

**RANCANG BANGUN PENGIRIMAN DAN
PENERIMAAN DATA MELALUI PELAYANAN
PENSINYALAN SATELIT GARUDA UNTUK
KOMUNIKASI BERGERAK**

TUGAS AKHIR

Oleh

IQBAL ISKANDAR

06 06 04 267 0



**DEPARTEMEN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS INDONESIA
GENAP 2007/2008**

**RANCANG BANGUN PENGIRIMAN DAN
PENERIMAAN DATA MELALUI PELAYANAN
PENSINYALAN SATELIT GARUDA UNTUK
KOMUNIKASI BERGERAK**

TUGAS AKHIR

Oleh

IQBAL ISKANDAR

06 06 04 267 0



**TUGAS AKHIR INI DIAJUKAN UNTUK MELENGKAPI
SEBAGIAN PERSYARATAN MENJADI SARJANA TEKNIK**

**DEPARTEMEN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS INDONESIA
GENAP 2007/2008**

PERNYATAAN KEASLIAN TUGAS AKHIR

Saya menyatakan dengan sesungguhnya bahwa tugas akhir dengan judul :

RANCANG BANGUN PENGIRIMAN DAN PENERIMAAN DATA MELALUI PELAYANAN PENSINYALAN SATELIT GARUDA UNTUK KOMUNIKASI BERGERAK

yang dibuat untuk melengkapi sebagian persyaratan menjadi Sarjana Teknik pada Program Studi Teknik Elektro Departemen Teknik Elektro Fakultas Teknik Universitas Indonesia, sejauh yang saya ketahui bukan merupakan tiruan atau duplikasi dari tugas akhir yang sudah dipublikasikan dan atau pernah dipakai untuk mendapatkan gelar kesarjanaan di lingkungan Universitas Indonesia maupun di Perguruan Tinggi atau Instansi manapun, kecuali bagian yang sumber informasinya dicantumkan sebagaimana mestinya.

Depok, 9 Juli 2008

IQBAL ISKANDAR
NPM 06 06 04 267 0

PENGESAHAN

Tugas akhir dengan judul :

RANCANG BANGUN PENGIRIMAN DAN PENERIMAAN DATA MELALUI PELAYANAN PENSINYALAN SATELIT GARUDA UNTUK KOMUNIKASI BERGERAK

dibuat untuk melengkapi sebagian persyaratan menjadi Sarjana Teknik pada program studi Teknik Elektro Departemen Teknik Elektro Fakultas Teknik Universitas Indonesia. Tugas akhir ini telah diajukan pada tanggal 22 April 2008 dan dinyatakan memenuhi syarat/sah sebagai tugas akhir pada Departemen Teknik Elektro Fakultas Teknik Universitas Indonesia.

Depok, 9 Juli 2008

Dosen Pembimbing,

Pembimbing 1

Pembimbing 2

(Ir. Rochmah N Sukardi Ny MSc)

(Dr. Ir. Arman D. Diponegoro)

NIP: 130 536 625

NIP : 131 476 472

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada :

Ir. Rochmah N Sukardi Ny MSc

Dr. Ir. Arman Djohan Diponegoro

selaku Dosen pembimbing 1 dan 2 yang telah bersedia meluangkan waktu untuk memberi pengarahan, diskusi dan bimbingan serta persetujuan sehingga tugas akhir ini dapat selesai dengan baik.

IQBAL ISKANDAR
NPM 06 06 04 267 0
Departemen Teknik Elektro

Dosen Pembimbing
I.Ir. Rochmah N Sukardi Ny MSc
II.Dr.Ir.Arman Djohan Diponegoro

**RANCANG BANGUN PENGIRIMAN DAN PENERIMAAN DATA
MELALUI PELAYANAN PENSINYALAN SATELIT GARUDA PADA
KOMUNIKASI BERGERAK**

ABSTRAK

Tugas akhir ini merancang sistem pengambilan dan pengiriman data GPS dengan menggunakan mikrokontroler sebagai pengolah data GPS. Data yang diperoleh dari GPS berupa format NMEA 0183. Data tersebut disimpan di dalam memori mikrokontroler. Mikrokontroler akan memproses data menjadi data posisi. Informasi ini dikirimkan melalui perangkat PASTI/*Byru Marine* ke satelit ACeS Garuda 1. Proses ini berlangsung terus-menerus, sehingga informasi posisi bersifat *real time*. Dari satelit ACeS Garuda 1, informasi tersebut ditransmisikan dan diterima oleh stasiun Bumi.

Dalam tugas akhir ini kami menggunakan mikrokontroler untuk mengambil, menyimpan, memproses, dan mengirimkan data. Kami menggunakan bahasa pemrograman assembler yang sesuai dengan spesifikasi mikrokontroler. RS232 digunakan sebagai antarmuka antara GPS, mikrokontroler, dan perangkat PASTI/*Byru Marine* untuk mengambil dan mengirimkan data.

Informasi yang terdiri dari data posisi yang terdapat pada *server* stasiun Bumi, dapat diakses menggunakan FTP *client*.

Kata kunci: GPS, NMEA 0183, Satelit ACeS Garuda 1, Mikrokontroler, Perangkat PASTI/*Byru Marine*, RS232, FTP Client

IQBAL ISKANDAR
NPM 06 06 04 267 0
Electrical Engineering Department

Supervisor
I.Ir. Rochmah N Sukardi Ny MSc
II.Dr.Ir.Arman Djohan Diponegoro

**DESIGNS AND BUILDS DATA TRANSMIT AND ACCEPTANCE
THROUGH SIGNALLING SERVICES OF GARUDA SATELLITE FOR
MOBILE COMMUNICATION**

ABSTRACT

This final project designs and builds the system for retrieving and transmitting the GPS data using microcontroller. The data is retrieved from GPS in NMEA 0183 format. The data will be saved in the microcontroller. The microcontroller will process the data to set the position information. This information then sent through PASTI/Byru Marine terminal to ACeS Garuda 1 satellite. This process will run continuously so the position will be real time. From ACeS Garuda 1 satellite, the information is transmitted and received by earth station.

In this project we use the microcontroller for retrieving, saving, processing, and sending the data. We build this functionality using assembly programming language, which suitable with microcontroller specification. We use RS232 converter as interface between GPS, microcontroller, and PASTI/Byru Marine terminal to retrieving and sending the data.

The information consists position data at earth station server, can be access using FTP client.

Keyword: GPS, NMEA 0183, ACeS Garuda 1 Satellite, Microcontroller, PASTI /Byru Marine Terminal, RS232 Converter, FTP Client

DAFTAR ISI

	Halaman
PERNYATAAN KEASLIAN TUGAS AKHIR	ii
PENGESAHAN	iii
UCAPAN TERIMA KASIH	iv
ABSTRAK	v
ABSTRACT	vi
DAFTAR ISI	vii
DAFTAR GAMBAR	x
DAFTAR TABEL	xii
DAFTAR LAMPIRAN	xiii
DAFTAR SINGKATAN	xiv
BAB I PENDAHULUAN	1
1.1 LATAR BELAKANG	1
1.2 PERUMUSAN MASALAH	2
1.3 TUJUAN	2
1.4 BATASAN MASALAH	2
1.5 SISTEMATIKA PENULISAN	3
BAB II LANDASAN TEORI	4
2.1 MIKROKONTROLER AT89S51	4
2.1.1 Arsitektur AT89S51	4
2.1.2 Fitur AT89S51	6
2.1.3 Konfigurasi Pin AT89S51	7
2.1.4 Peta Memori	8
2.1.5 Counter dan Timer	12
2.1.6 Masukan / Keluaran Data Serial	16
2.1.7 Komunikasi Serial AT89S51	17
2.1.7.1 Sistem Komunikasi Serial	17
2.1.7.2 Protokol Serial	18
2.1.7.3 Inisialisasi UART	18
2.1.7.4 Pengiriman Data	19

2.1.7.5	<i>Penerimaan Data</i>	19
2.2	KONVERTER RS232	20
2.3	GLOBAL POSITIONING SYSTEM (GPS)	23
2.3.1	Segmen Penyusun GPS	24
2.3.1.1	<i>Segmen Angkasa (Space Segment)</i>	25
2.3.1.2	<i>Segmen Sistem Kontrol (Control Segment)</i> ..	26
2.3.1.3	<i>Segmen Pengguna (User Segment)</i>	27
2.3.2	Perhitungan Posisi GPS	27
2.3.2.1	<i>Prinsip Penentuan Posisi Absolut dengan GPS</i>	27
2.3.2.2	<i>Ketelitian Posisi Absolut</i>	29
2.3.3	Format Data GPS	30
2.4	KOMUNIKASI SATELIT ACeS GARUDA 1	31
2.4.1	Spesifikasi Satelit ACeS Garuda 1	31
2.4.2	Metode Pengiriman Data pada Perangkat PASTI/ <i>Byru Marine</i>	33
2.4.2.1	<i>Tampilan Data</i>	33
2.5	FILE TRANSFER PROTOCOL (FTP)	34
2.5.1	Protokol FTP	35
2.5.2	Antarmuka FTP	37
BAB III	RANCANG BANGUN	39
3.1	PRINSIP KERJA SISTEM	39
3.1.1	Blok Diagram dan Fungsinya	39
3.2	PERANGKAT KERAS	40
3.3	PERANGKAT LUNAK	41
3.3.1	Pemrograman Mikrokontroler	41
3.3.2	Pemrograman FTP <i>Client</i>	45
3.4	FORMAT DATA	47
3.4.1	Format <i>Serial Data Output</i> (SDO)	47
3.4.2	Format Pengiriman Data	48
BAB IV	PENGUJIAN DAN ANALISIS SISTEM	50
4.1	HASIL RANCANG BANGUN	50

4.2 PENGUJIAN SISTEM	50
4.2.1 Menampilkan Data Keluaran GPS	51
4.2.2 Menampilkan Data Keluaran Kontroler	53
4.2.2.1 <i>Data Keluaran Tanpa Diberikan Masking pada Data Mentah GPS</i>	53
4.2.2.2 <i>Data Keluaran dengan Memberikan Masking \$GPRMC pada Data Mentah GPS</i>	54
4.2.2.3 <i>Data Keluaran dengan Memberikan Masking Posisi pada Data Masking \$GPRMC</i>	55
4.2.2.4 <i>Data Keluaran dengan Memberikan Masking Angka pada Data Masking Posisi</i>	56
4.2.3 Menampilkan Data Final yang Dikirimkan dengan Mengakses <i>Server</i> Menggunakan <i>FTP Client</i>	57
4.3 ANALISIS SISTEM	58
4.3.1 Analisis Pengujian Pertama	58
4.3.2 Analisis Pengujian Kedua	58
4.3.3 Analisis Pengujian Ketiga	72
BAB V KESIMPULAN	75
DAFTAR ACUAN	76
DAFTAR PUSTAKA	78
LAMPIRAN	79

DAFTAR GAMBAR

	Halaman
Gambar 2.1 Blok diagram fungsional AT89S51	6
Gambar 2.2 Pin AT89S51	7
Gambar 2.3 Peta memori AT89S51	9
Gambar 2.4 Internal memori AT89S51	9
Gambar 2.5 Susunan bit register PSW	11
Gambar 2.6 Peta memori RAM AT89S51	12
Gambar 2.7 Susunan bit register TMOD	12
Gambar 2.8 Mode 0 pencacah biner 13 bit	13
Gambar 2.9 Mode 1 pencacah biner 16 bit	14
Gambar 2.10 Mode 2 pencacah biner 8 bit dengan isi ulang	14
Gambar 2.11 Mode 3 gabungan pencacah biner 16 bit dan 8 bit	15
Gambar 2.12 Susunan bit register TCON	15
Gambar 2.13 Susunan bit register SCON	16
Gambar 2.14 Susunan bit register PCON	17
Gambar 2.15 Format waktu untuk <i>asynchronous</i> UART	18
Gambar 2.16 Spesifikasi level logik RS-232C	21
Gambar 2.17 Konektor DB9	22
Gambar 2.18 Fungsi dasar GPS	23
Gambar 2.19 Orbit satelit GPS	24
Gambar 2.20 Tiga segmen pada GPS	25
Gambar 2.21 Posisi pergerakan 28 satelit GPS pada tanggal 14 April 2001	26
Gambar 2.22 Distribusi 28 satelit GPS pada tanggal 14 april 2001	26
Gambar 2.23 Sinyal dari 4 satelit	27
Gambar 2.24 Sistem koordinat 3 dimensi	28
Gambar 2.25 Data keluaran GPS ditampilkan pada komputer dengan menggunakan <i>hyperterminal</i>	31
Gambar 2.26 Satelit ACeS Garuda 1	32

Gambar 2.27 Pengiriman data melalui perangkat PASTI/Byru Marine dengan menggunakan <i>hyper terminal</i>	34
Gambar 2.28 Pola hubungan FTP <i>client</i> dan FTP <i>server</i>	37
Gambar 2.29 Salah satu tampilan FTP grafikal	38
Gambar 3.1 Blok diagram sistem	39
Gambar 3.2 Skematik diagram mikrokontroler	41
Gambar 3.3 Diagram alir program mikrokontroler	42
Gambar 3.4 Diagram alir program FTP <i>client</i>	45
Gambar 3.5 Tampilan akses FTP pada <i>server</i> dengan alamat 152.118.101.8	46
Gambar 3.6 Pengiriman data GPS setelah diolah di mikrokontroler	49
Gambar 4.1 Perangkat keras sistem	50
Gambar 4.2 Blok diagram tempat pengujian sistem	51
Gambar 4.3 Data keluaran GPS yang ditampilkan pada <i>hyper terminal</i> ..	52
Gambar 4.4 Data keluaran GPS pada kondisi tanpa sinyal (<i>void</i>)	52
Gambar 4.5 Tampilan data dari kontroler pada <i>hyper terminal</i> tanpa <i>masking</i>	53
Gambar 4.6 Tampilan data dari kontroler pada <i>hyper terminal</i> dengan <i>masking</i> data \$GPRMC	54
Gambar 4.7 Tampilan data dari kontroler pada <i>hyper terminal</i> dengan <i>masking</i> data posisi	55
Gambar 4.8 Tampilan data posisi dalam bentuk angka	57
Gambar 4.9 Data GPS pada <i>server</i> dengan alamat 152.118.101.8	58
Gambar 4.10 Pengiriman data <i>latitude</i> dan <i>longitude</i> dengan sekali pengiriman	72

DAFTAR TABEL

	Halaman
Tabel 2.1 Fungsi-Fungsi Pin pada Port 3 AT89S51	8
Tabel 2.2 Alamat SFR	10
Tabel 2.3 Fungsi – Fungsi Bit pada PSW	11
Tabel 2.4 Konfigurasi Bit RS0 dan RS1	11
Tabel 2.5 Fungsi-Fungsi Bit TMOD	12
Tabel 2.6 Konfigurasi Mode <i>Timer/Counter</i>	13
Tabel 2.7 Fungsi-Fungsi Bit TCON	15
Tabel 2.8 Fungsi-Fungsi Bit SCON	16
Tabel 2.9 Konfigurasi Mode SM0 dan SM1	17
Tabel 2.10 Fungsi-Fungsi Bit PCON	17
Tabel 2.11 Spesifikasi RS-232C	21
Tabel 2.12 Konverter RS232C pada DB9	22
Tabel 2.13 Format Data GPS	30
Tabel 2.14 Perintah-Perintah FTP	35
Tabel 4.1 Hasil Perhitungan Total Waktu Pengiriman Data	65

DAFTAR LAMPIRAN

	Halaman
LAMPIRAN 1 Program Pengolahan Data GPS tanpa <i>Masking</i>	79
LAMPIRAN 2 Program Pengolahan Data GPS dengan <i>Masking</i> \$GPRMC	81
LAMPIRAN 3 Program Pengolahan Data Posisi GPS dengan <i>Masking</i> \$GPRMC	83
LAMPIRAN 4 Skematik Diagram Modul Mikrokontroler AT89S51 ...	86

DAFTAR SINGKATAN

ALE	Address Latch Enable
ACeS	Asia Cellular Satellite
CISC	Complex Instruction Set Computing
CPU	Central Processing Unit
CR	Carriage Return
CTS	Clear To Send
DCD	Data Carrier Detect
DCE	Communication Equipment
DPH	Pointer High
DPL	Data Pointer Low
DOP	Dilution of Precision
DPTR	Data Pointer
DSR	Data Set Ready
DTE	Data Terminal Equipment
DTR	Data Terminal Ready
EA	External Access Enable
EIA	Electronic Industries Association
FTP	File Transfer Protocol
GCS	Ground Control Station
GDOP	Geometrical DOP
GND	Signal Ground
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
HDOP	Horizontal DOP
I/O	Input/Output
IP	Internet Protocol
ISP	In System Programmable
LF	Line Feed
LSB	Least Significant Bit
MCS	Master Control Station
MS	Monitor Station
MSB	Most Significant Bit
NAVSTAR	NAVigation System with Timing And Ranging

NCC	Network Control Center
NGS	National Gateway Station
NMEA	National Marine Electronics Association
PASTI	Pasang Telepon Sendiri
PC	Program Counter
PCON	Power Mode Control
PDOP	Positional DOP
PPS	Precise Positioning Service
PSEN	Program Store Enable
PSW	Program Status Word
RD	Receive Data
RDRF	Receiver Data Register Full
RI	Ring Indicator
RTS	Ready To Send
SCON	Serial Port Control
SDO	Serial Data Output
SFR	Special Function Register
SP	Stack Pointer
SPS	Standard Positioning Service
TCON	Timer Control
TD	Transmit Data
TDRE	Transmitter Data Register Empty
TL0	Timer 0 low byte
TL1	Timer 1 low byte
TH0	Timer 0 high byte
TH1	Timer 1 <i>high</i> byte
TMOD	Timer Mode Control
UART	Universal Asynchronous Receiver/Transmitter
UTC	Universal Time Coordinat
VDOP	Vertical DOP

BAB I

PENDAHULUAN

1.1 LATAR BELAKANG

Beberapa tahun belakangan, Indonesia sering ditimpa oleh musibah di wilayah perairannya. Seperti kecelakaan transportasi kelautan.

Sehubungan dengan fakta tersebut, diperlukan usaha untuk mengantisipasi agar kecelakaan transportasi kelautan tidak terjadi. Untuk itu, diperlukan suatu sistem pemantau transportasi kelautan untuk pengawasan atau pemantauan transportasi laut di wilayah perairan Indonesia. Pemantauan ini berupa pengiriman data lokasi kapal atau alat transportasi laut lainnya. Sistem pemantauan lokasi transportasi kelautan yang digunakan saat ini adalah *Global Positioning System* (GPS) melalui satelit. Akan tetapi, penggunaan teknologi ini memerlukan biaya mahal, baik untuk biaya pembelian perangkat untuk GPS maupun untuk penyewaan jasa satelit. Oleh karena itu, hanya perusahaan transportasi laut skala besar yang bisa memanfaatkan teknologi ini.

Sementara itu, teknologi komunikasi (menggunakan satelit) dewasa ini mengalami perkembangan yang ditandai dengan penggunaan jasa telekomunikasi satelit yang juga semakin meningkat. Dampak perkembangan ini juga berpengaruh kepada penurunan biaya operasional sistem teknologi komunikasi ini. Namun demikian, kita bisa menyiasati biaya operasional untuk penerapan sistem komunikasi melalui satelit dengan cara memanfaatkan celah dalam sistem ini.

Satelit sebagai alat komunikasi memiliki spesifikasi dalam segi pemanfaatan. Salah satunya adalah jasa pelayanan data dan suara. Alat pemantau transportasi laut dapat memanfaatkan jasa layanan data yang disediakan oleh satelit dengan fungsi untuk itu. Salah satu satelit yang beredar di wilayah udara Indonesia dan yang menyediakan layanan seperti disebutkan adalah satelit Garuda 1.

Dengan memanfaatkan celah yang ada pada sistem komunikasi satelit Garuda 1, kami akan merancang dan membuat sistem pemantauan transportasi

laut pada daerah perairan Indonesia dengan memanfaatkan perangkat yang dapat mengolah data GPS dan dapat berkomunikasi dengan satelit. Dengan demikian, biaya operasional pun bisa ditekan bahkan bisa mendekati nol.

1.2. PERUMUSAN MASALAH

Berdasarkan latar belakang dapat dirumuskan masalah yaitu bagaimana merancang perangkat sistem pemantauan lokasi alat transportasi laut berbiaya murah dengan memanfaatkan satelit Garuda 1. Rumusan masalah dapat diperinci menjadi tiga pertanyaan sebagai berikut.

1. Bagaimana merancang dan membuat sistem pengiriman data GPS ke satelit Garuda 1 dengan memanfaatkan mikrokontroler?
2. Bagaimana mengolah data GPS agar dikenali oleh satelit Garuda 1 sebelum dikirimkan dengan memanfaatkan celah yang ada?
3. Bagaimana menerima data yang telah dikirimkan ke satelit Garuda 1 melalui koneksi internet?

1.3. TUJUAN

Tujuan tugas akhir ini adalah merancang dan membuat sistem pengiriman dan penerimaan data lokasi melalui pelayanan satelit ACeS Garuda 1 dan komunikasi antara mikrokontroler 89S51 dengan Perangkat PASTI (Pasang Telepon Sendiri)/*Byru Marine* dan GPS dengan biaya murah.

1.4. BATASAN MASALAH

Dalam perancangan sistem ini, terdapat beberapa pembatasan masalah, antara lain:

1. Perancangan sistem ini menggunakan modul MCS51 yang diproduksi oleh ATMEL.
2. Komunikasi antara modul dengan GPS dan perangkat PASTI/*Byru Marine* adalah melalui antarmuka RS232.
3. Sistem ini mengirimkan sirkulasi data GPS secara terus-menerus.
4. Untuk menerima data, sistem ini menggunakan koneksi internet dengan memanfaatkan fasilitas *FTP client*.

1.5. SISTEMATIKA PENULISAN

Penulisan tugas akhir ini disusun berdasarkan sistematika sebagai berikut.

Bab I Pendahuluan

Berisi latar belakang masalah, perumusan masalah, tujuan penulisan, batasan masalah, dan sistematika penulisan.

Bab II Landasan Teori

Bagian pertama menjelaskan teori dasar mikrokontroler tipe AT89S51. Bagian kedua menjelaskan konverter RS232. Bagian ketiga menjelaskan tentang GPS. Bagian keempat menjelaskan satelit ACeS Garuda 1, dan bagian kelima menjelaskan *File Transfer Protocol* (FTP) .

Bab III Rancang Bangun

Bagian pertama menjelaskan prinsip kerja sistem secara keseluruhan. Bagian kedua menjelaskan perancangan perangkat keras. Bagian ketiga menjelaskan perancangan perangkat lunak. Bagian keempat menjelaskan format data GPS dan format pengiriman data ke satelit.

Bab IV Pengujian dan Analisis Sistem

Berisikan pengujian sistem dan analisis pada setiap pengujian.

BabV Kesimpulan

Berisikan kesimpulan yang didapat dari hasil pengujian dan analisis.

BAB II

LANDASAN TEORI

2.1 MIKROKONTROLER AT89S51

Atmel sebagai salah satu *vendor* yang mengembangkan dan memasarkan produk mikroelektronika telah menjadi suatu standar bagi para perancang sistem elektronika saat ini. Mikrokontroler AT89S51 merupakan salah satu dari banyak produk yang dikeluarkan oleh Atmel. Mikrokontroler ini memiliki instruksi yang dikemas dalam kode 16 bit (16 bit *word*). AT89S51 merupakan satu dari keluarga mikrokontroler dengan kode seri MCS51 yang mempunyai teknologi *Complex Instruction Set Computing* (CISC). Mikrokontroler MCS secara umum dapat dikelompokkan menjadi dua, yaitu keluarga AT89Cxx dan AT89Sxx. Pada dasarnya yang membedakan masing-masing kelas tersebut adalah memori, perifer, dan fungsinya. Dari segi arsitektur dan instruksi yang digunakan, bisa dikatakan sama.

2.1.1 Arsitektur AT89S51

AT89S51 mempunyai arsitektur seperti yang diperlihatkan pada gambar 2.1 (pada halaman 6). Dari gambar 2.1 tersebut dapat dilihat bahwa AT89S51 memiliki bagian sebagaimana berikut.

1. 32 buah saluran I/O (*Input/Output* ‘Masukan/Keluaran’)

Jumlah saluran I/O pada mikrokontroler AT89S51 adalah sebanyak 32 saluran yang dibagi menjadi 4 keluaran *Port*, yaitu *Port 0*, *Port 1*, *Port 2*, *Port 3*.

2. 8 bit *Central Processing Unit* (CPU)

Mikrokontroler AT89S51 terdiri dari 8 bit CPU yang berisi register A (akumulator) dan register B.

3. *Program Counter* (PC) dan *Data Pointer* (DPTR).

PC dan DPTR memiliki 16 bit data. DPTR terbagi dua, yaitu 8 bit *Data Pointer High* (DPH) dan 8 bit *Data Pointer Low* (DPL).

4. *Stack Pointer* (SP).

Register SP pada mikrokontroler AT89S51 adalah sebesar 8 bit. Register ini akan bertambah tiap kali data disimpan pada saat program *PUSH* dan pada saat pemanggilan subrutin.

5. *In System Programmable (ISP) flash memory.*

Besarnya memori ISP pada tiap tipe mikrokontroler berbeda-beda. Pada tipe AT89S51 besarnya adalah 4K Byte.

6. RAM Internal.

Mikrokontroler AT89S51 memiliki RAM internal sebesar 128 byte.

7. *Timer/Counter.*

Mikrokontroler AT89S51 memiliki 2 x 16 bit *timer/counter*

8. *Port Serial*

Port serial mikrokontroler bekerja pada mode *full duplex*.

9. *Control Register*

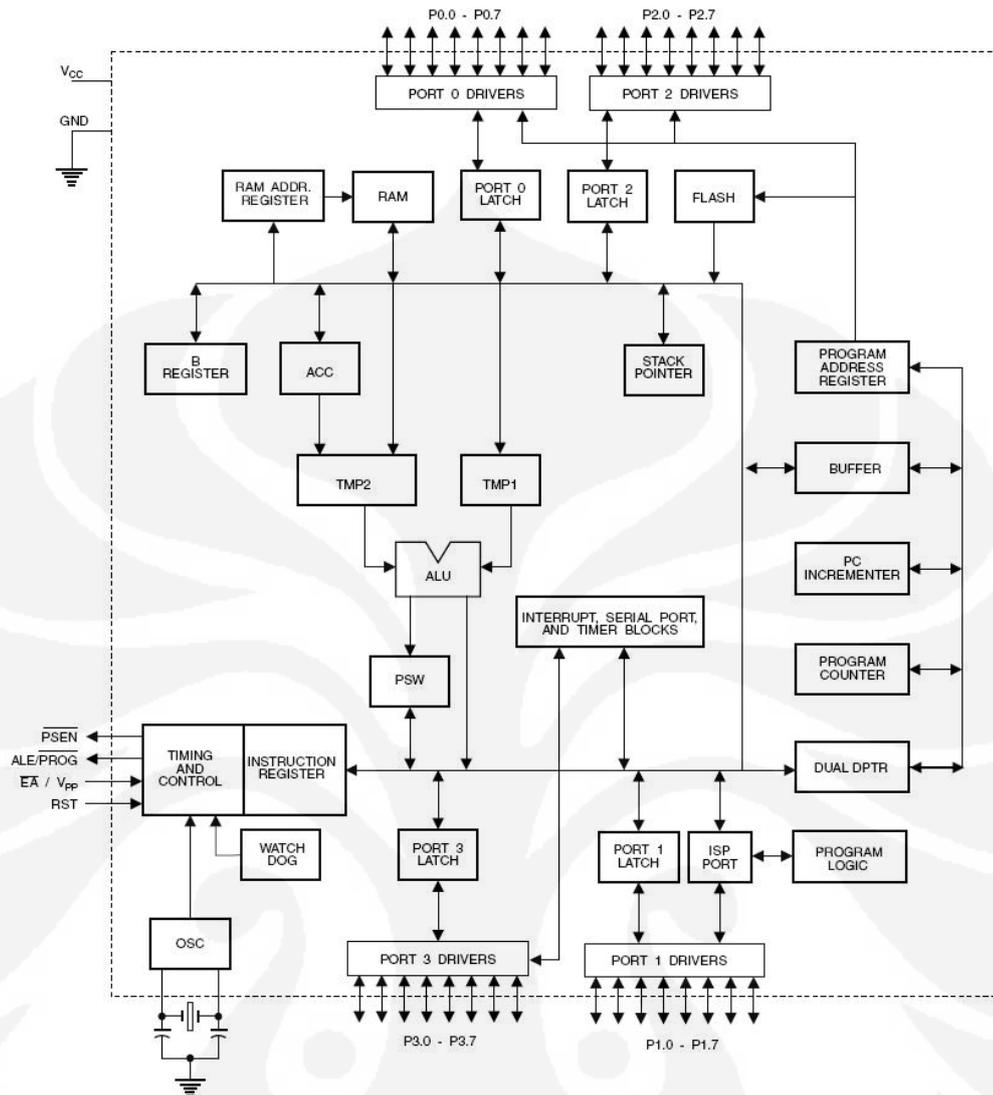
Mikrokontroler dalam bekerja didukung oleh register-register yang berfungsi untuk mengontrol masukan atau keluaran yang diberikan. Register-register ini adalah: TCON, TMOD, SCON, PCON, IP, dan IE.

10. Rangkaian osilator dan *clock*.

Rangkaian osilator dan *clock* dihasilkan oleh rangkaian osilasi kristal.

11. Lima unit interupsi

Interupsi pada mikrokontroler terdiri dari: dua interupsi eksternal, dua interupsi *timer*, dan interupsi *port serial*.



Gambar 2.1. Blok diagram fungsional AT89S51 [1]

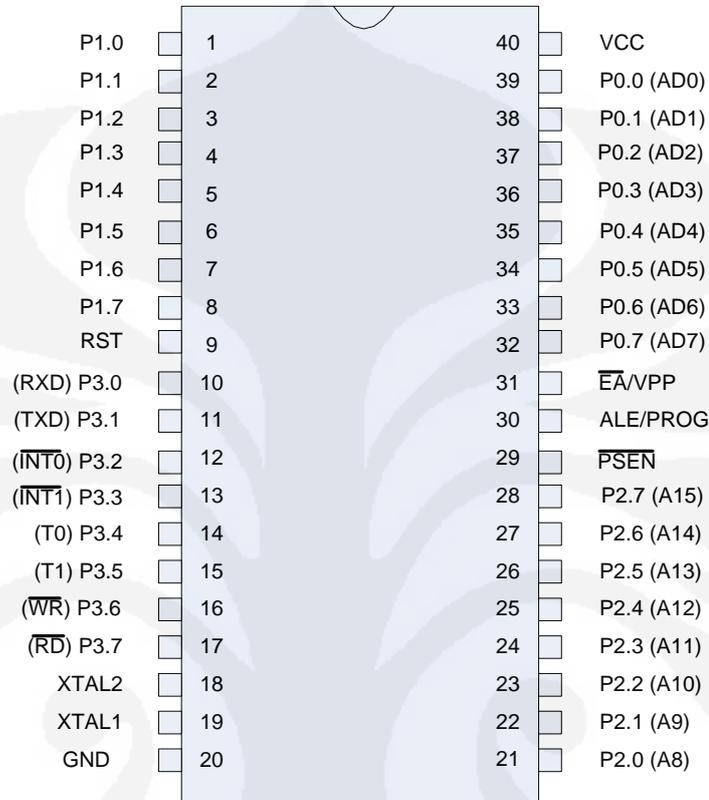
2.1.2 Fitur AT89S51

Mikrokontroler AT89S51 memiliki fitur-fitur sebagai berikut.

1. Kapasitas memori *flash* sebesar 4K byte dengan kemampuan tulis dan hapus 1000 kali.
2. Beroperasi pada jangkauan 4 V sampai 5,5 V dan 0 Hz sampai 33 MHz.
3. RAM internal sebesar 128 x 8 bit.
4. *Port* komunikasi serial *Universal Asynchronous Receiver/Transmitter* (UART) dengan kemampuan *full duplex*.

2.1.3 Konfigurasi Pin AT89S51

Konfigurasi pin AT89S51 dapat dilihat pada gambar 2.2. Dari gambar tersebut dapat dijelaskan secara fungsional konfigurasi pin AT89S51.



Gambar 2.2. Pin AT89S51 [2]

1. VCC
Merupakan pin yang berfungsi sebagai pin masukan catu daya.
2. GND
Merupakan pin *ground*.
3. Port 0 (P0.0...P0.7)
Merupakan pin I/O dua arah juga sebagai *multiplexer* yang memiliki alamat *low-order* atau *data bus* pada pengalamanatan memori eksternal.
4. Port 1 (P1.0...P1.7)
Merupakan pin I/O dua arah.
5. Port 2 (P2.0...P2.7)
Merupakan pin I/O dua arah juga memiliki alamat *high-order* pada pengalamanatan memori eksternal.

6. *Port 3 (P3.0...P3.7)*

Merupakan pin I/O dua arah dan memiliki fungsi-fungsi khusus pada tiap pinnya seperti yang diperlihatkan pada tabel 2.1.

Tabel 2.1 Fungsi-Fungsi Pin pada Port 3 AT89S51

Pin Port 3	Fungsi
P3.0	RXD (masukan <i>port</i> serial)
P3.1	TXD (keluaran <i>port</i> serial)
P3.2	$\overline{INT0}$ (Interupsi eksternal 0)
P3.3	$\overline{INT1}$ (Interupsi eksternal 1)
P3.4	T0 (masukan <i>timer</i> eksternal 0)
P3.5	T1 (masukan <i>timer</i> eksternal 1)
P3.6	\overline{WR} (Penulisan <i>strobe</i> pada memori data)
P3.7	\overline{RD} (Pembacaan <i>strobe</i> pada memori data)

7. RST

Merupakan pin yang digunakan untuk *mereset* mikrokontroler.

8. ALE / \overline{PROG} .

Sebagai *Address Latch Enable* (ALE) pin akan mengeluarkan pulsa untuk *melatching* alamat byte rendah pada saat mengakses eksternal memori.

Sebagai \overline{PROG} pin akan menerima masukan pulsa pada saat pemrograman *flash*.

9. *Program Store Enable* (\overline{PSEN})

Merupakan pin yang membaca *strobe* pada memori eksternal.

10. *External Access Enable* (\overline{EA} / VPP)

Pin \overline{EA} harus dihubungkan ke *ground* (GND) agar dapat mengambil program dari memori eksternal dan menerima tegangan sebesar 12 V dari *programming enable voltage* (V_{PP}) saat pemrograman *flash*.

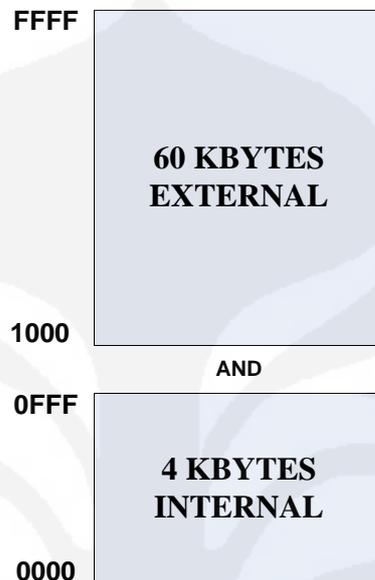
11. XTAL1 dan XTAL2

Merupakan pin masukan *clock* eksternal. Kaki XTAL1 dan XTAL2 terhubung pada kristal.

2.1.4 Peta Memori

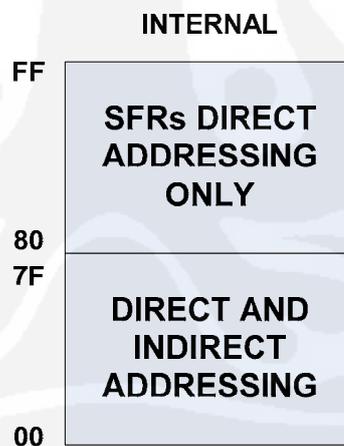
AT89S51 memiliki ruang pengalamatan memori data dan memori program yang terpisah. Gambar 2.3 memperlihatkan peta dari memori pada AT89S51

yang terdiri dari 4K byte internal (0000H sampai 0FFFH) memori dan 60K byte eksternal memori (1000H sampai FFFFH). Total 64K byte data memori eksternal dapat dialamatkan langsung AT89S51 ke dalam *chip*.



Gambar 2.3. Peta memori AT89S51 [3]

AT89S51 memiliki 128 byte internal RAM ditambah alamat dari *Special Function Register* (SFR) seperti yang diperlihatkan gambar 2.4.



Gambar 2.4. Internal memori AT89S51 [4]

Gambar 2.4 menunjukkan SFR dan pengalamatan langsung (*direct*) RAM memiliki alamat yang sama, yaitu 80H sampai 0FFH. Akan tetapi, keduanya berada pada daerah yang berlainan yang diakses dengan cara yang berbeda pula.

Fungsi dan alamat SFR dapat dilihat pada tabel 2.2 berikut ini.

Tabel 2.2. Alamat SFR

Simbol	Nama	Alamat
ACC	<i>Accumulator</i>	0E0H
B	<i>Register B</i>	0F0H
PSW	<i>Program Status Word</i>	0D0H
SP	<i>Stack Pointer</i>	81H
DPTR	<i>Data Pointer 2 Bytes</i>	
DPL	<i>Low Byte</i>	82H
DPH	<i>High Byte</i>	83H
P0	<i>Port 0</i>	80H
P1	<i>Port 1</i>	90H
P2	<i>Port 2</i>	0A0H
P3	<i>Port 3</i>	0B0H
IP	<i>Interrupt Priority Control</i>	0B8H
IE	<i>Interrupt Enable Control</i>	0A0H
TMOD	<i>Timer/Counter Mode Control</i>	89H
TCON	<i>Timer/Counter Control</i>	88H
TH0	<i>Timer/Counter 0 High Byte</i>	8CH
TL0	<i>Timer/Counter 0 Low Byte</i>	8AH
TH1	<i>Timer/Counter 1 High Byte</i>	8DH
TL1	<i>Timer/Counter 1 Low Byte</i>	8BH
SCON	<i>Serial Control</i>	98H
SBUF	<i>Serial Data Buffer</i>	99H
PCON	<i>Power Control</i>	87H

Sementara itu, 128 byte RAM pada AT89S51 dapat diakses secara langsung (*direct*) atau tidak langsung (*indirect*). Peta memori RAM dibagi menjadi tiga area seperti disebutkan di bawah ini.

1. Register *Bank 0 – 3*.

Memiliki alamat dari 00H sampai 1FH (32 byte). Tiap register *bank* terdiri dari 8 register (R0 sampai R7). Untuk memilih register *bank* yang dipakai, bit RS0 dan RS1 pada *Program Status Word* (PSW) dapat diatur. Bentuk dan isi PSW dapat dideskripsikan seperti dalam gambar 2.5.

MSB							LSB
CY	AC	F0	RS1	RS0	OV	-	P

Gambar 2.5. Susunan bit register PSW

Fungsi-fungsi bit pada PSW dapat dipaparkan sebagaimana dalam tabel 2.3 berikut ini.

Tabel 2.3. Fungsi – Fungsi Bit pada PSW

Simbol	Alamat Bit	Fungsi
CY	PSW.7	<i>Carry flag</i>
AC	PSW.6	<i>Auxillary carry flag</i>
F0	PSW.5	<i>User flag 0</i>
RS1	PSW.4	Register bank selector bit 1
RS0	PSW.3	Register bank selector bit 0
OV	PSW.2	<i>Overflow flag</i>
-	PSW.1	Untuk perancangan tambahan
P	PSW.0	<i>Parity flag</i>

Sementara itu, konfigurasi bit ketiga dan keempat, yaitu RS0 dan RS1, dapat dilihat pada tabel 2.4.

Tabel 2.4. Konfigurasi Bit RS0 dan RS1

RS1	RS0	Register Bank	Alamat
0	0	0	00H-07H
0	1	1	08H-0FH
1	0	2	10H-17H
1	1	3	18H-1FH

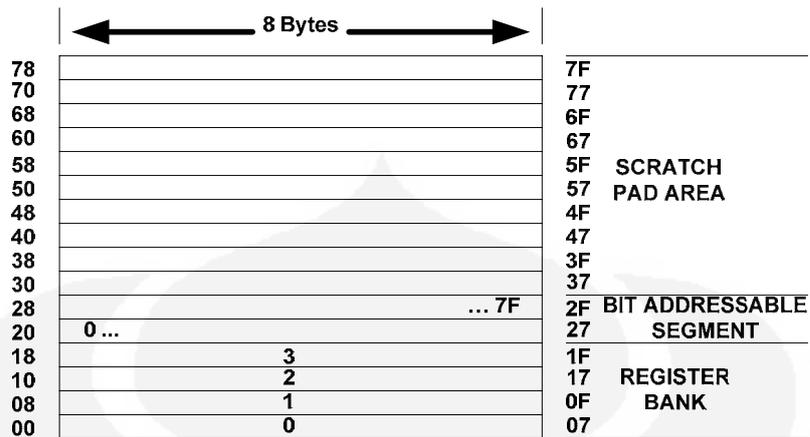
2. *Bit Addressable Area.*

Terdapat 16 byte alamat dari 20H sampai 2FH yang membentuk total 128 alamat bit.

3. *Scrath Pad Area.*

Byte dari alamat 30H sampai 7FH digunakan sebagai data RAM.

Peta memori RAM yang dibagi menjadi tiga area seperti disebutkan di atas dapat divisualisasikan dalam gambar 2.6 pada halaman berikut ini.



Gambar 2.6. Peta memori RAM AT89S51 [5]

2.1.5 Counter dan Timer

AT89S51 dilengkapi dengan dua perangkat *Timer/Counter* yang masing-masing dinamakan *Timer 0* dan *Timer 1*. Pencacah biner untuk *Timer 0* dibentuk dengan register *Timer 0 Low byte* (TL0) dan register *Timer 0 High byte* (TH0). Pencacah biner untuk *Timer 1* dibentuk dengan register *Timer 1 Low byte* (TL1) dan register *Timer 1 High byte* (TH1). Untuk mengatur kerja *Timer/Counter* dipakai dua register tambahan yang dipakai bersama oleh *Timer 0* dan *Timer 1*. Register tersebut adalah *Timer Mode Control* (TMOD) dan *Timer Control* (TCON). Bentuk dan isi dari register TMOD dapat dilihat pada gambar 2.7.

MSB							LSB
GATE	C/\bar{T}	M1	M0	GATE	C/\bar{T}	M1	M0
<i>TIMER 1</i>				<i>TIMER 0</i>			

Gambar 2.7. Susunan bit register TMOD

Sementara itu, fungsi bit-bit TMOD dapat dilihat pada tabel 2.5.

Tabel 2.5. Fungsi-Fungsi Bit TMOD

Simbol	Alamat Bit	Fungsi
GATE	TMOD.7 dan TMOD.3	Mengatur saluran sinyal denyut. Bila <i>0</i> , saluran sinyal denyut hanya diatur bit TRx. Bila <i>1</i> , kaki INT0 atau INT1 dipakai juga untuk mengatur saluran sinyal denyut.

C/\bar{T}	TMOD.6 dan TMOD.2	Mengatur sumber sinyal denyut yang diumpankan ke pencacah biner. Bila 0 , sinyal denyut diperoleh dari osilator kristal yang frekuensinya sudah dibagi 12. Bila 1 , maka sinyal denyut diperoleh dari kaki T0 atau kaki T1.
M1	TMOD.5 dan TMOD.1	Mode bit 1
M0	TMOD.4 dan TMOD.0	Mode bit 0

Tabel 2.6 berikut ini menjelaskan konfigurasi mode *Timer/Counter* dari fungsi bit register TMOD.

Tabel 2.6. Konfigurasi Mode *Timer/Counter*

M1	M0	Mode
0	0	0
0	1	1
1	0	2
1	1	3

1. Mode 0

Pencacah biner dibentuk dengan TLx (TL0 atau TL1) sebagai pencacah biner 5 bit. Limpahan dari pencacah biner 5 bit dihubungkan ke THx (TH0 atau TH1) membentuk untaian pencacah biner 13 bit. Limpahan dari pencacah 13 bit ini ditampung di *flip-flop* TFx (TF0 atau TF1) pada register TCON. Gambar 2.8 memperlihatkan *Timer/Counter* yang bekerja pada mode 0.



Gambar 2.8. Mode 0 pencacah biner 13 bit

2. Mode 1

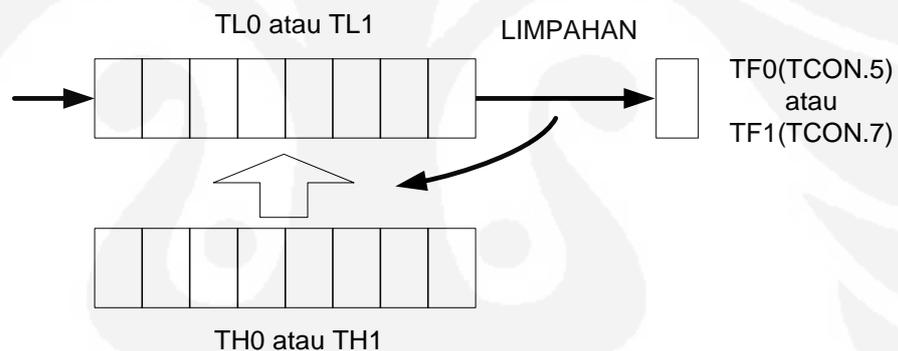
Sama halnya dengan mode 0, hanya saja register TLx dipakai sepenuhnya sebagai pencacah biner 8 bit sehingga kapasitas pencacah biner yang terbentuk adalah 16 bit. Gambar 2.9 memperlihatkan *Timer/Counter* yang bekerja pada mode 1.



Gambar 2.9. Mode 1 pencacah biner 16 bit

3. Mode 2

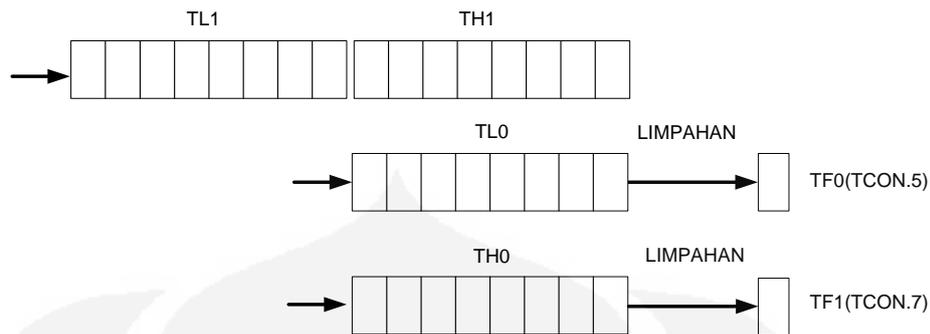
TLx dipakai sebagai pencacah biner 8 bit dan THx dipakai untuk menyimpan nilai yang diisikan ulang ke TLx setiap kali kedudukan TLx melimpah (berubah dari FFH ke 00H). Gambar 2.10 memperlihatkan *Timer/Counter* yang bekerja pada mode 2.



Gambar 2.10. Mode 2 pencacah biner 8 bit dengan isi ulang

4. Mode 3

Pada mode 3 TL0, TH0, TL1, dan TH1 dipakai untuk membentuk tiga untaian pencacah. *Pertama*, untaian pencacah biner 16 bit tanpa fasilitas pemantauan sinyal limpahan yang dibentuk dengan TL1 dan TH1. *Kedua*, TL0 dipakai sebagai pencacah biner 8 bit dengan TF0 sebagai sarana pemantau limpahan. *Ketiga*, TH0 yang dipakai sebagai pencacah biner 8 bit dengan TF1 sebagai sarana pemantau limpahan. Gambar 2.11 memperlihatkan *Timer/Counter* yang bekerja pada mode 3.



Gambar 2.11. Mode 3 gabungan pencacah biner 16 bit dan 8 bit

Mode 3 merupakan gabungan dari 2 pencacahan yaitu pencacahan 16 bit dan pencacahan 8 bit. TL1 dan TH1 membentuk susunan pencacahan 16 bit, sedangkan pencacah 8 bit disusun oleh TL0 dan TH0. Limpahan dari pencacahan 8 bit TL0 ditampung di TF0 yang terdapat pada alamat bit TCON.5 dan limpahan 8 bit TH0 ditampung di TF1 yang terdapat pada alamat bit TCON.7.

Bentuk dan isi dari register TCON dapat dilihat pada gambar 2.12.

MSB							LSB
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

Gambar 2.12. Susunan bit register TCON

Fungsi-fungsi bit TCON dapat diperinci sebagaimana terlihat pada tabel 2.7 berikut.

Tabel 2.7. Fungsi-Fungsi Bit TCON

Simbol	Posisi Bit	Fungsi
TF1	TCON.7	<i>Timer 1 overflow flag</i>
TR1	TCON.6	<i>Timer 1 run control bit</i>
TF0	TCON.5	<i>Timer 0 overflow flag</i>
TR0	TCON.4	<i>Timer 0 run control bit</i>
IE1	TCON.3	<i>Interrupt 1 edge flag</i>
IT1	TCON.2	<i>Interrupt 1 type control bit</i>
IE0	TCON.1	<i>Interrupt 0 edge flag</i>
IT0	TCON.0	<i>Interrupt 0 type control bit</i>

Sisa 4 bit dari register TCON (bit 4...bit 7) dibagi menjadi dua bagian secara simetris yang dipakai untuk mengatur *Timer 0/Timer 1* sebagai berikut.

- a. Bit TFX (TF0 atau TF1) merupakan bit penampung limpahan. TFX akan menjadi *1* setiap kali pencacah biner yang terhubung padanya melimpah.
- b. Bit TRx (TR0 atau TR1) merupakan bit pengatur saluran sinyal denyut. Bila bit *0* sinyal denyut tidak disalurkan ke pencacah biner, pencacah akan berhenti mencacah.

2.1.6 Masukan/Keluaran Data Serial

AT89S51 memiliki komunikasi data serial memiliki parameter yang terdiri dari register SBUF, Register *Serial Port Control* (SCON), dan register *Power Mode Control* (PCON). Register SBUF untuk menahan data dan merupakan bit dalam register SFR. Register *Serial Port Control* (SCON) seperti yang diperlihatkan Gambar 2.13 berfungsi untuk mengontrol komunikasi data.

MSB							LSB
SM0	SM1	SM2	REN	TB8	RB8	T1	R1

Gambar 2.13. Susunan bit register SCON

Tabel 2.8 pada halaman berikut ini memperlihatkan fungsi-fungsi pada bit SCON.

Tabel 2.8. Fungsi-Fungsi Bit SCON

Simbol	Posisi	Fungsi
SM0	SCON.7	<i>Port</i> serial mode 0
SM1	SCON.6	<i>Port</i> serial mode 1
SM2	SCON.5	Mengaktifkan komunikasi multiprosesor dalam mode 2 dan 3
REN	SCON.4	<i>Reception enable</i>
TB8	SCON.3	Mengirimkan bit ke-9 yang diaktifkan pada mode 2 dan 3
RB8	SCON.2	Menerima bit ke-9 yang diaktifkan pada mode 2 dan 3
T1	SCON.1	Mengirimkan <i>flag</i> interupsi
R1	SCON.0	Menerima <i>flag</i> interupsi

Sementara itu, tabel 2.9 di bawah ini berisi konfigurasi mode SM0 dan SM1.

Tabel 2.9. Konfigurasi Mode SM0 dan SM1

SM0	SM1	Mode	Deskripsi	Baud Rate
0	0	0	<i>Shift Register</i>	Tetap ($f_{osc}/12$)
0	1	1	8 bit UART	Bervariasi (Diset dengan
1	0	2	9 bit UART	Tetap ($f_{osc}/64$ atau $f_{osc}/32$)
1	1	3	9 bit UART	Bervariasi (Diset dengan

Register *Power Mode Control* (PCON) seperti yang terlihat pada gambar 2.14 berfungsi untuk mengontrol *data rate*, dan pin RXD dan TXD yang terhubung pada rangkaian data serial.

MSB							LSB
SMOD	-	-	-	GF1	GF0	PD	IDL

Gambar 2.14. Susunan bit register PCON

Tabel 2.10 di bawah ini memperlihatkan fungsi-fungsi dari bit PCON.

Tabel 2.10. Fungsi-Fungsi Bit PCON

Simbol	Posisi Bit	Fungsi
SMOD	PCON.7	Modifikasi <i>baud rate</i> . Diset 1 untuk <i>baud rate</i> 2 kali lipat menggunakan <i>Timer 1</i> untuk mode 1, 2, dan 3. Diset 0 untuk mendapatkan nilai <i>baud rate</i> dari <i>Timer 1</i>
-	PCON.6	-
-	PCON.5	-
-	PCON.4	-
GF1	PCON.3	<i>Flag</i> untuk fungsi umum
GF0	PCON.2	<i>Flag</i> untuk fungsi umum
PD	PCON.1	Bit untuk fungsi operasi <i>power down</i>
IDL	PCON.0	Bit untuk fungsi operasi <i>idle mode</i>

2.1.7 Komunikasi Serial AT89S51

Untuk membentuk komunikasi serial AT89S51, ada beberapa ketentuan yang harus diperhatikan.

2.1.7.1 Sistem Transmisi Serial

Dalam komunikasi serial AT89S51 terdapat dua jenis transmisi serial.

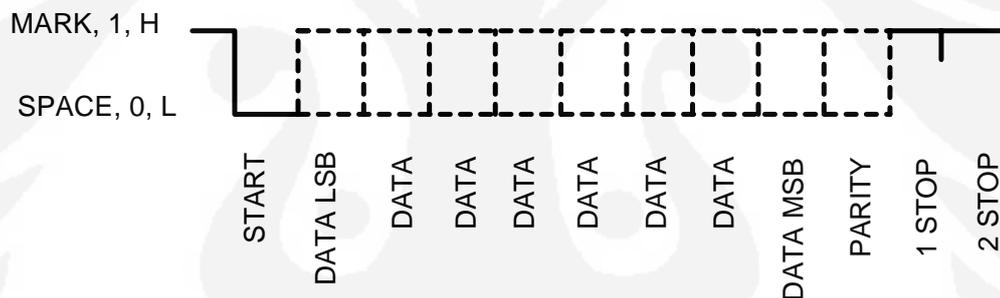
1. *Synchronous*, yaitu detak (*clock*) yang dikirim bersama dengan data serial itu sendiri.

2. *Asynchronous*, yaitu detak dibangkitkan oleh masing-masing sistem, baik pengirim maupun penerima.

2.1.7.2 Protokol Serial

Proses komunikasi serial membutuhkan protokol yang disebut dengan protokol serial. Protokol serial terdapat pada komunikasi *asynchronous*. Format yang dipakai dalam protokol serial adalah 12 bit seperti yang diperlihatkan pada gambar 2.15.

1. *Start bit* berfungsi untuk menginisialisasikan rangkaian pewaktu. Hal ini terdeteksi dengan perubahan bit dari *high* ke *low*.
2. 8 bit data diawali *Least Significant Bit (LSB)* dan diakhiri *Most Significant Bit (MSB)*.
3. *Optional bit parity*.
4. *Stop bit*. Pada *stop bit*, *line* menjadi *high* untuk 1 atau 2 bit yang menandakan karakter telah habis.



Gambar 2.15. Format waktu untuk *asynchronous* UART [6]

2.1.7.3 Inisialisai UART

Dalam proses inisialisasi ada beberapa buah register yang perlu ditentukan nilainya, yaitu TMOD, SCON, dan PCON.

TMOD merupakan register 8 bit yang berfungsi untuk mengatur kerja *Timer/Counter*. Dengan memanfaatkan bit TMOD.5 dan TMOD.1 (*Timer 1*) atau dengan TMOD.4 dan TMOD.0 (*Timer 0*), kita dapat memilih mode operasi pencacah biner yang diinginkan.

Dengan bantuan register SCON, kita dapat menentukan besarnya laju baud (*baud rate*) yang diinginkan dengan memanfaatkan bit SCON.7 dan SCON.6

untuk memilih mode jenis *baud rate*. Perhitungan *baud rate* dari tiap mode adalah[7]:

$$\text{Mode 0 baud rate} = \frac{\text{frekuensi osilator}}{12} \dots\dots\dots (2.1)$$

$$\text{Mode 2 baud rate} = \frac{2^{\text{SMOD}}}{64} \times \text{frekuensi osilator} \dots\dots\dots (2.2)$$

$$\text{Mode 1 dan 3 baud rate} = \frac{2^{\text{SMOD}}}{32} \times \frac{\text{frekuensi osilator}}{12 \times [256 - (\text{TH1})]} \dots\dots\dots (2.3)$$

Nilai SMOD didapat dari pengaturan bit PCON.7 pada register PCON.

2.1.7.4 Pengiriman Data

Proses pengiriman data serial dilakukan per byte data. Proses tersebut menggunakan bit yang ada pada register SCON, yaitu bit TI. Bit TI merupakan petanda yang setara dengan petanda *Transmitter Data Register Empty* (TDRE) yang umum dijumpai pada UART standar. Data yang dikirim disimpan terlebih dahulu pada register SBUF. Berikut adalah subrutin pengiriman data serial.

01:SerialOut:

```
02:JNB     TI,$
03:MOV    SBUF,A
04:CLR    TI
05:RET
```

Baris pertama adalah subrutin dengan nama *SerialOut*. Baris kedua menunggu data sebelumnya selesai dikirim. Baris ketiga mangirimkan data melalui *port* serial dengan cara meletakkan data yang telah disimpan di akumulator A ke register SBUF. Setelah itu, nilai TI dinolkan kembali.

2.1.7.5 Penerimaan Data

Proses penerimaan data serial dilakukan dengan mengecek bit RI pada register SCON. Bit RI merupakan petanda yang setara dengan petanda *Receiver Data Register Full* (RDRF). Setelah register SBUF menerima data dari *port* serial, bit RI akan bernilai *1* dengan sendirinya kemudian harus dinolkan dengan program agar bisa dipakai untuk memantau keadaan SBUF dalam penerimaan data berikutnya. Berikut adalah subrutin penerimaan data serial.

01:SerialIn:

```
02:JNB     RI,$
```

03:MOV A,SBUF

04:CLR RI

05:RET

Baris pertama adalah subrutin dengan nama *SerialIn*. Baris kedua menunggu register RI bernilai *1* untuk memastikan sudah ada data baru yang diterima pada SBUF. Baris ketiga, SBUF ditempatkan pada akumulator A supaya RI dapat dipakai untuk memantau keadaan SBUF pada pengiriman data berikutnya. Selanjutnya, pada baris empat RI dinolkan.

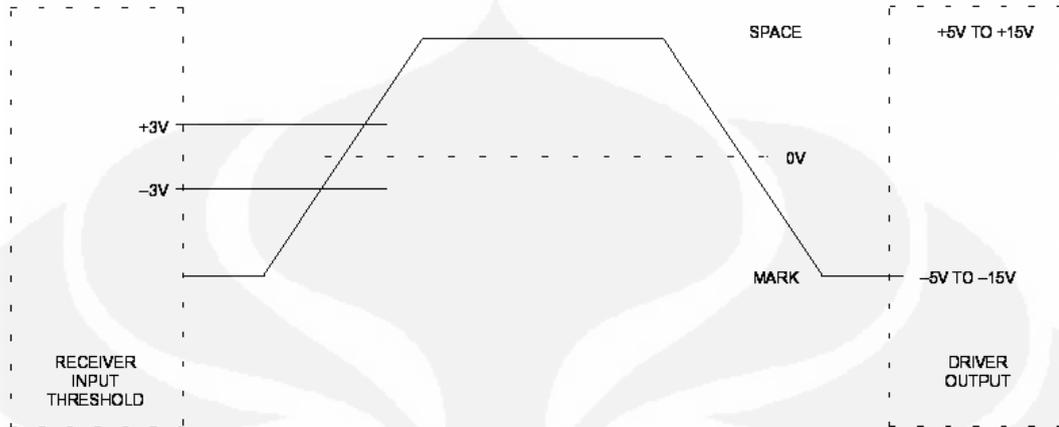
2.2 KONVERTER RS232

Konverter RS232 merupakan standar *Electronic Industries Association* (EIA) untuk komunikasi data binari serial. Sistem konverter RS232 pada umumnya digolongkan menjadi dua macam perangkat, yaitu *Data Communication Equipment* (DCE) dan *Data Terminal Equipment* (DTE). DCE berfungsi sebagai perangkat yang menyediakan kanal komunikasi antara dua jenis DTE seperti modem, printer, mouse, dan *plotter*. Sementara itu, DTE berfungsi sebagai sumber komunikasi, seperti komputer dan terminal. DTE terdiri dari *plug* (*male*) dan *socket* (*female*). Versi yang paling banyak digunakan adalah konverter RS232C (kadang dikenal dengan EIA232) dan yang terbaru adalah versi konverter RS232E.

Karakteristik transmisi data serial konverter versi RS232C pada level logika 1 (*Mark*) dinyatakan dengan tegangan antara -3 sampai 15 Volt dengan beban (-25 Volt tanpa beban), dan level logika 0 (*Space*) dinyatakan dengan tegangan antara +3 sampai +15 Volt dengan beban (+25 Volt tanpa beban). Untuk daerah +3 sampai -3 Volt tidak didefinisikan. Sedangkan paralel mentransmisikan level logika 0 dalam tegangan 0 sampai 0,8 Volt dan level logika 1 dalam tegangan 3,8 sampai 5 Volt.

Kecepatan komunikasi konverter RS232 dinyatakan dalam baud. Sesuai dengan standar yang berlaku, kecepatannya mencapai 20kbps dalam jarak kurang dari 15 meter. Beban impedansi pada *driver* harus di antara 3000 dan 7000 ohm serta tidak melebihi 2500pF.

Gambar 2.16 mengilustrasikan spesifikasi level logik konverter RS232C sebagaimana diuraikan dalam paragraf-paragraf sebelumnya.

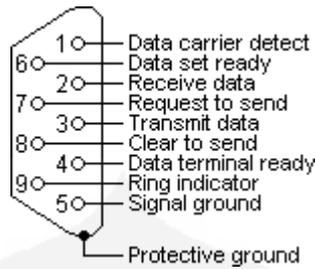


Gambar 2.16. Spesifikasi level logik konverter RS232C [8]

Pada komunikasi serial konverter RS232C, panjang maksimal kabel antara *transmitter* dan *receiver* tidak boleh melebihi 100 kaki (sekitar 30,48 meter). Spesifikasi konverter RS232C dapat dilihat pada tabel 2.11 dan perangkat konverter RS232C tipe konektor DB9 dapat dilihat pada gambar 2.17.

Tabel 2.11. Spesifikasi RS-232C

Spesifikasi	Keterangan
Mode operasi	<i>Single ended</i>
Jumlah Tx dan Rx yang diijinkan	1 Tx, 1 Rx (<i>point to point</i>)
Panjang kabel maksimum	Kurang dari 100 kaki
<i>Data rate</i> maksimum	20kbps
<i>Driver output range</i> minimum	± 5 Volt sampai ± 15 Volt
<i>Driver output range</i> maksimum	± 25 Volt
Impedansi pada Tx	3k Ω sampai 7k Ω
Sensitivitas masukan Rx	± 3 Volt
<i>Range</i> tegangan masukan Rx	± 15 Volt
Tahanan <i>input</i> Rx maksimum	3k Ω sampai 7k Ω
<i>Receiver threshold</i>	± 3 Volt



Gambar 2.17. Konektor DB9

Tabel 2.12 memperlihatkan posisi dan deskripsi pin dari konverter RS232C jenis DB9.

Tabel 2.12. Konverter RS232C pada DB9

Nomor Pin	Sinyal	Deskripsi
1	DCD	<i>Data Carrier Detect</i>
2	RD	<i>Receive Data</i>
3	TD	<i>Transmit Data</i>
4	DTR	<i>Data Terminal Ready</i>
5	GND	<i>Signal Ground</i>
6	DSR	<i>Data Set Ready</i>
7	RTS	<i>Ready To Send</i>
8	CTS	<i>Clear To Send</i>
9	RI	<i>Ring Indicator</i>

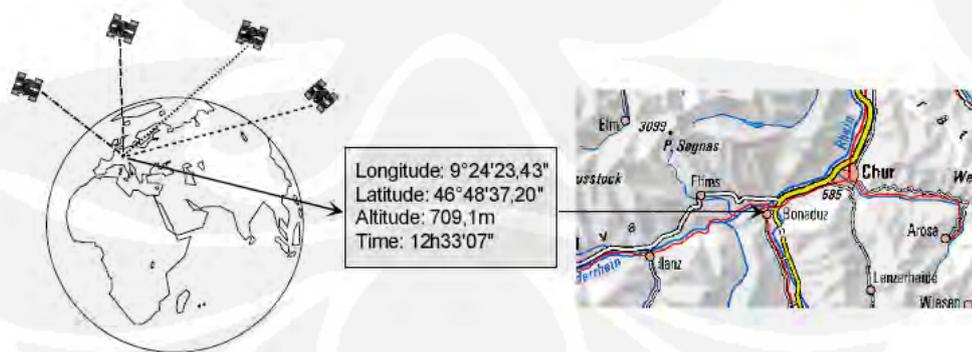
Sinyal	Fungsi
DCD	Saat modem mendeteksi sinyal <i>carrier</i> dari modem ujung yang lain pada jalur telepon, maka jalur ini akan aktif.
RD	Sinyal ini dihasilkan DCE dan diterima oleh DTE
TD	Sinyal ini dihasilkan DTE dan diterima oleh DCE
DTR	Mengindikasikan kesiapan dari DTE. Sinyal ini <i>ON</i> oleh DTE saat siap untuk mengirim atau menerima data.
DSR	Mengindikasikan kesiapan dari DCE. Sinyal ini <i>ON</i> oleh DCE saat siap untuk mengirim atau menerima data.
RTS	Saat DTE siap untuk mengirim data ke DCE, RTS akan <i>ON</i> . Pada sistem <i>simplex</i> dan <i>duplex</i> , kondisi ini menunjukkan DCE pada mode <i>receive</i> . Pada sistem <i>half-duplex</i> , kondisi ini menunjukkan DCE pada mode pengiriman. Setelah RTS diaktifkan, DCE harus mengaktifkan CTS sebelum memulai komunikasi.

CTS Digunakan bersama dengan RTS untuk mengadakan *handshaking* antara DTE dan DCE. Setelah DCE menerima perintah RTS, hal ini menyebabkan CTS *ON* saat siap untuk memulai komunikasi.

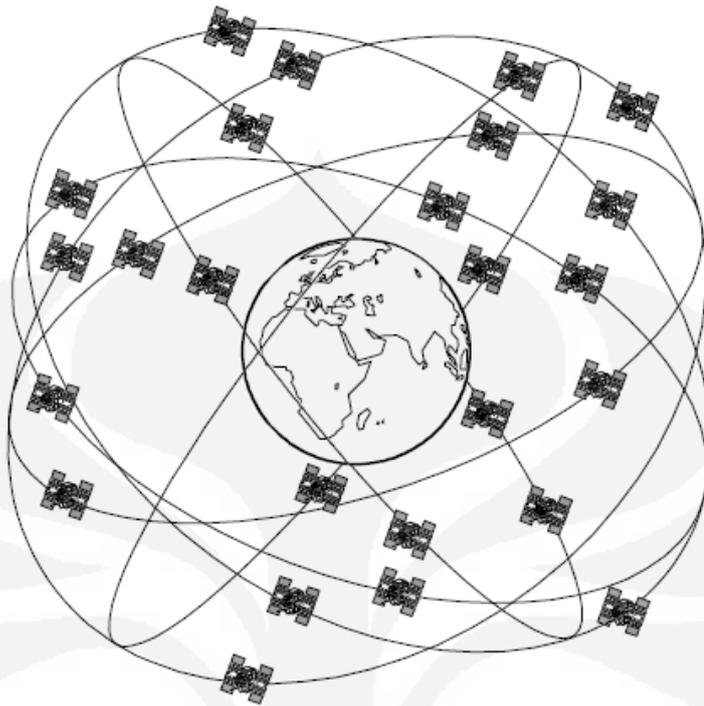
2.3 GLOBAL POSITIONING SYSTEM (GPS)

GPS adalah sistem navigasi satelit dan penentuan posisi menggunakan satelit. Istilah GPS juga bisa disebut dengan *NAVigation System with Timing And Ranging Global Positioning System* (NAVSTAR-GPS). Dikembangkan oleh Departemen Pertahanan Amerika Serikat (DoD) dan bisa diakses, baik oleh sipil maupun militer. Oleh karena itu, ada dua jenis akses, yaitu *The Civil Signal Standard Positioning Service* (SPS) dan *military signal Precise Positioning Service* (PPS). SPS dapat digunakan secara bebas oleh masyarakat tanpa dipungut biaya dan PPS hanya bisa digunakan oleh pihak militer Amerika Serikat dan pihak-pihak yang diijinkan. Sebagai jenis GPS yang bisa diakses oleh sipil, tingkat ketelitian SPS adalah ± 100 m pada saat kebijakan *Selective Availability* (SA) masih berlaku dan ± 20 m setelah kebijakannya dihapus (1 Mei 2000, 00:00 EDT).

Satelit pertama untuk fungsi dasar seperti gambar 2.18 diluncurkan ke orbit pada tanggal 22 Februari 1978 dan sekarang sebanyak 28 satelit beroperasi mengelilingi Bumi pada ketinggian 20.180 km pada 6 orbit seperti gambar 2.19.



Gambar 2.18. Fungsi dasar GPS [9]



Gambar 2.19. Orbit satelit GPS [10]

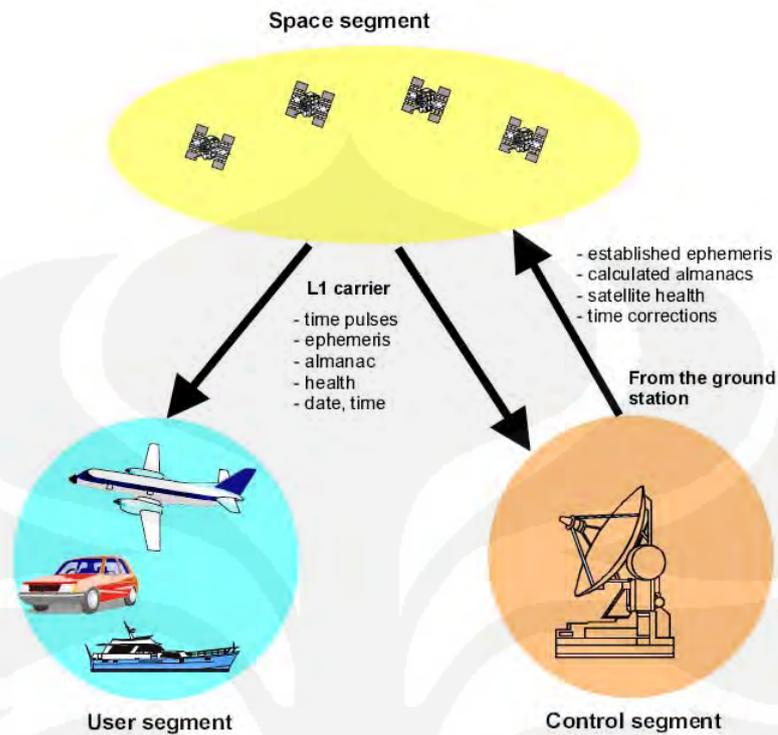
2.3.1 Segmen Penyusun GPS

Sebagai sistem navigasi, GPS bekerja dengan melibatkan beberapa segmen yang terkait satu dengan lainnya. Segmen-segmen yang mendukung kinerja GPS adalah sebagai berikut.

1. Segmen angkasa (*space segment*)
2. Segmen sistem kontrol (*control segment*)
3. Segmen pengguna (*user segment*)

Di antara segmen angkasa dan segmen sistem kontrol terjadi komunikasi dua arah. Sementara itu, antara segmen pengguna dan segmen angkasa hanya terjadi komunikasi satu arah, yaitu dari segmen angkasa ke segmen pengguna.

Seperti yang terlihat pada gambar 2.20, pemakai GPS hanya melakukan komunikasi satu arah, yaitu dari satelit-satelit GPS yang berada pada segmen angkasa.



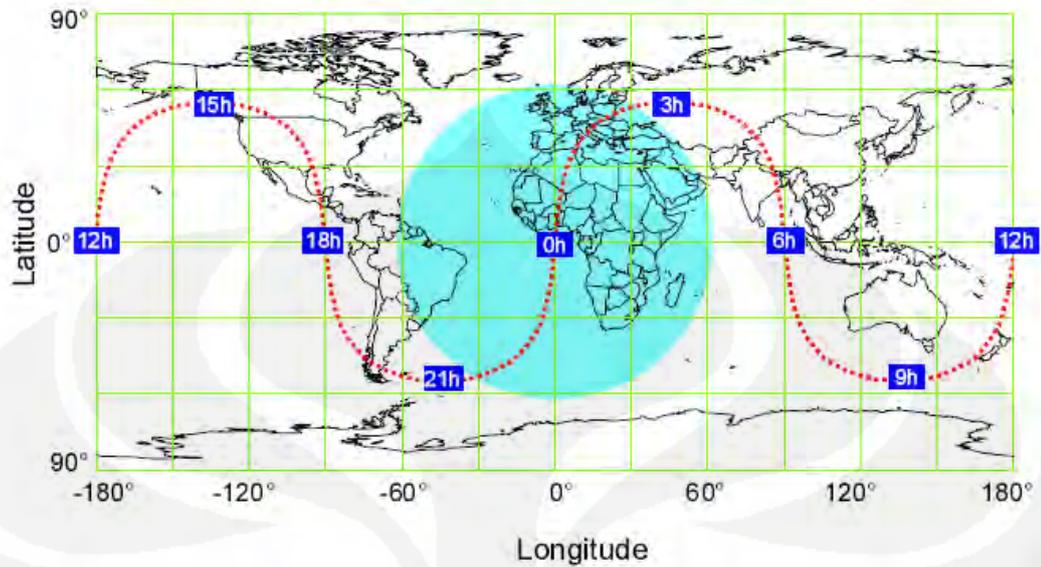
Gambar 2.20. Tiga segmen pada GPS [11]

2.3.1.1 Segmen Angkasa (Space Segment)

Segmen ini mencakup semua satelit yang beroperasi, yaitu sebanyak 28 satelit yang ditempatkan pada 6 bidang orbit (4 sampai 5 satelit per bidang orbit) pada ketinggian 20.180 km di atas permukaan Bumi. Masing-masing satelit memiliki jarak sama dan posisi antarsatelit tersebut adalah 55° condong ke ekuator. Tiap satelit akan memutari orbitnya tepat 12 jam. Berdasarkan rotasi Bumi, satelit akan berada pada titik awalnya setelah 24 jam (atau tepatnya 23 jam 56 menit).

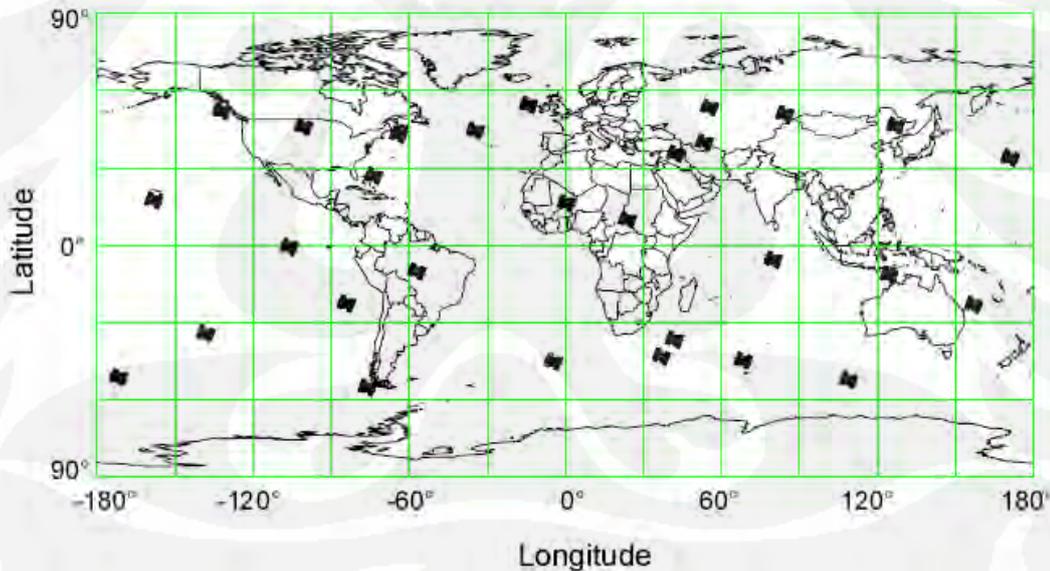
Gambar 2.21 berikut ini memperlihatkan posisi 28 satelit GPS pada rentang waktu 12 jam *Universal Time Coordinat* (UTC). Gambar 2.22 memperlihatkan distribusi ke-28 satelit GPS pada tanggal 14 April 2001.

Pada gambar 2.21, diperlihatkan arah pergerakan satelit dalam duarsi jam dari titik awal pergerakannya yang divisualisasikan dengan garis putus-putus berwarna merah. Satelit akan kembali ke posisi semula dalam rentang waktu 24 jam.



Gambar 2.21. Posisi pergerakan 28 satelit GPS pada tanggal 14 April 2001 [12]

Sementara itu, distribusi persebaran 28 satelit GPS pada tanggal 14 april 2001 divisualisasikan dengan gambar 2.22 berikut.



Gambar 2.22. Distribusi 28 satelit GPS pada tanggal 14 april 2001 [13]

2.3.1.2 Segmen Sistem Kontrol (Control Segment)

Sistem kontrol terdiri dari tiga macam dan terletak beberapa tempat. *Master Control Station* (MCS) terletak di Colorado. Lima *Monitor Station* (MS) yang dilengkapi dengan jam atom tersebar di penjuru bumi di sekitar ekuator.

Ground Control Station (GCS) yang akan mentransmisikan informasi ke satelit juga tersebar di penjuru bumi.

Tugas dari sistem kontrol adalah sebagai berikut.

1. Observasi pergerakan satelit dan perhitungan data orbit (*ephemeris*).
2. Memantau waktu satelit.
3. Sinkronisasi waktu satelit.
4. Mentransmisikan data orbit yang tepat dari satelit pada saat berkomunikasi.
5. Mentransmisikan perkiraan data orbit dari semua satelit (*almanac*).
6. Mentransmisikan informasi lainnya seperti keadaan satelit, kesalahan waktu dan lainnya.

2.3.1.3 Segmen Pengguna (*User Segment*)

Yang termasuk dalam segmen pengguna adalah semua pengguna GPS, baik sipil maupun militer. Dalam hal ini *receiver* GPS dibutuhkan untuk menerima dan memproses sinyal-sinyal dari GPS untuk digunakan dalam penentuan posisi, kecepatan, dan waktu.

2.3.2 Perhitungan Posisi GPS

Posisi GPS adalah bagian terpenting dari data GPS itu. Keakuratan posisi GPS dapat ditentukan dengan menentukan posisi absolut.

2.3.2.1 Prinsip Penentuan Posisi Absolut dengan GPS

Agar GPS dapat menerima sinyal untuk menentukan posisi pengguna, GPS harus menerima sinyal waktu dari empat satelit berbeda (Sat1...Sat4) sehingga bisa menghitung waktu transit $\Delta t_1 \dots \Delta t_4$ (gambar 2.23).



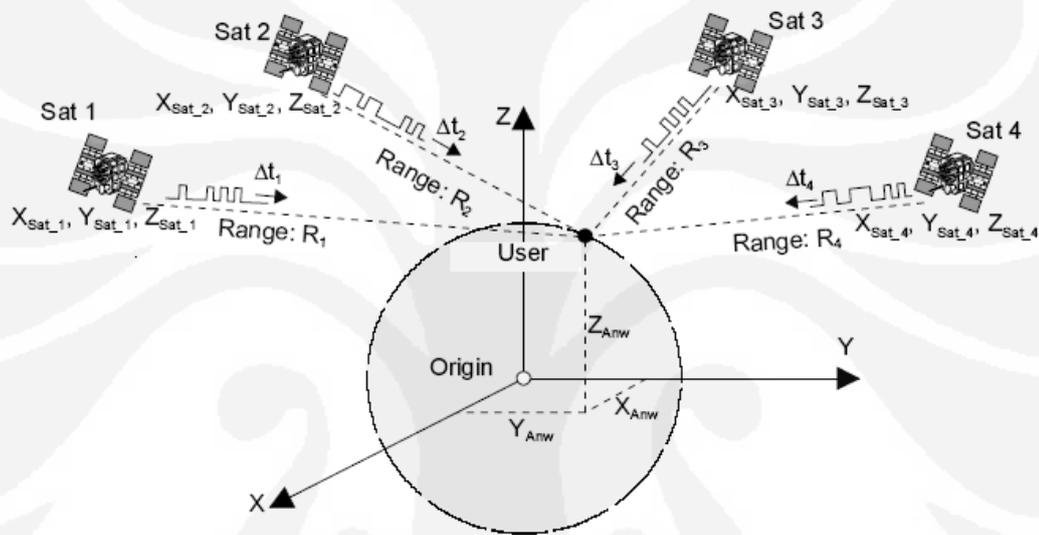
Gambar 2.23. Sinyal dari 4 satelit [14]

Perhitungan dinyatakan dalam bidang kartesian (X, Y, Z) dan koordinat 3 dimensi dengan sumbu *geocentric* (gambar 2.24). Jarak pengguna dengan 4 satelit $R_1, R_2, R_3,$ dan R_4 dapat ditentukan dengan bantuan sinyal transit waktu $\Delta t_1, \Delta t_2, \Delta t_3,$ dan Δt_4 antar-4 satelit dan pengguna. Setelah lokasi $X_{SAT}, Y_{SAT},$ dan Z_{SAT} dari ke-4 satelit diketahui, koordinat pengguna dapat diketahui [15].

$$\Delta t_{diukur} = \Delta t + \Delta t_0 \dots\dots\dots (2.4)$$

$$PSR = \Delta t_{diukur} \cdot c = (\Delta t + \Delta t_0) \dots\dots\dots (2.5)$$

$$PSR = R + \Delta t_0 \cdot c \dots\dots\dots (2.6)$$



Gambar 2.24. Sistem koordinat 3 dimensi [16]

Keterangan :

R = jarak sebenarnya satelit dengan pengguna

c = kecepatan cahaya = 3×10^8 m/s

Δt = waktu transit sinyal dari satelit ke pengguna

Δt_0 = perbedaan waktu satelit dengan waktu pengguna

$PSR = pseudo\ range$

Jarak R dari satelit ke pengguna dapat dihitung dengan sistem kartesian sebagai berikut [17].

$$R = \sqrt{(X_{SAT} - X_{USER})^2 + (Y_{SAT} - Y_{USER})^2 + (Z_{SAT} - Z_{USER})^2} \dots\dots\dots (2.7)$$

Persamaan (2.7) ke persamaan (2.6) menghasilkan persamaan (2.8).

$$PSR = \sqrt{(X_{SAT} - X_{USER})^2 + (Y_{SAT} - Y_{USER})^2 + (Z_{SAT} - Z_{USER})^2} + \Delta t_0 \cdot c \dots \quad (2.8)$$

Untuk menentukan empat variabel yang tidak diketahui (Δt_0 , X_{ANW} , Y_{ANW} , dan Z_{ANW}), diperlukan empat persamaan terpisah.

Persamaan berikut adalah sama untuk setiap satelit ($i = 1 \dots 4$) [18]

$$PSR_i = \sqrt{(X_{SAT_i} - X_{USER})^2 + (Y_{SAT_i} - Y_{USER})^2 + (Z_{SAT_i} - Z_{USER})^2} + \Delta t_0 \cdot c \quad (2.9)$$

2.3.2.2 Ketelitian Posisi Absolut

Ketelitian posisi GPS bergantung pada ketelitian data *pseudo range* dan pengukuran geometri satelit.

Ketelitian posisi GPS = Geometri satelit \times Ketelitian *pseudo range*

Ketelitian ini diwakili oleh nilai *Dilution of Precision* (DOP). Ada empat macam DOP.

1. *Geometrical DOP* (GDOP) adalah posisi satelit dalam ruang tiga dimensi, termasuk deviasi waktu.
2. *Positional DOP* (PDOP) adalah posisi dalam ruang tiga dimensi.
3. *Horizontal DOP* (HDOP) adalah posisi satelit pada bidang.
4. *Vertical DOP* (VDOP) adalah ketinggian satelit.

Berikut ini adalah faktor-faktor yang mempengaruhi ketelitian dalam penentuan posisi GPS.

1. Satelit, yaitu kesalahan orbit (*ephemeris*) dan waktu satelit.
2. Media propagasi, yaitu bias ionosfer dan bias troposfer yang mempengaruhi kecepatan (memperlambat) dan arah perlambatan sinyal GPS.
3. *Receiver* GPS (GPS pengguna), yaitu kesalahan waktu pada *receiver* (pengguna), yakni kesalahan yang terkait dengan antena dan *noise* (derau). Kesalahan ini tergantung pada kualitas *receiver* GPS dan berbanding lurus dengan harga dari *receiver* GPS, semakin tinggi harga *receiver* semakin tinggi pula kualitasnya.
4. Lingkungan sekitar *receiver* GPS, seperti *multipath*. *Multipath* yang merupakan fenomena sinyal GPS yang sampai ke antena *receiver* GPS adalah resultan sinyal langsung dari GPS dan satelit yang dipantulkan oleh benda-benda di sekeliling *receiver* GPS.

2.3.3 Format Data GPS

Untuk menampilkan variabel GPS, seperti waktu dan kecepatan ke perangkat, GPS menggunakan perangkat antarmuka konverter RS232C atau TTL. Akan tetapi, yang paling penting pada saat GPS menerima informasi adalah variabel tersebut ditampilkan dalam format data khusus. Format data GPS ini ditentukan oleh *National Marine Electronics Association* (NMEA). Saat ini data yang ditampilkan GPS sesuai dengan format NMEA-0183. NMEA memiliki format data untuk beberapa jenis aplikasi seperti *Global Navigation Satellite System* (GNSS), GPS, Loran, Omega, dan Transit. Pada GPS sendiri terdapat tujuh macam data yang dapat ditampilkan.

1. GGA adalah data tetap GPS.
2. GLL adalah posisi geografis, yaitu *latitude/longitude*.
3. GSA adalah GNSS DOP dan satelit yang aktif, yaitu penurunan akurasi dan jumlah satelit yang aktif pada GNSS DOP.
4. GSV adalah satelit GNSS dalam jangkauan.
5. RMC adalah spesifikasi data minimal GNSS yang direkomendasikan.
6. VTG adalah jalur dan kecepatan.
7. ZDA adalah waktu dan penanggalan.

Format data GPS berdasarkan standar NMEA-0183 adalah

\$GPDTS,Inf_1,Inf_2,Inf_3,Inf_4,Inf_5,Inf_6,Inf_n*CS<CR><LF> [19]

Keterangan simbol-simbol dalam format data GPS tersebut dapat dilihat dalam tabel 2.13 di bawah ini. Sementara itu keluaran data GPS berdasarkan format NMEA-0183 dapat dilihat pada gambar 2.25.

Tabel 2.13. Format Data GPS

Simbol	Keterangan
\$	Awal data
GP	Informasi peralatan navigasi
DTS	Jenis informasi yang terkandung
Inf_1 bis Inf_n	Data navigasi
,	Pemisahan antara informasi
*	Pemisahan <i>checksum</i>
CS	Pengecekan kesalahan pada kalimat

<CR><LF>	Akhir dari data: <i>carriage return</i> (<CR>), dan <i>line feed</i> (<LF>)
----------	---

```

$GPGLL,0621.9896,S,10649.2145,E,043224,A,A*5A
$GPBOD,,T,,M,,*47
$PGRME,3.0,M,8.2,M,8.7,M*28
$PGRMZ,314,f,3*1D
$GPRTE,1.1,c,*37
$GPRMC,043226,A,0621.9896,S,10649.2145,E,0.0,81.9,090608,0.6,E,A*35
$GPRMB,A,,,,,,,,,V,A*1C
$GPGGA,043226,0621.9896,S,10649.2145,E,1.12,1.1,95.7,M,7.3,M,,*51
$GPGSA,A,3,01,05,06,09,12,14,18,21,22,24,30,31,2.4,1.1,2.2*34
$GPGSV,3,1,12,01,11,300,00,05,42,077,35,06,03,261,00,09,05,026,00*77
$GPGSV,3,2,12,12,26,069,35,14,15,324,28,18,40,029,35,21,73,200,39*74
$GPGSV,3,3,12,22,20,350,00,24,36,153,29,30,54,109,41,31,42,243,29*74
$GPGLL,0621.9896,S,10649.2145,E,043226,A,A*58
$GPBOD,,T,,M,,*47
$PGRME,3.0,M,8.1,M,8.7,M*2B
$PGRMZ,314,f,3*1D
$GPRTE,1.1,c,*37
$GPRMC,043228,A,0621.9895,S,10649.2145,E,0.0,81.9,090608,0.6,E,A*38
$GPRMB,A,,,,,,,,,V,A*1C
$GPGGA,043228,0621.9895,S,10649.2145,E,1.12,1.1,95.7,M,7.3,M,,*5C
$GPGSA,A,3,01,05,06,09,12,14,18,21,22,24,30,31,2.4,1.1,2.2*34
$GPGSV,3,1,12,01,11,300,00,05,42,077,35,06,03,261,00,09,05,026,00*77
$GPGSV,3,2,12,12,26,069,36,14,15,324,28,18,40,029,35,21,73,200,39*77
$GPGSV,3,3,12,22,21,350,00,24,36,-

```

Gambar 2.25. Data keluaran GPS ditampilkan pada komputer dengan menggunakan *hyperterminal*

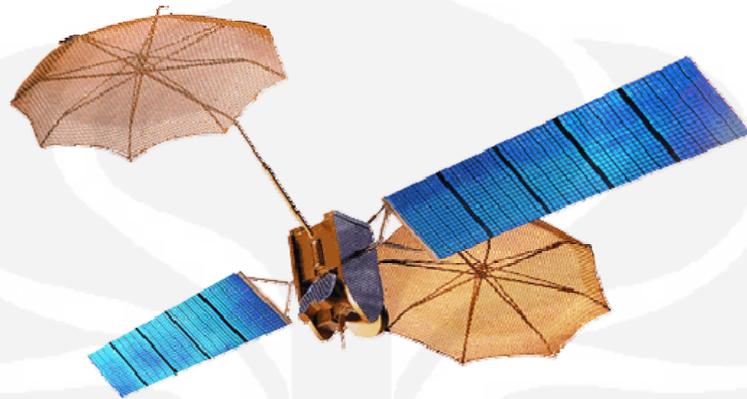
2.4 KOMUNIKASI SATELIT ACeS GARUDA 1

Asia Cellular Satellite (ACeS) memiliki beberapa macam satelit, yaitu Garuda 1, Palapa C, dan Agila 2. Satelit Garuda 1 diluncurkan pada tanggal 12 Februari 2000 di Moscow, Rusia. Satelit ini merupakan *link* komunikasi antara pelanggan dan *Network Control Center* (NCC) untuk pensinyalan, pelanggan dan *gateway* untuk suara dan pensinyalan, NCC dan masing-masing *gateway* untuk pensinyalan, dan antara *gateway* yang satu dengan *gateway* lainnya dengan pensinyalan.

2.4.1 Spesifikasi Satelit ACeS Garuda 1[20]

Satelit Garuda 1 (gambar 2.26) memiliki dua antena L-band dengan diameter 12 m untuk berkomunikasi dengan pelanggan (terdapat 140 *spotbeams* yang dicakup pada wilayah Asia Pasifik). Orbit satelit Garuda 1 berada pada 123⁰

bujur timur, $\pm 0,3^0$ ke utara dan selatan. Dan akan stabil pada 0^0 (garis khatulistiwa) pada saat umur satelit melebihi 3,7 tahun.



Gambar 2.26. Satelit ACeS Garuda 1

Satelit ACeS Garuda 1 memiliki spesifikasi sebagaimana berikut.

1. Dimensi: 1317,74 *feet* x 1244,37 *feet* (401,647152 m x 379,283976 m).
2. Berat: 2737,5 kg.
3. *Antenna coverage*: L-band (140 *spotbeams*) dan C-band.
4. Kapasitas: 11.000 (kira-kira 10,000 untuk *traffic*, 1000 untuk pensinyalan).
5. pelanggan: 2.000.000 (maksimal 12 juta).
6. Jumlah *National Gateway Station* (NGS): 3 untuk IOC (maksimal 26).
7. Jaringan: Satelit ACeS dan GSM.
8. *Frequency Multiplexing*
 - a. *Forward direction*
 - i. Modulasi: OQPSK
 - ii. *Burst rate*: 270,8 kbps
 - iii. *Carrier spacing*: 200KHz
 - iv. *Timeslot per frame*: 8
 - v. *Timeslot duration*: 577 μ sec
 - b. *Return direction*
 - i. Modulasi: GMSK (BT = 0,3)
 - ii. *Burst rate*: 67,7 kbps

- iii. *Carrier spacing*: 50KHz
- iv. *Timeslot per frame per carrier* 2
- v. *Timeslot duration*: 2,3 msec

9. Teknik *multiple access*: FDMA/TDMA.

2.4.2 Metode Pengiriman Data pada Perangkat PASTI/Byru Marine

Untuk mengirimkan data ke satelit melalui perangkat PASTI/Byru Marine diperlukan format tertentu sehingga satelit dapat mengenali data yang dikirimkan. Ada dua metode yang dapat dipakai saat pengiriman data yang dikenali oleh satelit.

1. Menghubungi nomor satelit teregistrasi

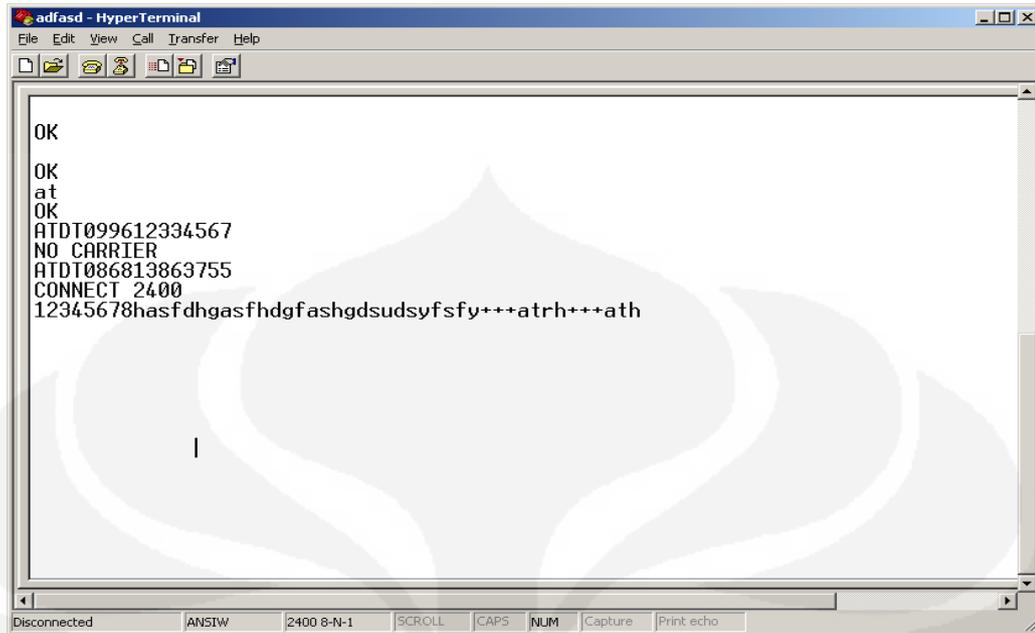
Metode ini dilakukan dengan menghubungi nomor data yang dikenali oleh satelit. Program akan memanggil nomor data perangkat PASTI/Byru Marine lain. Setelah berhasil berkomunikasi, barulah data GPS dikirimkan. Pada pemanggilan ini, *header* ATDT disisipkan di depan nomor yang dituju. Setelah terhubung, *header* ATDT dapat tidak disertakan dalam pengiriman data. Dengan metode ini, jumlah data yang dikirim tidak dibatasi. Data tersebut juga bisa dalam bentuk huruf atau karakter ASCII lainnya.

2. Menyisipkan data GPS sebagai nomor fiktif

Metode ini dilakukan dengan menumpangkan data yang hanya berisi angka dalam rentang 0-9 dengan total jumlah 16 angka ditambah *header* ATDT yang disisipkan di depan penomoran. Metode ini dapat dianalogikan dengan sistem salah panggil. Oleh karena itu, data GPS diatur sedemikian rupa sehingga hanya diwakili oleh angka saja.

2.4.2.1 Tampilan Data

Tampilan untuk pengiriman data berupa karakter ASCII ke satelit melalui perangkat PASTI/Byru Marine dapat dilihat pada Gambar 2.27.



```
adfasd - HyperTerminal
File Edit View Call Transfer Help
[Icons]
OK
OK
at
OK
ATDT099612334567
NO CARRIER
ATDT086813863755
CONNECT 2400
12345678hasfdhgasfhdgfashgdsudsyfsfy+++atrh+++ath
|
Disconnected ANSIR 2400 8-N-1 SCROLL CAPS NUM Capture Print echo
```

Gambar 2.27. Pengiriman data melalui perangkat PASTI/Byru Marine dengan menggunakan *hyper terminal*

Data pada baris kelima merupakan pengiriman dengan menggunakan metode kedua, yakni dua belas angka di belakang *header* ATDT merupakan nomor fiktif. Baris di bawahnya *NO CARRIER* menunjukkan bahwa tidak dapat dideteksinya nomor tujuan. Akan tetapi, data yang dikirimkan saat pemanggilan disimpan pada *server*. Di pihak lain, pada baris ketujuh merupakan pemanggilan dengan menggunakan nomor tujuan yang ada. Dua belas digit angka dibelakang *header* ATDT merupakan nomor yang aktif dan terdaftar. Dengan demikian, pada baris berikutnya terlihat pesan *CONNECT 2400* yang berarti koneksi berjalan pada kecepatan 2400 bps. Dengan metode ini, perangkat bebas mengirimkan data apapun dengan format karakter ASCII.

2.5 FILE TRANSFER PROTOCOL (FTP)

FTP merupakan metode untuk melakukan transfer *file* antarkomputer tanpa tergantung pada lokasi komputer, cara koneksinya, dan sistem operasinya. FTP dapat digunakan apabila komputer memiliki akses ke internet. Artinya, selama komputer tersebut memiliki protokol TCP/IP, FTP dapat digunakan.

2.5.1 Protokol FTP

FTP menggunakan paling banyak empat karakter ASCII untuk mendeskripsikan perintah-perintahnya. Tabel 2.14 memperlihatkan jenis-jenis dan deskripsi perintah yang dapat digunakan pada FTP.

Tabel 2.14. Perintah-Perintah FTP[21]

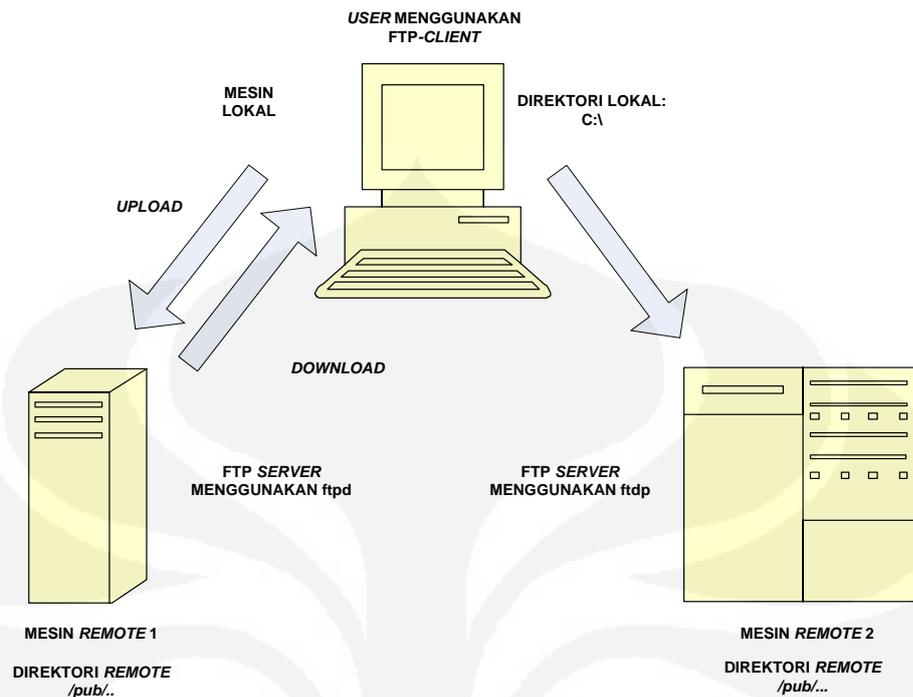
Perintah	Deskripsi
ABOR	Menggagalkan perintah FTP sebelumnya
ACCT	Menetapkan <i>account</i> pengguna
ALLO	Mengatakan pada <i>server</i> untuk mengalokasi ruang penyimpanan tambahan untuk berkas baru
APPE	Mengatakan kepada <i>server</i> untuk membubuhkan ke sebuah berkas yang eksis
CDUP	Beralih ke direktori induk
CWD	Beralih direktori
DELE	Menghapus sebuah berkas
HELP	Meminta bantuan
LIST	Mendaftarkan berkas-berkas untuk ditampilkan
MKD	Membuat sebuah direktori
MODE	Menentukan mode transfer data
NLST	Mendaftarkan berkas-berkas untuk prosesi tambahan
NOOP	No-op
PASS	Menetapkan <i>password</i> pengguna
PASV	Mengatakan kepada <i>server</i> untuk beranjak ke mode <i>passive</i>
PORT	Menetapkan <i>port</i> data untuk koneksi
PWD	Mencetak direktori kerja
QUIT	Menutup sesi koneksi
REIN	Membangun ulang koneksi
REST	Merestart transfer berkas
RETR	Mengembalikan berkas
RMD	Menghapus sebuah direktori
RNFR	Menetapkan nama <i>path</i> berkas untuk diganti

RNTO	Menetapkan nama baru sebuah berkas yang diganti
SITE	Memberikan layanan-layanan tambahan spesifik
SMNT	Melakukan <i>mount</i> sebuah berkas sistem berbeda
STAT	Memperoleh status
STOR	Mengatakan pada <i>server</i> untuk menerima penyimpanan sebuah berkas
STOU	Mengatakan pada <i>server</i> untuk membuat nama unik bagi sebuah berkas baru
STRU	Menetapkan struktur berkas
SYST	Mengatakan pada <i>server</i> untuk mendeklarasikan sistem operasi
TYPE	Menetapkan tipe representasi data
USER	Menetapkan <i>username</i>

FTP merupakan salah satu pelayanan internet, pemakai dapat menyalin berkas-berkas dari suatu komputer ke komputer lain. Misalnya, teks ASCII, teks berformat, gambar, suara, dan lain-lain.

Gambar 2.28 memperlihatkan pola hubungan antara FTP *client* dan FTP *server*. *User* (pengguna) menggunakan fasilitas FTP dengan menggunakan FTP *client*. Ketika seorang pengguna menjalankan FTP *client* dan melakukan koneksi dengan FTP *server*, setelah terjadinya koneksi biasanya diikuti dengan[22]:

1. *Login*. Pengguna diminta untuk memverifikasi *user ID* dan *password*.
2. Penentuan direktori. Pengguna diminta menentukan direktori permulaan. Direktori di FTP *client* dikenal dengan direktori lokal, sedangkan direktori di FTP *server* disebut direktori *remote*.
3. Penentuan modus transfer berkas. Pengguna menentukan format berkas yang dikirimkan, binari atau ASCII.
4. Memulai transfer data. Proses menyalin berkas dari komputer lain ke komputer pemakai disebut dengan *download*, sedangkan proses menyalin berkas dari komputer pemakai ke komputer lain disebut dengan *upload*.
5. Mengakhiri transfer data.



Gambar 2.28. Pola hubungan FTP *client* dan FTP *server*[23]

Di dalam pemakaian FTP, ada dua macam hak akses, yaitu:

1. *Anonymous* FTP. Yaitu, setiap pemakai dapat menggunakan FTP untuk mengakses *host* dengan gratis. Caranya, pemakai FTP akan diminta memasukkan *anonymous* sebagai *login*nya dan alamat email sebagai *password* (bisa juga dengan mengetik *guest* atau tanpa *password*, tergantung *host* yang dituju).
2. FTP yang pemakaiannya hanya oleh pihak tertentu (biasanya anggota dengan biaya tertentu). Pemakai harus memasukkan *login* dan *password* yang telah diberikan sebelumnya.

2.5.2 Antarmuka FTP

Program-program FTP telah mengalami kemajuan yang pesat. Jika dilihat dari sisi antarmukanya, FTP dapat dikelompokkan menjadi dua kategori.

1. *FTP client command line*

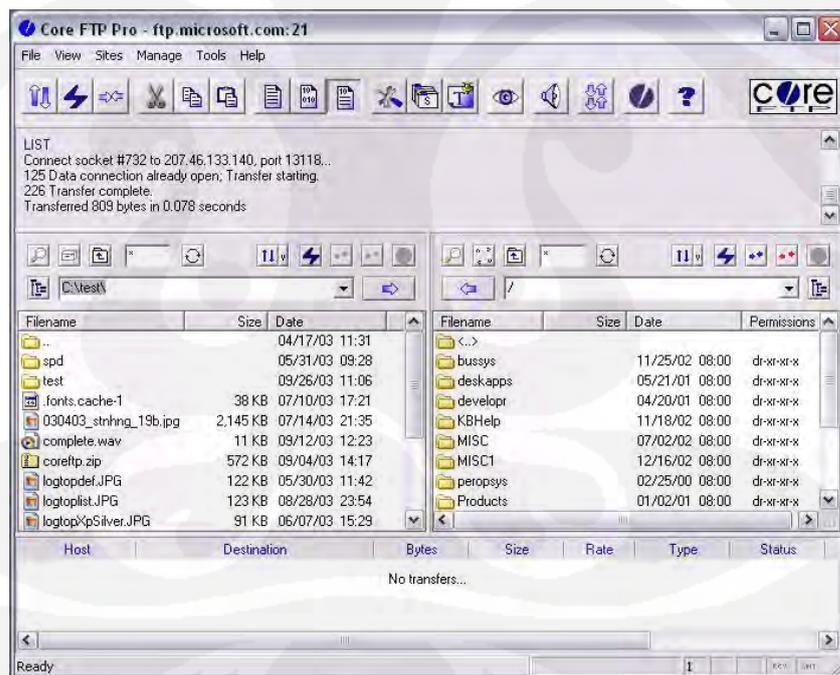
Antarmuka FTP jenis ini merupakan bentuk orisinal dari *FTP client*. FTP ini dapat ditemui disemua *platform*, baik itu pada MS Windows, UNIX, LINUX, dan BSD. Tidak diperlukan perangkat lunak tambahan untuk memakai *FTP client* ini.

2. FTP *client graphics* (GUI)

FTP grafikal menawarkan kemudahan dalam segi penggunaan, dikarenakan lebih mudah untuk dipahami. Kebanyakan opsi-opsi FTP *client* grafikal sulit diimplementasikan pada FTP *client command line*. Beberapa keunggulan yang dapat diperoleh pemakai pada FTP grafikal seperti fitur-fitur berikut:

- a. *Thumbnail view*. Fitur yang menampilkan gambar secara *thumbnail*.
- b. *Active edit*. Fitur pengeditan berkas secara *remote*.
- c. *Browser integration*. Fitur yang menyertakan *browser* yang terintegrasi.
- d. Fitur berupa antarmuka *explorer* seperti halnya aplikasi *explorer* pada *platform windows*.
- e. Fitur pencarian berkas.
- f. Dan lain-lain.

Gambar 2.29 memperlihatkan salah satu bentuk FTP dengan antarmuka GUI.



Gambar 2.29. Salah satu tampilan FTP grafikal

BAB III

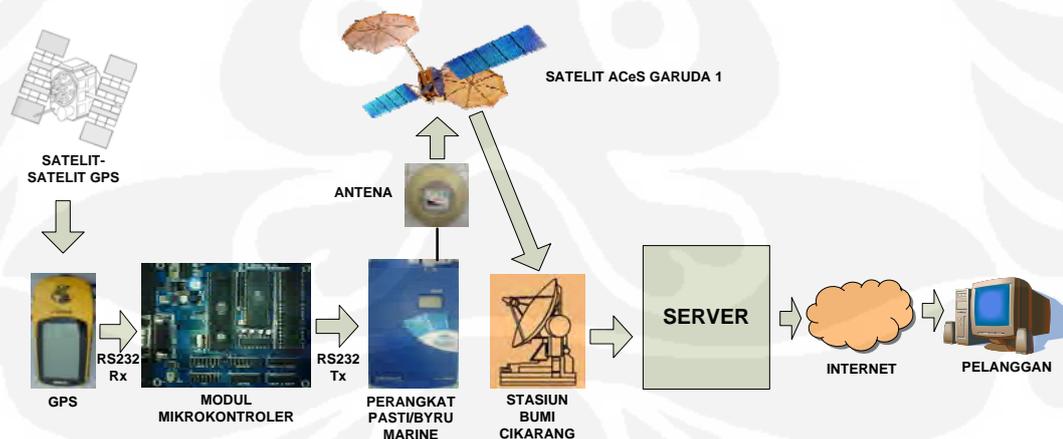
RANCANG BANGUN

3.1 PRINSIP KERJA SISTEM

Prinsip kerja sistem secara keseluruhan dimulai dari menempatkan sistem penerimaan GPS pada suatu tempat dan menunggu sampai GPS menerima data dengan baik (minimal menerima sinyal dari empat buah satelit GPS). Selanjutnya, mikrokontroler mulai mengolah data tersebut dengan cara mengambil data dari GPS melalui antarmuka konverter RS232C, kemudian menemukannya di suatu alamat memori RAM. Data mentah GPS tersebut *dimasking* dan dengan data *header ATDT* yang disimpan pada mikrokontroler, data dikirimkan melalui antarmuka konverter RS232C ke satelit ACeS Garuda 1 melalui perangkat PASTI/Byru Marine. Data yang dikirimkan disimpan dalam *server* ACeS yang terletak di Cikarang, kita dapat mengakses data pada *server* dengan memanfaatkan perangkat lunak *FTP Client*.

3.1.1 Blok Diagram dan Fungsinya

Sistem terdiri atas beberapa bagian yang dapat digambarkan menjadi blok diagram seperti yang terlihat pada gambar 3.1.



Gambar 3.1. Blok diagram sistem

Sistem yang dirancang dibagi menjadi empat bagian, yaitu perangkat penerima GPS, mikrokontroler (kontroler), perangkat PASTI/Byru Marine, dan Pelanggan (FTP Client).

1. Perangkat penerima GPS

Perangkat penerima GPS digunakan untuk menerima data dari satelit GPS.

2. Kontroler

Kontroler yang digunakan adalah mikrokontroler AT89S51 yang akan melakukan proses pengolahan data termasuk *masking* data GPS dan mengirimkannya ke satelit melalui perangkat PASTI/*Byru Marine*.

3. Perangkat PASTI/*Byru Marine*

Perangkat mengirimkan data GPS yang telah diolah kontroler ke satelit ACeS Garuda 1.

4. Pelanggan (FTP *Client*)

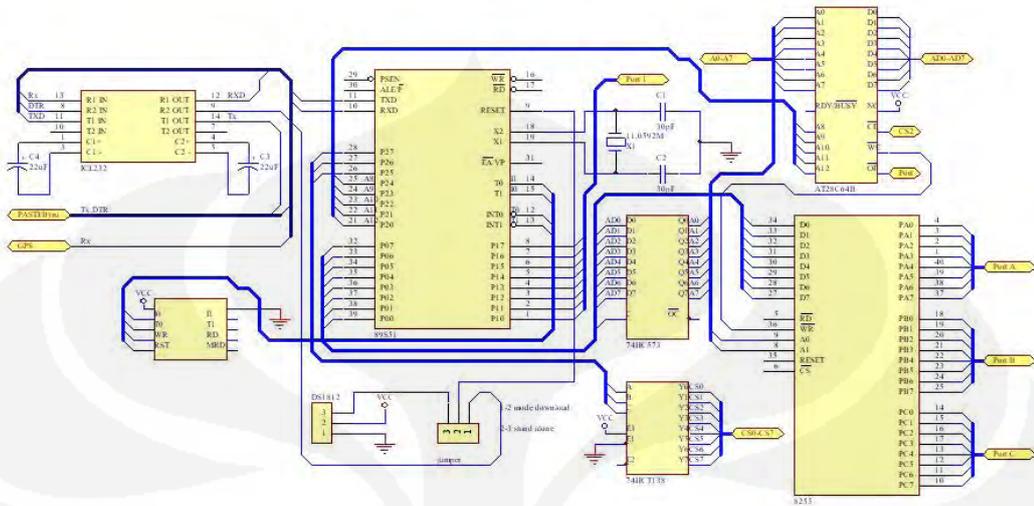
FTP *Client* mengakses data GPS yang telah dikirimkan sistem yang mana datanya berada pada *server*.

3.2 PERANGKAT KERAS

Perangkat keras yang digunakan sesuai dengan fungsi blok-blok sistem yang digambarkan pada gambar 3.1. Penerima GPS yang digunakan adalah GPS merk Garmin seri Etrex-H; Kontroler yang digunakan adalah modul mikrokontroler AT89S51; dan perangkat PASTI/*Byru Marine* untuk berkomunikasi dengan satelit Garuda 1.

Gambar 3.2 memperlihatkan diagram dari perangkat keras sistem. Perangkat keras sistem terdiri dari mikrokontroler AT89S51 sebagai kontroler, kristal sebesar 11,0592 MHz untuk penghasil denyut eksternal, dan ICL232 untuk mengkonversi komunikasi serial mikrokontroler ke RS232C jenis DB9.

Konverter RS232C dihubungkan dengan perangkat navigasi GPS dan perangkat PASTI/*Byru Marine*. Perangkat navigasi GPS difungsikan sebagai masukan data dan perangkat PASTI/*Byru Marine* difungsikan untuk mengirimkan data ke satelit ACeS Garuda 1.



Gambar 3.2. Skematik diagram mikrokontroler (lampiran 4)

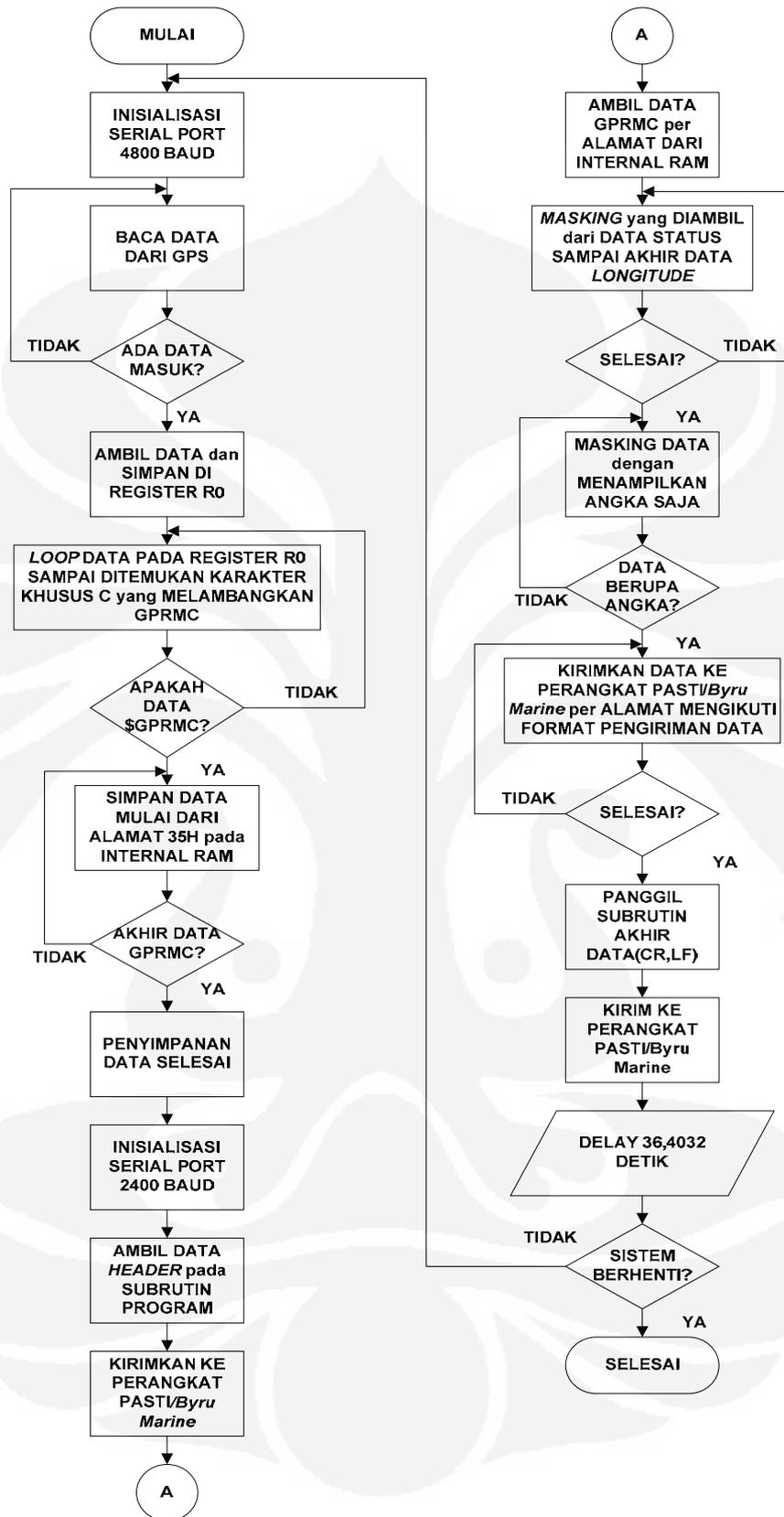
3.3 PERANGKAT LUNAK

Perangkat lunak merupakan pemrograman pada mikrokontroler AT89S51. Pemrograman yang dipakai adalah dengan menggunakan bahasa standar pemrograman mikrokontroler, yaitu bahasa assembler.

3.3.1 Pemrograman Mikrokontroler

Pemrograman pada mikrokontroler dilakukan dengan membuat subrutin-subrutin sebagai berikut: inisialisasi port serial, pengambilan data GPS, penyimpanan data pada RAM, *masking* data, dan pengiriman data ke perangkat PASTI/Byru Marine.

Algoritma pemrosesan data secara keseluruhan yang menggunakan mikrokontroler dapat digambarkan pada gambar 3.3 berikut.



Gambar 3.3. Diagram alir program mikrokontroler

Pada awal diagram alir, mikrokontroler akan melakukan inialisasi *port* serial yang digunakan yang berupa kecepatan baud, mode serial yang digunakan, dan besarnya nilai *timer* yang dipakai.

Pada perancangan ini digunakan dua kecepatan baud yang berbeda, yang pertama sebesar 4800 baud dan 2400 baud. Kecepatan 4800 baud difungsikan untuk mengambil data dari GPS. Kecepatan 2400 baud difungsikan untuk mengirimkan data ke satelit. Besaran kecepatan 4800 dan 2400 baud didapat dengan menentukan besarnya *timer* yang digunakan oleh mikrokontroler. Untuk menghitung nilai *timer* kedua kecepatan dapat dipergunakan modifikasi rumus pada persamaan 2.3.

1. Untuk kecepatan 4800 baud, digunakan *timer* sebesar

$$TH1 = 256d - \left(\frac{2^{SMOD}}{32d} \times \frac{frekuensi\ OSC}{12 \times 4800d} \right) \dots\dots\dots (3.1)$$

Dalam perencanaan dipergunakan SMOD bit 0 dan kristal sebesar 11,0592 MHz. Maka didapatkan.

$$TH1 = 256d - \left(\frac{2^0}{32d} \times \frac{11,0592 \times 10^6}{12 \times 4800d} \right)$$

$$TH1 = 250d$$

$$TH1 = FAh$$

2. Untuk kecepatan 2400 baud, digunakan *timer* sebesar

$$TH1 = 256d - \left(\frac{2^{SMOD}}{32d} \times \frac{frekuensi\ OSC}{12 \times 2400d} \right) \dots\dots\dots (3.2)$$

Besarnya Parameter dan nilai yang digunakan pada kecepatan 2400 baud sama dengan parameter dan nilai pada kecepatan 4800 baud. Maka didapatkan.

$$TH1 = 256d - \left(\frac{2^0}{32d} \times \frac{11,0592 \times 10^6}{12 \times 2400d} \right)$$

$$TH1 = 244d$$

$$TH1 = F4h$$

Pada Saat melakukan pengambilan data, mikrokontroler akan menset kecepatannya pada 4800 baud. Hal ini dikarenakan kecepatan data yang dipakai GPS adalah 4800 baud. Perlu dua kecepatan yang sama dalam melakukan tranmisi data antarperangkat, agar dapat melakukan komunikasi dan menghindari

adanya kesalahan pada saat penerimaan data. Sedangkan pada saat mengirim data ke satelit Garuda 1, kecepatan data yang digunakan perangkat PASTI/*Byru Marine* untuk berkomunikasi dengan satelit Garuda 1 adalah 2400 baud. Untuk sinkronisasi, mikrokontroler harus menset kecepataannya juga pada 2400 baud.

Mikrokontroler mengambil data dari GPS yang telah aktif (menerima sinyal sekurang-kurangnya dari empat satelit GPS). Mikrokontroler akan mengambil data mentah GPS melalui konektor data yang terhubung dengan antarmuka RS232C mikrokontroler. Data yang masuk ke mikrokontroler adalah data NMEA-0183 (sepuluh macam data). Tiap data yang diambil mikrokontroler, disimpan di register *R0*. Pada register ini diberikan *loop* untuk mencari data yang mempunyai karakter *C* di dalamnya. Karakter *C* ini melambangkan format data GPRMC.

Setelah ditemukan, tiap karakter setelah karakter *C* akan disimpan pada memori internal RAM, mulai alamat 35H. Data yang disimpan ini akan dibatasi dengan data *LF* (data 10d).

Setelah data *masking* GPRMC disimpan, mikrokontroler diubah kecepataannya menjadi 2400 baud. Ini menandakan data siap dikirimkan. Data yang dikirim pertama kali adalah data *ATDT*. Data tersebut dapat diambil dari subrutin yang telah disediakan. Setelah dikirimkan ke perangkat PASTI/*Byru Marine*, data GPRMC disiapkan untuk dikirimkan.

Data yang diambil dari internal RAM *dimasking* untuk mendapatkan data baru, yaitu, data dari status sampai dengan data akhir *longitude*. Program akan mengulang proses pada alamat internal RAM sampai didapat akhir data tersebut.

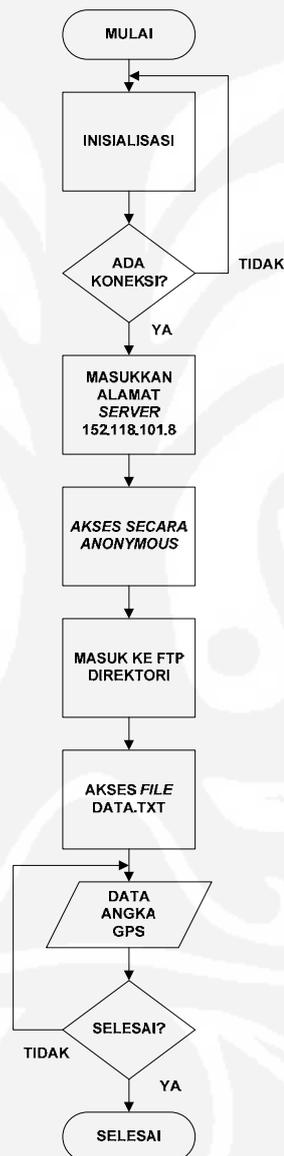
Setelah didapat data seperti yang dijelaskan pada paragraf sebelumnya, mikrokontroler kembali melakukan *masking* dari data tersebut. Tujuannya untuk mengambil data angka dari *masking* sebelumnya. Setelah didapatkan, data tersebut dikirimkan ke perangkat PASTI/*Byru Marine* per alamat. Untuk mengakhiri pengiriman satu siklus data GPS yang telah diolah, subrutin yang mengandung data *<CR><LF>* dikirimkan ke perangkat PASTI/*Byru Marine*.

Mikrokontroler akan *mendelay* sistemnya selama 36,4032 detik sebelum mengambil data baru dari GPS.

3.3.2 Pemrograman FTP Client

Pemrograman FTP *client* direncanakan supaya pelanggan dapat mengambil berkas berupa data posisi GPS secara otomatis. Artinya, pelanggan cukup melakukan *login* ke alamat *server*. Setelah mengakses alamat *server*, FTP *client* akan mengakses isi *server* secara otomatis.

Gambar 3.4 menunjukkan diagram alir program FTP *client*. Diagram alir di bawah ini menjelaskan cara pelanggan mengakses data pada *server*.



Gambar 3.4. Diagram alir program FTP *client*

Pertama kali sebelum mengakses *server*, pelanggan harus melakukan inisialisasi pada komputer, atau perangkat yang dapat mengakses dan melakukan

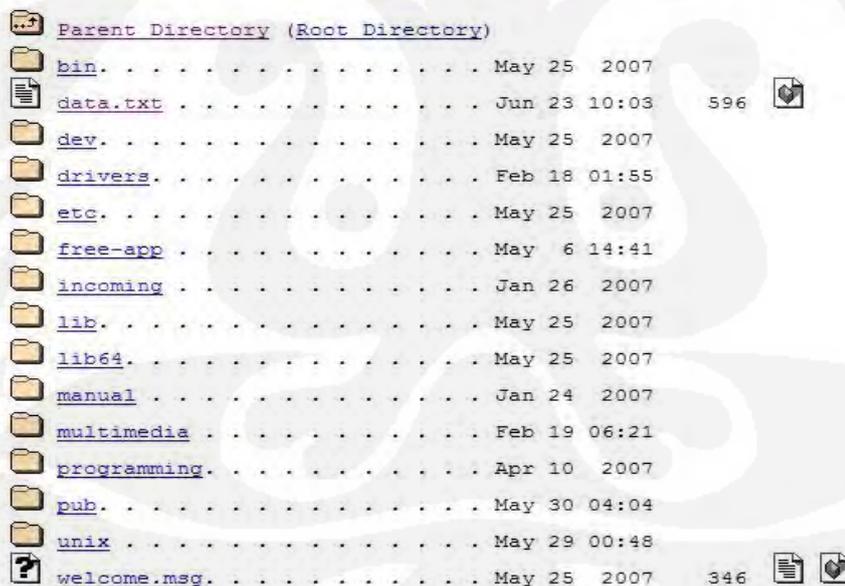
koneksi ke internet. Inisialisasi ini berupa pengaturan nomor *Internet Protocol* (IP) dan koneksi ke internet. Apabila ada koneksi, pelanggan dapat mengakses alamat *server 152.118.101.8*. Gambar 3.5 memperlihatkan akses *server* dengan menggunakan FTP.

Pelanggan diberi hak akses *anonymous*, yaitu hak akses gratis dan terbatas. Pelanggan dapat langsung mengakses direktori yang berisi data lokasi GPS yang telah dikirimkan ke satelit Garuda 1. Dengan membuka *file data.txt*, pelanggan dapat melihat data GPS dalam bentuk angka.

Untuk mengakhiri koneksi dengan *server*, pelanggan dapat menutup FTP *client*.

```
----- Welcome to Pure-FTPd [privsep] [TLS] -----  
You are user number 1 of 50 allowed.  
Local time is now 11:29. Server port: 21.  
IPv6 connections are also welcome on this server.
```

FTP Directory: <ftp://152.118.101.8/>



Icon	Name	Size	Date	Actions
Parent Directory	Parent Directory (Root Directory)			
Folder	bin.		May 25 2007	
File	data.txt	596	Jun 23 10:03	Download
Folder	dev.		May 25 2007	
Folder	drivers.		Feb 18 01:55	
Folder	etc.		May 25 2007	
Folder	free-app		May 6 14:41	
Folder	incoming		Jan 26 2007	
Folder	lib.		May 25 2007	
Folder	lib64.		May 25 2007	
Folder	manual		Jan 24 2007	
Folder	multimedia		Feb 19 06:21	
Folder	programming.		Apr 10 2007	
Folder	pub.		May 30 04:04	
Folder	unix		May 29 00:48	
File	welcome.msg.	346	May 25 2007	Download

Gambar 3.5. Tampilan akses FTP pada *server* dengan alamat 152.118.101.8

3.4 FORMAT DATA

Pada perancangan sistem di atas, dapat disimpulkan bahwa ada dua jenis data yang yang digunakan. Data tersebut adalah format data SDO dan format pengiriman data.

3.4.1 Format *Serial Data Output* (SDO)

Format data SDO adalah istilah format data serial yang dikirim perangkat GPS. Format data SDO terdiri atas beberapa segmen yang disusun dalam satu *frame* yang dilengkapi sinyal *flag*, *header*, dan lainnya. Semua data yang ada dalam *frame* merupakan sederetan data dalam kode ASCII 8 bit. Data diawali dengan simbol \$, kemudian disusul dengan data GPS (lintang, bujur, dan kecepatan). Untuk memisahkan setiap data digunakan tanda koma dan diakhiri dengan karakter *. Pada akhir *frame* diberi *carriage return/line feed* (CR/LF).

Format data dari GPS dalam bentuk SDO adalah

```
$GPDTS,Inf_1,Inf_2,Inf_3,Inf_4,Inf_5,Inf_6,Inf_n*CS<CR><LF>
```

Keterangan:

\$: Awal data
GP	: Informasi peralatan navigasi
DTS	: Jenis informasi yang terkandung
Inf_1 bis Inf_n	: Data navigasi
,	: Pemisahan antara informasi
*	: Pemisahan <i>checksum</i>
CS	: Pengecekan kesalahan pada kalimat
<CR><LF>	: Akhir dari data, yaitu <i>carriage return</i> (<CR>), dan <i>line feed</i> (<LF>)

Selain format di atas, terdapat berbagai macam format pengiriman data GPS, salah satu contohnya adalah \$GPRMC

```
$GPRMC,hhmmss,s,lll.lll,a,yyyyy.yyy,b,c,d,ddmmyy,e*f<CR><LF>[24]
```

Keterangan:

RMC	: <i>Recommended Minimum Specific GPS/TRANSIT Data</i>
hhmmss	: UTC of <i>position fix</i> dalam format jam/menit/detik
s	: Data Status (A untuk aktif dan V untuk <i>void</i>)
lll.lll	: Posisi <i>latitude</i>

- a : N or S (Utara atau Selatan)
- yyyyy.yyy : Posisi *longitude*
- b : E or W (Timur atau Barat)
- c : Kecepatan bergerak (knots)
- d : Sudut *track* dalam derajat
- ddmmyy : Tanggal UTC dalam hari/bulan/tahun
- e* : *Magnetic variation* E or W (Timur atau Barat)
- f : *Checksum*
- <CR><LF> : Akhir dari data: *carriage return* (<CR>), dan *line feed* (<LF>)

Misalnya GPS mengirimkan data sebagai berikut:

```
$GPRMC,123519,A,4807.038,N,01131.000,E,022.4,084.4,230394,003.1,W*6A
```

Data tersebut dapat dibaca sebagai berikut:

RMC : *recommended minimum sentence C*

123519 : diambil pada 12:35:19 UTC

A : status A = *active* atau V = *void*

4807.038,N : *latitude* 48 deg 07.038' N

01131.000,E : *longitude* 11 deg 31.000' E

022.4 : kecepatan dalam knots

084.4 : *track angle in degrees true*

230394: *date -23rd of March 1994*

003.1,W : *magnetic variation*

*6A : *the checksum data*, selalu diawali dengan *

3.4.2 Format Pengiriman Data

Berdasarkan pembahasan pada bagian 2.4.2 mengenai metode pengiriman data pada perangkat PASTI/Byru Marine, perancangan sistem menggunakan metode yang kedua. Walaupun menganalogikan data sebagai nomor fiktif, tersebut tersimpan dalam *server*. Untuk dapat mewakili data GPS dalam bentuk angka, diperlukan *masking* data mentah GPS. Dari data \$GPRMC yang disimpan di memori, data tersebut diberi *masking* menjadi data lokasi *latitude* dan *longitude*. Data lokasi tersebut *dimasking* lagi sehingga hanya data berupa angka saja yang diambil.

Berikut ini merupakan format pengiriman data ke satelit Garuda 1 menggunakan metode kedua:

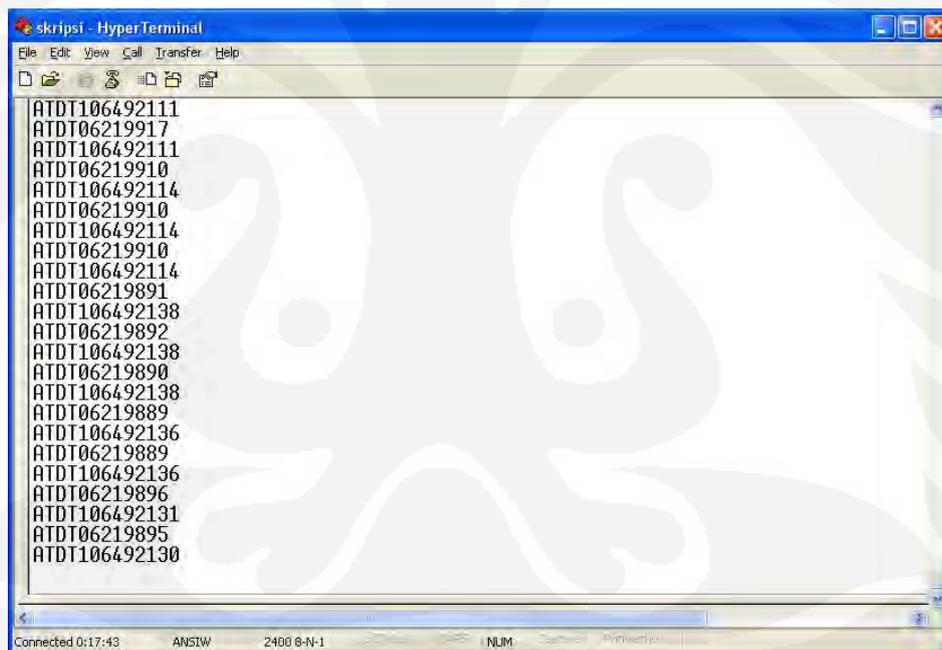
```
ATDTdataGPS<CR><LF>
```

Keterangan:

- ATDT : *Header* pengiriman data ke satelit Garuda 1
- dataGPS : Data GPS lokasi yang diberi *masking* berupa angka
- <CR><LF> : Akhir dari data, yaitu *carriage return* (<CR>), dan *line feed* (<LF>)

Gambar 3.4 memperlihatkan pengiriman data GPS yang telah disisipkan *header* ATDT pada tampilan *hyper terminal*. Metode pengiriman yang digunakan adalah metode kedua dengan format data GPS lokasi yang hanya berupa angka.

Jika data telah terkirim dan disimpan pada *server* satelit Garuda 1, maka pada layar perangkat PASTI/*Byru Marine* akan terlihat tulisan *DATA MODE*.



Gambar 3.6. Pengiriman data GPS setelah diolah di mikrokontroler

BAB IV

PENGUJIAN DAN ANALISIS SISTEM

4.1 HASIL RANCANG BANGUN

Berdasarkan rancang bangun pada Bab 3, dibuatlah sistem secara keseluruhan. Gambar 4.1 memperlihatkan perangkat keras yang digunakan pada sistem yang dirancang pada Bab sebelumnya.



Gambar 4.1. Perangkat keras sistem

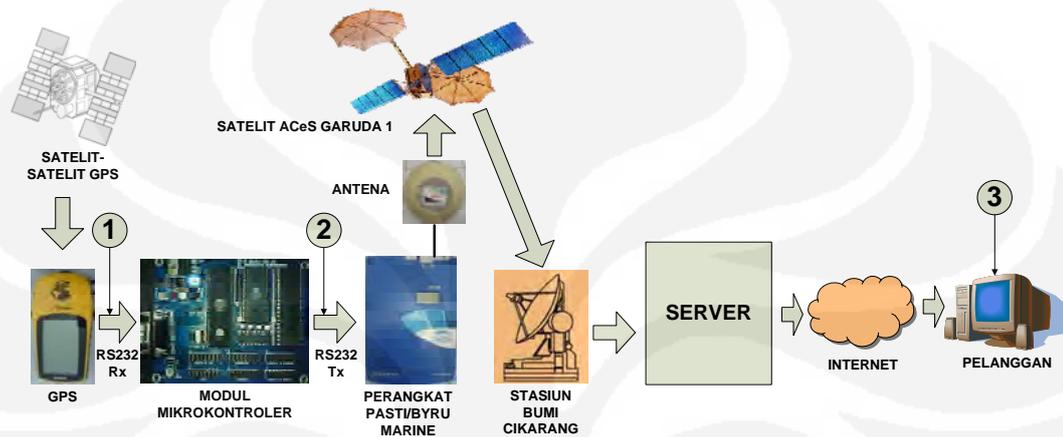
4.2 PENGUJIAN SISTEM

Pada sistem yang telah dibuat, dilakukan beberapa skenario pengujian, antara lain:

1. Menampilkan data keluaran GPS (NMEA-0183)
2. Menampilkan data keluaran pada kontroler dengan memberikan suatu kondisi yang telah ditetapkan pada diagram alir pemrograman, antara lain:
 - a. Data keluaran tanpa diberikan *masking* pada data mentah GPS.
 - b. Data keluaran dengan memberi *masking* \$GPRMC pada data mentah GPS.
 - c. Data keluaran dengan memberikan *masking* data posisi pada data *masking* \$GPRMC.
 - d. Data keluaran dengan memberikan *masking* data berupa angka pada data *masking* posisi.

3. Menampilkan data final yang dikirimkan, dengan mengakses *server* menggunakan perangkat lunak FTP *client*.

Untuk mendapatkan hasil dari skenario pengujian yang dipaparkan pada poin-poin di atas, kami menentukan tempat melakukan uji coba. Tempat yang dimaksudkan adalah poin-poin yang ditunjukkan (poin 1 dan poin 2) pada gambar 4.2.



Gambar 4.2. Blok diagram tempat pengujian sistem

4.2.1 Menampilkan Data Keluaran GPS

Menampilkan data keluaran GPS pada layar komputer bertujuan untuk mengetahui apakah GPS telah mengeluarkan data dari terminalnya atau belum. Ini dilakukan dengan cara menghubungkan konektor dari GPS dengan komputer melalui *port* serial PC. Pada komputer dijalankan program *hyper terminal* dan diset sesuai dengan konfigurasi pada GPS. Gambar 4.3 merupakan tampilan dari hasil pengujian keluaran data GPS pada *hyper terminal* dengan format NMEA-0183.

```

skripsi - HyperTerminal
File Edit View Call Transfer Help
$GPGLL,0621.9896,S,10649.2145,E,043224,A,A*5A
$GPBOD,,T,M,,*47
$PGRME,3.0,M,8.2,M,8.7,M*28
$PGRMZ,314,f,3*1D
$GPRTE,1,1,c,*37
$GPRMC,043226,A,0621.9896,S,10649.2145,E,0.0,81.9,090608,0.6,E,A*35
$GPRMB,A,,,,,,,,,V,A*1C
$GPGGA,043226,0621.9896,S,10649.2145,E,1,12,1.1,95.7,M,7.3,M,,*51
$GPGSA,A,3,01,05,06,09,12,14,18,21,22,24,30,31,2.4,1.1,2.2,*34
$GPGSV,3,1,12,01,11,300,00,05,42,077,35,06,03,261,00,09,05,026,00*77
$GPGSV,3,2,12,12,26,069,35,14,15,324,28,18,40,029,35,21,73,200,39*74
$GPGSV,3,3,12,22,20,350,00,24,36,153,29,30,54,109,41,31,42,243,29*74
$GPGLL,0621.9896,S,10649.2145,E,043226,A,A*58
$GPBOD,,T,M,,*47
$PGRME,3.0,M,8.1,M,8.7,M*2B
$PGRMZ,314,f,3*1D
$GPRTE,1,1,c,*37
$GPRMC,043228,A,0621.9896,S,10649.2145,E,0.0,81.9,090608,0.6,E,A*38
$GPRMB,A,,,,,,,,,V,A*1C
$GPGGA,043228,0621.9896,S,10649.2145,E,1,12,1.1,95.7,M,7.3,M,,*5C
$GPGSA,A,3,01,05,06,09,12,14,18,21,22,24,30,31,2.4,1.1,2.2,*34
$GPGSV,3,1,12,01,11,300,00,05,42,077,35,06,03,261,00,09,05,026,00*77
$GPGSV,3,2,12,12,26,069,36,14,15,324,28,18,40,029,35,21,73,200,39*77
$GPGSV,3,3,12,22,21,350,00,24,36,

```

Gambar 4.3. Data keluaran GPS yang ditampilkan pada *hyper terminal*

Format data GPS NMEA 0183 memiliki beberapa data berbeda. Terdapat dua belas pengiriman data sebelum kembali ke format data awal, yaitu \$GPRMC.....<CR><LF>. Siklus pengiriman data NMEA 0183 adalah 2 detik dihitung dari pengiriman data \$GPRMC awal sampai \$GPRMC berikutnya. Data posisi berada pada deretan data jika GPS bebas dari halangan (GPS tidak dapat menerima sinyal dari satelit-satelit GPS). Gambar 4.4 memperlihatkan kondisi GPS tidak mendapatkan sinyal dari satelit-satelit GPS (*void*).

```

skripsi - HyperTerminal
File Edit View Call Transfer Help
$GPRMB,V,,,,,,,,,V,N*04
$GPGGA,,,,,0,12,M,M,*65
$GPGSA,A,1,01,05,06,12,14,16,18,21,22,24,29,30,,*19
$GPGSV,3,1,12,01,08,322,00,05,29,051,00,06,08,239,00,12,14,049,00*75
$GPGSV,3,2,12,14,12,345,00,16,16,209,00,18,54,061,00,21,50,175,31*7C
$GPGSV,3,3,12,22,39,001,00,24,15,150,00,29,19,133,00,30,49,068,00*75
$GPGLL,,,,,V,N*64
$GPBOD,,T,M,,*47
$PGRME,,M,M,M*00
$PGRMZ,,f,1*29
$GPRTE,1,1,c,*37
$GPRMC,,V,,,,,150608,0.6,E,N*34
$GPRMB,V,,,,,,,,,V,N*04
$GPGGA,,,,,0,12,M,M,*65
$GPGSA,A,1,01,05,06,12,14,16,18,21,22,24,29,30,,*19
$GPGSV,3,1,12,01,08,322,00,05,29,051,00,06,08,239,00,12,14,049,00*75
$GPGSV,3,2,12,14,12,345,00,16,16,209,00,18,54,061,00,21,50,175,32*7F
$GPGSV,3,3,12,22,39,001,00,24,15,150,00,29,19,133,00,30,49,068,00*75
$GPGLL,,,,,V,N*64
$GPBOD,,T,M,,*47
$PGRME,,M,M,M*00
$PGRMZ,,f,1*29
$GPRTE,1,1,c,*37

```

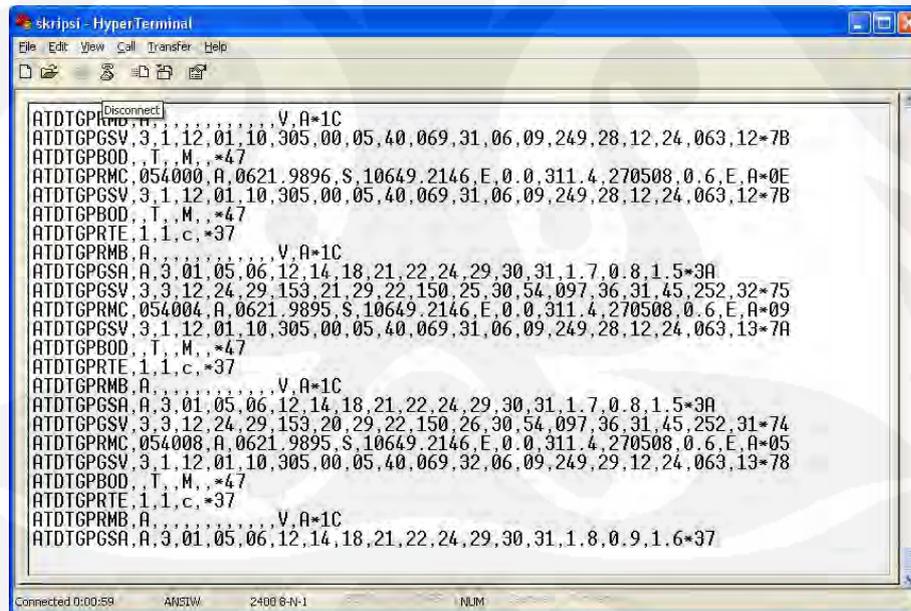
Gambar 4.4. Data keluaran GPS pada kondisi tanpa sinyal (*void*)

4.2.2 Menampilkan Data Keluaran Kontroler

Pada pengujian ini dilakukan empat macam *masking* terhadap data mentahan GPS seperti gambar 4.3 pada bagian sebelumnya. Pengujian ini bertujuan untuk menampilkan data dengan *masking* yang berbeda sehingga diperoleh data GPS yang sesuai untuk dikirimkan ke satelit Garuda 1. Data mentah GPS yang digunakan adalah data GPS dengan format NMEA-0183. Langkah pengujian dilakukan dengan menghubungkan konektor data GPS dengan modul mikrokontroler yang telah diprogram melalui *port* serial dan menghubungkan mikrokontroler dengan PC melalui *port* serial. PC akan menampilkan data olahan dari mikrokontroler pada program *hyper terminal*.

4.2.2.1 Data Keluaran Tanpa Memberikan Masking pada Data Mentah GPS

Pengujian ini bertujuan untuk melihat kemampuan kontroler dalam mengambil data mentah dari GPS. Gambar 4.5 memperlihatkan hasil olahan mikrokontroler yang hanya difungsikan melewati data mentahan GPS untuk ditampilkan pada *hyper terminal*.



```
skripsi - HyperTerminal
File Edit View Call Transfer Help
[Icons]
[Disconnect]
ATDTGPRMB,A,,,,,,,,,V,A=1C
ATDTGPGSV,3,1,12,01,10,305,00,05,40,069,31,06,09,249,28,12,24,063,12=7B
ATDTGPBOD,,T,,M,,*47
ATDTGPRMC,054000,A,0621.9896,S,10649.2146,E,0.0,311.4,270508,0.6,E,A=0E
ATDTGPGSV,3,1,12,01,10,305,00,05,40,069,31,06,09,249,28,12,24,063,12=7B
ATDTGPBOD,,T,,M,,*47
ATDTGPRTE,1,1,c,*37
ATDTGPRMB,A,,,,,,,,,V,A=1C
ATDTGPGSA,A,3,01,05,06,12,14,18,21,22,24,29,30,31,1.7,0.8,1.5*3A
ATDTGPGSV,3,3,12,24,29,153,21,29,22,150,25,30,54,097,36,31,45,252,32=75
ATDTGPRMC,054004,A,0621.9895,S,10649.2146,E,0.0,311.4,270508,0.6,E,A=09
ATDTGPGSV,3,1,12,01,10,305,00,05,40,069,31,06,09,249,28,12,24,063,13=7A
ATDTGPBOD,,T,,M,,*47
ATDTGPRTE,1,1,c,*37
ATDTGPRMB,A,,,,,,,,,V,A=1C
ATDTGPGSA,A,3,01,05,06,12,14,18,21,22,24,29,30,31,1.7,0.8,1.5*3A
ATDTGPGSV,3,3,12,24,29,153,20,29,22,150,26,30,54,097,36,31,45,252,31=74
ATDTGPRMC,054008,A,0621.9895,S,10649.2146,E,0.0,311.4,270508,0.6,E,A=05
ATDTGPGSV,3,1,12,01,10,305,00,05,40,069,32,06,09,249,29,12,24,063,13=78
ATDTGPBOD,,T,,M,,*47
ATDTGPRTE,1,1,c,*37
ATDTGPRMB,A,,,,,,,,,V,A=1C
ATDTGPGSA,A,3,01,05,06,12,14,18,21,22,24,29,30,31,1.8,0.9,1.6*37
Connected 0:00:58 ANSTW 2400 6-IV-1 NILM
```

Gambar 4.5. Tampilan data dari kontroler pada *hyper terminal* tanpa *masking*

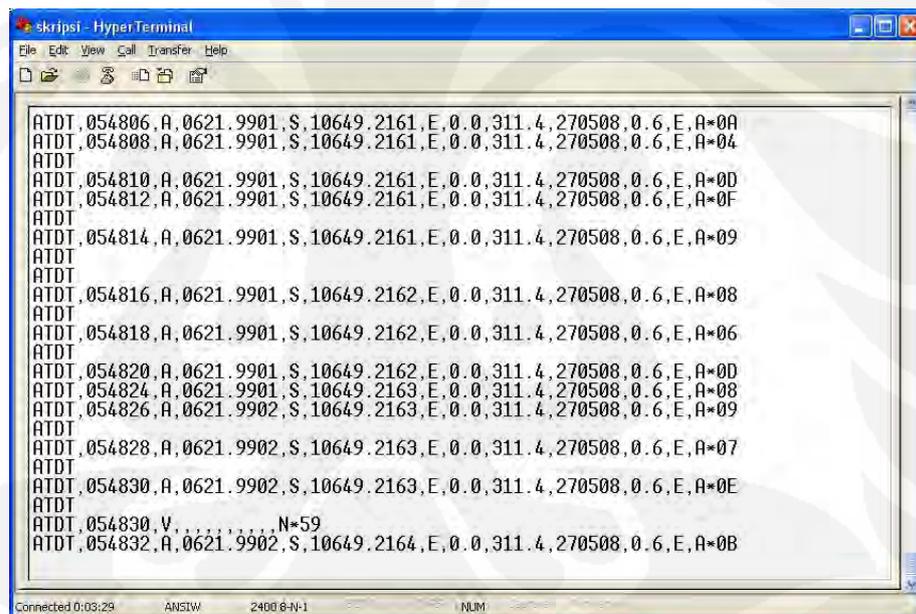
Gambar di atas memperlihatkan tidak semua data dapat diambil dari data mentahan GPS. Dari dua belas data yang dikirimkan tiap satu siklus pengiriman dengan format NMEA-0183. Hanya tujuh data yang dapat diambil per dua siklus.

Ini terlihat dari data waktu pada \$GPRMC pertama dengan \$GPRMC berikutnya yang berdurasi 4 detik.

Pada data \$GPRMC pertama pada baris keempat tampilan *hyper terminal*, data waktunya menunjukkan 054000 yang berarti jam 05, 40 menit 00 detik. Pada data \$GPRMC berikutnya yaitu pada baris kesebelas tampilan *hyper terminal* menunjukkan data waktu 054004 yang berarti jam 05, 40 menit 04 detik.

4.2.2.2 Data Keluaran dengan Memberikan Masking \$GPRMC pada Data Mentah GPS

Pengujian ini bertujuan untuk mengambil data \$GPRMC dari format data GPS NMEA-0183 yang dikirimkan. Mikrokontroler akan melakukan *masking* terhadap data mentah GPS yang diterimanya. Hasilnya hanya data \$GPRMC yang dikirimkan. Gambar 4.6 memperlihatkan data GPS yang telah diolah pada mikrokontroler sehingga hanya data \$GPRMC yang dilewatkan.



```
skripsi - HyperTerminal
File Edit View Call Transfer Help
[Icons]
ATDT,054806,A,0621.9901,S,10649.2161,E,0.0,311.4,270508,0.6,E,A*0A
ATDT,054808,A,0621.9901,S,10649.2161,E,0.0,311.4,270508,0.6,E,A*04
ATDT
ATDT,054810,A,0621.9901,S,10649.2161,E,0.0,311.4,270508,0.6,E,A*0D
ATDT,054812,A,0621.9901,S,10649.2161,E,0.0,311.4,270508,0.6,E,A*0F
ATDT
ATDT,054814,A,0621.9901,S,10649.2161,E,0.0,311.4,270508,0.6,E,A*09
ATDT
ATDT
ATDT,054816,A,0621.9901,S,10649.2162,E,0.0,311.4,270508,0.6,E,A*08
ATDT
ATDT,054818,A,0621.9901,S,10649.2162,E,0.0,311.4,270508,0.6,E,A*06
ATDT
ATDT,054820,A,0621.9901,S,10649.2162,E,0.0,311.4,270508,0.6,E,A*0D
ATDT,054824,A,0621.9901,S,10649.2163,E,0.0,311.4,270508,0.6,E,A*08
ATDT,054826,A,0621.9902,S,10649.2163,E,0.0,311.4,270508,0.6,E,A*09
ATDT
ATDT,054828,A,0621.9902,S,10649.2163,E,0.0,311.4,270508,0.6,E,A*07
ATDT
ATDT,054830,A,0621.9902,S,10649.2163,E,0.0,311.4,270508,0.6,E,A*0E
ATDT
ATDT,054830,V,,,,,,,,,N*59
ATDT,054832,A,0621.9902,S,10649.2164,E,0.0,311.4,270508,0.6,E,A*0B
Connected 0:03:29 ANSW 2400 8-N-1 NUM
```

Gambar 4.6. Tampilan data dari kontroler pada *hyper terminal* dengan *masking* data \$GPRMC

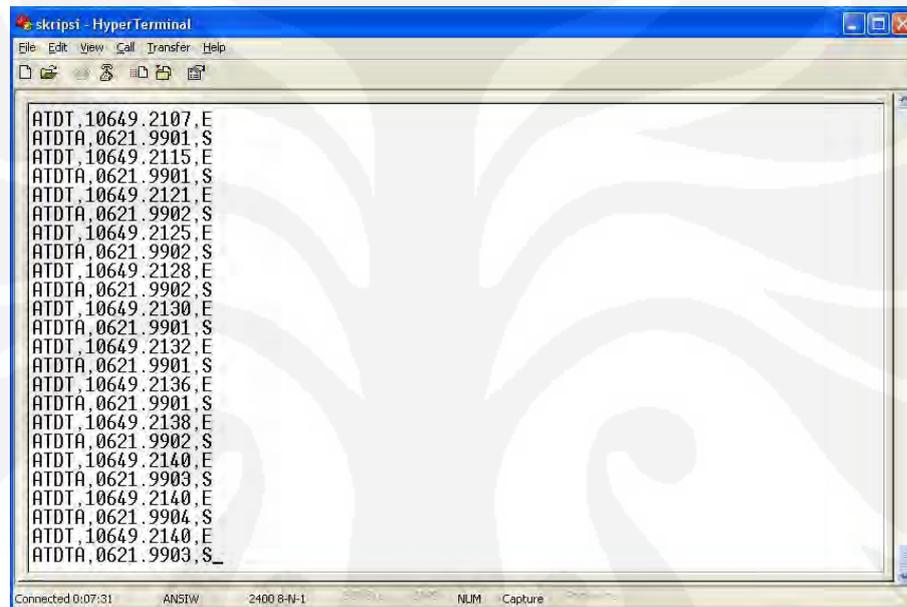
Pada gambar tersebut data \$GPRMC yang dilewatkan sesuai dengan siklus pengiriman data mentah GPS dengan format NMEA-0183, yaitu setiap 2 detik. Ini terlihat dari data waktu antara \$GPRMC pertama dengan yang berikutnya.

Pada data \$GPRMC pertama pada baris pertama tampilan *hyper terminal* data waktunya menunjukkan 054806 yang berarti jam 05, 48 menit 06 detik.

Sedangkan pada baris kedua *hyper terminal* data waktunya menunjukkan 054808 yang berarti jam 05, 48 menit 08 detik.

4.2.2.3 Data Keluaran dengan Memberikan Masking Posisi pada Data Masking \$GPRMC

Pengujian ini bertujuan untuk mengambil data posisi *latitude* dan *longitude* pada data \$GPRMC yang telah *dimasking* sebelumnya. Gambar 4.7 memperlihatkan data posisi yang telah diolah pada mikrokontroler dan ditampilkan pada PC dengan menggunakan program *hyper terminal*.



```
ATDT,10649,2107,E
ATDTA,0621,9901,S
ATDT,10649,2115,E
ATDTA,0621,9901,S
ATDT,10649,2121,E
ATDTA,0621,9902,S
ATDT,10649,2125,E
ATDTA,0621,9902,S
ATDT,10649,2128,E
ATDTA,0621,9902,S
ATDT,10649,2130,E
ATDTA,0621,9901,S
ATDT,10649,2132,E
ATDTA,0621,9901,S
ATDT,10649,2136,E
ATDTA,0621,9901,S
ATDT,10649,2138,E
ATDTA,0621,9902,S
ATDT,10649,2140,E
ATDTA,0621,9903,S
ATDT,10649,2140,E
ATDTA,0621,9904,S
ATDT,10649,2140,E
ATDTA,0621,9903,S
```

Gambar 4.7 Tampilan data dari kontroler pada *hyper terminal* dengan *masking* data posisi

Pada gambar 4.7 di atas hanya data posisi saja yang dilewatkan dari data \$GPRMC. Sebagai contoh, data posisi *latitude* seperti pada baris kedua tampilan *hyper terminal* adalah 0621.9901,S yang berarti *latitude* $06^{\circ} 21.9901'S$ atau lebih tepatnya data posisi GPS dapat dikonversi menjadi data dalam bentuk *Geographic Information Systems* (GIS) dalam dua langkah[25].

Langkah 1

Membagi koordinat dengan konstanta 100

$$\frac{0621.9901}{100} = 6.219901 = 6.2199$$

Langkah 2

Mengalikan angka di belakang koma (.) dengan konstanta 10 dan membaginya dengan konstanta 6. Hasilnya ditambahkan dengan angka di depan koma pada hasil langkah 1.

$$\text{Data posisi peta GIS} = 6 + \frac{0.2199 \times 10}{6} = 6.3665 \text{LS}$$

Karena kota Depok berada pada bagian selatan dari garis khatulistiwa maka di depan angka lintang selatan ditambahkan tanda minus (-) sehingga didapat -6.3665 LS.

Sedangkan data A di depan data *latitude* menunjukkan GPS dalam keadaan aktif atau menerima sinyal dari satelit-satelit GPS.

Untuk contoh data posisi *longitude* seperti pada baris pertama tampilan *hyper terminal* adalah 10649.2107,E yang berarti *longitude* 106⁰ 49.2107'E. Apabila dikonversi ke posisi peta GIS dengan menggunakan langkah yang sama, didapatkan:

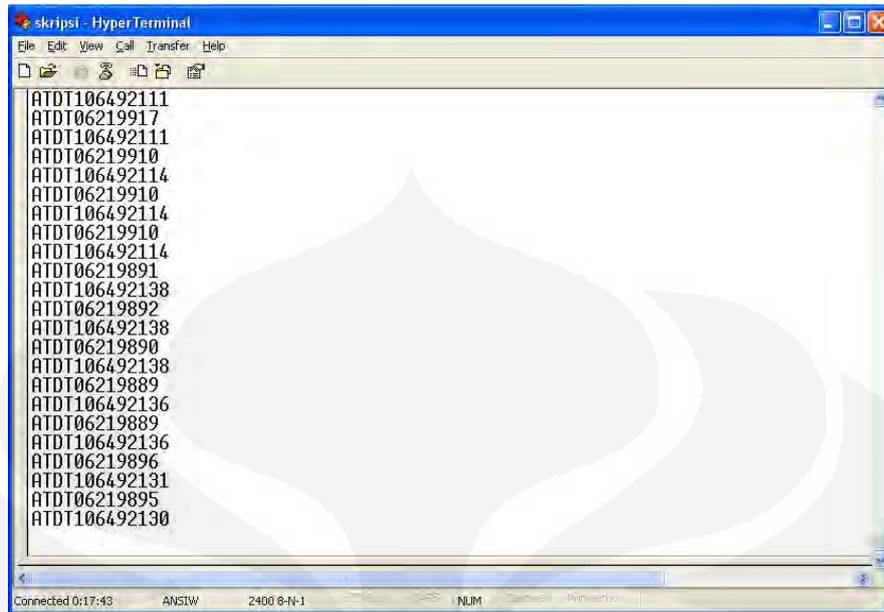
$$\frac{10649.2107}{100} = 106.492107 = 106.4921$$

$$\text{Data posisi peta GIS} = 106 + \frac{0.4921 \times 10}{6} = 106.8202 \text{BT}$$

4.2.2.4 Data Keluaran dengan Memberikan Masking Angka pada Data Masking Posisi

Pengujian ini bertujuan untuk mengolah data yang akan dikirimkan ke satelit Garuda 1 dengan menggunakan metode pengiriman nomor fiktif seperti yang dibahas pada Bab 2. Kontroler akan mengolah data posisi dari data \$GPRMC dalam bentuk angka. Artinya, huruf atau karakter yang dikirimkan dalam data posisi seperti pada gambar 4.7, harus ditiadakan. Gambar 4.8 memperlihatkan data posisi yang telah diolah, sehingga yang ditampilkan dari data tersebut hanya data berupa angka saja.

Apabila telah disusun seperti gambar 4.8, data tersebut siap untuk dikirimkan ke satelit Garuda 1 melalui perangkat PASTI/Byru Marine. Format ini seakan-akan seperti bentuk penomoran data biasa saat melakukan panggilan ke nomor tertentu.

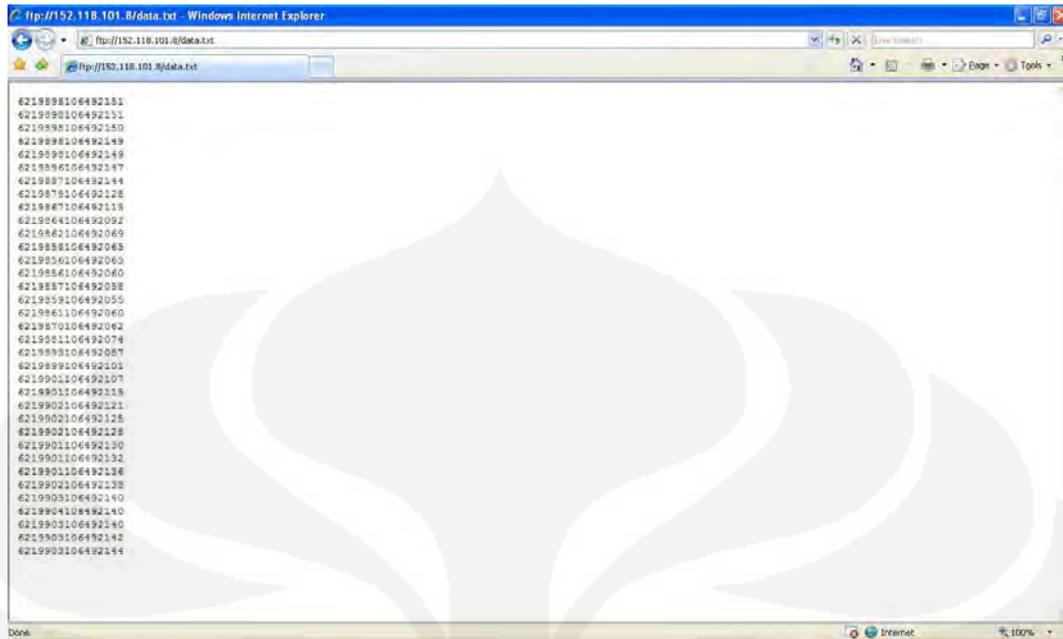


```
skripsi - HyperTerminal
File Edit View Call Transfer Help
ATDT106492111
ATDT06219917
ATDT106492111
ATDT06219910
ATDT106492114
ATDT06219910
ATDT106492114
ATDT06219910
ATDT106492114
ATDT06219891
ATDT106492138
ATDT06219892
ATDT106492138
ATDT06219890
ATDT106492138
ATDT06219889
ATDT106492136
ATDT06219889
ATDT106492136
ATDT06219896
ATDT106492131
ATDT06219895
ATDT106492130
Connected 0:17:43 ANSLW 2400 8-N-1 NLM
```

Gambar 4.8 Tampilan data posisi dalam bentuk angka

4.2.3 Menampilkan Data Final yang Dikirimkan dengan Mengakses *Server* Menggunakan *FTP Client*

Pengujian ini bertujuan melihat data GPS yang telah dikirimkan ke satelit Garuda 1 melalui *server* dengan alamat *152.118.101.8*. Gambar 4.9 memperlihatkan data GPS yang terdapat di *server*. Untuk mengakses data tersebut, kita harus menjalankan perangkat lunak *FTP client*, kemudian mengakses *server* dengan alur yang diterangkan pada bagian 3.3.2. Setelah mengakses direktori data yang ada, dapat dilihat bahwasannya data tersebut tidak lagi memiliki *header* ATDT seperti pada saat dikirimkan. Ini dikarenakan, ATDT merupakan suatu perintah pengiriman data, bukan data itu sendiri. Data yang masuk merupakan data lokasi GPS dalam bentuk angka.



Gambar 4.9 Data GPS pada *server* dengan alamat 152.118.101.8

4.3 ANALISIS SISTEM

Dari skenario pengujian yang dilakukan pada bagian 4.2, kita dapat menganalisis pengujian yang dilakukan di atas. Analisis yang dihasilkan adalah sebagai berikut.

4.3.1 Analisis Pengujian Pertama

Dari hasil pengujian pertama ada dua kondisi keluaran dari GPS, yaitu pada kondisi aktif (GPS mendapat sinyal dari satelit-satelit GPS) dan kondisi *void* (GPS tidak mendapatkan sinyal yang cukup dari satelit-satelit GPS). Pada gambar 4.4 beberapa data tidak dapat mengirimkan status posisi, waktu, dan kecepatannya dikarenakan kondisi *void* pada GPS. Untuk mendapatkan data-data seperti yang disebutkan di atas, GPS sekurang-kurangnya mendapatkan sinyal dari empat satelit GPS.

4.3.2 Analisis Pengujian Kedua

Pada pengujian kedua yang terbagi atas empat kali pengujian dengan mencoba beberapa skenario.

1. Pada saat menampilkan data mentah yang dikirimkan oleh GPS, mikrokontroler dikondisikan untuk mengambil semua data yang diambil dari GPS. Kondisi idealnya adalah semua data yang dikirimkan GPS akan diambil dan dikirimkan sesuai dengan data yang diterima mikrokontroler yaitu dua

belas macam data NMEA-0183 dengan siklus pengiriman 2 detik. Pada kenyataannya, mikrokontroler hanya mampu menangkap tujuh macam data NMEA-0183 dengan durasi 4 detik (dua kali siklus). Jika dirumuskan, berarti untuk mengirimkan satu data \$GPRMC, misalnya, diperlukan waktu sebesar:

$$\text{waktu untuk 1 data} = \frac{2 \text{ det } ik}{12 \text{ data}} = 0,1667 \text{ det } ik / \text{data}$$

Apabila dibandingkan dengan program untuk pengujian ini pada lampiran 1, maka besarnya waktu yang dibutuhkan untuk tiap data apabila GPS diposisikan dalam keadaan tidak bergerak adalah sebagai berikut:

Perhitungan besarnya nilai 1 mesin *cycle* (T) adalah[26]:

$$T = \frac{12}{\text{frekuensi osilator}} \dots\dots\dots (4.1)$$

$$T = \frac{12}{11,0592 \text{ MHz}}$$

$$T = 1,0850 \mu s$$

Apabila nilai ini dimasukkan sesuai dengan besarnya mesin *cycle* tiap instruksi didapatkan waktu tiap data sebagai berikut.

a. \$GPRMC

i. \$GPRMC terima = (23,8700 + A)μs

A merupakan banyaknya karakter yang diterima mikrokontroler. Apabila dalam keadaan diam, GPS menerima data \$GPRMC sebanyak 69 karakter termasuk <CR><LF> tanpa karakter \$ karena telah dimasking terlebih dahulu. Sebelum data \$GPRMC mencapai nilai #10d (<LF>), program mengulang proses pengambilan selama 68 kali dari subrutin *aaa* sampai akhir subrutin *bbb*. Setelah mencapai nilai #10d, program akan meloncat ke subrutin *ccc* untuk mengirimkan karakter.

$$A = (68x (\text{inst.1}+\text{inst.2}+\text{inst.5}+\text{inst.6}+\text{inst.7})) + (\text{inst.1}+\text{inst.2}+\text{inst.3}+\text{inst.4})$$

$$A = 1119,7200 \mu s$$

Dalam hal ini inst.1-inst.7, merupakan barisan instruksi pada total waktu A pada lampiran 1. Besarnya waktu pada saat mikrokontroler mengambil data adalah

$$\text{\$GPRMC terima} = (23,8700 + 1119,7200)\mu\text{s} = 1143,5900 \mu\text{s}$$

ii. $\text{\$GPRMC kirim} = (26,0400 + B + 19,5300)\mu\text{s}$

B merupakan banyaknya karakter yang dikirim mikrokontroler. Pada saat mengirim data \\$GPRMC, jumlah karakter yang dikirimkan adalah 68 karakter tanpa <LF>.

$$B = (67 \times (\text{inst.1} + \text{inst.2} + \text{inst.5} + \text{inst.6} + \text{inst.7})) + (\text{inst.1} + \text{inst.2} + \text{inst.3} + \text{inst.4})$$

$$B = 1105,6150\mu\text{s}$$

$$\text{\$GPRMC kirim} = (26,0400 + 1105,6150 + 19,5300)\mu\text{s} = 1151,1850\mu\text{s}$$

iii. $\text{\$GPRMC total} = \text{\$GPRMC terima} + \text{\$GPRMC kirim}$

$$\text{\$GPRMC total} = (1143,5900 + 1151,1850)\mu\text{s} = 2292,7750 \mu\text{s}$$

iv. $\text{Delay} = \text{\$GPRMC kirim} - \text{\$GPRMC terima}$

$$\text{Delay} = (1151,1850 - 1143,5900)\mu\text{s} = 7,5950\mu\text{s}$$

b. $\text{\$GPRMB}$

$\text{\$GPRMB}$ total karakter 27

i. $\text{\$GPRMB terima} = (23,8700 + A)\mu\text{s}$

$$A = (26 \times (\text{inst.1} + \text{inst.2} + \text{inst.5} + \text{inst.6} + \text{inst.7})) + (\text{inst.1} + \text{inst.2} + \text{inst.3} + \text{inst.4})$$

$$A = 436,1700\mu\text{s}$$

$$\text{\$GPRMB terima} = (23,8700 + 436,1700)\mu\text{s} = 460,0400\mu\text{s}$$

ii. $\text{\$GPRMB kirim} = (26,0400 + B + 19,5300)\mu\text{s}$

$$B = (25 \times (\text{inst.1} + \text{inst.2} + \text{inst.5} + \text{inst.6} + \text{inst.7})) + (\text{inst.1} + \text{inst.2} + \text{inst.3} + \text{inst.4})$$

$$B = 422,0650\mu\text{s}$$

$$\text{\$GPRMB kirim} = (26,0400 + 422,0650 + 19,5300)\mu\text{s} = 467,6350\mu\text{s}$$

iii. $\text{\$GPRMB total} = \text{\$GPRMB terima} + \text{\$GPRMB kirim}$

$$\text{\$GPRMB total} = 460,0400\mu\text{s} + 467,6350\mu\text{s} = 927,6750\mu\text{s}$$

iv. $\text{Delay} = \text{\$GPRMB kirim} - \text{\$GPRMB terima}$

$$\text{Delay} = 7,5950\mu\text{s}$$

c. \$GPGGA

GPGGA total karakter 66

i. \$GPGGA terima = $(23,8700 + A)\mu\text{s}$

$$A = (65 \times (\text{inst.1} + \text{inst.2} + \text{inst.5} + \text{inst.6} + \text{inst.7})) + (\text{inst.1} + \text{inst.2} + \text{inst.3} + \text{inst.4})$$

$$A = 1070,8950\mu\text{s}$$

$$\text{\$GPGGA terima} = (23,8700 + 1070,8950)\mu\text{s} = 1094,7650\mu\text{s}$$

ii. \$GPGGA kirim = $(26,0400 + B + 19,5300)\mu\text{s}$

$$B = (64 \times (\text{inst.1} + \text{inst.2} + \text{inst.5} + \text{inst.6} + \text{inst.7})) + (\text{inst.1} + \text{inst.2} + \text{inst.3} + \text{inst.4})$$

$$B = 1056,7900\mu\text{s}$$

$$\text{\$GPGGA kirim} = (26,0400 + 1056,7900 + 19,5300)\mu\text{s} = 1102,3600\mu\text{s}$$

iii. \$GPGGA total = \$GPGGA terima + \$GPGGA kirim

$$\text{\$GPGGA total} = 2197,1250\mu\text{s}$$

iv. *Delay* = \$GPGGA kirim - \$GPGGA terima

$$\text{Delay} = 7,5950\mu\text{s}$$

d. \$GPGSA

\$GPGSA total karakter 46

i. \$GPGSA terima = $(23,8700 + A)\mu\text{s}$

$$A = (45 \times (\text{inst.1} + \text{inst.2} + \text{inst.5} + \text{inst.6} + \text{inst.7})) + (\text{inst.1} + \text{inst.2} + \text{inst.3} + \text{inst.4})$$

$$A = 745,3950\mu\text{s}$$

$$\text{\$GPGSA terima} = (23,87 + 745,395)\mu\text{s} = 769,2650\mu\text{s}$$

ii. \$GPGSA kirim = $(26,0400 + B + 19,5300)\mu\text{s}$

$$B = (44 \times (\text{inst.1} + \text{inst.2} + \text{inst.5} + \text{inst.6} + \text{inst.7})) + (\text{inst.1} + \text{inst.2} + \text{inst.3} + \text{inst.4})$$

$$B = 731,2900\mu\text{s}$$

$$\text{\$GPGSA kirim} = (26,0400 + 731,2900 + 19,5300)\mu\text{s} = 776,8600\mu\text{s}$$

iii. \$GPGSA total = \$GPGSA terima + \$GPGSA kirim

$$\text{\$GPGSA total} = 1546,1250\mu\text{s}$$

iv. *Delay* = \$GPGSA kirim - \$GPGSA terima

$$\text{Delay} = 7,5950\mu\text{s}$$

e. \$GPGSV

\$GPGSV total karakter 69

i. \$GPGSV terima = $(23,8700 + A)\mu\text{s}$

$$A = (68 \times (\text{inst.1} + \text{inst.2} + \text{inst.5} + \text{inst.6} + \text{inst.7})) + (\text{inst.1} + \text{inst.2} + \text{inst.3} + \text{inst.4})$$

$$A = 1119,7200\mu\text{s}$$

$$\text{\$GPGSV terima} = (23,8700 + 1119,7200)\mu\text{s} = 1143,5900 \mu\text{s.}$$

ii. \$GPGSV kirim = $(26,0400 + B + 19,5300)\mu\text{s}$

$$B = (67 \times (\text{inst.1} + \text{inst.2} + \text{inst.5} + \text{inst.6} + \text{inst.7})) + (\text{inst.1} + \text{inst.2} + \text{inst.3} + \text{inst.4})$$

$$B = 1105,6150\mu\text{s}$$

$$\text{\$GPGSV kirim} = (26,0400 + 1105,6150 + 19,5300)\mu\text{s} = 1151,1850\mu\text{s}$$

iii. \$GPGSV total = \$GPGSV terima + \$GPGSV kirim

$$\text{\$GPGSV total} = 2292,7750 \mu\text{s}$$

iv. *Delay* = \$GPGSV kirim - \$GPGSV terima

$$\text{Delay} = 7,5950\mu\text{s}$$

f. \$GPGLL

\$GPGLL total karakter 46

i. \$GPGLL terima = $(23,8700 + A)\mu\text{s}$

$$A = (45 \times (\text{inst.1} + \text{inst.2} + \text{inst.5} + \text{inst.6} + \text{inst.7})) + (\text{inst.1} + \text{inst.2} + \text{inst.3} + \text{inst.4})$$

$$A = 745,3950\mu\text{s}$$

$$\text{\$GPGLL terima} = (23,8700 + 745,3950)\mu\text{s} = 769,2650\mu\text{s}$$

ii. \$GPGLL kirim = $(26,0400 + B + 19,5300)\mu\text{s}$

$$B = (44 \times (\text{inst.1} + \text{inst.2} + \text{inst.5} + \text{inst.6} + \text{inst.7})) + (\text{inst.1} + \text{inst.2} + \text{inst.3} + \text{inst.4})$$

$$B = 731,2900\mu\text{s}$$

$$\text{\$GPGLL kirim} = (26,0400 + 731,2900 + 19,5300)\mu\text{s} = 776,8600\mu\text{s}$$

iii. \$GPGLL total = \$GPGLL terima + \$GPGLL kirim

$$\text{\$GPGLL total} = 1546,1250\mu\text{s}$$

iv. *Delay* = \$GPGLL kirim - \$GPGLL terima

$$\text{Delay} = 7,5950\mu\text{s}$$

g. \$GPBOD

\$GPBOD total karakter 18

i. \$GPBOD terima = $(23,8700 + A)\mu\text{s}$

$$A = (17 \times (\text{inst.1} + \text{inst.2} + \text{inst.5} + \text{inst.6} + \text{inst.7})) + (\text{inst.1} + \text{inst.2} + \text{inst.3} + \text{inst.4})$$

$$A = 289,6950\mu\text{s}$$

$$\text{\$GPBOD terima} = (23,8700 + 289,6950)\mu\text{s} = 313,5650\mu\text{s}$$

ii. \$GPBOD kirim = $(26,0400 + B + 19,5300)\mu\text{s}$

$$B = (16 \times (\text{inst.1} + \text{inst.2} + \text{inst.5} + \text{inst.6} + \text{inst.7})) + (\text{inst.1} + \text{inst.2} + \text{inst.3} + \text{inst.4})$$

$$B = 275,5900\mu\text{s}$$

$$\text{\$GPBOD kirim} = (26,0400 + 275,5900 + 19,5300)\mu\text{s} = 321,1600\mu\text{s}$$

iii. \$GPBOD total = \$GPBOD terima + \$GPBOD kirim

$$\text{\$GPBOD total} = 634,7250\mu\text{s}$$

iv. *Delay* = \$GPBOD kirim - \$GPBOD terima

$$\text{Delay} = 7,5950\mu\text{s}$$

h. \$GPRME

\$GPRME total karakter 31

i. \$GPBOD terima = $(23,8700 + A)\mu\text{s}$

$$A = (30 \times (\text{inst.1} + \text{inst.2} + \text{inst.5} + \text{inst.6} + \text{inst.7})) + (\text{inst.1} + \text{inst.2} + \text{inst.3} + \text{inst.4})$$

$$A = 501,2700\mu\text{s}$$

$$\text{\$GPRME terima} = (23,8700 + 501,2700)\mu\text{s} = 525,1400\mu\text{s}$$

ii. \$GPRME kirim = $(26,0400 + B + 19,5300)\mu\text{s}$

$$B = (29 \times (\text{inst.1} + \text{inst.2} + \text{inst.5} + \text{inst.6} + \text{inst.7})) + (\text{inst.1} + \text{inst.2} + \text{inst.3} + \text{inst.4})$$

$$B = 487,1650\mu\text{s}$$

$$\text{\$GPRME kirim} = (26,0400 + 487,1650 + 19,5300)\mu\text{s} = 532,7350\mu\text{s}$$

iii. \$GPRME total = \$GPRME terima + \$GPRME kirim

$$\text{\$GPRME total} = 1057,8750\mu\text{s}$$

iv. *Delay* = \$GPRME kirim - \$GPRME terima

$$\text{Delay} = 7,5950\mu\text{s}$$

i. \$GPRMZ

\$GPRMZ total karakter 18

i. \$GPRMZ terima = $(23,8700 + A)\mu\text{s}$

$$A = (17 \times (\text{inst.1} + \text{inst.2} + \text{inst.5} + \text{inst.6} + \text{inst.7})) + (\text{inst.1} + \text{inst.2} + \text{inst.3} + \text{inst.4})$$

$$A = 289,6950\mu\text{s}$$

$$\text{\$GPRMZ terima} = (23,8700 + 289,6950)\mu\text{s} = 313,5650\mu\text{s}$$

ii. \$GPRMZ kirim = $(26,0400 + B + 19,5300)\mu\text{s}$

$$B = (16 \times (\text{inst.1} + \text{inst.2} + \text{inst.5} + \text{inst.6} + \text{inst.7})) + (\text{inst.1} + \text{inst.2} + \text{inst.3} + \text{inst.4})$$

$$B = 275,5900\mu\text{s}$$

$$\text{\$GPRMZ kirim} = (26,0400 + 275,5900 + 19,5300)\mu\text{s} = 321,1600\mu\text{s}$$

iii. \$GPRMZ total = \$GPRMZ terima + \$GPRMZ kirim

$$\text{\$GPRMZ total} = 634,7250\mu\text{s}$$

iv. *Delay* = \$GPRMZ kirim - \$GPRMZ terima

$$\text{Delay} = 7,5950\mu\text{s}$$

j. \$GPRTE

\$GPRTE total karakter 17

i. \$GPRTE terima = $(23,8700 + A)\mu\text{s}$

$$A = (16 \times (\text{inst.1} + \text{inst.2} + \text{inst.5} + \text{inst.6} + \text{inst.7})) + (\text{inst.1} + \text{inst.2} + \text{inst.3} + \text{inst.4})$$

$$A = 273,4200\mu\text{s}$$

$$\text{\$GPRTE terima} = (23,8700 + 273,4200)\mu\text{s} = 297,2900\mu\text{s}$$

ii. \$GPRTE kirim = $(26,0400 + B + 19,5300)\mu\text{s}$

$$B = (15 \times (\text{inst.1} + \text{inst.2} + \text{inst.5} + \text{inst.6} + \text{inst.7})) + (\text{inst.1} + \text{inst.2} + \text{inst.3} + \text{inst.4})$$

$$B = 259,3150\mu\text{s}$$

$$\text{\$GPRTE kirim} = (26,0400 + 259,3100 + 19,5300)\mu\text{s} = 304,8850\mu\text{s}$$

iii. \$GPRTE total = \$GPRTE terima + \$GPRTE kirim

$$\text{\$GPRTE total} = 602,1750\mu\text{s}$$

iv. *Delay* = \$GPRTE kirim - \$GPRTE terima

$$\text{Delay} = 7,5950\mu\text{s}$$

Dari perhitungan di atas dapat dibuat tabel 4.1.

Tabel 4.1. Hasil Perhitungan Total Waktu Pengiriman Data

Jenis Data	Total Waktu (μ s)	Delay (μ s)
\$GPRMC	2292,7750	7,5950
\$GPRMB	927,6750	7,5950
\$GPGGA	2197,1250	7,5950
\$GPGSA	1546,1250	7,5950
\$GPGSV	2292,7750	7,5950
\$GPGLL	1546,1250	7,5950
\$GPBOD	634,7250	7,5950
\$GPRME	1057,8750	7,5950
\$GPRMZ	634,7250	7,5950
\$GPRTE	602,1750	7,5950

Total waktu yang dibutuhkan oleh mikrokontroler, bila semua data dapat diambil dari GPS, adalah:

$$(2292,775+927,675+2197,125+1546,125+2292,775+2292,775+2292,775+1546,125+634,725+1057,875+634,725+602,175) = 18317,6500\mu\text{s} = 0,0183\text{s}$$

Namun demikian, waktu ideal untuk satu siklus dari \$GPRMC ke \$GPRMC berikutnya, diperlukan waktu 2 detik atau 0,1677 detik per data. Ini menyebabkan adanya perbedaan waktu yang sangat jauh antara siklus pada GPS dengan mikrokontroler, akibatnya, tidak semua data dapat diambil oleh mikrokontroler. Hal ini dapat dilihat pada Gambar 4.5.

2. Mikrokontroler dikondisikan untuk melakukan *masking* terhadap data mentah dari GPS. Data yang akan diambil dan dilewati hanya format \$GPRMC, yaitu data minimum yang direkomendasikan pada sistem navigasi GPS. Tujuannya adalah agar mikrokontroler dapat mengambil data \$GPRMC persiklus pengiriman data GPS. Hal ini merupakan kendala dari pengujian pertama.

Untuk mengetahui besarnya waktu yang diperlukan untuk melakukan pengambilan dan pengiriman data pada mikrokontroler, dapat dengan menghitung total mesin *cycle* pada programnya. Besarnya mesin *cycle* dan waktu yang diperlukan dapat dilihat pada lampiran 2. Untuk memudahkan *masking*, program dibatasi dengan jangkauan antara karakter khusus data \$GPRMC (karakter C) dengan data akhir <LF>.

Lama waktu pengambilan data dari GPS oleh mikrokontroler dengan *masking* \$GPRMC adalah:

a. \$GPRMC terima = (11,9350 + A + B) μ s

A adalah banyaknya pengulangan sampai mikrokontroler mengenali karakter C pada data GPS.

$$A = (5 \times (\text{Inst.1} + \text{Inst.2})) + (\text{Inst.1} + \text{Inst.2} + \text{Inst.3})$$

$$A = (5 \times (9,7650)) + (11,9350)$$

$$A = 60,7600\mu\text{s}$$

B adalah banyaknya pengulangan sampai mikrokontroler memperoleh semua data \$GPRMC yaitu dengan ditandai dengan <LF>.

$$B = (63 \times (\text{Inst.1} + \text{Inst.2} + \text{Inst.5} + \text{Inst.6} + \text{Inst.7}) + (\text{Inst.1} + \text{Inst.2} + \text{Inst.3} + \text{Inst.4}))$$

$$B = 1038,3450\mu\text{s}$$

$$\text{\$GPRMC terima} = (11,9350 + 60,7600 + 1038,3450)\mu\text{s}$$

$$\text{\$GPRMC terima} = 1111,0400\mu\text{s}$$

b. \$GPRMC kirim = (26,0400 + C + 19,5300) μ s

C adalah banyaknya pengulangan sampai akhirnya mikrokontroler mengirimkan data terakhir \$GPRMC.

$$C = (62 \times (\text{Inst.1} + \text{Inst.2} + \text{Inst.5} + \text{Inst.6} + \text{Inst.7}) + (\text{Inst.1} + \text{Inst.2} + \text{Inst.3} + \text{Inst.4}))$$

$$C = 1024,2400\mu\text{s}$$

$$\text{\$GPRMC kirim} = (26,0600 + 1024,2400 + 19,5300)\mu\text{s}$$

$$\text{\$GPRMC kirim} = 1069,8300\mu\text{s}$$

c. \$GPRMC total = \$GPRMC terima + \$GPRMC kirim

$$\text{\$GPRMC total} = (1111,0400 + 1069,8300)\mu\text{s}$$

$$\text{\$GPRMC total} = 2180,8700\mu\text{s}$$

d. *Delay* = \$GPRMC kirim - \$GPRMC terima

$$\text{Delay} = (1069,8300 - 1111,0400)\mu\text{s}$$

$$\text{Delay} = -41,2100\mu\text{s}$$

Nilai *delay* yang menunjukkan hasil minus (-) berarti, waktu yang diperlukan untuk mengirimkan data yang telah disimpan lebih cepat 41,2100 μ s. Sedangkan untuk mengambil data \$GPRMC diperlukan waktu 2180,8700 μ s

pada pengambilan pertama. Pada pengujian ini, mikrokontroler tetap mengambil data-data lain dari format NMEA-0183. Oleh karena itu waktu pengiriman data lainnya tetap diperhitungkan. Pada pengujian pertama, dapat diketahui besarnya waktu pengiriman data dari GPS, yaitu 0,1670 detik per data. Berarti ada sekitar sebelas data lagi yang dikirimkan oleh GPS ke mikrokontroler. Besarnya waktu untuk mengirimkan ke sebelas data lainnya adalah:

$$\text{Waktu total} = 11 \times 0,1670 \text{ detik} = 1,8370 \text{ detik}$$

Pada *lampiran 2* dapat dilihat bahwa program melakukan pengulangan untuk sampai menemukan karakter khusus \$GPRMC. Berarti total besarnya waktu yang digunakan untuk mengambil data \$GPRMC berikutnya adalah:

$$\text{\$GPRMC berikutnya} = \text{\$GPRMC total} + 11,9350\mu\text{s} + X$$

Nilai X pada rumus di atas adalah waktu total dari 11 data yang dikirimkan GPS.

$$\text{\$GPRMC berikutnya} = 2180,8700\mu\text{s} + 11,9350\mu\text{s} + 1,8370\text{s}$$

$$\text{\$GPRMC berikutnya} = 2180,8700\mu\text{s} + 11,9350\mu\text{s} + 1837000\mu\text{s}$$

$$\text{\$GPRMC berikutnya} = 1839192,8050\mu\text{s} = 1,8392\text{s}$$

Ini memperlihatkan bahwa kecepatan mikrokontroler dalam mengambil dan mengirimkan data \$GPRMC tiap siklusnya mendekati besar siklus data GPS sebesar 2 detik, yaitu berbeda sebesar 0,1608s. Perbedaan ini bahkan lebih kecil dari nilai waktu pengiriman data GPS yang sebesar 0,1677 detik per datanya. Rentang waktu yang sedikit ini masih mempengaruhi pengiriman data dari mikrokontroler. Seperti yang terlihat pada gambar 4.6, walau dapat mengambil data persiklus GPS, rentang waktu yang masih ada menyebabkan mikrokontroler mengirimkan *header* ATDT tanpa data \$GPRMC.

3. Pengujian ketiga bertujuan untuk mengambil data posisi dari \$GPRMC sehingga kendala pemakaian memori yang cukup besar pada pengujian pertama dan kedua dapat diatasi. Dari lampiran 3 dapat dihitung besarnya nilai mesin *cycle* dari program yang digunakan.

- a. \$GPRMC terima = \$GPRMC terima pengujian kedua

$$\text{\$GPRMC terima} = 1111,0400\mu\text{s}$$

- b. \$GPRMC kirim (Status + *latitude*) = (26,0400 + C + D + 19,5300) μs

C adalah banyaknya pengulangan pada program sampai ditemukannya nilai yang menyatakan GPS dalam keadaan aktif (karakter A). Banyaknya pengulangan adalah berjumlah 9 kali.

$$C = (8 \times (\text{Inst.1} + \text{Inst.2} + \text{Inst.6} + \text{Inst.7}) + (\text{Inst.1} + \text{Inst.2} + \text{Inst.3} + \text{Inst.4} + \text{Inst.5}))\mu\text{s}$$

$$C = (8 \times (3,2550+2,1700+1,0850+2,1700) + (3,2550+2,1700+7,5950+1,0850+2,1700))\mu\text{s}$$

$$C = 85,7150\mu\text{s}$$

D merupakan banyaknya pengulangan pada program sampai akhir data posisi *latitude*. Banyaknya pengulangan adalah sebanyak 12 kali.

$$D = (11 \times (\text{Inst.1} + \text{Inst.2} + \text{Inst.5} + \text{Inst.6} + \text{Inst.7}) + (\text{Inst.1} + \text{Inst.2} + \text{Inst.3} + \text{Inst.4}))\mu\text{s}$$

$$D = (11 \times (3,2550+2,1700+7,5950+1,0850+2,1700) + (3,2550+2,1700+7,5950+2,1700))\mu\text{s}$$

$$D = 194,2150\mu\text{s}$$

$$\text{\$GPRMC kirim (Status + latitude)} = (26,0400 + 85,7150 + 194,2150 + 19,5300)\mu\text{s}$$

$$\text{\$GPRMC kirim (Status + latitude)} = 325,5000\mu\text{s}$$

c. $\text{\$GPRMC kirim (longitude)} = (21,7000 + E + F + 19,5300)\mu\text{s}$

E merupakan banyaknya pengulangan pada program sampai akhir data posisi *longitude*. Banyaknya pengulangan adalah sebanyak 13 kali.

$$E = (12 \times (\text{Inst.1} + \text{Inst.2} + \text{Inst.6} + \text{Inst.7} + \text{Inst.8}) + (\text{Inst.1} + \text{Inst.2} + \text{Inst.3} + \text{Inst.4} + \text{Inst.5}))\mu\text{s}$$

$$E = (12 \times (3,2550+2,1700+7,5950+1,0850+2,1700) + ((3,2550+2,1700+7,5950+1,0850+2,1700))\mu\text{s}$$

$$E = 211,5750\mu\text{s}$$

F merupakan banyaknya pengulangan pada program sampai data akhir $\text{\$GPRMC}$ didapat, yaitu #13d atau <CR>. Banyaknya pengulangan adalah sebanyak 29 kali.

$$F = (28 \times (\text{Inst.1} + \text{Inst.2} + \text{Inst.5} + \text{Inst.6}) + (\text{Inst.1} + \text{Inst.2} + \text{Inst.3} + \text{Inst.4}))\mu\text{s}$$

$$F = (28 \times (3,2550+2,1700+1,0850+2,1700) +$$

$$(3,2550+2,1700+7,5950+2,1700))\mu\text{s}$$

$$F = 258,2300\mu\text{s}$$

$$\text{\$GPRMC kirim (longitude) = (21,7000 + 211,5750 + 258,2300 + 19,5300)\mu\text{s}}$$

$$\text{\$GPRMC kirim (longitude) = 511,0350\mu\text{s}}$$

- d. Total waktu pengiriman data = \\$GPRMC terima +
 \\$GPRMC kirim (Status + *latitude*) + \\$GPRMC kirim (*longitude*)

$$\text{Total waktu pengiriman data = 1111,0400\mu\text{s} + 325,5000\mu\text{s} + 511,0350\mu\text{s}}$$

$$\text{Total waktu pengiriman data = 1947,5700\mu\text{s}}$$

Besarnya waktu pengiriman tersebut adalah untuk data \\$GPRMC saja. Sama halnya dengan pengujian kedua, sebelas data yang lain juga dikirimkan. Dengan demikian, total waktu untuk pengiriman data posisi untuk satu siklus pada mikrokontroler adalah:

- e. Data posisi berikutnya = waktu total awal + 11,9350 μs + X

Nilai X pada rumus di atas adalah waktu total dari 11 data yang dikirimkan GPS.

$$\text{Data posisi berikutnya = 1947,5700\mu\text{s} + 11,9350 \mu\text{s} + 1,8370\text{s}}$$

$$\text{Data posisi berikutnya = 1947,5700\mu\text{s} + 11,9350 \mu\text{s} + 1837000\mu\text{s}}$$

$$\text{Data posisi berikutnya = 1838959,5050\mu\text{s} = 1,8389\text{s}}$$

Ini memperlihatkan bahwa kecepatan mikrokontroler dalam mengambil dan mengirimkan data posisi pada \\$GPRMC tiap siklusnya mendekati besar siklus data GPS sebesar 2 detik, yaitu berbeda sebesar 0,1611s. Perbedaan ini bahkan lebih kecil dari nilai waktu pengiriman data GPS yang sebesar 0,1677 detik per datanya. Gambar 4.7 memperlihatkan tidak adanya data yang tidak dikirimkan pada tiap siklus pengiriman GPS.

4. Pengujian keempat bertujuan untuk menyiapkan data untuk dikirimkan ke satelit ACeS Garuda 1 yaitu dengan menggunakan metode penomoran fiktif. Data yang dikirim nantinya dianalogikan sebagai nomor tujuan yang akan dihubungi. Keuntungan ini membuat satelit Garuda 1 tetap menerima data yang telah diwakili oleh angka. Sebelum itu haruslah diketahui parameter seperti apa saja yang harus dipenuhi sebelum melakukan pengiriman ke satelit melewati perangkat PASTI/Byru Marine.

Berdasarkan pembahasan pada Bab 2 dan Bab 3 mengenai metode pengiriman dan format data pengiriman, kita dapat mengambil inti-inti dari parameter yang harus dipenuhi, antara lain:

- a. Panjang nominal angka adalah 16 digit yang disisipkan *header* ATDT di depannya.
- b. Kecepatan transmisi sebesar 2400 bps.
- c. Perangkat PASTI/*Byru Marine* akan menampilkan kalimat *DATA MODE* apabila pengiriman data sukses.
- d. Pada metode penomoran fiktif, perangkat PASTI/*Byru Marine* akan *restart* sistem apabila *server* ACeS tidak mengenali nomor yang dikirimkan.
- e. Lamanya waktu *restart* sistem adalah selama 30 detik.

Berdasarkan parameter di atas, diketahui bahwa mikrokontroler harus mengirimkan data sebanyak maksimal 20 karakter dengan tambahan *delay* waktu mendekati 30 detik sebelum melakukan pengambilan dan pengiriman data kembali. Pemberian *delay* bertujuan untuk mengatasi kekurangan dari metode penomoran fiktif tersebut. Lama waktu yang terpakai pada saat pengambilan dan pengiriman data oleh mikrokontroler, tanpa *delay* menjadi tidak berpengaruh, besarnya waktu yang digunakan untuk pengambilan data posisi dalam bentuk angka oleh mikrokontroler, tidak akan jauh berbeda dengan waktu yang didapat dari pengujian 1, 2, dan 3. Dapat dipastikan waktu total yang didapat tanpa diberikan *delay* tidak melebihi waktu 2 detik. Akan tetapi, dengan metode penomoran fiktif ini, sistem akan *restart* perangkat PASTI/*Byru Marine* selama 30 detik. Untuk itu diperlukan *delay* waktu yang mendekati 30 detik. Berikut program subrutin yang dapat ditambahkan untuk menghasilkan *delay* waktu mendekati 30 detik pada mikrokontroler:

```
Delay1:  DJNZ      R2,$
          DJNZ      R3,Delay1
          DJNZ      R4,Delay1
          Ret
```

Nilai awal dari register R2, R3, dan R4 adalah 0, sedangkan instruksi DJNZ bernilai 2 mesin *cycle* atau setara dengan $2,1700\mu\text{s}$. Simbol \$ berarti program akan lanjut ke baris berikutnya bila bernilai 0 setelah mengalami pengurangan. Baris pertama program waktunya adalah sebesar $= 2,1700\mu\text{s} \times 256 = 0,5555\text{ms}$. Baris kedua program waktunya adalah sebesar $= 256 \times 0,5555\text{ms} = 0,1422\text{s}$. Baris ketiga program waktunya adalah sebesar $= 256 \times 0,1422\text{s} = 36,4032\text{s}$.

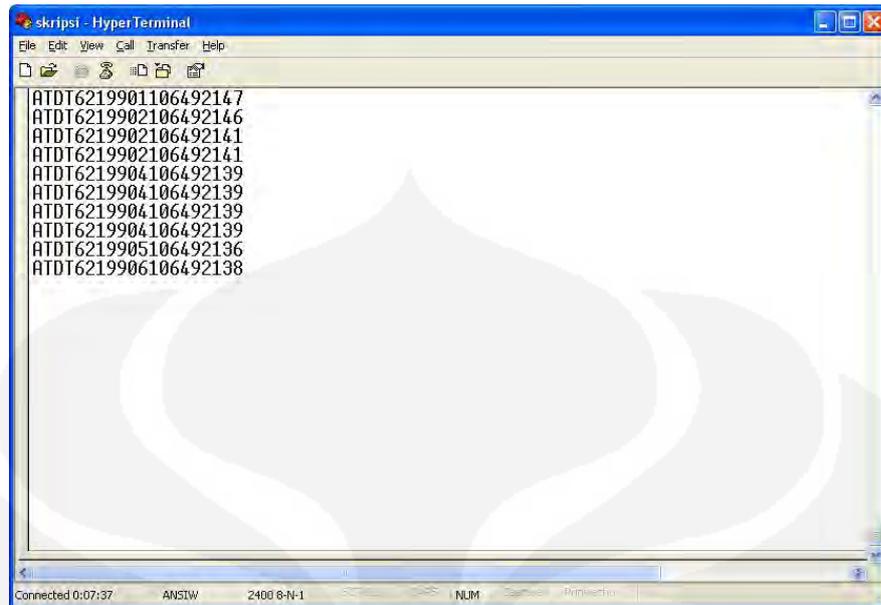
Jadi total waktu yang dapat dimanfaatkan untuk *delay* adalah selama 36,4032 detik. Setelah selama 36,4032 detik mikrokontroler akan mengambil dan mengirimkan data baru. Gambar 4.8 memperlihatkan pengiriman data dua kali, yaitu *latitude* dan setelah 36,4032 detik data *longitude* dikirim.

Cara pembacaan posisinya tetap sama dengan yang sebelumnya, misalnya:

06219921 dibaca *latitude* $06^{\circ} 21,9921' \text{S}$

106492160 dibaca *longitude* $106^{\circ} 49,2160' \text{E}$

Berarti pengiriman data posisi \$GPRMC untuk satu siklus memerlukan waktu 72,8064 detik atau 1 menit 12,8064 detik dan hampir senilai pengiriman 37 kali data mentah GPS. Untuk mengatasi penggunaan waktu berlebihan ini maka data posisi *latitude* dan *longitude* harus dikirimkan bersamaan. Dari Gambar 4.8 diketahui jumlah karakter *latitude* dan *longitude* tanpa disisipkan *header* ATDT berjumlah 17 karakter, sehingga tidak mencukupi kuota maksimal yang ditetapkan. Apabila dicermati data pada gambar 4.8, kita dapat menghilangkan satu nominal angka paling depan pada posisi *latitude*. Ini tidaklah berpengaruh, pertama karena nominal paling depan adalah 0 dan kedua posisi bujur hanya berkisar pada derajat dengan nominal 2 angka. Sebagai contoh, nilai *latitude* yang semula 06219922 dapat diubah menjadi 6219922. Cara ini akan menghemat sekitar 36,4032 detik atau hampir senilai pengiriman 19 kali data mentah GPS.



Gambar 4.10 Pengiriman data *latitude* dan *longitude* dengan sekali pengiriman

Dari gambar 4.9 dapat dilihat data posisi yang digabungkan kondisinya memenuhi parameter yang telah ada.

ATDT6219914106492157 merupakan contoh data yang siap dikirimkan, data ini terdiri dari:

1. 4 karakter ATDT.
2. 7 angka setelah *header* ATDT yang mewakili posisi *latitude* (6219914).
3. 9 angka setelah angka *latitude* merupakan angka yang mewakili posisi *longitude* (106492157).

4.3.3 Analisis Pengujian Ketiga

Pada pengujian ketiga diketahui bahwasannya data yang terdapat pada *server* hanya berupa nomor saja. *Header* ATDT yang disisipkan pada pengiriman data ke satelit Garuda 1 hanya berfungsi sebagai perintah pengiriman saja.

Pada gambar 4.9, data GPS yang ditampilkan di *server* adalah sebagai berikut.

```
6219898106492151
6219898106492151
6219898106492150
6219898106492149
6219898106492149
6219896106492147
6219887106492144
6219878106492128
```

6219867106492115
6219864106492092
6219862106492069
6219858106492065
6219856106492065
6219856106492060
6219857106492058
6219859106492055
6219861106492060
6219870106492062
6219881106492074
6219893106492087
6219899106492101
6219901106492107
6219901106492115
6219902106492121
6219902106492125
6219902106492128
6219901106492130
6219901106492132
6219901106492136
6219902106492138
6219903106492140
6219904106492140
6219903106492140
6219903106492142
6219903106492144

Data yang berupa angka di atas mewakili data *longitude* dan *latitude* dari posisi GPS. Terdapat enam belas digit angka, yang terdiri dari tujuh digit angka yang mewakili posisi *latitude* dan sembilan digit angka yang mewakili posisi *longitude*. Misal, data *6219898106492151* pada baris pertama dapat diartikan sebagai berikut:

1. Tujuh digit di awal, yaitu *6219898*, menunjukkan data posisi *latitude*. Karena pada pengujian, Depok terletak pada bagian selatan garis khatulistiwa, maka *6219898* diartikan sebagai berikut:
 $6^{\circ} 21,9898'$ Selatan atau dengan menggunakan langkah 1 dan langkah 2 seperti pada bagian 4.2.2.3, didapatkan $-6,3665^{\circ}$ lintang selatan. Tanda minus (-) diberikan, karena kota Depok terletak pada garis bawah khatulistiwa.
2. Sembilan digit setelah data *latitude*, menunjukkan data posisi *longitude*. Data *106492151* dapat diartikan sebagai berikut:

106° 49,2151' Timur atau dengan menggunakan langkah yang sama seperti pada bagian 4.2.2.3, didapatkan 106,8203⁰ bujur timur.



BAB V

KESIMPULAN

Berdasarkan hasil pengujian dan analisis sistem pada Bab 4, dapat diambil beberapa kesimpulan sebagai berikut:

1. Sesuai dengan tujuan dari tugas akhir, kami berhasil merancang dan membuat sistem pengiriman data lokasi berbiaya murah menggunakan perangkat mikrokontroler sebagai penerima, pengolah, dan pengirim data GPS, melalui perangkat PASTI/*Byru Marine* memanfaatkan pelayanan pensinyalan satelit Garuda 1.
2. Mikrokontroler tidak dapat mengambil semua jenis data, jika difungsikan untuk melalui data mentah GPS karena jarak waktu antara total waktu pengiriman data oleh mikrokontroler, jauh lebih kecil daripada satu siklus pengiriman data mentah oleh GPS, yaitu 0,0183 detik dan 2 detik.
3. Untuk mengatasi hal sebagaimana dinyatakan dalam kesimpulan kedua di atas, data mentah GPS harus *dimasking* sehingga data yang dikirimkan hanya GPRMC. Dengan cara ini lama waktu yang diperlukan mikrokontroler tiap siklusnya mendekati 2 detik, yaitu 1,8392s.
4. Untuk mengatasi lamanya waktu pengiriman pada pengujian keempat, data *latitude* dan *longitude* diatur sedemikian rupa sehingga dikirimkan secara bersamaan. Ini akan menghemat waktu kirim selama 36,4032 detik atau sama dengan pengiriman 19 kali data mentah oleh GPS.
5. Data yang terdapat pada *server* dapat diakses melalui perangkat lunak FTP *client*. Data tersebut berupa angka yang mewakili posisi *longitude* dan *latitude* GPS.

DAFTAR ACUAN

- [1] Kenneth J Ayala, *The 8051 Microcontroller Architecture, Programming, and Application 2nd Edition* (New York: West Publishing Company, 1997), hal. 55
- [2] Kenneth J Ayala, *The 8051 Microcontroller Architecture, Programming, and Application 2nd Edition* (New York: West Publishing Company, 1997), hal. 58
- [3] Kenneth J Ayala, *The 8051 Microcontroller Architecture, Programming, and Application 2nd Edition* (New York: West Publishing Company, 1997), hal. 63
- [4] Kenneth J Ayala, *The 8051 Microcontroller Architecture, Programming, and Application 2nd Edition* (New York: West Publishing Company, 1997), hal. 63
- [5] Kenneth J Ayala, *The 8051 Microcontroller Architecture, Programming, and Application 2nd Edition* (New York: West Publishing Company, 1997), hal. 63
- [6] Taryudi. “Rancang Bangun Tampilan Data GPS pada Peta GIS dan Data Eksternal pada Monitor PC Berbasis Mikrokontroler”. Skripsi, Program Sarjana Fakultas Teknik UI, Depok, 2007, hal. 22
- [7] Taryudi. “Rancang Bangun Tampilan Data GPS pada Peta GIS dan Data Eksternal pada Monitor PC Berbasis Mikrokontroler”. Skripsi, Program Sarjana Fakultas Teknik UI, Depok, 2007, hal. 80
- [8] Taryudi. “Rancang Bangun Tampilan Data GPS pada Peta GIS dan Data Eksternal pada Monitor PC Berbasis Mikrokontroler”. Skripsi, Program Sarjana Fakultas Teknik UI, Depok, 2007, hal. 26
- [9] Jean Marie Zogg (2007). *GPS Basics*. Diakses 8 April 2008 dari microblox. <http://www.u-blox.com>
- [10] Jean Marie Zogg (2007). *GPS Basics*. Diakses 8 April 2008 dari microblox. <http://www.u-blox.com>
- [11] Jean Marie Zogg (2007). *GPS Basics*. Diakses 8 April 2008 dari microblox. <http://www.u-blox.com>
- [12] Jean Marie Zogg (2007). *GPS Basics*. Diakses 8 April 2008 dari microblox. <http://www.u-blox.com>
- [13] Jean Marie Zogg (2007). *GPS Basics*. Diakses 8 April 2008 dari microblox. <http://www.u-blox.com>
- [14] Jean Marie Zogg (2007). *GPS Basics*. Diakses 8 April 2008 dari microblox. <http://www.u-blox.com>

- [15] Jean Marie Zogg (2007). *GPS Basics*. Diakses 8 April 2008 dari microblox. <http://www.u-blox.com>
- [16] Jean Marie Zogg (2007). *GPS Basics*. Diakses 8 April 2008 dari microblox. <http://www.u-blox.com>
- [17] Jean Marie Zogg (2007). *GPS Basics*. Diakses 8 April 2008 dari microblox. <http://www.u-blox.com>
- [18] Jean Marie Zogg (2007). *GPS Basics*. Diakses 8 April 2008 dari microblox. <http://www.u-blox.com>
- [19] Taryudi. “Rancang Bangun Tampilan Data GPS pada Peta GIS dan Data Eksternal pada Monitor PC Berbasis Mikrokontroler”. Skripsi, Program Sarjana Fakultas Teknik UI, Depok, 2007, hal. 33.
- [20] Lee Wismer. *ACeS System Overview*(lockheed martin, 2000), hal.21.
- [21] Rahmat Rafiudin, *Membangun Server FTP* (Yogyakarta: Andi, 2005), hal. 6
- [22] I Made Wiryana (2007). *File Transfer Protocol*. Diakses 13 Mei 2008 dari google. <http://www.google.com/FTP>
- [23] I Made Wiryana (2007). *File Transfer Protocol*. Diakses 13 Mei 2008 dari google. <http://www.google.com/FTP>
- [24] Peter, Joe (2007). *NMEA Data*. Diakses 6 Mei 2008 dari National Marine Electronics Association. www.nmea.org
- [25] Taryudi. “Rancang Bangun Tampilan Data GPS pada Peta GIS dan Data Eksternal pada Monitor PC Berbasis Mikrokontroler”. Skripsi, Program Sarjana Fakultas Teknik UI, Depok, 2007, hal. 41.
- [26] Kenneth J Ayala, *The 8051 Microcontroller Architecture, Programming, and Application 2nd Edition* (New York: West Publishing Company, 1997), hal. 60.

DAFTAR PUSTAKA

Ayala, Kenneth J, *The 8051 Microcontroller Architecture, Programming, and Application 2nd Edition* (New York: West Publishing Company, 1997)

Peter, Joe. *NMEA Data*. Diakses 6 Mei 2008 dari National Marine Electronics Association.
www.nmea.org

Rafiudin Rahmat, *Membangun Server FTP* (Yogyakarta: Andi, 2005)

Taryudi. “ Rancang Bangun Tampilan Data GPS pada Peta GIS dan Data Eksternal pada Monitor PC Berbasis Mikrokontroler”. Skripsi , Program Sarjana Fakultas Teknik UI, Depok, 2007.

Wiryana, I Made (2007). *File Transfer Protocol*. Diakses 13 Mei 2008 dari google.
<http://www.google.com/FTP>

Wismer, Lee. *ACeS System Overview* (lockheed martin, 2000)

Zogg , Jean Marie (2007). *GPS Basics*. Diakses 8 April 2008 dari microblox.
<http://www.u-blox.com>

LAMPIRAN 1. Program Pengolahan Data GPS tanpa Masking

		Mesin Cycle	Waktu(μ s)	Total Waktu(μ s)
kirim:	mov	sbuf,A	2	2,1700
	jnb	ti,\$	2	2,1700
	clr	ti	1	1,0850
	ret		2	2,1700
terima:	clr	ri	1	1,0850
	mov	A,sbuf	2	2,1700
	jnb	ri,\$	2	2,1700
	ret		2	2,1700
simpan:	mov	@r0,A	1	1,0850
	ret		2	2,1700
ambil:	mov	A,@r0	1	1,0850
	ret		2	2,1700
header:	mov	A,'#A'	1	1,0850
	acall	kirim		7,5950
	mov	A,'#T'	1	1,0850
	acall	kirim		7,5950
	mov	A,'#D'	1	1,0850
	acall	kirim		7,5950
	mov	A,'#T'	1	1,0850
	acall	kirim		7,5950
	ret		2	2,1700
baudK:	mov	TMOD,#20h	1	1,0850
	mov	TH1,#0F4h	1	1,0850
	mov	SCON,#50h	1	1,0850
	mov	PCON,#00h	1	1,0850
	setb	TR1	1	1,0850
	setb	ren	1	1,0850
	ret		2	2,1700
baudT:	mov	TMOD,#20h	1	1,0850
	mov	TH1,#0FAh	1	1,0850
	mov	SCON,#50h	1	1,0850
	mov	PCON,#00h	1	1,0850
	setb	TR1	1	1,0850
	setb	ren	1	1,0850
	ret		2	2,1700
mulai:	mov	sp,#20h	1	1,0850
yyy:	mov	r0,#35h	1	1,0850
	acall	baudT		9,7650
xxx:	acall	terima		7,5950
	cjne	A,'#\$',xxx	2	2,1700
	ajmp	aaa	2	2,1700

			Mesin Cycle	Waktu (μ s)	Total Waktu (μ s)
aaa:	acall	terima		7,5950	1
	cjne	A,#10,bbb	2	2,1700	2
	mov	r0,A	1	1,0850	3
	ajmp	ccc	2	2,1700	4
bbb:	acall	simpan		3,2550	5
	inc	r0	1	1,0850	6
	ajmp	aaa	2	2,1700	7
ccc:	mov	r0,#35h	1	1,0850	26,0400
	acall	baudK		9,7650	
	acall	header		15,1900	
zzz:	acall	ambil		3,2550	1
	cjne	A,#13,ddd	2	2,1700	2
	acall	kirim		7,5950	3
	ajmp	eee	2	2,1700	4
ddd:	acall	kirim		7,5950	5
	inc	r0	1	1,0850	6
	ajmp	zzz	2	2,1700	7
eee:	mov	A,#13	1	1,0850	19,5300
	acall	kirim		7,5950	
	mov	A,#10	1	1,0850	
	acall	kirim		7,5950	
	ajmp	yyy	2	2,1700	
	end				

LAMPIRAN 2. Program Pengolahan Data GPS dengan *Masking* \$GPRMC

		Mesin	Cycle	Waktu (µs)	Total Waktu (µs)
kirim:	mov	sbuf,A	2	2,1700	7,5950
	jnb	ti,\$	2	2,1700	
	clr	ti	1	1,0850	
	ret		2	2,1700	
terima:	clr	ri	1	1,0850	7,5950
	mov	A,sbuf	2	2,1700	
	jnb	ri,\$	2	2,1700	
	ret		2	2,1700	
simpan:	mov	@r0,A	1	1,0850	3,2550
	ret		2	2,1700	
ambil:	mov	A,@r0	1	1,0850	3,2550
	ret		2	2,1700	
header:	mov	A,'#A'	1	1,0850	36,8900
	acall	kirim		7,5950	
	mov	A,'#T'	1	1,0850	
	acall	kirim		7,5950	
	mov	A,'#D'	1	1,0850	
	acall	kirim		7,5950	
	mov	A,'#T'	1	1,0850	
	acall	kirim		7,5950	
	ret		2	2,1700	
	baudK:	mov	TMOD,#20h	1	
mov		TH1,#0F4h	1	1,0850	
mov		SCON,#50h	1	1,0850	
mov		PCON,#00h	1	1,0850	
setb		TR1	1	1,0850	
setb		ren	1	1,0850	
ret			2	2,1700	
baudT:		mov	TMOD,#20h	1	1,0850
	mov	TH1,#0FAh	1	1,0850	
	mov	SCON,#50h	1	1,0850	
	mov	PCON,#00h	1	1,0850	
	setb	TR1	1	1,0850	
	setb	ren	1	1,0850	
	ret		2	2,1700	
	mulai:	mov	sp,#20h	1	1,0850
yyy:		mov	r0,#35h	1	1,0850
xxx:	acall	baudT		9,7650	1
	acall	terima		7,5950	
	cjne	A,'#C',xxx	2	2,1700	
	ajmp	aaa	2	2,1700	
				3	

			Mesin Cycle	Waktu (μs)	Total Waktu (μs)
aaa:	acall	terima		7,5950	<div style="border: 1px solid black; padding: 5px;"> 1 2 3 4 B 5 6 7 </div>
	cjne	A,#10,bbb	2	2,1700	
	mov	r0,A	1	1,0850	
	ajmp	ccc	2	2,1700	
bbb:	acall	simpan		3,2550	
	inc	r0	1	1,0850	
	ajmp	aaa	2	2,1700	
ccc:	mov	r0,#35h	1	1,0850	<div style="border: 1px solid black; padding: 5px; text-align: center;"> 26,040 </div>
	acall	baudK		9,7650	
	acall	header		15,1900	
zzz:	acall	ambil		3,2550	<div style="border: 1px solid black; padding: 5px;"> 1 2 3 4 C 5 6 7 </div>
	cjne	A,#13,ddd	2	2,1700	
	acall	kirim		7,5950	
	ajmp	eee	2	2,1700	
ddd:	acall	kirim		7,5950	
	inc	r0	1	1,0850	
	ajmp	zzz	2	2,1700	
eee:	mov	A,#13	1	1,0850	<div style="border: 1px solid black; padding: 5px; text-align: center;"> 19,5300 </div>
	acall	kirim		7,5950	
	mov	A,#10	1	1,0850	
	acall	kirim		7,5950	
	ajmp	yyy	2	2,1700	
	end				

**LAMPIRAN 3. Program Pengolahan Data Posisi GPS dengan Masking
\$GPRMC**

		Mesin Cycle	Waktu (µs)	Total Waktu (µs)		
kirim:	mov	sbuf,A	2	2,1700	7,5950	
	jnb	ti,\$	2	2,1700		
	clr	ti	1	1,0850		
	ret		2	2,1700		
terima:	clr	ri	1	1,0850	7,5950	
	mov	A,sbuf	2	2,1700		
	jnb	ri,\$	2	2,1700		
	ret		2	2,1700		
simpan:	mov	@r0,A	1	1,0850	3,2550	
	ret		2	2,1700		
ambil:	mov	A,@r0	1	1,0850	3,2550	
	ret		2	2,1700		
header:	mov	A,#'A'	1	1,0850	36,8900	
	acall	kirim		7,5950		
	mov	A,#'T'	1	1,0850		
	acall	kirim		7,5950		
	mov	A,#'D'	1	1,0850		
	acall	kirim		7,5950		
	mov	A,#'T'	1	1,0850		
	acall	kirim		7,5950		
	ret		2	2,1700		
baudK:	mov	TMOD,#20h	1	1,0850		
	mov	TH1,#0F4h	1	1,0850	9,7650	
	mov	SCON,#50h	1	1,0850		
	mov	PCON,#00h	1	1,0850		
	setb	TR1	1	1,0850		
	setb	ren	1	1,0850		
	ret		2	2,1700		
baudT:	mov	TMOD,#20h	1	1,0850		
	mov	TH1,#0FAh	1	1,0850		
	mov	SCON,#50h	1	1,0850	9,7650	
	mov	PCON,#00h	1	1,0850		
	setb	TR1	1	1,0850		
	setb	ren	1	1,0850		
	ret		2	2,1700		
mulai:	mov	sp,#20h	1	1,0850		
yyy:	mov	r0,#35h	1	1,0850		11,9350
	acall	baudT		9,7650		
xxx:	acall	terima		7,5950	1 2 A 3	
	cjne	A,#'C',xxx	2	2,1700		
	ajmp	aaa	2	2,1700		

			Mesin Cycle	Waktu (µs)	Total Waktu (µs)
aaa:	acall	terima		7,5950	1
	cjne	A,#10,bbb	2	2,1700	2
	mov	r0,A	1	1,0850	3
	ajmp	ccc	2	2,1700	4
bbb:	acall	simpan		3,2550	5
	inc	r0	1	1,0850	6
	ajmp	aaa	2	2,1700	7
ccc:	mov	r0,#35h	1	1,0850	
	acall	baudK		9,7650	26,0400
	acall	header		15,1900	
zzz:	acall	ambil		3,2550	1
	cjne	A,#'A',mmm	2	2,1700	2
	acall	kirim		7,5950	3
	inc	r0	1	1,0850	4
	ajmp	ttt	2	2,1700	5
mmm:	inc	r0	1	1,0850	6
	ajmp	zzz	2	2,1700	7
ttt:	acall	ambil		3,2550	1
	cjne	A,#'S',ddd	2	2,1700	2
	acall	kirim		7,5950	3
	ajmp	sss	2	2,1700	4
ddd:	acall	kirim		7,5950	5
	inc	r0	1	1,0850	6
	ajmp	ttt	2	2,1700	7
sss:	mov	A,#13	1	1,0850	
	acall	kirim		7,5950	19,5300
	mov	A,#10	1	1,0850	
	acall	kirim		7,5950	
	ajmp	uuu	2	2,1700	
uuu:	acall	ambil		3,2550	21,7000
	cjne	A,#'S',uuu	2	2,1700	
	inc	r0	1	1,0850	
	acall	header		15,1900	
www:	acall	ambil		3,2550	1
	cjne	A,#'E',eee	2	2,1700	2
	acall	kirim		7,5950	3
	inc	r0	1	1,0850	4
	ajmp	fff	2	2,1700	5
eee:	acall	kirim		7,5950	6
	inc	r0	1	1,0850	7
	ajmp	www	2	2,1700	8

			Mesin Cycle	Waktu (μ s)	Total Waktu (μ s)
fff:	acall	ambil		3,2550	1
	cjne	A,#13,ggg	2	2,1700	2
	acall	kirim		7,5950	3
	ajmp	nnn	2	2,1700	4 F
ggg:	inc	r0	1	1,0850	5
	ajmp	fff	2	2,1700	6
nnn:	mov	A,#13	1	1,0850	
	acall	kirim		7,5950	
	mov	A,#10	1	1,0850	
	acall	kirim		7,5950	
	ajmp	yyy	2	2,1700	
	end				
					19,5300