



**UNIVERSITAS INDONESIA**

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

**SKRIPSI**

**GUSNI HARIYANTO  
04 04 03 0415**

**FAKULTAS TEKNIK ELEKTRO  
PROGRAM STUDI TEKNIK ELEKTRO  
DEPOK  
DESEMBER 2008**



**UNIVERSITAS INDONESIA**

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

**SKRIPSI**

**Diajukan sebagai salah satu syarat untuk memperoleh gelar  
Sarjana Teknik Elektro**

**GUSNI HARIYANTO  
04 04 03 0415**

**FAKULTAS TEKNIK ELEKTRO  
PROGRAM STUDI TEKNIK ELEKTRO  
DEPOK  
DESEMBER 2008**

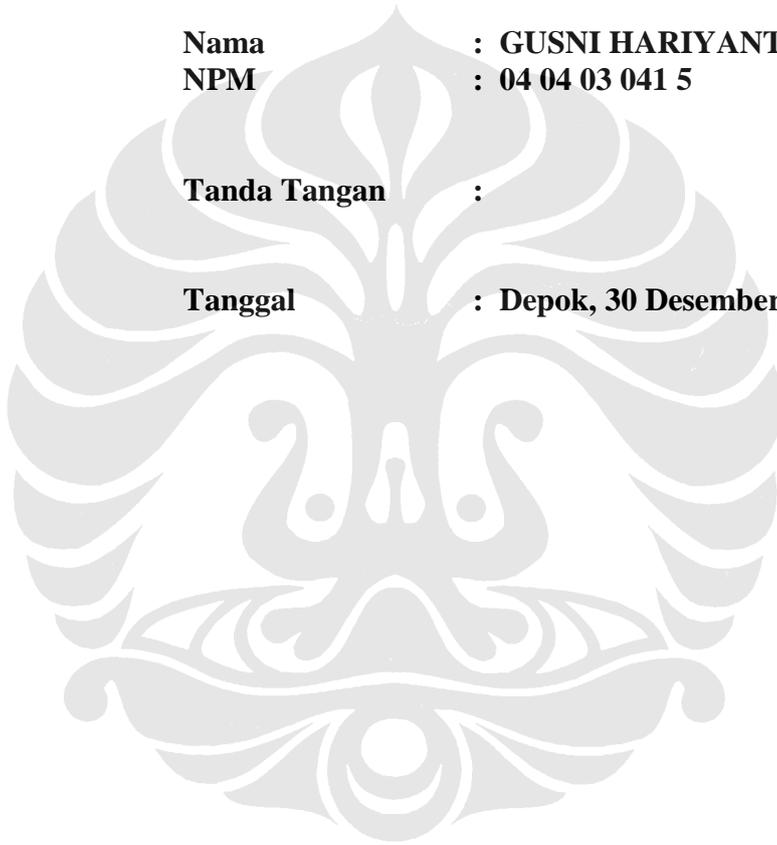
## HALAMAN PERNYATAAN ORISINALITAS

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

**Nama : GUSNI HARIYANTO**  
**NPM : 04 04 03 041 5**

**Tanda Tangan :**

**Tanggal : Depok, 30 Desember 2008**



## HALAMAN PENGESAHAN

Tugas akhir ini diajukan oleh :

Nama : GUSNI HARIYANTO  
NPM : 04 04 03 0415  
Judul Skripsi :  
Tugas akhir dengan judul : RANCANG BANGUN PENGIRIMAN DAN  
PENERIMAAN DATA MELALUI  
PELAYANAN PENSINYALAN SATELIT  
GARUDA ANTAR ALAT KOMUNIKASI  
BERGERAK

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

### DEWAN PENGUJI

Pembimbing 1 : Dr. Ir. Arman D. Diponegoro (.....)

Penguji : Ir. Rochmah N Sukardi Ny, MSc (.....)

Penguji : Dr. Ir. Kalamullah Ramli, M.Eng (.....)

Ditetapkan di : Depok

Tanggal : 30 Desember 2008

## KATA PENGANTAR/UCAPAN TERIMA KASIH

Puji syukur saya panjatkan kepada Tuhan Yang Maha Esa, karena atas berkat dan rahmat-Nya, saya dapat menyelesaikan tugas akhir ini. Penulisan tugas akhir ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Sarjana Teknik Jurusan Teknik Elektro pada Fakultas Teknik Universitas Indonesia. Saya menyadari bahwa, tanpa bantuan dan bimbingan dari berbagai pihak, dari masa perkuliahan sampai pada penyusunan tugas akhir ini, sangatlah sulit bagi saya untuk menyelesaikan tugas akhir ini. Oleh karena itu, saya mengucapkan terima kasih kepada:

- (1) Dr. Ir. Arman D. Diponegoro, selaku dosen pembimbing 1 yang telah menyediakan waktu, tenaga, dan pikiran untuk mengarahkan saya dalam penyusunan tugas akhir ini;
- (2) pihak PT.PSN yang terutama Pa kukun telah banyak membantu dalam usaha memperoleh data dan peralatan yang saya perlukan;
- (3) orang tua saya, Ibunda Mugiriyanti, Ibu yang mendoakan siang dan malam , Ayahanda Tamril Ayah bekerja keras demi member nafkah pada keluarga agar aku dapat berhasil,sehingga aku tak akan bisa membalasnya dengan apapun;
- (4) Risma,Taofik dan Iqbal Iskandar, teman dalam ikatan persaudaraan. Selalu membantu dikala kesulitan dalam skripsi beliau banyak berjasa dalam tenaga dan pikiran dalam penyelesaian tugas akhir ini,sehingga dapat selesai.
- (5) Sahabat-sahabat kampus.Anggi,Rofan yang bersedia meminjamkan alat Dwi Rilli,Mardianyang baik dalam meminjamkan laptop ,Selly,Yunanto,Akhmad Fauzi,M Ghazali,Yunanto, dan semua yang tidak dapat kusebutkan, terima kasih telah terlibat dalam jalan maupun ide yang kuambil;
- (9) Sahabat-sahabat BBA99.yang selalu mendoakan aku terutama pa Awi dan Marwan yang tak pernah lupa mendoakan aku
- (10) semua pihak yang telah membantu dalam pembuatan dan penyusunan tugas akhir ini.

Akhir kata, saya berharap Tuhan Yang Maha Esa berkenan membalas segala kebaikan semua pihak yang telah membantu. Semoga tugas akhir ini membawa manfaat bagi pengembangan ilmu.

Depok, 30 Desember 2008

Penulis

**GUSNI HARIYANTO**



## HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS

---

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

Nama : GUSNI HARIYANTO  
NPM : 04 04 03 0415  
Program Studi : Teknik Elektro  
Departemen : Teknik Elektro  
Fakultas : Teknik  
Jenis karya : Tugas Akhir

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

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

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

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok  
Pada Tanggal : 30 Desember 2008  
Yang menyatakan

(GUSNI HARIYANTO)

## ABSTRAK

Nama : GUSNI HARIYANTO

Program Studi : Teknik Elektro

Judul : RANCANG BANGUN PENGIRIMAN DAN PENERIMAAN  
DATA MELALUI PELAYANAN PENSINYALAN SATELIT  
GARUDA ANTAR ALAT KOMUNIKASI BERGERAK

Tugas akhir ini merancang sistem pengambilan dan pengiriman data GPS dengan menggunakan mikrokontroler sebagai pengolah data GPS melalui komunikasi satelit mobile Garuda. 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 kemudian dikirim lagi ke perangkat PASTI/byru marine yang lain melalui satelit ACeS Garuda 1 dengan menggunakan penomoran khusus data tanpa bayar.

Informasi yang terdiri dari data posisi yang terdapat pada server stasiun Bumi, dapat diakses oleh perangkat PASTI/byru marine. Dari hasil uji coba diperoleh data GPS berhasil dilakukan dengan kecepatan 2400 bps.

Kata Kunci: GPS, NMEA 0183, Satelit ACeS Garuda 1, Mikrokontroler, Perangkat PASTI/Byru Marine.

## ABSTRACT

Name : GUSNI HARIYANTO  
Study Program: Electrical Engineering  
Title : DESIGNS AND BUILDS DATA TRANSMIT AND  
ACCEPTANCE THROUGH SIGNALLING SERVICES OF  
GARUDA WITH MOBILE COMMUNICATION DEVICES

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 and transmitted again to another PASTI/ Byru Marine through ACeS Garuda 1 satellite use free payment number..

The information consists position data at earth station server, can be access by PASTI/byru marine.From research,GPS data successfully with speed 2400 bps.

Keyword: GPS, NMEA 0183, ACeS Garuda 1 Satellite, Microcontroller, PASTI  
Byru Marine Terminal

# DAFTAR ISI

	Halaman
HALAMAN JUDUL .....	i
HALAMAN PERNYATAAN ORISINALITAS .....	ii
LEMBAR PENGESAHAN .....	iii
KATA PENGANTAR .....	iv
LEMBAR PERSETUJUAN PUBLIKASI KARYA ILMIAH .....	vi
ABSTRAK .....	vii
ABSTRACT .....	viii
DAFTAR ISI .....	ix
DAFTAR GAMBAR .....	xii
DAFTAR TABEL .....	xiv
DAFTAR LAMPIRAN .....	xv
DAFTAR SINGKATAN .....	xvi
<b>BAB I PENDAHULUAN .....</b>	<b>1</b>
1.1 LATAR BELAKANG .....	1
1.2 PERUMUSAN MASALAH .....	2
1.3 TUJUAN .....	2
1.4 BATASAN MASALAH .....	2
1.5 SISTEMATIKA PENULISAN .....	3
<b>BAB II LANDASAN TEORI .....</b>	<b>4</b>
2.1 MIKROKONTROLER AT89S51 .....	4
2.1.1 Arsitektur AT89S51 .....	4
2.1.2 Fitur AT89S51 .....	6
2.1.3 Konfigurasi Pin AT89S51 .....	6
2.1.4 Peta Memori .....	8
2.1.5 <i>Counter</i> dan <i>Timer</i> .....	12
2.1.6 Masukan / Keluaran Data Serial .....	16
2.1.7 Komunikasi Serial AT89S51 .....	17
2.1.7.1 <i>Sistem Komunikasi Serial</i> .....	17
2.1.7.2 <i>Protokol Serial</i> .....	18

2.1.7.3	<i>Inisialisasi UART</i> .....	18
2.1.7.4	<i>Pengiriman Data</i> .....	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 Angka (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
<b>BAB III</b>	<b>RANCANG BANGUN</b> .....	<b>35</b>
3.1	PRINSIP KERJA SISTEM .....	35
3.1.1	Blok Diagram dan Fungsinya .....	35
3.2	PERANGKAT KERAS .....	36
3.3	PERANGKAT LUNAK .....	37
3.3.1	Pemrograman Mikrokontroler .....	38
3.4	FORMAT DATA .....	40
3.4.1	Format <i>Serial Data Output</i> (SDO) .....	41
3.4.2	Format Pengiriman Data .....	42
<b>BAB IV</b>	<b>PENGUJIAN DAN ANALISIS SISTEM</b> .....	<b>44</b>
4.1	HASIL RANCANG BANGUN .....	44
4.2	PENGUJIAN SISTEM .....	44
4.2.1	Menampilkan Data Keluaran GPS .....	45



## 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

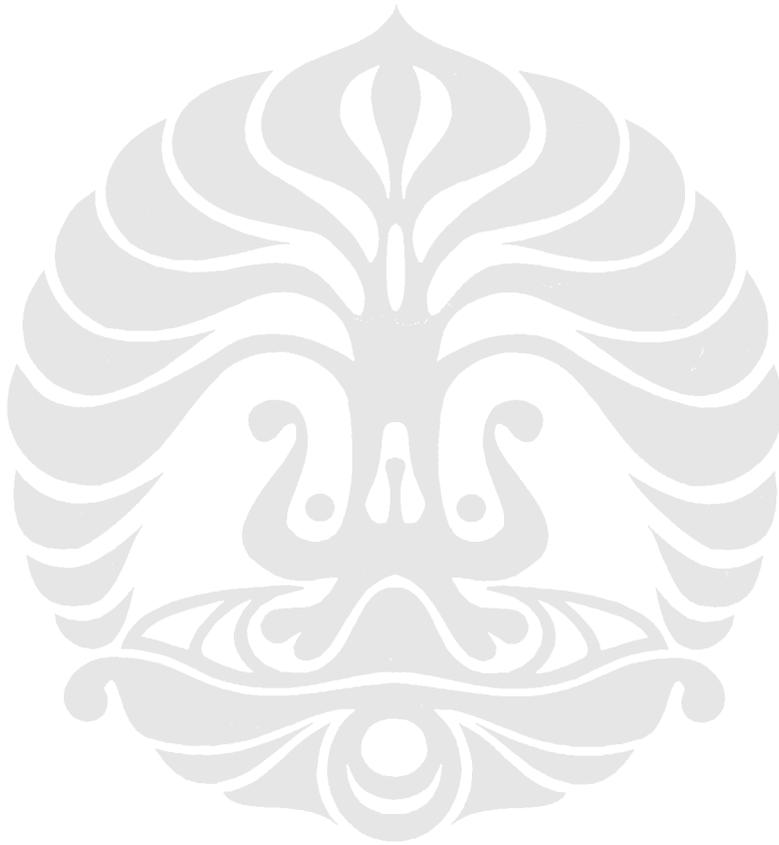
Gambar 2.27 Pengiriman data melalui perangkat PASTI/Byru Marine dengan menggunakan <i>hyper terminal</i> .....	34
Gambar 3.1 Blok diagram sistem .....	35
Gambar 3.2 Skematik diagram mikrokontroler .....	38
Gambar 3.3 Diagram alir program mikrokontroler .....	43
Gambar 4.1 Perangkat keras sistem .....	44
Gambar 4.2 Blok diagram tempat pengujian sistem .....	45
Gambar 4.3 Data keluaran GPS yang ditampilkan pada <i>hyper terminal</i> ..	45
Gambar 4.4 Data keluaran GPS pada kondisi tanpa sinyal ( <i>void</i> ) .....	46
Gambar 4.5 Tampilan data dari kontroler pada <i>hyper terminal</i> tanpa <i>masking</i> .....	46
Gambar 4.6 Tampilan data dari kontroler pada <i>hyper terminal</i> dengan kecepatan 4800 baud.....	47
Gambar 4.7 Tampilan data dari kontroler pada <i>hyper terminal</i> dengan kecepatan 1200 baud.....	47
Gambar 4.8 Tampilan data \$GPRMC pada PASTI/ <i>user terminal</i> .....	48

## 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

## DAFTAR LAMPIRAN

	Halaman
LAMPIRAN 1 Program Pengolahan Data GPS dengan <i>Masking</i>	
\$GPRMC .....	57



## 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
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
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. Dengan menggunakan pelayanan tersebut permasalahan sistem charging pengiriman data berdasarkan waktu, cara ini tidak praktis untuk armada perkapalan dimana setiap kapal harus membayar berdasarkan waktu pemakaian yang seharusnya keperluannya bukan untuk armada sendiri tapi untuk pengawasan.

Dengan memanfaatkan komunikasi data pada sistem komunikasi satelit Garuda 1, kami akan merancang dan membuat sistem pemantauan transportasi.

## **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.

## **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 dan penggunaan nomor khusus tanpa charging biaya perwaktu pemakaian.

## **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. Komunikasi data menggunakan dua perangkat PASTI/*byru marine*

## **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.

### **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, periferal, 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*

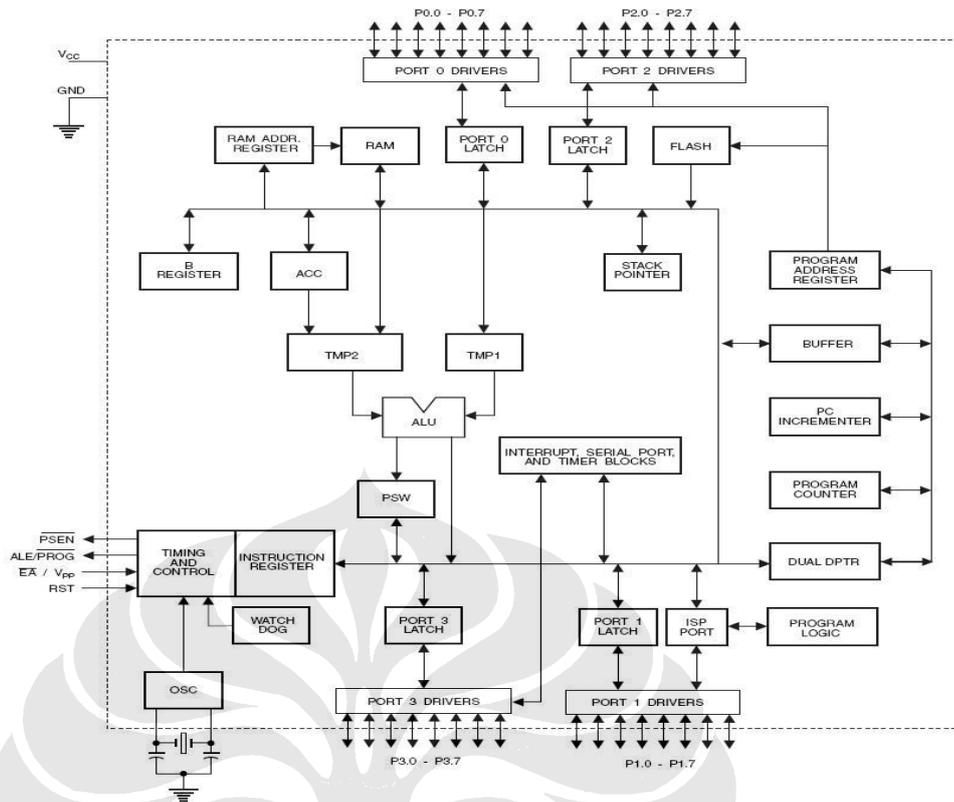
Mikrokontroler dalam bekerja didukung oleh register-register yang berfungsi untuk mengontrol masukan atau keluaran yang diberikan. Register-register ini adalah: TCON, TMOD, SCON, PCON, IP, dan IE.

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]

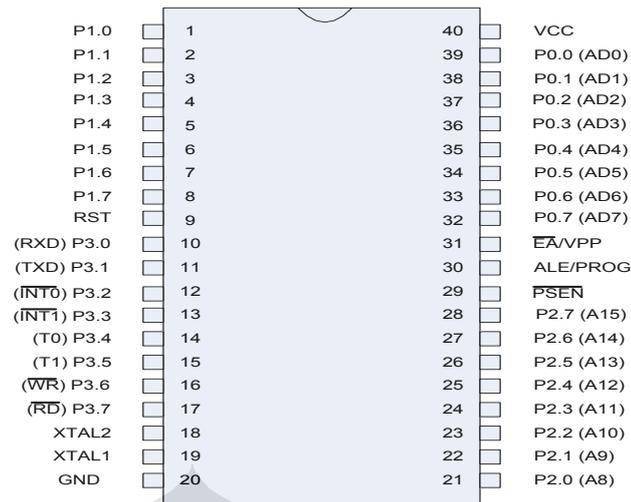
### 2.1.2 Fitur AT89S51

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 pengalamatan 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 pengalamatan 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* ( $\overline{EA} / VPP$ )

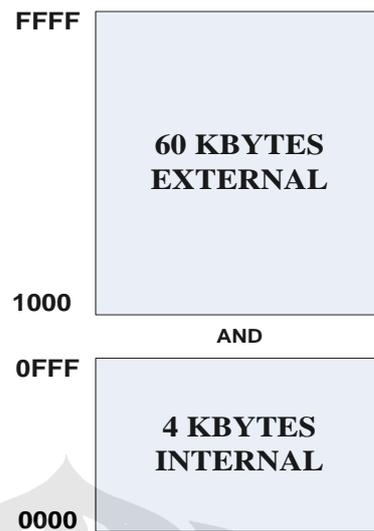
Pin  $\overline{EA}$  harus dihubungkan ke *ground* (GND) agar dapat mengambil program dari memori eksternal dan menerima tegangan sebesar 12 V dari *programming enable voltage* ( $V_{PP}$ ) saat pemrograman *flash*.

#### 11. XTAL1 dan XTAL2

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

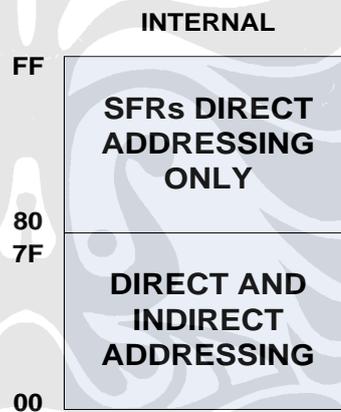
### 2.1.4 Peta Memori

AT89S51 memiliki ruang pengalaman 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

<b>Simbol</b>	<b>Nama</b>	<b>Alamat</b>
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

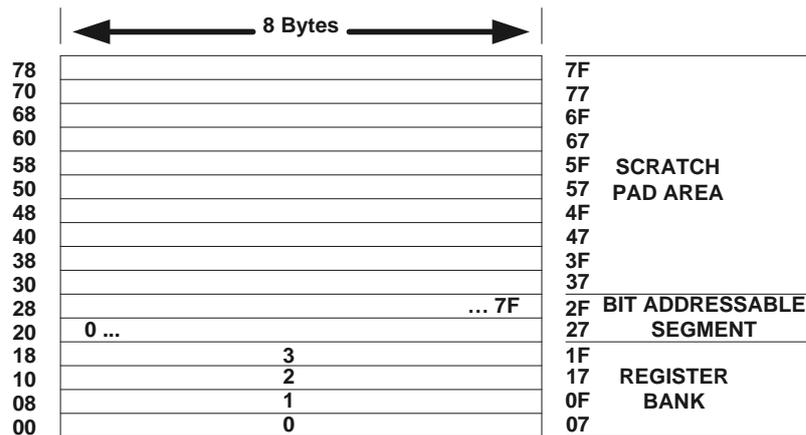
## 2. Bit Addressable Area.

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

## 3. Scratch 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

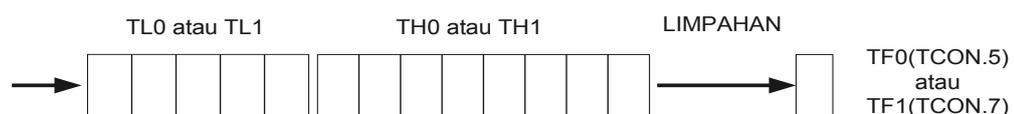
Tabel 2.6 berikut ini menjelaskan konfigurasi mode *Timer/Counter* dari fungsi bit register TMOD.

**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

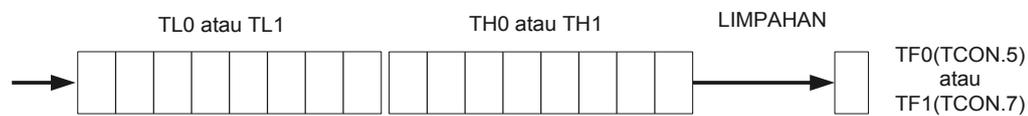
Pencacah biner dibentuk dengan TLx (TL0 atau TL1) sebagai pencacah biner 5 bit. Limpahan dari pencacah biner 5 bit dihubungkan ke THx (TH0 atau TH1) membentuk untaian pencacah biner 13 bit. Limpahan dari pencacah 13 bit ini ditampung di *flip-flop* TFx (TF0 atau TF1) pada register TCON. Gambar 2.8 memperlihatkan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



**Gambar 2.8.** Mode 0 pencacah biner 13 bit

## 2. Mode 1

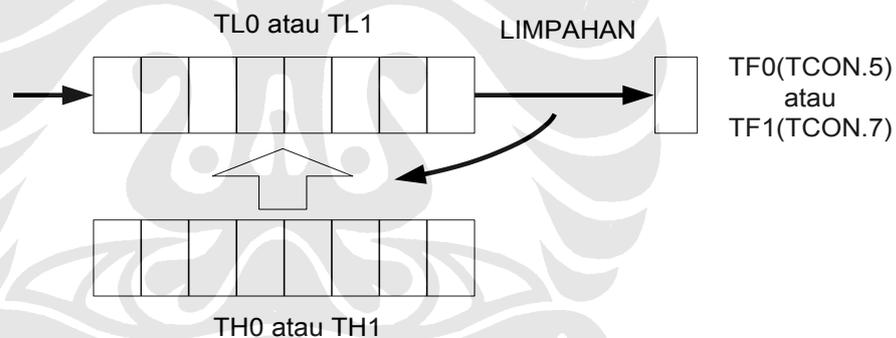
Sama halnya dengan mode 0, hanya saja register TLx dipakai sepenuhnya sebagai pencacah biner 8 bit sehingga kapasitas pencacah biner yang terbentuk adalah 16 bit. Gambar 2.9 memperlihatkan *Timer/Counter* yang bekerja pada mode 1.



**Gambar 2.9.** Mode 1 pencacah biner 16 bit

## 3. Mode 2

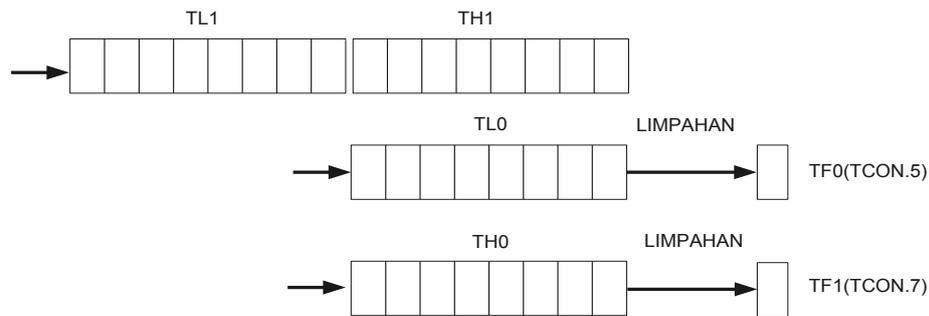
TLx dipakai sebagai pencacah biner 8 bit dan THx dipakai untuk menyimpan nilai yang diisikan ulang ke TLx setiap kali kedudukan TLx melimpah (berubah dari FFH ke 00H). Gambar 2.10 memperlihatkan *Timer/Counter* yang bekerja pada mode 2.



**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 T<sub>Fx</sub> (TF0 atau TF1) merupakan bit penampung limpahan. T<sub>Fx</sub> akan menjadi *1* setiap kali pencacah biner yang terhubung padanya melimpah.
- b. Bit TR<sub>x</sub> (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 serial mode 0</i>
SM1	SCON.6	<i>Port serial mode 1</i>
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

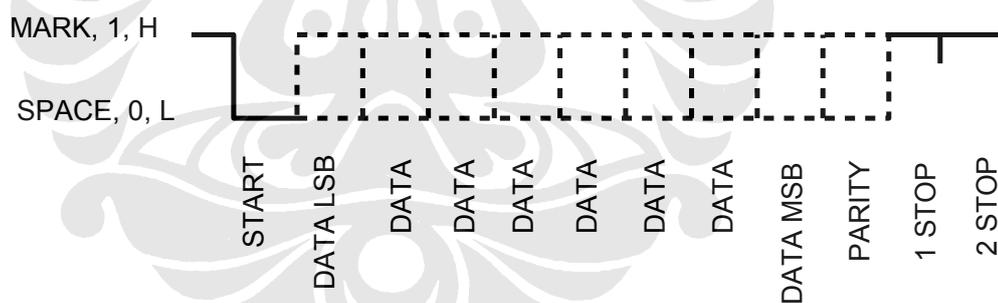
Dalam komunikasi serial AT89S51 terdapat dua jenis transmisi serial.

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]:

$$\text{Mode0baudrate} = \frac{\text{frekuensi osilator}}{12} \dots\dots\dots (2.1)$$

$$\text{Mode2baudrate} = \frac{2^{\text{SMOD}}}{64} \times \text{frekuensi osilator} \dots\dots\dots (2.2)$$

$$\text{Mode1 dan 3baudrate} = \frac{2^{\text{SMOD}}}{32} \times \frac{\text{frekuensi osilator}}{12 \times [56 - \text{CH1}]} \dots\dots\dots (2.3)$$

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 mengirimkan 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 *I* 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.

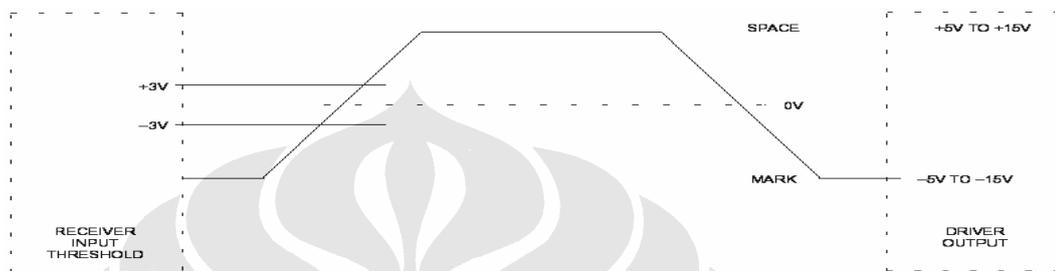
## 2.2 KONVERTER RS232

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.

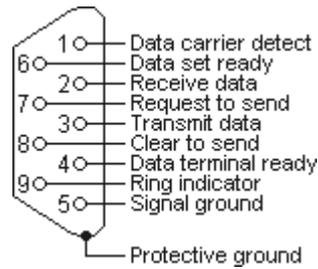


**Gambar 2.16.** Spesifikasi level logik konverter RS232C [8]

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	3kΩ sampai 7kΩ
Sensitivitas masukan Rx	±3 Volt
<i>Range</i> tegangan masukan Rx	±15 Volt
Tahanan <i>input</i> Rx maksimum	3kΩ sampai 7kΩ
<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

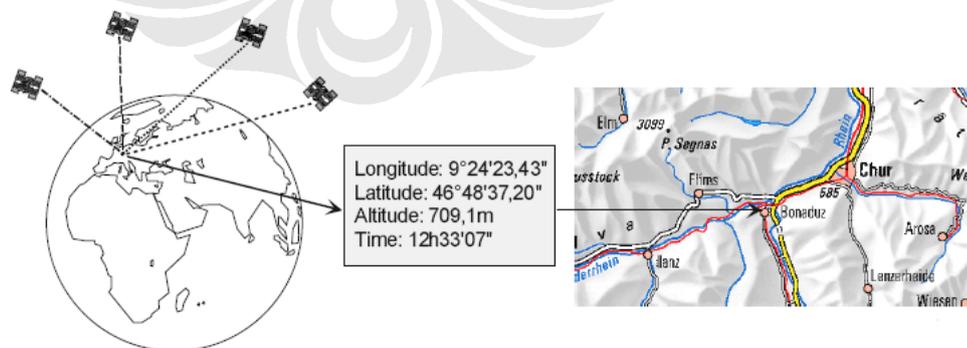
mengaktifkan CTS sebelum memulai komunikasi.

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.

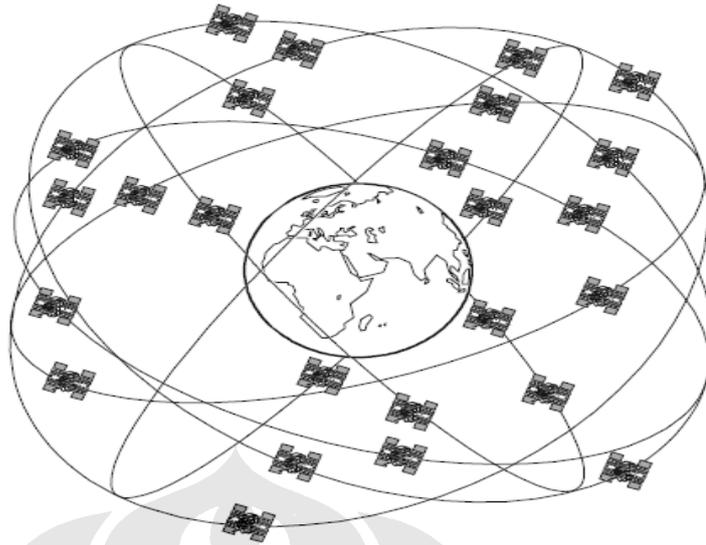
### 2.3 GLOBAL POSITIONING SYSTEM (GPS)

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  $\pm 100$  m pada saat kebijakan *Selective Availability* (SA) masih berlaku dan  $\pm 20$  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]

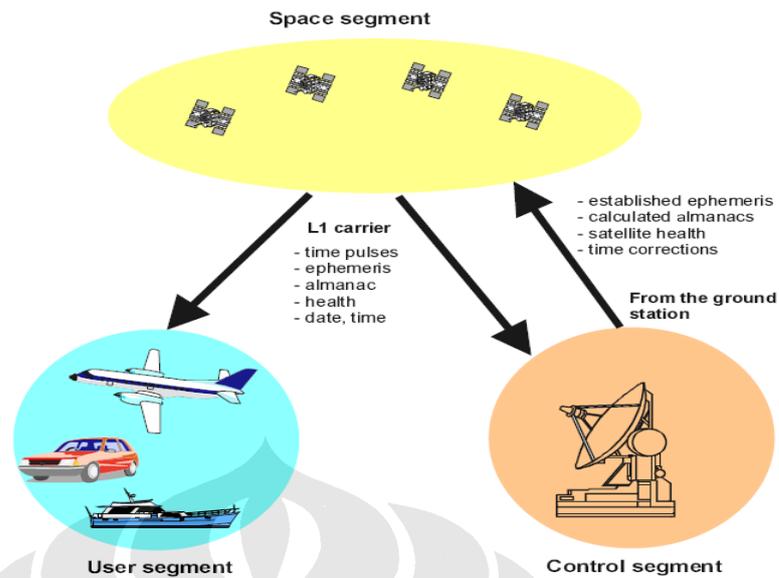
### 2.3.1 Segmen Penyusun GPS

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.



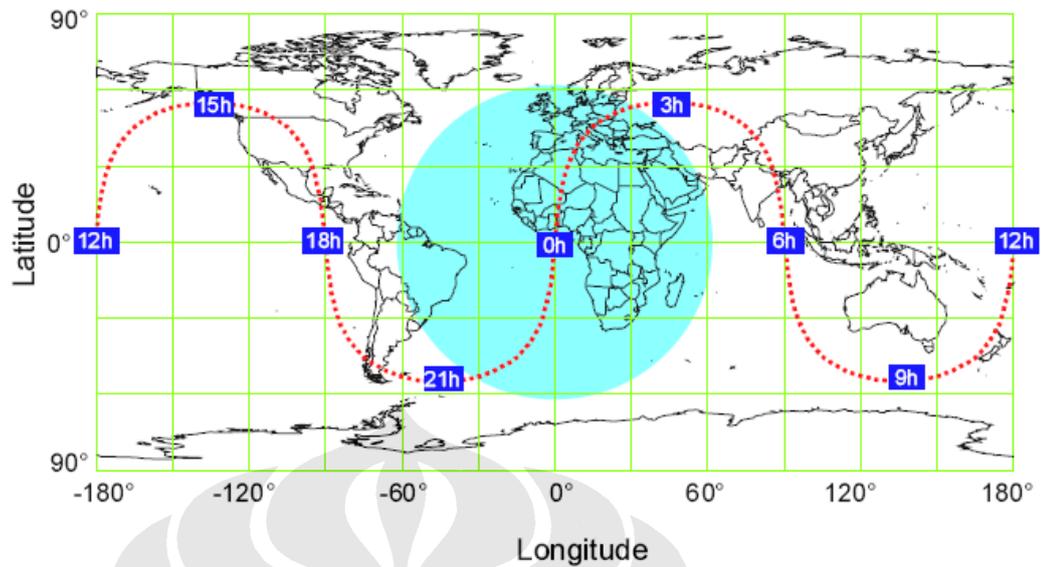
**Gambar 2.20.** Tiga segmen pada GPS [11]

### 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  $55^{\circ}$  condong ke ekuator. Tiap satelit akan memutar orbitnya tepat 12 jam. Berdasarkan rotasi Bumi, satelit akan berada pada titik awalnya setelah 24 jam (atau tepatnya 23 jam 56 menit).

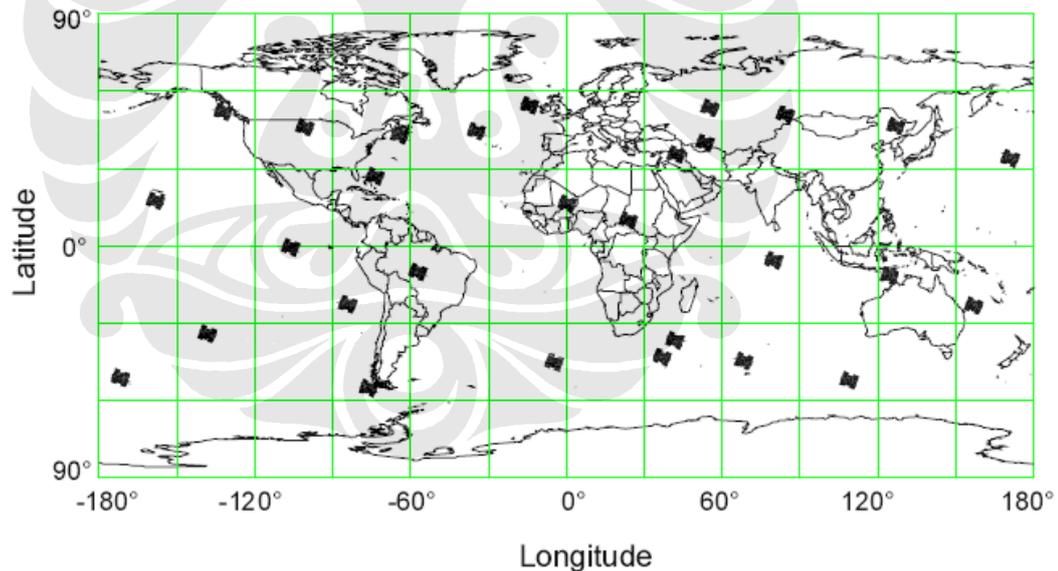
Gambar 2.21 berikut ini memperlihatkan posisi 28 satelit GPS pada rentang waktu 12 jam *Universal Time Coordinat* (UTC). Gambar 2.22 memperlihatkan distribusi ke-28 satelit GPS pada tanggal 14 April 2001.

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*)

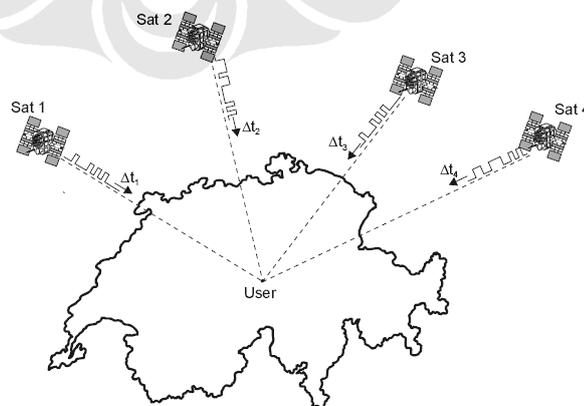
Yang termasuk dalam segmen pengguna adalah semua pengguna GPS, baik sipil maupun militer. Dalam hal ini *receiver* GPS dibutuhkan untuk menerima dan memproses sinyal-sinyal dari GPS untuk digunakan dalam penentuan posisi, kecepatan, dan waktu.

### 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).



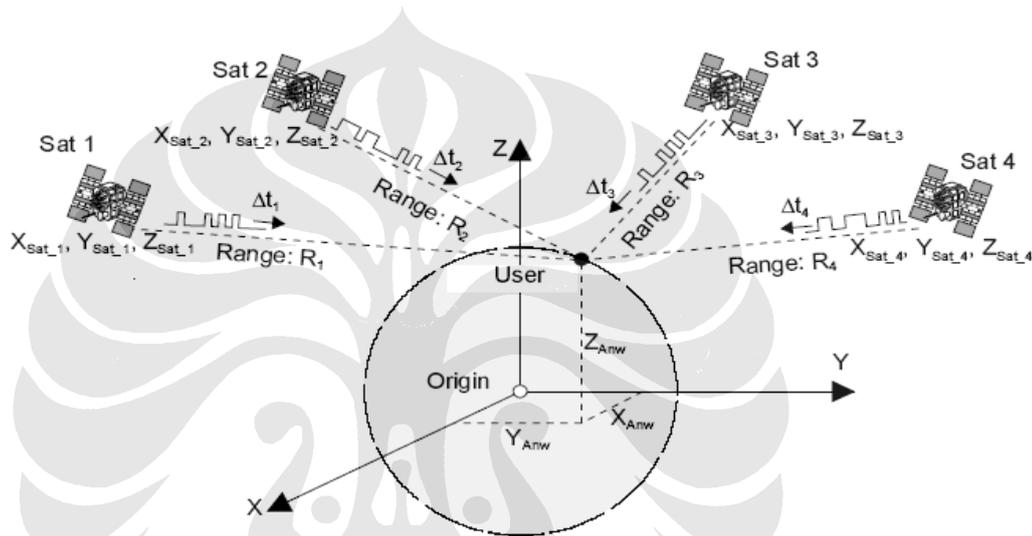
**Gambar 2.23.** Sinyal dari 4 satelit [14]

Perhitungan dinyatakan dalam bidang kartesian (X, Y, Z) dan koordinat 3 dimensi dengan sumbu *geocentric* (gambar 2.24). Jarak pengguna dengan 4 satelit R<sub>1</sub>, R<sub>2</sub>, R<sub>3</sub>, dan R<sub>4</sub> dapat ditentukan dengan bantuan sinyal transit waktu Δt<sub>1</sub>, Δt<sub>2</sub>, Δt<sub>3</sub>, dan Δt<sub>4</sub> antar-4 satelit dan pengguna. Setelah lokasi X<sub>SAT</sub>, Y<sub>SAT</sub>, dan Z<sub>SAT</sub> 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 x 10<sup>8</sup> m/s

Δt = waktu transit sinyal dari satelit ke pengguna

Δt<sub>0</sub> = 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).

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

Untuk menentukan empat variabel yang tidak diketahui ( $\Delta t_0$ ,  $X_{ANW}$ ,  $Y_{ANW}$ , dan  $Z_{ANW}$ ), diperlukan empat persamaan terpisah.

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

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

### 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
$GPGSV,3,3,12,22,21,350,00,24,36,
$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 123<sup>0</sup>

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 $\mu$ 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 2 perangkat PASTI/Byru Marine dapat dilihat pada Gambar 2.27.

```

test - HyperTerminal
File Edit View Call Transfer Help
OK
OK
OK
atdt
NO CARRIER
ATDT0868190014111
NO CARRIER
ATDT0868190014111
NO CARRIER
RING
RING
RING
ATA
CONNECT 2400
LDFKJGFKLDJGKLGFDKJHDFKJFHKJGKDFJDGHKDJhjsdgsjgsfhsdjskchkd|Xn
NO CARRIER
-
Connected 0:26:12 Auto detect 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 keenam merupakan pengiriman dengan menggunakan metode kedua, yakni tiga 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. Tiga 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.

## BAB III

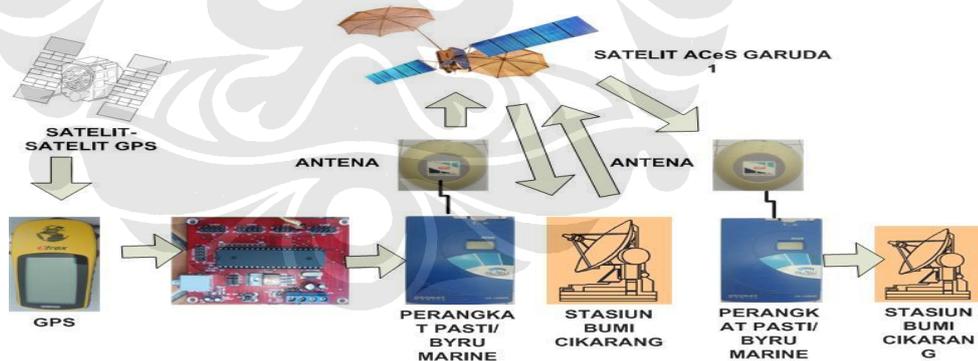
### RANCANG BANGUN

#### 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 dengan penghubung konverter RS232C, kemudian menempatkannya di suatu alamat memori RAM. Data mentah GPS tersebut dengan data *header* ATDT yang disimpan pada mikrokontroler, data dikirimkan melalui antarmuka konverter RS232C ke satelit ACeS Garuda 1 melalui perangkat PASTI/Byru Marine kemudian data tersebut dikirim ke PASTI/ Byru Marine yang lain

##### 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

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. Kartu data khusus

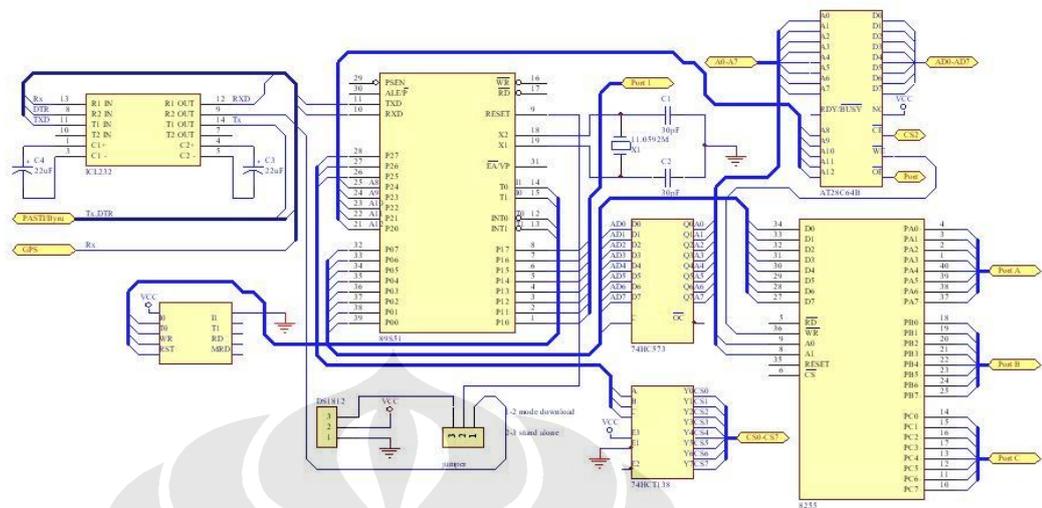
Kartu yang telah diatur sedemikian rupa oleh produsen PSN yang menonaktifkan sistem pembayaran perpemakaian.

### 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

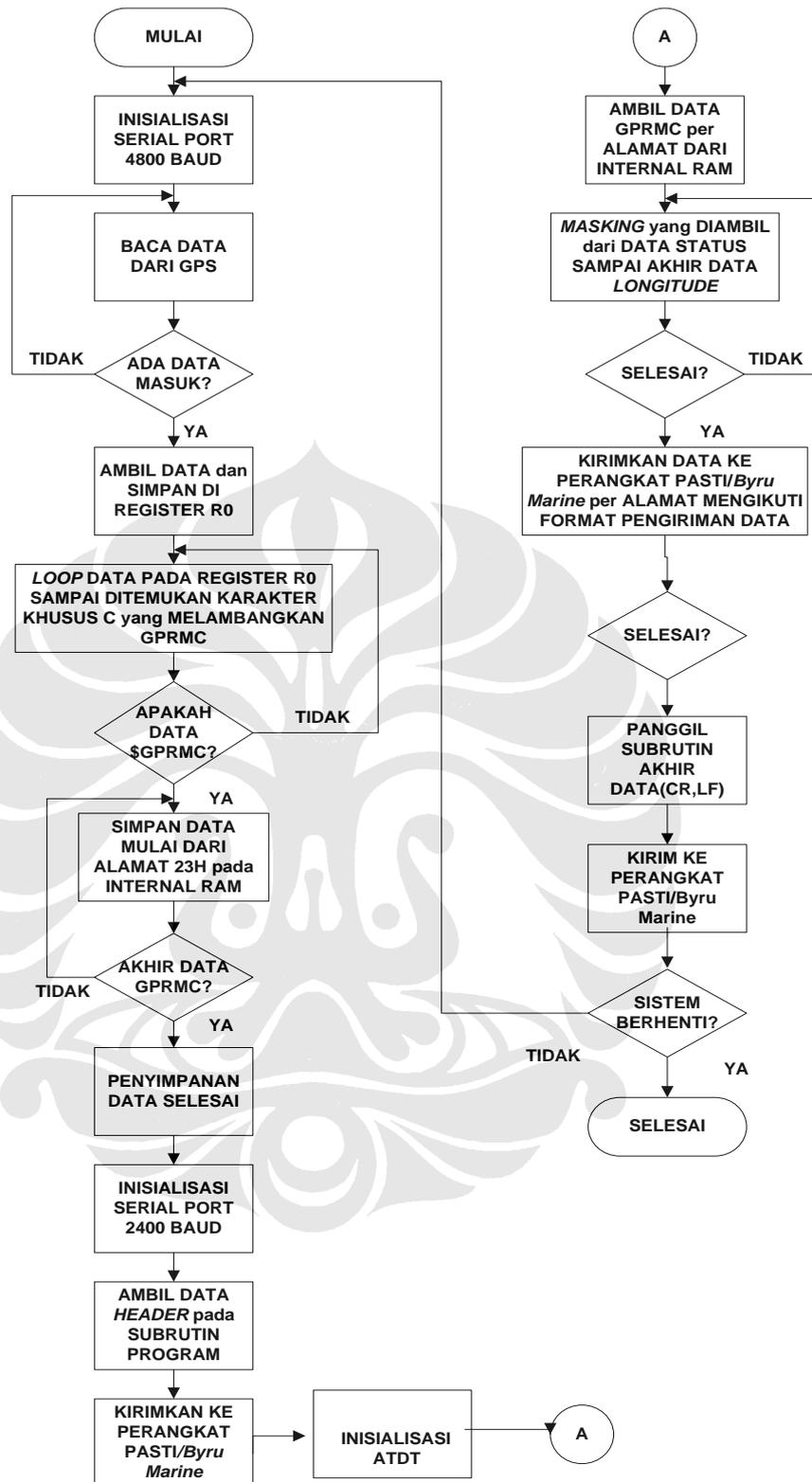
### 3.3 PERANGKAT LUNAK

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, 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 inisialisasi *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

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

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

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

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

transmisi 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 kecepatannya juga pada 2400 baud.

Mikrokontroler mengambil data dari GPS (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 baik GPS dan keypad 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 23H. Data yang disimpan ini akan dibatasi dengan data *CR* (data 0d).

Setelah data *masking* GPRMC disimpan, mikrokontroler diubah kecepatannya 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.

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*.

### 3.4 FORMAT DATA

Pada perancangan sistem di atas, dapat disimpulkan bahwa ada dua jenis data 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 *checksum*  
 CS : Pengecekan kesalahan pada kalimat  
 <CR><LF> : Akhir dari data, yaitu *carriage return* (<CR>), dan *line feed* (<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>[21]
```

#### Keterangan:

RMC : *Recommended Minimum Specific GPS/TRANSIT Data*  
 hhmmss : UTC of *position fix* dalam format jam/menit/detik  
 s : Data Status (A untuk aktif dan V untuk *void*)  
 lll.lll : Posisi *latitude*  
 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* -23<sup>rd</sup> of March 1994

003.1,W : *magnetic variation*

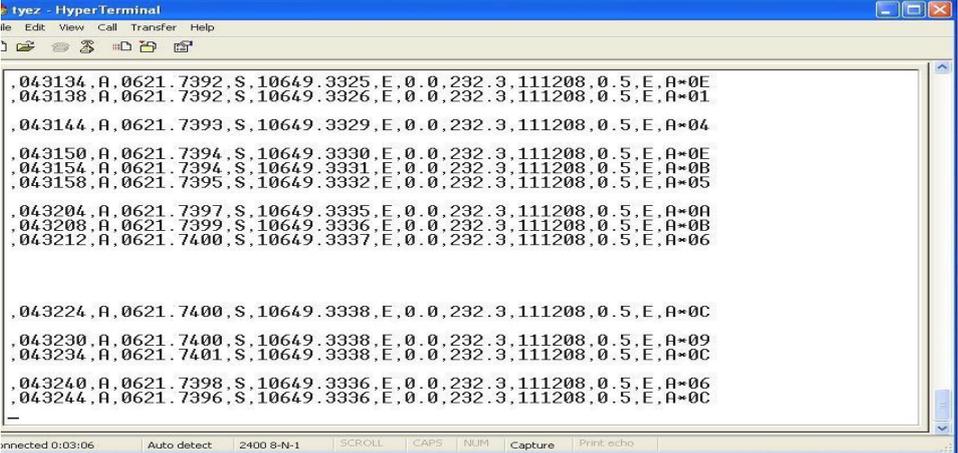
\*6A : *the checksum data*, selalu diawali dengan \*

### 3.3.2 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*. Dari data \$GPRMC yang disimpan di memori, data tersebut berisi data lokasi *latitude* dan *longitude*.

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*.



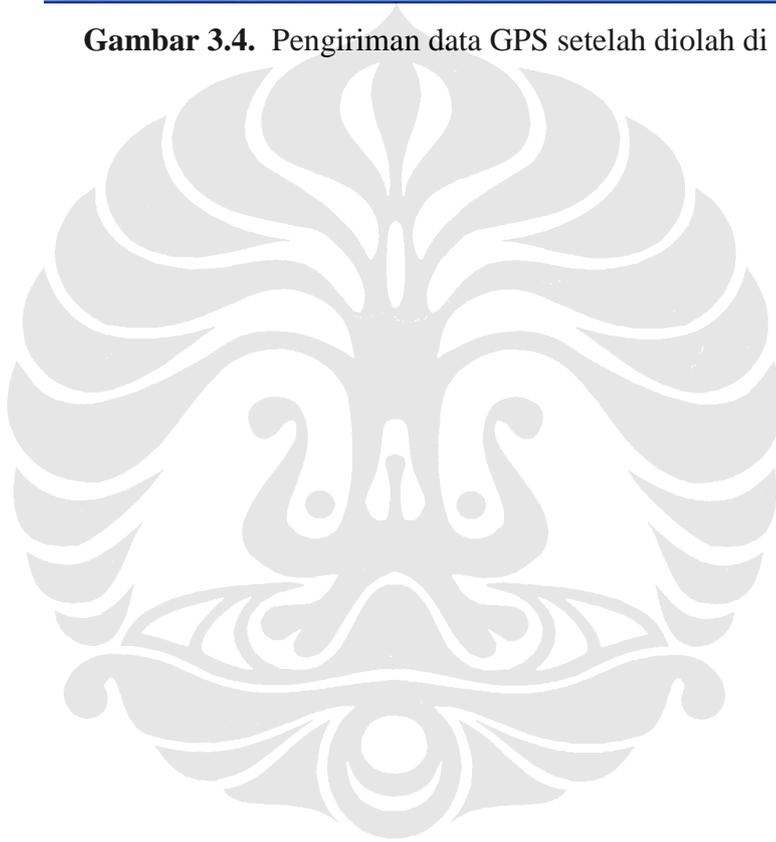
```
tyez - HyperTerminal
File Edit View Call Transfer Help

.043134,A,0621.7392,S,10649.3325,E,0.0,232.3,111208,0.5,E,A~0E
.043138,A,0621.7392,S,10649.3326,E,0.0,232.3,111208,0.5,E,A~01
.043144,A,0621.7393,S,10649.3329,E,0.0,232.3,111208,0.5,E,A~04
.043150,A,0621.7394,S,10649.3330,E,0.0,232.3,111208,0.5,E,A~0E
.043154,A,0621.7394,S,10649.3331,E,0.0,232.3,111208,0.5,E,A~0B
.043158,A,0621.7395,S,10649.3332,E,0.0,232.3,111208,0.5,E,A~05
.043204,A,0621.7397,S,10649.3335,E,0.0,232.3,111208,0.5,E,A~0A
.043208,A,0621.7399,S,10649.3336,E,0.0,232.3,111208,0.5,E,A~0B
.043212,A,0621.7400,S,10649.3337,E,0.0,232.3,111208,0.5,E,A~06

.043224,A,0621.7400,S,10649.3338,E,0.0,232.3,111208,0.5,E,A~0C
.043230,A,0621.7400,S,10649.3338,E,0.0,232.3,111208,0.5,E,A~09
.043234,A,0621.7401,S,10649.3338,E,0.0,232.3,111208,0.5,E,A~0C
.043240,A,0621.7398,S,10649.3336,E,0.0,232.3,111208,0.5,E,A~06
.043244,A,0621.7396,S,10649.3336,E,0.0,232.3,111208,0.5,E,A~0C

connected 0:03:06 Auto detect 2400 8-N-1 SCROLL CAPS NUM Capture Print echo
```

**Gambar 3.4.** 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.



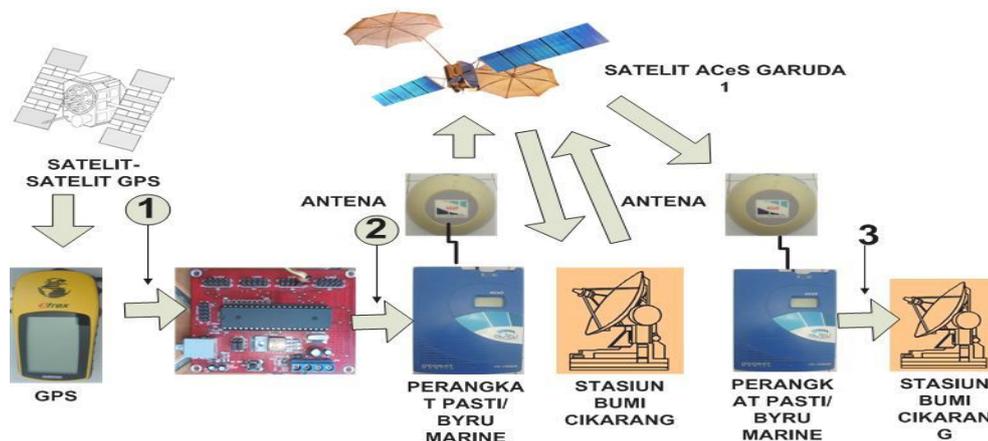
**Gambar 4.1.** Perangkat keras sistem

### 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 GPS berdasar kecepatan yang disetting dari *hyperterminal*.
3. Menampilkan data keluaran dari perangkat PASTI disisi receiver:

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.

```

$GPGGA,0621.9896,S,10649.2145,E,043224,A,A=5A
$GPBOD,T,M,*47
$GPRME,3.0,M,8.2,M,8.7,M=28
$GPRMZ,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
$GPRME,3.0,M,8.1,M,8.7,M=2B
$GPRMZ,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 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,0.12,M,M,*04
$GPGGA,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
$GPGRL,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,0.12,M,M,*04
$GPGGA,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
$GPGRL,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

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 4.5 memperlihatkan data GPS yang telah diolah pada mikrokontroler sehingga hanya data \$GPRMC yang dilewatkan.

```

,072342,A,0619.6069,S,10708.8055,E,0.0,191.1,241208,0.6,E,A*04
,072344,A,0619.6069,S,10708.8055,E,0.0,191.1,241208,0.6,E,A*02
,072346,A,0619.6069,S,10708.8055,E,0.0,191.1,241208,0.6,E,A*00
,072348,A,0619.6071,S,10708.8055,E,0.0,191.1,241208,0.6,E,A*07
,072350,A,0619.6071,S,10708.8055,E,0.0,191.1,241208,0.6,E,A*0E
,072352,A,0619.6071,S,10708.8055,E,0.0,191.1,241208,0.6,E,A*0C
,072354,A,0619.6073,S,10708.8056,E,0.0,191.1,241208,0.6,E,A*0B
,072356,A,0619.6073,S,10708.8055,E,0.0,191.1,241208,0.6,E,A*0A
,072358,A,0619.6074,S,10708.8055,E,0.0,191.1,241208,0.6,E,A*03

,072400,A,0619.6075,S,10708.8055,E,0.0,191.1,241208,0.6,E,A*08

,072402,A,0619.6075,S,10708.8055,E,0.0,191.1,241208,0.6,E,A*0A

,072404,A,0619.6075,S,10708.8055,E,0.0,191.1,241208,0.6,E,A*0C

,072406,A,0619.6074,S,10708.8055,E,0.0,191.1,241208,0.6,E,A*0F
,072408,A,0619.6074,S,10708.8055,E,0.0,191.1,241208,0.6,E,A*01
,072410,A,0619.6073,S,10708.8055,E,0.0,191.1,241208,0.6,E,A*0F
,072412,A,0619.6071,S,10708.8055,E,0.0,191.1,241208,0.6,E,A*0F
,072414,A,0619.6071,S,10708.8056,E,0.0,191.1,241208,0.6,E,A*0A

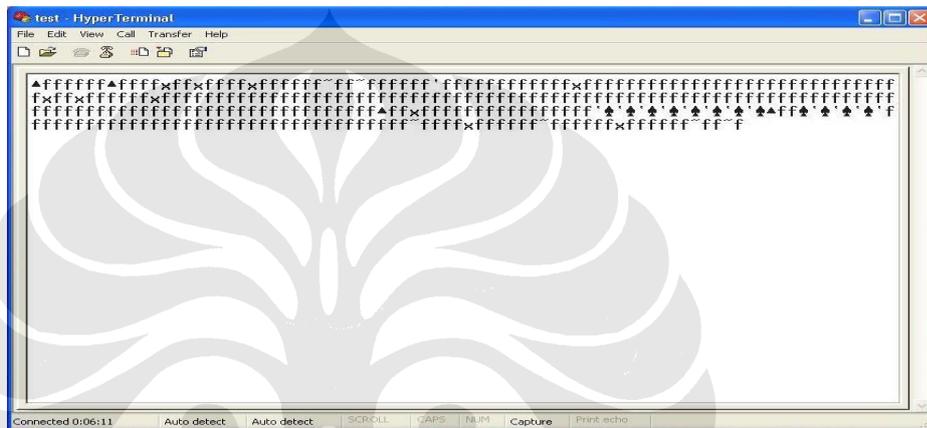
```

**Gambar 4.5.** Tampilan data dari kontroler pada *hyperterminal* dengan *masking* data \$GPRMC

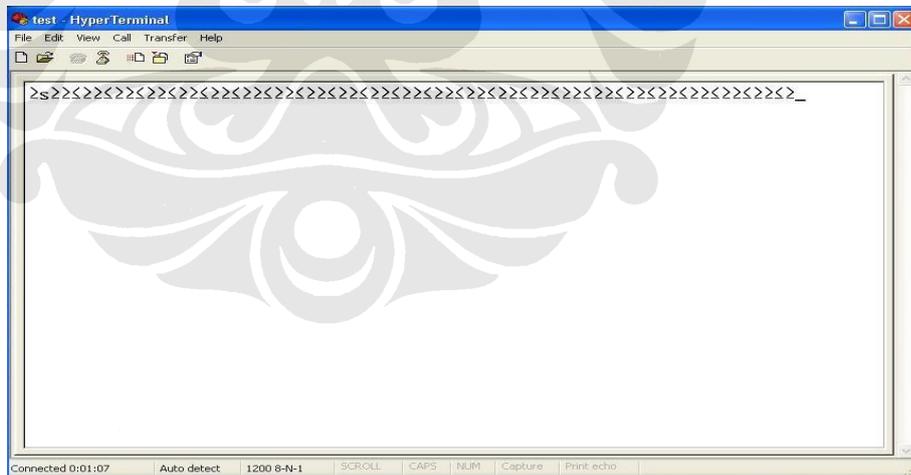
Pada baris pertama waktunya menunjukkan 072342 yang berarti jam 07, 23 menit 42 detik. Sedangkan pada baris kedua *hyper terminal* data waktunya menunjukkan 072344 yang berarti jam 07,23 menit 44 detik.

#### 4.2.2.1 Data Keluaran dengan memberikan variasi kecepatan hyperterminal

Pengujian ini bertujuan untuk melihat data yang akan dikirimkan ke satelit Garuda 1 dengan menggunakan kecepatan pada hyperterminal dengan kecepatan yang digunakan adalah 4800 baud dan 1200 baud.



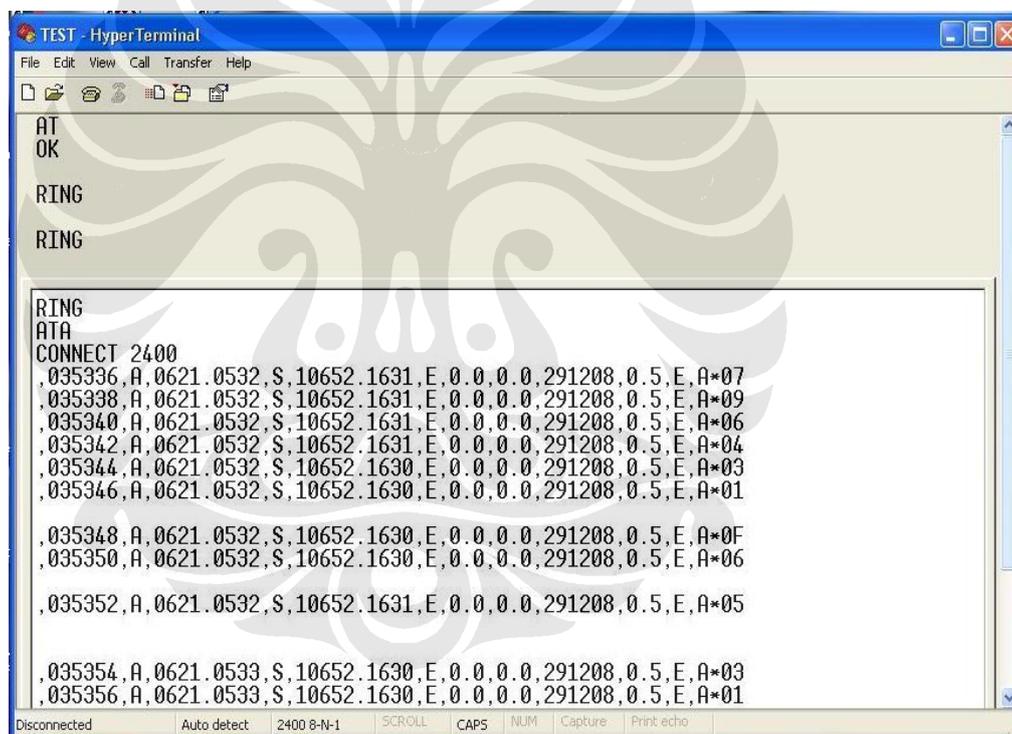
**Gambar 4.6.** Data keluaran dengan kecepatan 4800 baud



**Gambar 4.6.** Data keluaran dengan kecepatan 1200 baud

### 4.2.3 Menampilkan Data Final yang diterima PASTI/*User Terminal* sebagai sisi receiver

Pengujian ini dilakukan untuk melihat apakah data yang dikirim akan sampai tujuan sesuai dengan yang diinginkan dengan data GPS dalam bentuk \$GPRMC tidak berubah datanya. Dalam pengujian ini dilakukan beberapa langkah seperti gambar 4.7 yaitu dengan mengetik kata AT pada *hyperterminal*, kata OK adalah balasan bahwa perangkat PASTI/*user terminal* siap melakukan komunikasi yang dilanjutkan dengan kata RING tiga kali ini menyatakan sistem komunikasi yang dilakukan adalah antara perangkat PASTI/*user terminal*. Agar didapatkan datanya maka perlu mengetik kata ATA berarti perangkat siap menerima data. Terakhir akan didapatkan data \$GPRMC secara terus menerus.



```

TEST - HyperTerminal
File Edit View Call Transfer Help
[Icons]
AT
OK
RING
RING
RING
ATA
CONNECT 2400
,035336,A,0621.0532,S,10652.1631,E,0.0,0.0,291208,0.5,E,A*07
,035338,A,0621.0532,S,10652.1631,E,0.0,0.0,291208,0.5,E,A*09
,035340,A,0621.0532,S,10652.1631,E,0.0,0.0,291208,0.5,E,A*06
,035342,A,0621.0532,S,10652.1631,E,0.0,0.0,291208,0.5,E,A*04
,035344,A,0621.0532,S,10652.1630,E,0.0,0.0,291208,0.5,E,A*03
,035346,A,0621.0532,S,10652.1630,E,0.0,0.0,291208,0.5,E,A*01
,035348,A,0621.0532,S,10652.1630,E,0.0,0.0,291208,0.5,E,A*0F
,035350,A,0621.0532,S,10652.1630,E,0.0,0.0,291208,0.5,E,A*06
,035352,A,0621.0532,S,10652.1631,E,0.0,0.0,291208,0.5,E,A*05
,035354,A,0621.0533,S,10652.1630,E,0.0,0.0,291208,0.5,E,A*03
,035356,A,0621.0533,S,10652.1630,E,0.0,0.0,291208,0.5,E,A*01
Disconnected Auto detect 2400 8-N-1 SCROLL CAPS NUM Capture Print echo

```

**Gambar 4.6.** Tampilan data \$GPRMC pada PASTI/*user terminal* sisi receiver

## 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 dua pengujian dengan mencoba beberapa skenario.

1. 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. 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 1*. 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:

$$a. \text{ \$GPRMC terima} = (11,9350 + A + B)\mu\text{s}$$

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>.

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

$$(Inst.1 + Inst.2 + Inst.3 + Inst.4)$$

$$B = 1038,3450\mu s$$

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

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

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

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

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

$$C = 1024,2400\mu s$$

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

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

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

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

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

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

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

$$Delay = -41,2100\mu s$$

Nilai *delay* yang menunjukkan hasil minus (-) berarti, waktu yang diperlukan untuk mengirimkan data yang telah disimpan lebih cepat 41,2100 $\mu s$ . Sedangkan untuk mengambil data  $\text{\$GPRMC}$  diperlukan waktu 2180,8700 $\mu s$  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 I* dapat dilihat bahwa program melakukan pengulangan untuk sampai menemukan karakter khusus  $\text{\$GPRMC}$ . Berarti total besarnya waktu yang digunakan untuk mengambil data  $\text{\$GPRMC}$  berikutnya adalah:

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

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

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

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

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

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  $\$GPRMC$ .

2. Pengaturan kecepatan pada *hyperteminal* yaitu dengan kecepatan 4800 baud dengan maksud apakah data yang dikirim akan semakin cepat diterima dan 1200 baud apakah data yang diterima semakin lambat. Berdasarkan sistem yang dipakai yaitu perangkat PASTI dengan standar kecepatan 2400 baud sehingga jika diatur kecepatannya yang tidak sesuai maka akan didapatkan data yang berbeda.

### 4.3.3 Analisis Pengujian Ketiga

Pengujian ini dilakukan dengan maksud apakah data  $\$GPRMC$  yang dikirim dari perangkat PASTI/byru marine (*sisi transmitter*) sama dengan data  $\$GPRMC$  yang diterima perangkat PASTI/byru marine (*sisi receiver*). Berdasarkan Gambar 4.8 maka bisa diartikan untuk data posisi *latitude* seperti pada baris kedua tampilan *hyper terminal* adalah  $06210532,S$  yang berarti *latitude*  $06^{\circ} 21.0532'S$  atau lebih tepatnya data posisi GPS dapat dikonversi menjadi data dalam bentuk *Geographic Information Systems* (GIS) dalam dua langkah[22].

*Langkah 1*

Membagi koordinat dengan konstanta 100

$$\frac{0621.0532}{100} = 6.210532 = 6.210$$

### 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.

$$\text{Data posisi peta GIS} = 6 + \frac{0.210 \times 10}{6} = 6.35LS$$

Karena kota Jakarta 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 165021631,E yang berarti *longitude* 106° 02.1631'E. Apabila dikonversi ke posisi peta GIS dengan menggunakan langkah yang sama, didapatkan:

$$\frac{105021631}{100} = 165.021631 = 165.0216$$

$$\text{Data posisi peta GIS} = 165 + \frac{0.021631 \times 10}{6} = 165.21631BT$$

Jika dilihat pada gambar 4.8 terlihat terdapat jeda kosong walaupun selang waktu antara masing-masing data \$GPRMC tetap 2 sekon. Hal ini dikarenakan perbedaan waktu pengambilan dan pengiriman data \$GPRMC yang berbeda sedikit oleh mikrokontroler.

## **BAB V**

### **KESIMPULAN DAN SARAN**

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. Penggunaan kecepatan kirim data dan terima data untuk komunikasi PASTI/*user terminal* adalah 2400 baud dan tidak bisa diatur dengan maksud ingin lebih cepat atau lebih lambat dalam menerima dan mengirim data.
3. Data \$GPRMC yang diterima adalah sama dengan yang dikirim dengan beda waktu antara masing-masing data 2 sekon dari semula. Perbedaan kecepatan mikrokontroler dalam mengambil dan mengirimkan data \$GPRMC menyebabkan jeda dengan adanya data tanpa data navigasi.

## DAFTAR REFERENSI

- [1] Kenneth J Ayala, *The 8051 Microcontroller Architecture, Programming, and Application 2<sup>nd</sup> Edition* (New York: West Publishing Company, 1997), hal. 55
- [2] Kenneth J Ayala, *The 8051 Microcontroller Architecture, Programming, and Application 2<sup>nd</sup> Edition* (New York: West Publishing Company, 1997), hal. 58
- [3] Kenneth J Ayala, *The 8051 Microcontroller Architecture, Programming, and Application 2<sup>nd</sup> Edition* (New York: West Publishing Company, 1997), hal. 63
- [4] Kenneth J Ayala, *The 8051 Microcontroller Architecture, Programming, and Application 2<sup>nd</sup> Edition* (New York: West Publishing Company, 1997), hal. 63
- [5] Kenneth J Ayala, *The 8051 Microcontroller Architecture, Programming, and Application 2<sup>nd</sup> 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] Peter, Joe (2007). *NMEA Data*. Diakses 6 Mei 2008 dari National Marine Electronics Association. [www.nmea.org](http://www.nmea.org)
- [22] 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.

## DAFTAR PUSTAKA

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

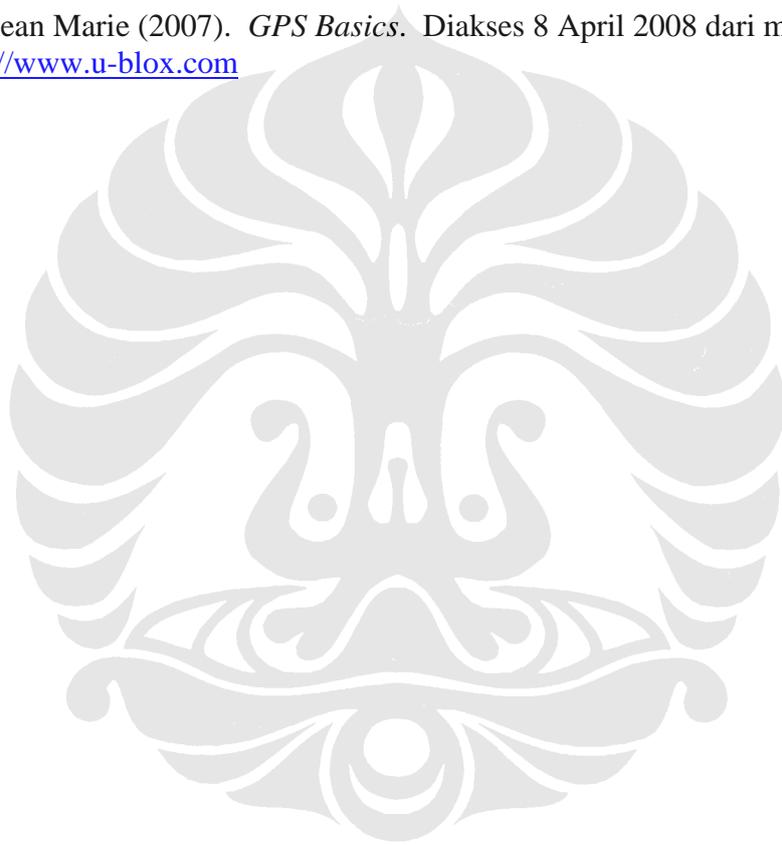
Peter, Joe. *NMEA Data*. Diakses 6 Mei 2008 dari National Marine Electronics Association.

[www.nmea.org](http://www.nmea.org)

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 Posisi GPS dengan Masking  
\$GPRMC**

		Mesin Cycle	Waktu ( $\mu$ s)	Total Waktu ( $\mu$ s)	
kirim:	mov	sbuf,A	2	2,1700	<b>7,5950</b>
	jnb	ti,\$	2	2,1700	
	clr	ti	1	1,0850	
	ret		2	2,1700	
terima:	clr	ri	1	1,0850	<b>7,5950</b>
	mov	A,sbuf	2	2,1700	
	jnb	ri,\$	2	2,1700	
	ret		2	2,1700	
simpan:	mov	@r0,A	1	1,0850	<b>3,2550</b>
	ret		2	2,1700	
ambil:	mov	A,@r0	1	1,0850	<b>3,2550</b>
	ret		2	2,1700	
header:	mov	A,#'A'	1	1,0850	<b>36,8900</b>
	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	<b>9,7650</b>
	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	
	mov	TH1,#0FAh	1	1,0850	<b>9,7650</b>
	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	<b>1 2 3</b>
	cjne	A,#'C',xxx	2	2,1700	
	ajmp	aaa	2	2,1700	

		Mesin Cycle	Waktu ( $\mu$ s)	Total Waktu ( $\mu$ 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,#'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	
	acall	kirim	7,5950	
	ajmp	sss	2 2,1700	3
ddd:	acall	kirim	7,5950	4
	inc	r0	1 1,0850	5
	ajmp	ttt	2 2,1700	6
sss:	mov	A,#13	1 1,0850	7
	acall	kirim	7,5950	19,5300
	mov	A,#10	1 1,0850	
	acall	kirim	7,5950	
	ajmp	uuu	2 2,1700	21,7000
uuu:	acall	ambil	3,2550	
	cjne	A,#'S',uuu	2 2,1700	
	inc	r0	1 1,0850	15,1900
	acall	header	15,1900	
www:	acall	ambil	3,2550	
	cjne	A,#'E',eee	2 2,1700	1
	acall	kirim	7,5950	2
	inc	r0	1 1,0850	3
	ajmp	fff	2 2,1700	4
eee:	acall	kirim	7,5950	5
	inc	r0	1 1,0850	6
	ajmp	www	2 2,1700	7
				8

			<b>Mesin Cycle</b>	<b>Waktu (<math>\mu</math>s)</b>	<b>Total Waktu (<math>\mu</math>s)</b>
fff:	acall	ambil		3,2550	<b>1</b>
	cjne	A,#13,ggg	2	2,1700	<b>2</b>
	acall	kirim		7,5950	<b>3</b>
	ajmp	nnn	2	2,1700	<b>4</b>
ggg:	inc	r0	1	1,0850	<b>5</b>
	ajmp	fff	2	2,1700	<b>6</b>
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				
					<b>19,5300</b>

