedStatus.Text = StatLastTimeStatus
End If
End Sub
Private Sub UpdateTxtProgLog()
    txtProgLog.AppendText("Receiving message: " & SerialPortInputMessage
    & ", length: " & SerialPortInputMessage.Length & vbCrLf)
    txtProgLog.ScrollToCaret()
    'If (InStr(SerialPortInputMessage, "$GPRMC") Or
InStr(SerialPortInputMessage, "$gprmc")) Then
    ' txtGPSLog.AppendText(SerialPortInputMessage & vbCrLf)
    ' txtGPSLog.ScrollToCaret()
    'End If
End Sub

Private Sub WriteGPSLog()
    If txtGPSLog.InvokeRequired Then
        txtGPSLog.Invoke(New UpdateGPSLog(AddressOf WriteGPSLog))
    Else
        txtGPSLog.AppendText("Latitude = " & Latitude & ", Longitude = " &
Longitude & ", Speed = " & Speed & " kph, Direction = " & Direction &
Chr(176) & vbCrLf)
    End If
End Sub

Private Sub ReceiveNewSMS()
    txtProgLog.AppendText("Receive new SMS at index: " & SMSIndex &
vbCrLf)
    txtProgLog.ScrollToCaret()
End Sub

Private Sub UpdateProgLogCoordinate()
    txtProgLog.AppendText("Receive Coordinate from GPRMC data" & vbCrLf
    & "Latitude = " & Latitude & ", longitude = " & Longitude & vbCrLf)
    txtProgLog.ScrollToCaret()
End Sub

Private Sub frmMain_FormClosing(ByVal sender As Object, ByVal e As
System.Windows.Forms.FormClosingEventArgs) Handles Me.FormClosing
    If mySerialPort.IsOpen Then

```

```

        DisconnectingSerialPort()
    End If
End
End Sub

```

```

Private Sub frmMain_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

```

```

    Me.WindowState = FormWindowState.Maximized
    SerialPortSettingInit()
    DefaultSerialPortSetting()
    grpPortSetting.Visible = False
    grpNetSetting.Visible = False
    ControlGrpATCommandSender(False)
    stripCounter.Text = ""
    SerialPortRefresh()
    myTimerCounter.Enabled = False
    WaitingTimeOut = 60

```

```
End Sub
```

```

Private Sub SerialPortRefresh()
    cbbCOMPorts.Items.Clear()
    Dim mySerialPortCount As Integer =
My.Computer.Ports.SerialPortNames.Count
    If mySerialPortCount <> 0 Then
        For i As Integer = 0 To mySerialPortCount - 1
            cbbCOMPorts.Items.Add(My.Computer.Ports.SerialPortNames(i))
        Next
        cbbCOMPorts.Sorted = True
        cbbCOMPorts.SelectedIndex = 0
        btnConnect.Enabled = True
        btnDisconnect.Enabled = False
        grpPortSetting.Enabled = True
        StatusStrip.Text = "Ready"
    Else
        btnConnect.Enabled = False
        btnDisconnect.Enabled = False
        grpPortSetting.Enabled = False
        cbbCOMPorts.Items.Clear()
        StatusStrip.Text = "No COM ports detected"
    End If
    txtProgLog.AppendText("Checking available COM Ports" & vbCrLf)
    txtProgLog.ScrollToCaret()
End Sub

```

```

Private Sub DefaultSerialPortSetting()
    cbbBaudRate.SelectedIndex = 5
    cbbDataBits.SelectedIndex = 3
    cbbParity.SelectedIndex = 0

```

```

    cbbStopBits.SelectedIndex = 0
End Sub

```

```

Private Sub SerialPortSettingInit()

```

```

    With cbbBaudRate.Items

```

```

        .Add("300")
        .Add("600")
        .Add("1200")
        .Add("2400")
        .Add("4800")
        .Add("9600")
        .Add("19200")
        .Add("38400")
        .Add("57600")
        .Add("115200")

```

```

    End With

```

```

    With cbbDataBits.Items

```

```

        .Add("5")
        .Add("6")
        .Add("7")
        .Add("8")

```

```

    End With

```

```

    With cbbParity.Items

```

```

        .Add("None")
        .Add("Odd")
        .Add("Even")
        .Add("Space")
        .Add("Mark")

```

```

    End With

```

```

    With cbbStopBits.Items

```

```

        .Add("1")
        .Add("1.5")
        .Add("2")

```

```

    End With

```

```

End Sub

```

```

Private Sub ConnectingSerialPort()

```

```

    Try

```

```

        With mySerialPort

```

```

            .PortName = cbbCOMPorts.Text
            .BaudRate = Convert.ToInt16(cbbBaudRate.Text)
            .DataBits = Convert.ToInt16(cbbDataBits.Text)

```

```

            Select Case cbbParity.Text

```

```

                Case "None"

```

```

                    .Parity = IO.Ports.Parity.None

```

```

                Case "Odd"

```

```

                    .Parity = IO.Ports.Parity.Odd

```

```

                Case "Even"

```

```

        .Parity = IO.Ports.Parity.Even
    Case "Mark"
        .Parity = IO.Ports.Parity.Mark
    Case "Space"
        .Parity = IO.Ports.Parity.Space
End Select
Select Case cbbStopBits.Text
    Case "1"
        .StopBits = IO.Ports.StopBits.One
    Case "1.5"
        .StopBits = IO.Ports.StopBits.OnePointFive
    Case "2"
        .StopBits = IO.Ports.StopBits.Two
End Select
mySerialPort.Open()
btnConnect.Enabled = False
btnDisconnect.Enabled = True
cbbCOMPorts.Enabled = False
btnRefresh.Enabled = False
grpPortSetting.Enabled = False
ControlGrpATCommandSender(True)
txtProgLog.AppendText("Connected to port " &
mySerialPort.PortName.ToString & vbCrLf)
txtProgLog.ScrollToCaret()
StatusStrip.Text = mySerialPort.PortName.ToString & " connected."
TimerSerialPortChecker.Enabled = True
ReadLineThread = New Thread(AddressOf ReadingSerialPortInput)
ReadLineThread.Start()
End With
Catch ex As Exception
    MessageBox.Show(ex.Message, "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error)
    txtProgLog.AppendText("Failed to connect to port " &
mySerialPort.PortName.ToString & vbCrLf)
    txtProgLog.ScrollToCaret()
End Try
End Sub

Private Sub DisconnectingSerialPort()
    Try
        Try
            mySerialPort.Close()
        Catch ex As Exception
            Exit Try
        End Try
    Try
        mySerialPort.DiscardInBuffer()
    End Try
End Sub

```



```

Catch ex As Exception
    Exit Try
End Try
Try
    mySerialPort.DiscardOutBuffer()
Catch ex As Exception
    Exit Try
End Try
cbbCOMPorts.Enabled = True
btnRefresh.Enabled = True
ControlGrpATCommandSender(False)
txtProgLog.AppendText("Disconnected from port " &
mySerialPort.PortName & vbCrLf)
txtProgLog.ScrollToCaret()
StatusStrip.Text = mySerialPort.PortName & " disconnected."
TimerSerialPortChecker.Enabled = False
SerialPortRefresh()
ReadLineThread.Abort()
Catch ex As Exception
    MessageBox.Show(ex.Message, "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error)
    txtProgLog.AppendText("Failed to disconnect from port " &
mySerialPort.PortName & vbCrLf)
    txtProgLog.ScrollToCaret()
End Try
End Sub

Private Sub btnAdvSetting_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnAdvSetting.Click
    If grpPortSetting.Visible = False Then
        grpPortSetting.Visible = True
        grpNetSetting.Visible = True
        btnAdvSetting.Text = "Less <<"
    Else
        grpPortSetting.Visible = False
        grpNetSetting.Visible = False
        btnAdvSetting.Text = "More >>"
    End If
End Sub

Private Sub btnDefault_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnDefault.Click
    DefaultSerialPortSetting()
End Sub

Private Sub btnConnect_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnConnect.Click
    If mySerialPort.IsOpen Then

```

```

        DisconnectingSerialPort()
    End If
    txtProgLog.AppendText("Trying to connect to port " & cbbCOMPorts.Text
& vbCrLf)
    txtProgLog.ScrollToCaret()
    ConnectingSerialPort()
End Sub

Private Sub btnDisconnect_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnDisconnect.Click
    If mySerialPort.IsOpen Then
        DisconnectingSerialPort()
    End If
End Sub

Private Sub btnClearProgLog_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles btnClearProgLog.Click
    If MessageBox.Show("Delete all program logs?", "Delete Program Logs",
MessageBoxButtons.YesNo, MessageBoxIcon.Exclamation,
MessageBoxDefaultButton.Button2) = Windows.Forms.DialogResult.Yes Then
        txtProgLog.Text = String.Empty
    End If
End Sub

Private Sub btnClearGPSLog_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles btnClearGPSLog.Click
    If MessageBox.Show("Delete all GPS data?", "Delete GPS Data",
MessageBoxButtons.YesNo, MessageBoxIcon.Exclamation,
MessageBoxDefaultButton.Button2) = Windows.Forms.DialogResult.Yes Then
        txtGPSLog.Text = String.Empty
    End If
End Sub

Private Sub TimerSerialPortChecker_Tick(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles TimerSerialPortChecker.Tick
    If mySerialPort.PortName Is Nothing Then
        DisconnectingSerialPort()
        MessageBox.Show(mySerialPort.PortName & " lost connection",
"Warning", MessageBoxButtons.OK, MessageBoxIcon.Error)
    End If
    If mySerialPort.IsOpen = False Then
        DisconnectingSerialPort()
        MessageBox.Show(mySerialPort.PortName & " lost connection",
"Warning", MessageBoxButtons.OK, MessageBoxIcon.Error)
    End If
End Sub

```

```

Private Sub btnRefresh_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnRefresh.Click
    SerialPortRefresh()
End Sub

```

```

Private Sub btnGo_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnGo.Click
    Try
        If InStr(txtURL.Text, "$GPRMC") Then
            ReadingGPRMC(txtURL.Text)
            If MessageBox.Show("Do you want to add this data to GPS log?",
"Question", MessageBoxButtons.YesNo, MessageBoxIcon.Question) =
Windows.Forms.DialogResult.Yes Then
                WriteGPSLog()
            End If
            MapCoordinate()
        Else
            WebBrowser.Navigate(txtURL.Text)
        End If
    Catch ex As Exception
        MessageBox.Show(ex.ToString, "Can't navigate Web Browser",
MessageBoxButtons.OK, MessageBoxIcon.Error)
    End Try
End Sub

```

```

Private Sub btnSendATCommand_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles btnSendATCommand.Click
    If mySerialPort.IsOpen Then
        Try
            mySerialPort.Write(txtATCommand.Text & vbCr)
            txtProgLog.AppendText("Sending AT Command: " &
txtATCommand.Text & vbCrLf)
            txtProgLog.ScrollToCaret()
            txtATCommand.Text = String.Empty
            isSendingATCommand = True
            ControlGrpATCommandSender(False)
        Catch ex As Exception
            MessageBox.Show(ex.ToString, "Can't send AT Command",
MessageBoxButtons.OK, MessageBoxIcon.Error)
        End Try
    End If
End Sub

```

```

Private Sub btnSendSMS_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnSendSMS.Click
    If mySerialPort.IsOpen Then
        Try
            mySerialPort.Write("AT+CMGS=" & cbbPhoneNo.Text & vbCr)

```

```

mySerialPort.Write(txtMessage.Text & Chr(26) & vbCr)
txtProgLog.AppendText("Sending SMS to phone no: " &
cbbPhoneNo.Text & vbCrLf & "Send Message:" & vbCrLf & txtMessage.Text &
vbCrLf)

```

```

txtProgLog.ScrollToCaret()
isSendingATCommand = True
ControlGrpATCommandSender(False)
If cbbPhoneNo.Items.Contains(cbbPhoneNo.Text) = False Then
    cbbPhoneNo.Items.Add(cbbPhoneNo.Text)
End If
cbbPhoneNo.Text = String.Empty
txtMessage.Text = String.Empty
Catch ex As Exception
    MessageBox.Show(ex.ToString, "Can't send SMS",
MessageBoxButtons.OK, MessageBoxIcon.Error)
End Try
End If
End Sub

```

```

Private Sub menuAbout_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles menuAbout.Click
    MessageBox.Show("Vehicle Tracker" & vbCrLf & "Created by Syukron
Zahri" & vbCrLf & "Electrical Engineering, University of Indonesia" & vbCrLf &
vbCrLf & "This program works only with Wavecom modem", "About",
MessageBoxButtons.OK, MessageBoxIcon.Information)
End Sub

```

```

Private Sub menuExit_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles menuExit.Click
    Me.Close()
End Sub

```

```

Private Sub rdbNoProxy_CheckedChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles rdbNoProxy.CheckedChanged
    lblProxy.Enabled = False
    lblProxyPort.Enabled = False
    txtProxy.Enabled = False
    txtProxyPort.Enabled = False
End Sub

```

```

Private Sub rdbUseProxy_CheckedChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles rdbUseProxy.CheckedChanged
    lblProxy.Enabled = True
    lblProxyPort.Enabled = True
    txtProxy.Enabled = True
    txtProxyPort.Enabled = True
End Sub

```

```

Private Sub btnSetProxy_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnSetProxy.Click
    If rdbNoProxy.Checked = True Then
        UseProxy(":")
        txtProgLog.AppendText("Internet Setting: Not Using Proxy" & vbCrLf)
    ElseIf rdbUseProxy.Checked = True Then
        Try
            UseProxy(txtProxy.Text & ":" & txtProxyPort.Text)
            txtProgLog.AppendText("Internet Setting: Using proxy (" &
txtProxy.Text & ":" & txtProxyPort.Text & ") " & vbCrLf)
        Catch ex As Exception
            MessageBox.Show(ex.ToString, "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error)
        End Try
    End If
End Sub

Private Sub SendSMS(ByVal PhoneNo As String, ByVal SMSMessage As
String)
    If mySerialPort.IsOpen Then
        Try
            mySerialPort.Write("AT+CMGS=" & PhoneNo & vbCr)
            mySerialPort.Write(SMSMessage & Chr(26) & vbCr)
            isSendingATCommand = True
            ControlGrpATCommandSender(False)
            txtProgLog.AppendText("Sending SMS to phone no: " & PhoneNo &
vbCrLf & "Message: " & SMSMessage & vbCrLf)
            txtProgLog.ScrollToCaret()
        Catch ex As Exception
            MessageBox.Show(ex.ToString, "Can't send SMS",
MessageBoxButtons.OK, MessageBoxIcon.Error)
        End Try
    End If
End Sub

Private Sub UpdateGrpSendCommand()
    If cbbPhoneNumber.Items.Contains(cbbPhoneNumber.Text) = False Then
        cbbPhoneNumber.Items.Add(cbbPhoneNumber.Text)
    End If
    grpSendCommand.Enabled = False
End Sub

Private Sub ControlGrpATCommandSender(ByVal [Condition] As Boolean)
    If grpSendCommand.InvokeRequired Then
        Dim d As New UpdateFormControl(AddressOf
ControlGrpATCommandSender)
        Me.Invoke(d, New Object() {[Condition]})
    Else
        If Condition = True Then
            grpSendCommand.Enabled = True
        End If
    End Sub

```

```

        grpATCommand.Enabled = True
        grpSendSMS.Enabled = True
    Else
        grpSendCommand.Enabled = False
        grpATCommand.Enabled = False
        grpSendSMS.Enabled = False
    End If
End If
End Sub

```

```

Private Sub myTimerCounter_Tick(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles myTimerCounter.Tick
    TimeCounter += 0.1
    stripCounter.Text = "Waiting for response in: " &
TimeCounter.ToString("#.#") & " s"
    If TimeCounter >= WaitingTimeOut Then
        myTimerCounter.Enabled = False
        stripCounter.Text = "Response timeout"
    End If
End Sub

```

```

Private Sub btnEnableEngine_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles btnEnableEngine.Click
    If cbbPhoneNumber.Text = String.Empty Then
        MessageBox.Show("Please input the Phone No!", "Error",
MessageBoxButtons.OK, MessageBoxIcon.Error)
    Else
        SendSMS(cbbPhoneNumber.Text, "mesin on")
        UpdateGrpSendCommand()
    End If
End Sub

```

```

Private Sub btnDisableEngine_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles btnDisableEngine.Click
    If cbbPhoneNumber.Text = String.Empty Then
        MessageBox.Show("Please input the Phone No!", "Error",
MessageBoxButtons.OK, MessageBoxIcon.Error)
    Else
        SendSMS(cbbPhoneNumber.Text, "mesin off")
        UpdateGrpSendCommand()
    End If
End Sub

```

```

Private Sub btnActivateHorn_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles btnActivateHorn.Click
    If cbbPhoneNumber.Text = String.Empty Then
        MessageBox.Show("Please input the Phone No!", "Error",
MessageBoxButtons.OK, MessageBoxIcon.Error)
    Else

```

```

        SendSMS(cbbPhoneNumber.Text, "klakson on")
        UpdateGrpSendCommand()
    End If
End Sub

```

```

Private Sub btnDeactivateHorn_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnDeactivateHorn.Click
    If cbbPhoneNumber.Text = String.Empty Then
        MessageBox.Show("Please input the Phone No!", "Error",
        MessageBoxButtons.OK, MessageBoxIcon.Error)
    Else
        SendSMS(cbbPhoneNumber.Text, "klakson off")
        'UpdateGrpSendCommand()
    End If
End Sub

```

```

Private Sub btnGetPosition_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnGetPosition.Click
    If cbbPhoneNumber.Text = String.Empty Then
        MessageBox.Show("Please input the Phone No!", "Error",
        MessageBoxButtons.OK, MessageBoxIcon.Error)
    Else
        SendSMS(cbbPhoneNumber.Text, "posisi")
        'UpdateGrpSendCommand()
        TimeCounter = 0
        myTimerCounter.Enabled = True
    End If
End Sub

```

```

Private Sub btnGetStatus_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnGetStatus.Click
    If cbbPhoneNumber.Text = String.Empty Then
        MessageBox.Show("Please input the Phone No!", "Error",
        MessageBoxButtons.OK, MessageBoxIcon.Error)
    Else
        SendSMS(cbbPhoneNumber.Text, "status")
        'UpdateGrpSendCommand()
        TimeCounter = 0
        myTimerCounter.Enabled = True
    End If
End Sub

```

```

Private Sub btnReset_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnReset.Click
    If cbbPhoneNumber.Text = String.Empty Then
        MessageBox.Show("Please input the Phone No!", "Error",
        MessageBoxButtons.OK, MessageBoxIcon.Error)
    Else

```

```

        SendSMS(cbbPhoneNumber.Text, "reset")
        'UpdateGrpSendCommand()
    End If
End Sub

```

```

Private Sub btnSetEmergencyNumber_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles btnSetEmergencyNumber.Click
    If cbbPhoneNumber.Text = String.Empty Then
        MessageBox.Show("Please input the Phone No!", "Error",
        MessageBoxButtons.OK, MessageBoxIcon.Error)
    Else
        Dim tmpNumber As New StringBuilder
        If txtEmergencyNo.Text.StartsWith(Chr(34)) = False Then
            tmpNumber.Append(Chr(34))
        End If
        tmpNumber.Append(txtEmergencyNo.Text)
        If txtEmergencyNo.Text.EndsWith(Chr(34)) = False Then
            tmpNumber.Append(Chr(34))
        End If
        txtEmergencyNo.Text = tmpNumber.ToString
        SendSMS(cbbPhoneNumber.Text, "no " & tmpNumber.ToString)
        'UpdateGrpSendCommand()
    End If
End Sub

```

```

Private Sub stripCounter_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles stripCounter.Click
    If myTimerCounter.Enabled = False Then
        stripCounter.Text = String.Empty
    End If
End Sub

```

```

Private Sub btnEnableAlarm_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles btnEnableAlarm.Click
    If cbbPhoneNumber.Text = String.Empty Then
        MessageBox.Show("Please input the Phone No!", "Error",
        MessageBoxButtons.OK, MessageBoxIcon.Error)
    Else
        SendSMS(cbbPhoneNumber.Text, "alarm on")
        'UpdateGrpSendCommand()
    End If
End Sub

```

```

Private Sub btnDisableAlarm_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles btnDisableAlarm.Click
    If cbbPhoneNumber.Text = String.Empty Then
        MessageBox.Show("Please input the Phone No!", "Error",
        MessageBoxButtons.OK, MessageBoxIcon.Error)
    End If
End Sub

```



```

Else
    SendSMS(cbbPhoneNumber.Text, "alarm off")
    'UpdateGrpSendCommand()
End If
End Sub

```

```

Private Sub btnDeleteSMS_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnDeleteSMS.Click
    If mySerialPort.IsOpen Then
        Try
            mySerialPort.Write("AT+CMGD=1,4" & vbCr)
            txtProgLog.AppendText("Deleting all SMS" & vbCrLf)
            txtProgLog.ScrollToCaret()
            isSendingATCommand = True
            ControlGrpATCommandSender(False)
        Catch ex As Exception
            MessageBox.Show(ex.ToString, "Can't delete all SMS",
MessageBoxButtons.OK, MessageBoxIcon.Error)
        End Try
    End If
End Sub

```

```

Private Sub btnSaveProgLog_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles btnSaveProgLog.Click
    With mySaveFileDialog
        .Filter = "Text file (*.txt)|*.txt|All files|*.*"
        .AddExtension = True
        .FileName = "[Vehicle Tracker] program log"
        .OverwritePrompt = True
    End With
    If mySaveFileDialog.ShowDialog = Windows.Forms.DialogResult.OK Then
        System.IO.File.WriteAllText(mySaveFileDialog.FileName,
txtProgLog.Text)
    End If
End Sub

```

```

Private Sub btnSaveGPSLog_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles btnSaveGPSLog.Click
    With mySaveFileDialog
        .Filter = "Text file (*.txt)|*.txt|All files|*.*"
        .AddExtension = True
        .FileName = "[Vehicle Tracker] GPS log"
        .OverwritePrompt = True
    End With
    If mySaveFileDialog.ShowDialog = Windows.Forms.DialogResult.OK Then
        System.IO.File.WriteAllText(mySaveFileDialog.FileName,
txtGPSLog.Text)
    End If

```

End Sub
End Class

