

# BAB III

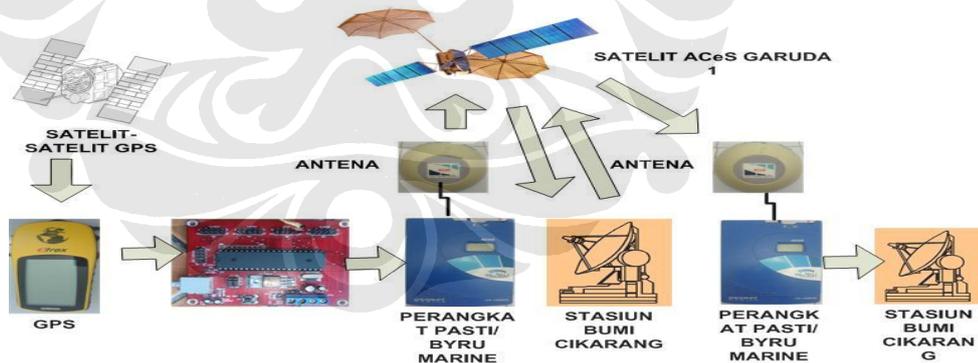
## RANCANG BANGUN

### 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

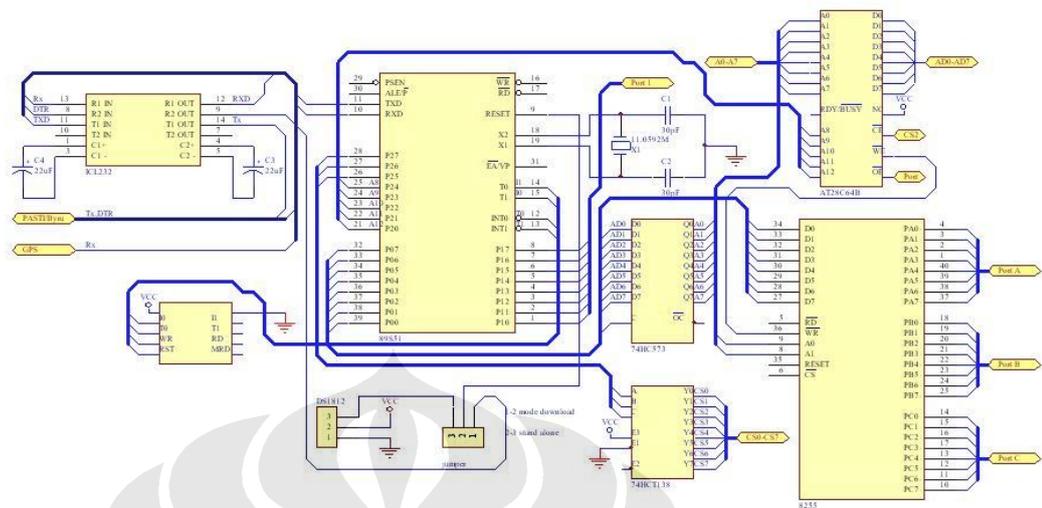
Kartu yang telah diatur sedemikian rupa oleh produsen PSN yang menonaktifkan sistem pembayaran perpemakaian.

### 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

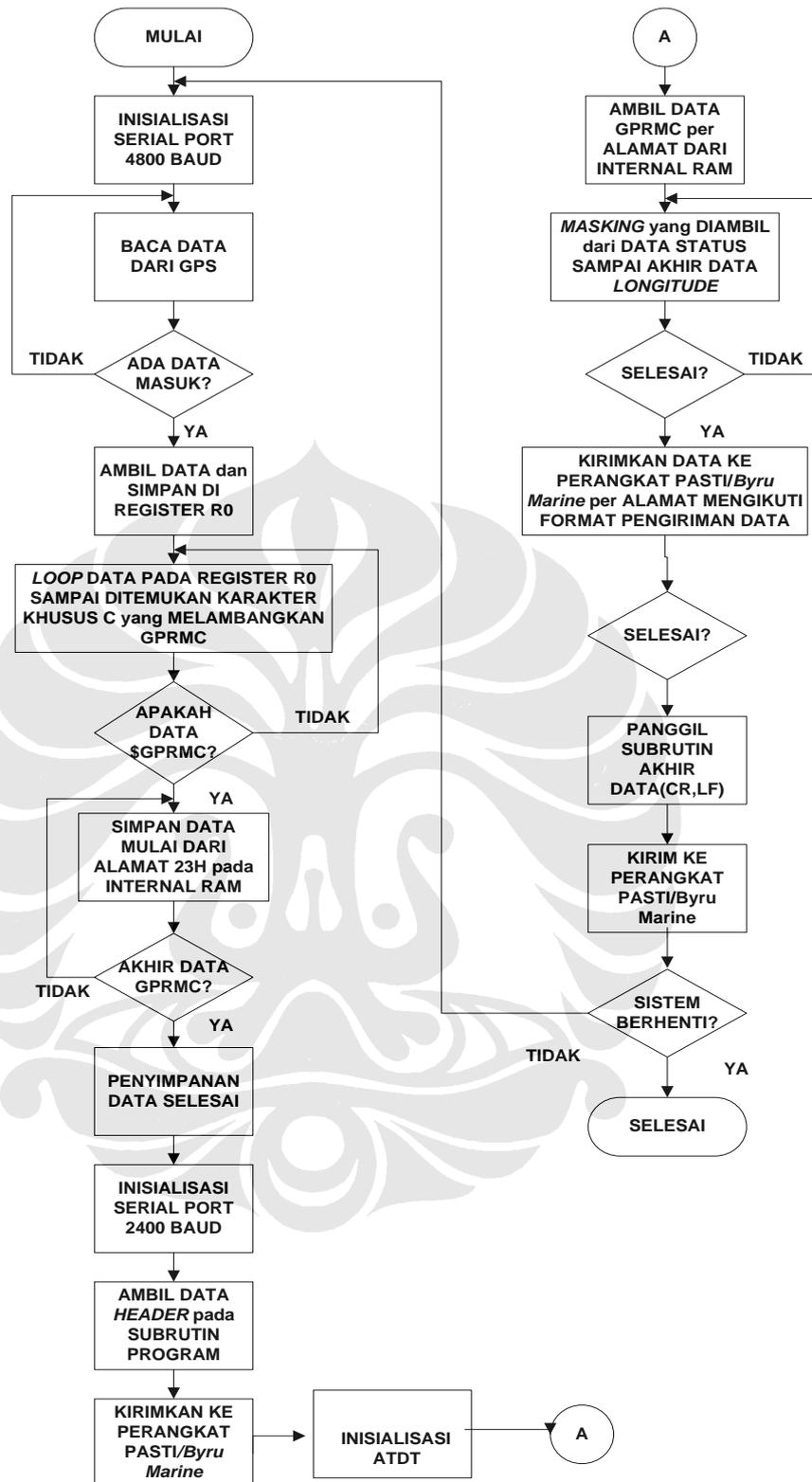
### 3.3 PERANGKAT LUNAK

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  $\langle CR \rangle \langle LF \rangle$  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*

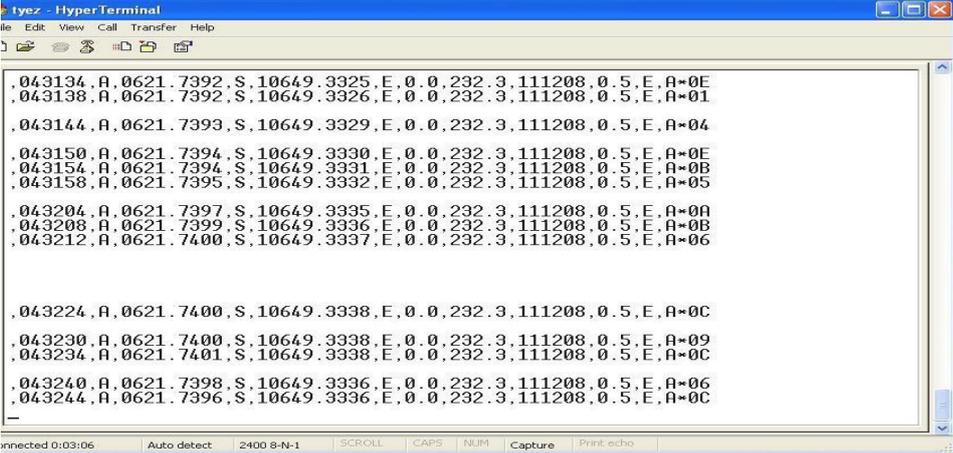
\*6A : *the checksum data*, selalu diawali dengan \*

### 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

