

## BAB III

### METODE PENELITIAN

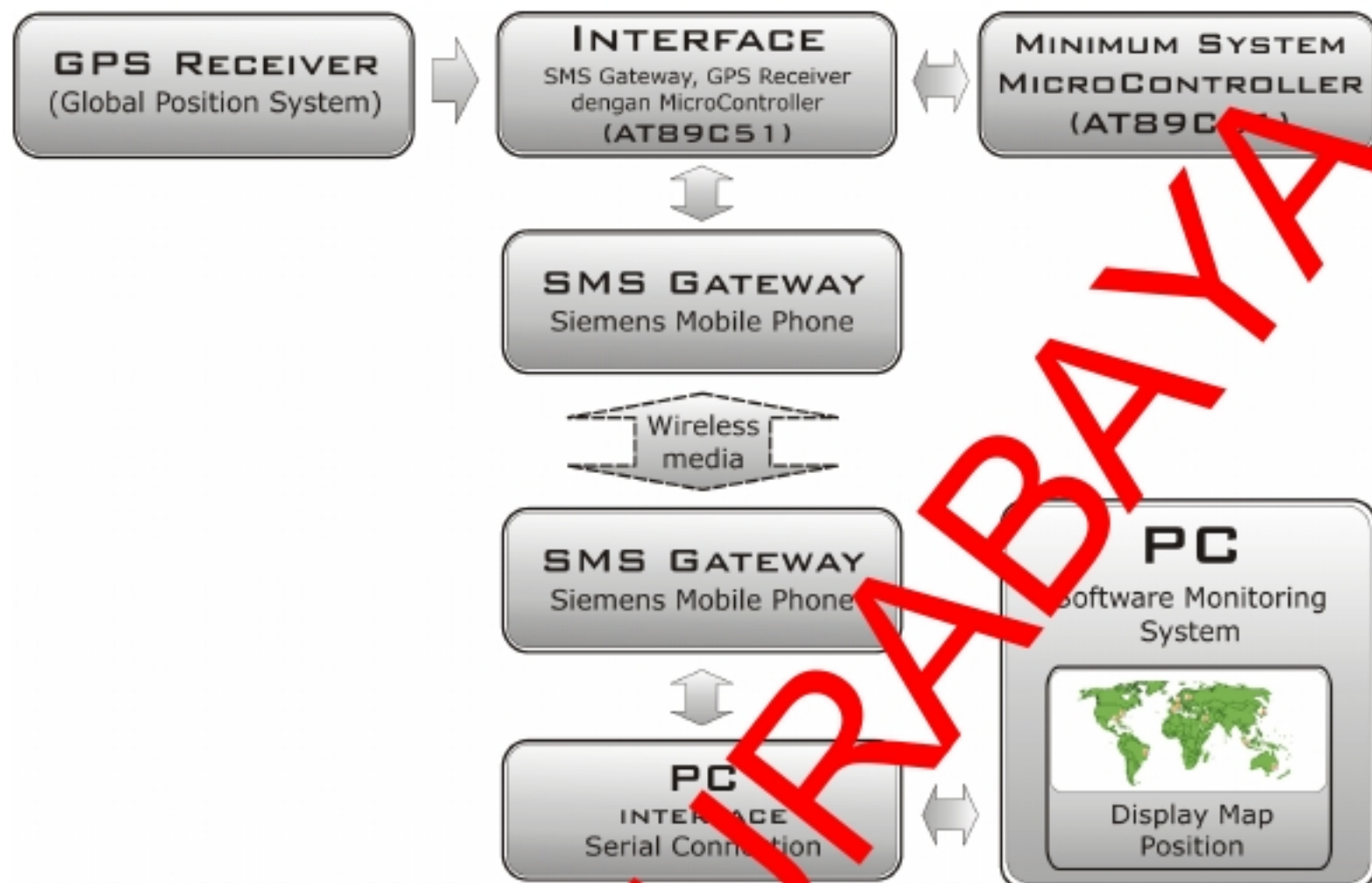
Metode penelitian yang digunakan pada pembuatan perangkat keras dan perangkat lunak yaitu dengan studi kepustakaan. Dengan cara ini penulis berusaha untuk mendapatkan dan mengumpulkan data-data, informasi, konsep-konsep yang bersifat teoritis dari buku, bahan-bahan kuliah dan internet yang berkaitan dengan permasalahan.

Dari data-data untuk memecahkan permasalahan yang diperoleh maka dilakukan perencanaan rangkaian perangkat keras. Setelah pembuatan perangkat keras, maka dilakukan pengujian terhadap perangkat keras tersebut dengan program-program kecil untuk menguji tiap-tiap bagiannya. Terakhir adalah pembuatan perangkat lunak yang sebenarnya sesuai dengan kerja perangkat keras yang diinginkan.

Perencanaan dan pembuatan yang dilakukan dalam tugas akhir ini bersifat praktis, yaitu perencanaan dan pembahasan yang mengarah pada pembuatan perangkat keras maupun perangkat lunak secara nyata dan aplikatif.

#### 3.1 Perancangan Perangkat Keras

Dalam perancangan dan pembuatan sistem penentuan lokasi kendaraan bermotor berbasis GPS dengan memanfaatkan media wireless yaitu berupa SMS sebagai komunikasi data dengan sebuah PC ini digunakan blok diagram seperti gambar berikut ini.



Gambar 3.1 Block Diagram *Minimum System Microcontroller* dan Monitoring *Software*

Dari gambar diatas dapat dilihat bahwa perancangan perangkat keras dibagi menjadi beberapa modul, diantaranya modul *minimum system Microcontroller* AT89C51 dengan RAM eksternal 8 Kbyte, modul *interface* antara SMS Gateway (*Siemens Mobile Phone*), GPS Receiver dengan *microcontroller* AT89C51.

### 3.1.1. *Minimum system AT89C51*

*Minimum system* AT89C51 dirancang dengan RAM eksternal 8 kbyte yang digunakan untuk menyimpan data-data bersifat sementara dari GPS receiver dan juga untuk menyimpan database nama-nama jalan yang akan di-download dari PC.

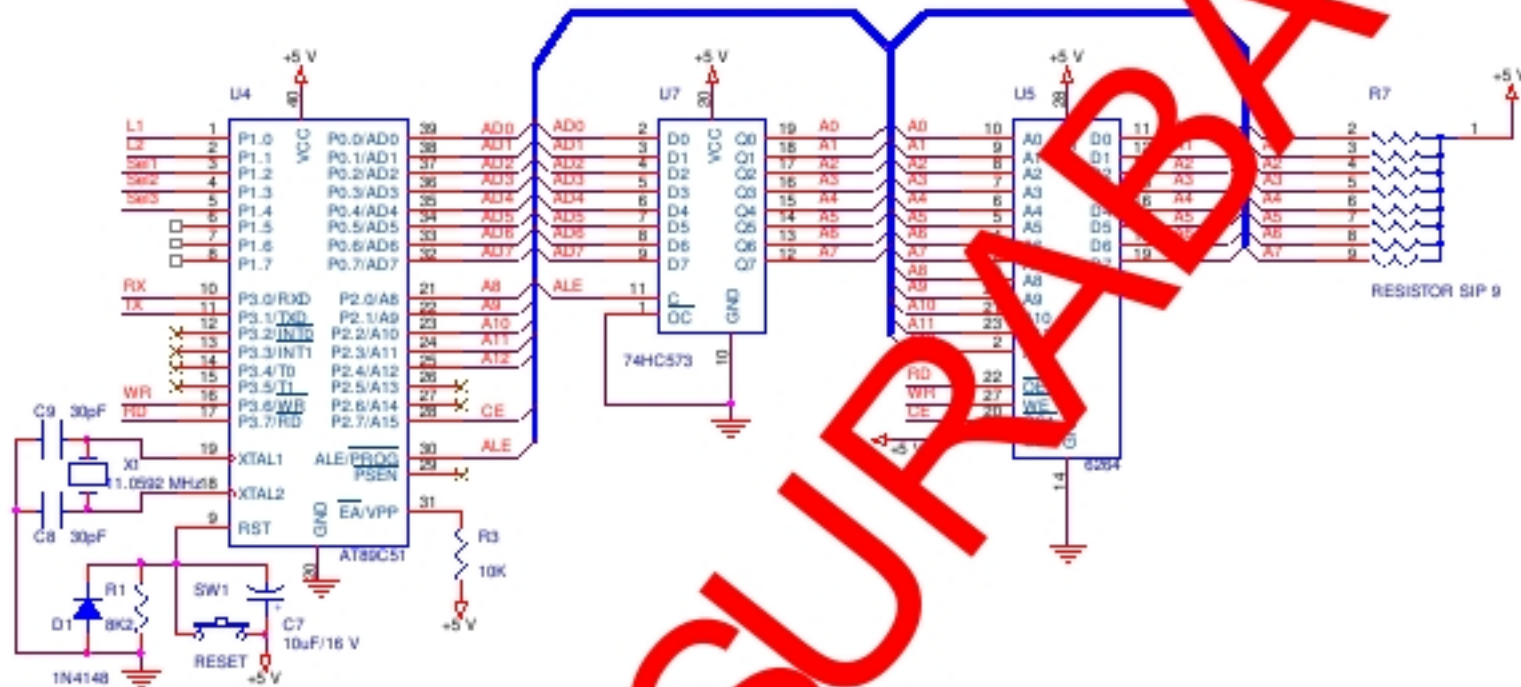
AT89C51 mempunyai 4 Kbyte Flash PEROM, yaitu ROM yang dapat ditulis ulang atau dihapus menggunakan sebuah perangkat *programmer*. Port 0 digunakan sebagai

bus data dan 8 bit bus alamat rendah dengan sistem *multiplexing addressing* (pengalamatan bergantian). Port 2 digunakan sebagai 8 bit bus alamat tinggi. Port 3 dari AT89C51 memiliki fungsi-fungsi khusus sebagai bus kontrol antara lain RXD, TXD,  $\overline{INT0}$ ,  $\overline{INT1}$ , T0, T1,  $\overline{WR}$ , dan  $\overline{RD}$ .

Port 3.0 (pin RXD) sebagai port serial input dan port 3.1 (pin TXD) sebagai *serial output* yang digunakan untuk komunikasi dengan PC dan GPS receiver. Port 3.2 (pin  $\overline{INT0}$ ) sebagai pin eksternal *interrupt* 0 yang digunakan untuk mendapatkan inputan dari tombol UP pada *Keypad*. Port 3.3 (pin  $\overline{INT1}$ ) sebagai pin eksternal *interrupt* 1 yang digunakan untuk mendapatkan inputan dari tombol DOWN pada *Keypad*. Sedangkan untuk port 3.6 (pin  $\overline{WR}$ ) digunakan untuk menulis data ke RAM. Dan port 3.7 (pin  $\overline{RD}$ ) digunakan untuk membaca data dari RAM.

Pin ALE atau *Address Latch Enable* berfungsi sebagai penunjuk keadaan bahwa port 0 sedang sebagai bus data atau sebagai bus alamat. Pin ini digunakan ENABLE dari IC 74571 (pin 11 / C) yang berfungsi sebagai *buffer* penampung alamat rendah dari bus alamat. Pin RESET berfungsi untuk mengembalikan ke keadaan awal proses dari AT89C51. Pin X1 dan X2 dihubungkan ke rangkaian osilator sebagai penentu kecepatan proses dari AT89C51. Pin  $\overline{EA}/VP$  berfungsi untuk menentukan ROM yang dipakai, eksternal atau internal ROM. Pada AT89C51 yang memiliki internal *Flash PEROM*, jadi pin ini langsung dihubungkan dengan +5 volt.

Program yang ada pada *Flash* PEROM akan dijalankan jika pada saat sistem di-reset. Pin  $\overline{EA}/VP$  berlogika satu sehingga *Microcontroller* akan dan mengeksekusi program yang ada pada *Flash* PEROMnya.



Gambar 3.2. Rangkaian *Minimum system* AT89C51 dengan RAM Eksternal 8 Kbyte

Seperti yang tampak pada gambar diatas, sistem akan mengakses RAM eksternal melalui Port 0 dan Port 2. Pin *chip select* ( $\overline{CS1}$ ) dan ( $\overline{CS2}$ ) dari RAM langsung terhubung dengan CE dan +5V, sehingga akan langsung aktif pada saat sistem dinyalakan. Pada Port 1 terdapat L1 dan L2. L1 dan L2 merupakan LED (*Light Emmitting Diode*). L1 tepatnya pada Port P1.0 dihubungkan dengan LED warna merah yakni sebagai indikator bahwa *minimum system* sedang mengakses *Port serial* yang terhubung dengan *GPS Receiver* dalam hal ini mengambil data dari *GPS Receiver*. Sedangkan pada Port P1.1 terhubung dengan LED warna putih, yakni sebagai indikator bahwa *minimum system* sedang mengakses *Serial Port* yang

terhubung dengan SMS Gateway dalam hal ini adalah *hanphone* Siemens. Sedangkan pada port P1.2, P1.3 dan P1.4 terhubung dengan selektor Mux 4051 hal ini diperlukan sebagai selektor serial *port* karena *minimum system* ini didesain mengakses 2 serial *port*, sedangkan *microcontroller* AT89C51 hanya memiliki kemampuan untuk mengakses satu serial port saja, Selanjutnya tentang perancangan Mux 4051 akan dijelaskan dalam sub bab 3.1.2.

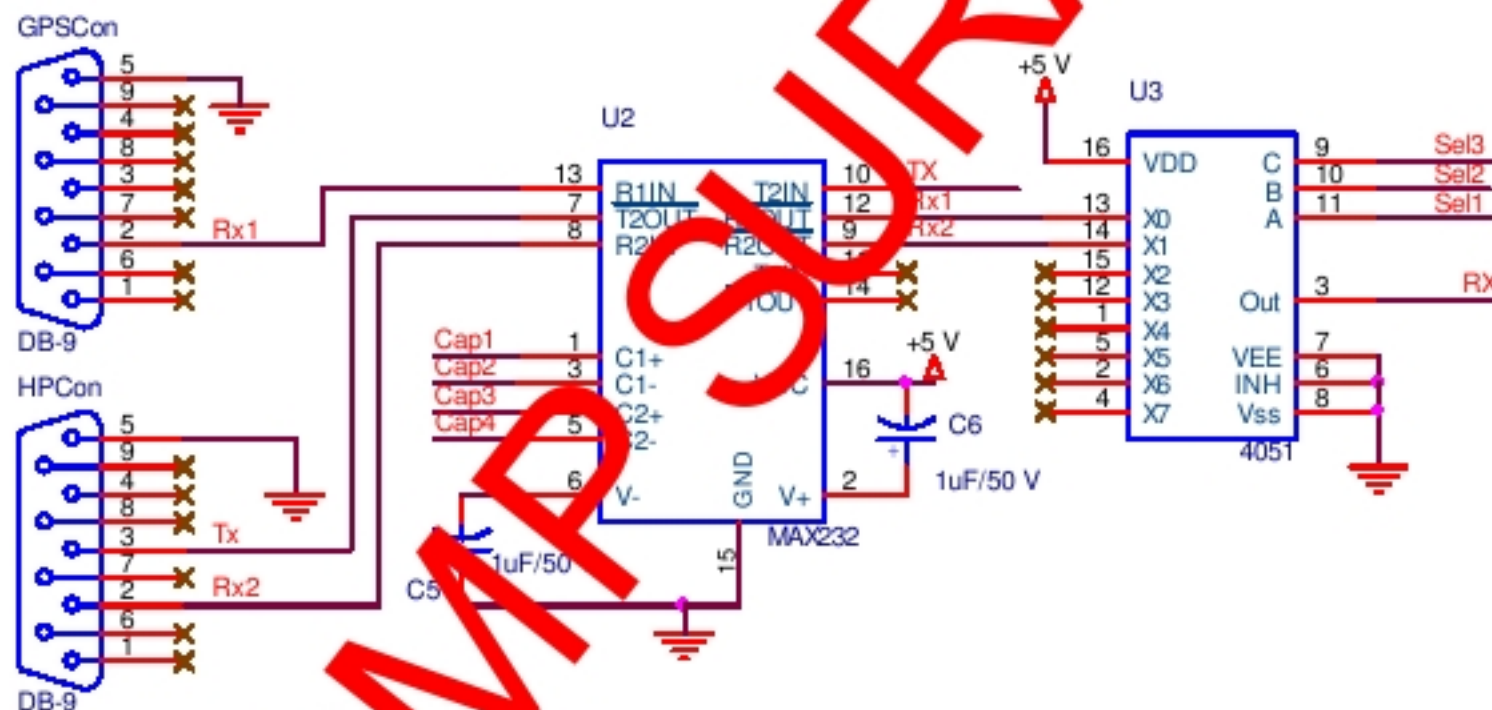
### 3.1.2. Interface Serial AT89C51 dengan SMS Gateway dan GPS Receiver

*Interface* serial AT89C51 dengan SMS Gateway dan GPS Receiver ini menggunakan saluran komunikasi serial. Untuk saluran komunikasi dengan GPS receiver sebenarnya bisa menggunakan *half-duplex*, tetapi sistem ini didesain untuk menerima data dari GPS receiver data, sehingga lebih tepatnya jika sistem ini dikatakan menggunakan *simplex* (komunikasi satu arah) dengan kecepatan transfer data (*baudrate*) 4.800 bps. Sedangkan untuk saluran komunikasi dengan SMS Gateway dalam hal ini adalah *Siemens Mobile phone* menggunakan komunikasi dua arah saling bergantian (*half-duplex*) dengan kecepatan transfer data ditentukan sesuai dengan SMS Gateway yang digunakan, untuk *Siemens* model C45 *baudrate* yang digunakan adalah 19.200 bps.

*Interface* yang diperlukan agar SMS Gateway dan GPS receiver ini dapat berkomunikasi dengan *minimum system microcontroller* AT89C51 diperlukan satu buah MUX 4051 sebagai selektor dan satu buah IC MAX 232, beberapa kapasitor dan dua buah konektor DB-9 Male.

MUX 4051 digunakan sebagai selektor untuk RX, karena sistem ini didesain mampu mengakses dua serial port tetapi *microcontroller* AT89C51 hanya mampu mengakses satu serial port saja. MUX 4051 ini memiliki 3 inputan dan menghasilkan satu *output* dari 8 *output* yang ada. 3 inputan tersebut dihubungkan dengan port P1.2 – P1.4 dari *microcontroller* AT89C51.

IC MAX232 digunakan sebagai pengubah tegangan pada *level* serial RS232 menjadi tegangan pada *level* TTL dan sebaliknya ia juga akan mengubah tegangan pada *level* TTL menjadi tegangan pada *level* serial RS232 seperti gambar dibawah ini

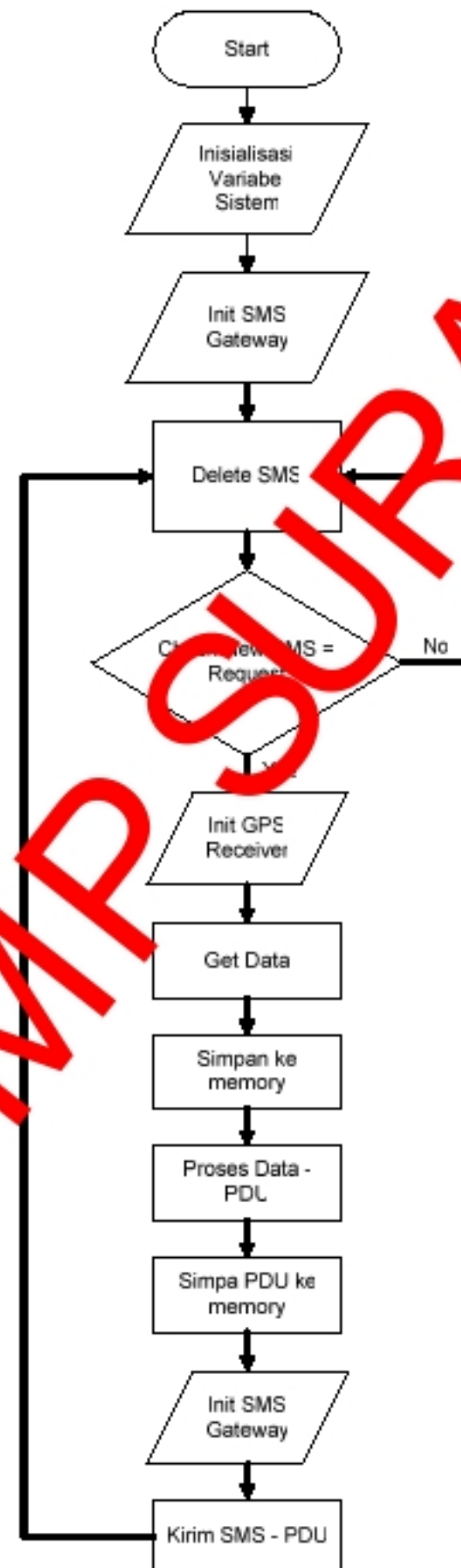


Gambar 3.3 Rangkaian *Interface* Serial AT89C51 dengan SMS Gateway dan GPS Receiver

### 3.2. Perancangan Perangkat Lunak

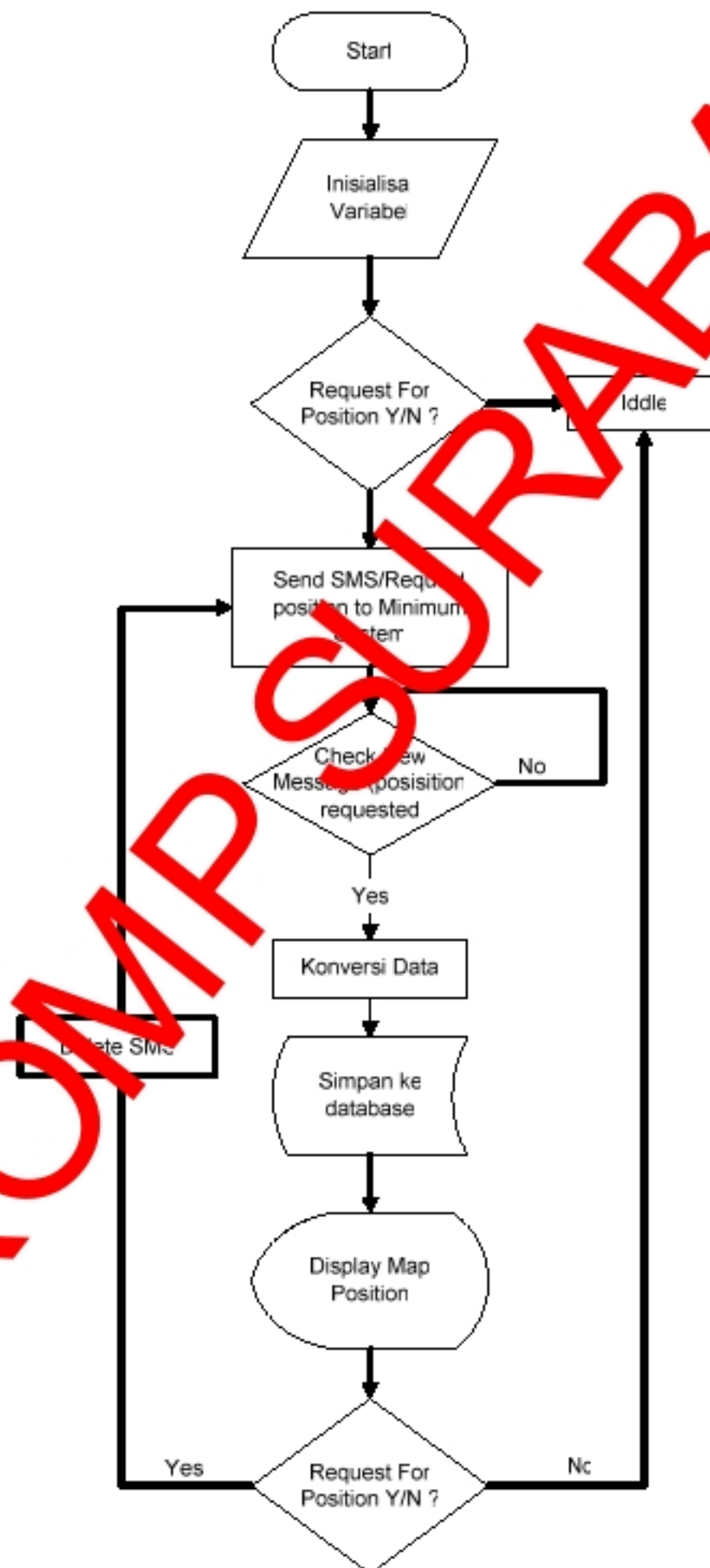
Selain *hardware* yang diperlukan pada perancangan dan pembuatan sistem penentuan lokasi kendaraan bermotor berbasis GPS dengan memanfaatkan media wireless yaitu berupa SMS sebagai komunikasi data dengan sebuah PC, juga diperlukan *software* / program pada *Microcontroller* dan PC untuk dapat bekerja.

Perancangan perangkat lunak meliputi algoritma dan program baik pada *microcontroller* itu sendiri juga pada *PC Monitoring Software*. *Flowchart* untuk program pada *Microcontroller* dijelaskan pada gambar 3.4.



Gambar 3.4 *Flowchart* untuk Program *Firmware*

Sedangkan Pada *PC Monitoring Software* algoritma pemrograman yang digunakan dijelaskan pada gambar dibawah ini.



Gambar 3.5 Flowchart untuk *PC Monitoring Software*



### 3.2.1. Program Serial

Dalam perancangan sistem ini, komunikasi serial terdiri dari dua bagian yaitu komunikasi serial *minimum system* dengan *GPS receiver* dan komunikasi serial antara *minimum system* dengan *SMS Gateway*. Kedua bagian memiliki perbedaan konfigurasi dan *register* yang digunakan. Berikut ini akan dijelaskan konfigurasi untuk masing-masing bagian.

a. komunikasi serial antara *minimum system* dengan *GPS receiver*

agar *minimum system* dapat berkomunikasi melalui serial dengan *GPS receiver* konfigurasi yang digunakan *baud rate 4800 bps*, *None parity*, *1 stop bit*.

Sedangkan untuk *register* yang digunakan adalah sebagai berikut :

- *Register TMOD (Timer Mode)* diisi dengan nilai 21 hexa

Timer 1				Timer 0			
Gate	C/T	M1	M0	Gate	C/T	M1	M0
0	0	0	0	0	0	0	1

Gate : 0, input interupsi ditentukan oleh *software*

C/T : 0, dipilih sebagai *Timer*

M1 : bit Selektor *mode Timer*

M0 : bit selektor *mode Timer*

Sesuai dengan tabel 2.9 maka untuk *Timer 0* dipilih mode 1 yaitu 16-bit timer,

hal ini dapat digunakan sebagai *timer* untuk prosedur *delay*. Sedangkan untuk

*Timer 1* dipilih *mode 2* yaitu 8-bit *auto reload* sebagai *timer* untuk

komunikasi serial *minimum system* dengan *GPS receiver*

- *Register TCON (Timer/Counter Control)* diisi dengan nilai 40 hexa

TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
0	1	0	0	0	0	0	0

TF1 : 0, penunjuk interupsi timer 1.

TR1 : 1, pengontrol *Timer* 1 dari *software*.

TF0 : 0, penunjuk interupsi timer 0.

TR0 : 0, pengontrol *Timer* 0 dari *software*.

IE1 : 0, Penunjuk jika terjadi interupsi eksternal 1.

IT1 : 0, Pengontrol interupsi eksternal 1 dari *software*.

IE0 : 0, Penunjuk jika terjadi interupsi eksternal 0.

IT0 : 0, Pengontrol interupsi eksternal 0 dari *software*.

Pada *register* ini bit TR1 diberi nilai 1 yang artinya *Timer* 1 dikontrol melalui *software*.

- *Register IE (Interrupt Enable)* diisi dengan nilai 80 hexa

EA	-	-	ES	ET1	EX1	ET0	EX0
1	0	0	0	0	0	0	0

EA : 1, Untuk mengaktifkan semua interupsi.

- : Tidak digunakan.

- : Tidak digunakan.

ES : 0, interupsi komunikasi serial.

ET1 : 0, interupsi timer 1.

EX1 : 0, interupsi eksternal 1.

ET0 : 0, interupsi timer 0.

EX0 : 0, interupsi eksternal 0.

Pada bit EA diberi nilai 1, yaitu untuk mengaktifkan semua interupsi.

- Register IP (*Interrupt Priority*) diisi dengan nilai 10 hexa

-	-	PT2	PS	PT1	PX1	PT0	PX0
0	0	0	1	0	0	0	0

- : Tidak digunakan.

- : Tidak digunakan.

- : Tidak digunakan.

PS : 1, prioritas interupsi komunikasi serial.

PT1 : 0, prioritas interupsi timer 1.

PX1 : 0, prioritas interupsi eksternal 1.

PT0 : 0, prioritas interupsi timer 0.

PX0 : 0, prioritas interupsi eksternal 0.

Pada bit PS diberi nilai 1, artinya untuk interupsi serial diberi prioritas tertinggi.

- Register SCON (*Serial Port Control*) diisi dengan nilai 50 hexa

SM0	SM1	SM2	REN	TB8	RB8	TI	RI
0	1	0	1	0	0	0	0

SM0 : 0, memilih mode komunikasi serial.

SM1 : 1, memilih mode komunikasi serial.

SM2 : 0, komunikasi multi prosesor.

REN : 1, Pengontrol penerima data dari *software*.

TB8 : 0, Tempat data ke 9 yang akan dikirim.

RB8 : 0, Tempat data ke 9 yang akan diterima.

TI : 0, Penunjuk jika terjadi pengiriman data.

RI : 0, Penunjuk jika terjadi penerimaan data.

Bit SM0 diberi nilai 0 dan bit SM1 diberi nilai 1, yang artinya adalah komunikasi serial menggunakan *mode 1*, dimana data yang dikirim mempunyai lebar 10 bit. Bit ke-0 adalah *start bit* kemudian disusul 8 *bit* data, terakhir adalah *stop bit*. Dan pada *mode 1* ini *Baud Rate* yang digunakan pada *mode* ini dapat diubah-ubah sesuai dengan nilai *register* pada TH1. Untuk bit REN diberi nilai 1 yaitu penerima data dikontrol melalui *software*.

- *Register PCON (Power Control)* diisi dengan nilai 00 hexa

SMOD	-	-	-	GF1	GF0	PD	IDL
0	0	0	0	0	0	0	0

SMOD: 0, Pengali dua kecepatan transmisi data serial (*Baud Rate*).

- : Tidak digunakan.

- : Tidak digunakan

- : Tidak digunakan

GF1 : 0, *Flag* untuk kegunaan umum.

GF0 : 0, *Flag* untuk kegunaan umum.

PD : 0, Untuk mengaktifkan mode *Power Down*.

IDL : 0, Untuk mengaktifkan *mode Idle*.

Pada *register* ini semua bit diberi nilai 0, sehingga pada bit SMOD bernilai 0, artinya tidak diperlukan pengali 2 kecepatan untuk *baud rate*.

- *Register* TH1 (*Timer 1 High Bit*) diisi dengan nilai FA hexa

Pada *register* ini nilai FA hexa merupakan variabel penentu *Baud Rate* yang digunakan pada komunikasi serial *mode 1* ini. Nilai FA didapat sesuai dengan persamaan 2.5 maka TH1 dapat dihitung dengan :

$$TH1 = 256 - \frac{K * Osc.Freq}{384 * BaudRate} \quad (3.1)$$

Sesuai dengan konfigurasi yang digunakan, maka apabila kita menggunakan *baud rate* 4800, Pada *register* PCON bit SMOD bernilai 0 maka K = 1, sehingga diperoleh nilai TH1 :

$$TH1 = 256 - \frac{1 * 11.0592}{84 * 4800}$$

$$TH1 = 256 - (0.000006 * 10^6)$$

$$TH1 = 256 - 6$$

$$TH1 = 250 \text{ atau FA hexa.}$$

Penjelasan program dapat dilihat pada listing program dibawah ini.

```
void init_serial_GPS()
{
    PCON = 0x50;           // serial mode 1, kontrol penerimaan data oleh software
    TH1 = 0xFA;           // baudrate yang digunakan 4800
    PCON = 0x00;           // SMOD = 0, K=1
    TI = 1;
    gantiSerial(2);        // Mux Selektor untuk Serial GPS receiver
    delay(100);
}
```

b. komunikasi antara *minimum system* dengan SMS gateway

Sedangkan untuk komunikasi serial *minimum system* dengan SMS gateway menggunakan konfigurasi *baud rate* 19200 bps, *None parity*, 1 *stop bit*. Sedangkan untuk *register* yang digunakan adalah sebagai berikut :

- *Register TMOD (Timer Mode)* diisi dengan nilai 21 hexa

Timer 1				Timer 0			
Gate	C/T	M1	M0	Gate	C/T	M1	M0
0	0	1	0	0	0	0	1

Gate : 0, input interupsi ditentukan oleh *software*

C/T : 0, Dipilih sebagai *Timer*

M1 : bit Selektor *mode Timer*

M0 : bit selektor *mode Timer*

Sesuai dengan tabel 2.9 maka untuk *Timer 0* dipilih *mode 1* yaitu 16-bit *timer*, hal ini dapat digunakan sebagai *timer* untuk prosedur *delay*. Sedangkan untuk *Timer 1* dipilih *mode 2* yaitu 8-bit *auto reload* sebagai *timer* untuk komunikasi serial *minimum system* dengan GPS Receiver.

- *Register TCON (Timer/Counter Control)* diisi dengan nilai 40 hexa

TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
0	1	0	0	0	0	0	0

TF1 : 0, penunjuk interupsi timer 1.

TR1 : 1, pengontrol *Timer 1* dari *software*.

TF0 : 0, penunjuk interupsi timer 0.

TR0 : 0, pengontrol *Timer* 0 dari *software*.

IE1 : 0, Penunjuk jika terjadi interupsi eksternal.

IT1 : 0, Pengontrol interupsi eksternal 1 dari *software*.

IE0 : 0, Penunjuk jika terjadi interupsi eksternal 0.

IT0 : 0, Pengontrol interupsi *eksternal* 0 dari *software*.

Pada *register* ini bit TR1 diberi nilai 1 yang artinya *Timer* 1 dikontrol melalui *software*.

- *Register IE (Interrupt Enable)* diisi dengan nilai 00 hexa

EA	-	-	ES	ET1	EX1	ET0	EX0
1	0	0	0	0	0	0	0

EA : 1, Untuk mengaktifkan semua interupsi.

- : Tidak digunakan.

- : Tidak digunakan.

ES : 0, interupsi komunikasi serial.

ET1 : 0, interupsi timer 1.

EX1 : 0, interupsi eksternal 1.

ET0 : 0, interupsi *timer* 0.

EX0 : 0, interupsi eksternal 0.

Pada bit EA diberi nilai 1, yaitu untuk mengaktifkan semua interupsi.

*Register IP (Interrupt Priority)* diisi dengan nilai 10 hexa

-	-	PT2	PS	PT1	PX1	PT0	PX0
0	0	0	1	0	0	0	0

- : Tidak digunakan.

- : Tidak digunakan.

- : Tidak digunakan.

PS : 1, prioritas interupsi komunikasi serial.

PT1 : 0, prioritas interupsi timer 1.

PX1 : 0, prioritas interupsi eksternal 1.

PT0 : 0, prioritas interupsi timer 0.

PX0 : 0, prioritas interupsi eksternal 0.

Pada bit PS diberi nilai 1, artinya untuk interupsi serial diberi prioritas tertinggi.

- Register SCON (Serial Port Control) diisi dengan nilai 50 hexa

SM0	SM1	SM2	REN	TB8	RB8	TI	RI
0	1	0	1	0	0	0	0

SM0 : 0, memilih *mode* komunikasi serial.

SM1 : 1, memilih *mode* komunikasi serial.

SM2 : 0, komunikasi multi prosesor.

REN : 1, Pengontrol penerima data dari *software*.

TB8 : 0, Tempat data ke 9 yang akan dikirim.

RB8 : 0, Tempat data ke 9 yang akan diterima.

TI : 0, Penunjuk jika terjadi pengiriman data.

RI : 0, Penunjuk jika terjadi penerimaan data.



Bit SM0 diberi nilai 0 dan bit SM1 diberi nilai 1, yang artinya adalah komunikasi serial menggunakan *mode 1*, dimana data yang dikirim mempunyai lebar 10 bit. Bit ke-0 adalah *start bit*, kemudian disusul 8 bit data, terakhir adalah *stop bit*. Dan pada *mode 1* ini *Baud Rate* yang digunakan pada *mode* ini dapat diubah-ubah sesuai dengan nilai *register* pada TH1. Untuk bit REN diberi nilai 1 yaitu penerimaan data dikontrol melalui *software*.

- *Register PCON (Power Control)* diisi dengan nilai 80 hexa

SMOD	-	-	-	GF1	GF0	PD	IDL
1	0	0	0	0	0	0	0

SMOD: 1, Pengali dua kecepatan transmisi data serial (*Baud Rate*).

- : Tidak digunakan.

- : Tidak digunakan

- : Tidak digunakan

GF1 : 0, *Flag* untuk kegunaan umum.

GF0 : 0, *Flag* untuk kegunaan umum.

PD : 0, Untuk mengaktifkan *mode Power Down*.

IDL : 0, Untuk mengaktifkan *mode Idle*.

Pada *register* ini bit SMOD bernilai 1, artinya diperlukan pengali 2 kecepatan untuk *baud rate*.

*Register TH1 (Timer 1 High Bit)* diisi dengan nilai FD hexa

Pada register ini nilai FD hexa merupakan variabel penentu *Baud Rate* yang digunakan pada komunikasi serial *mode 1* ini. Nilai FD hexa didapat sesuai

dengan persamaan 3.1. Untuk konfigurasi yang digunakan, maka apabila kita menggunakan *baud rate* 19200, Pada *register* PCON bit SMOD bit nilai 1 maka  $K = 2$ , sehingga diperoleh nilai TH1 :

$$TH1 = 256 - \frac{2 * 11.0592}{384 * 4800}$$

$$TH1 = 256 - (0.000003 * 10^6)$$

$$TH1 = 256 - 3$$

$$TH1 = 253 \text{ atau FD hexa.}$$

Penjelasan program dapat dilihat pada listing program dibawah ini.

```

void init_serial_HP()
{
    SCON = 0x50;           // Serial mode 1, Kontrol penerimaan data oleh software
    TH1 = 0xFD;           // baud rate di set ke 19200
    PCON = 0x80;         // SMOD = 1, K = 2, Pengali kecepatan baudrate (2x)
    TR1 = 1;
    gantiSerial(1);      // Mux Selektor untuk Serial HandPhone
    delay(100);
}

void gantiSerial(unsigned int x)
{
    if(x == 1)
        TR1 = 0x00;      // Mux Selektor untuk Serial HandPhone
    else if ( x == 2)
        TR1 = 0x01;      // Mux Selektor untuk Serial GPS receiver
}

```

Program penerimaan data melalui *serial port* dilakukan dengan memanfaatkan *interrupt serial* dimana apabila ada data yang akan masuk akan disimpan secara langsung kedalam RAM. Dapat dijelaskan pada listing program berikut :

```
void getString() interrupt 4 using 1
```

```
{
    if (RI==1) // start received data
    {
        sbuffer[cnt] = SBUF; // data yang diterima disimpan dalam RAM
        cnt++;
        RI = 0; // reset RI
    }
}
```

### 3.2.2. Program Check SMS

Program ini digunakan untuk melihat isi sms baru yang masuk diterima sms gateway. Apabila terdapat pesan baru pada SMS gateway maka *minimum system* akan secara otomatis membaca isi pesan dan membandingkannya apakah pesan tersebut berisi permintaan lokasi dari PC. Jika salah *minimum system* akan menghapus pesan tersebut dan melakukan prosedur check SMS berulang-ulang hingga mendapatkan pesan yang sesuai. Jika benar maka *minimum system* melakukan proses pengambilan data dari GPS receiver. Dapat dijelaskan pada listing program berikut :

```
bit check_SMS()
```

```
{
    unsigned int j;
    bit fl;
    fl = 0;
    send_AT();
    delay(500);
    cnt = 0;
    send_ATCMGL();
    TR = 0;
    ES = 1;
    delay(2000);
    ES = 0;
    delay(500);
    for(j=0;j<500;j++)
```

```

{
    if ((sbuffer[j] == '0') && (sbuffer[(j+1)] == 'C') && (sbuffer[(j+2)] == '9') &&
        (sbuffer[(j+3)] == '1'))
    {
        for(k=0;k<16;k++)
        {
            if (sbuffer[j+k] == HP_PENGINIRIM[k])
                fl = 1;
            else
            {
                fl = 0;
                delete_SMS();
                delay(2000);
                break;
            }
        }
    }
    if (fl == 1)
    for(k=0;k<16;k++)
    {
        if (sbuffer[j+k-34] == REQUEST_PDU[k])
            fl = 1;
        else
        {
            fl = 0;
            delete_SMS();
            delay(2000);
            break;
        }
    }
    break;
}
return (fl);
}

void send_AT() // Kirim perintah AT
{
    unsigned int j;
    for(j=0;j<2;j++)
    {

```

```

        TI = 0;
        SBUF = cmd_AT[j];
        while (!TI);
    }
    TI = 0;
    SBUF = 0x0D;
    while (!TI);
}

void send_ATCMGL() // kirim perintah AT+CMGL = 0
{
    unsigned int j;
    for(j=0;j<9;j++)
    {
        TI = 0;
        SBUF = cmd_ATCMGL[j];
        while (!TI);
    }
    TI = 0;
    SBUF = 0x0D;
    while (!TI);
}

void delete_SMS() // Hapus SMS dengan nomor referensi = 0
{
    unsigned int j;
    send_AT();
    delay(500);
    for(j=0;j<9;j++)
    {
        TI = 0;
        SBUF = cmd_ATCMGD[j];
        while (!TI);
    }
    TI = 0;
    SBUF = 0x0D;
    while (!TI);
    delay(1000);
}

```

### 3.2.3. Program Pengambilan Data dari GPS

Program ini digunakan untuk mengambil data dari GPS *receiver* dan mengubahnya menjadi format posisi yang siap akan dikirimkan melalui SMS *gateway*. Dapat dijelaskan pada listing program berikut ::

```

void getGPS() // menyimpan data interup serial dari GPS receiver
{
    TI = 0;
    cnt = 0;
    flag = 0;
    clear_buff();
    while(!RI);
    ES = 1;
    delay(4000);
    ES = 0;
}

void scan_posisi() // Ambil data lokasi S dan E
{
    int j,i;
    for (j=0;j<500;j++)
    {
        if ((sbuffer[j] == 0x10) && (sbuffer[j+13] == 'S') &&
            (sbuffer[j+24] == 'E'))
        {
            flag = 1;
            for (i=0;i<54;i++)
            {
                GPGLL[i] = sbuffer[(j+i)];
                delay(10);
            }
            break;
        }
        else
        {
            flag = 0;
        }
    }
}

```

```

}
void cat_data()
{
    unsigned int k;
    if (flag == 1)
    {
        tbuffer[0] = GPGLL[14];
        tbuffer[1] = GPGLL[15];
        tbuffer[2] = GPGLL[16];
        tbuffer[3] = GPGLL[17];
        tbuffer[4] = GPGLL[18];
        tbuffer[5] = GPGLL[19];
        tbuffer[6] = GPGLL[20];
        tbuffer[7] = GPGLL[22];
        tbuffer[8] = GPGLL[23];
        tbuffer[9] = GPGLL[24];
        tbuffer[10] = GPGLL[25];
        tbuffer[11] = GPGLL[26];
        tbuffer[12] = GPGLL[27];
        tbuffer[13] = GPGLL[28];
        tbuffer[14] = GPGLL[29];
    }
    else if (flag == 0)
    {
        for (k=0;k<15;k++)
            tbuffer[k] = '0';
    }
}

```

### 3.2.4. Program konversi data ke PDU

Program ini digunakan untuk merubah bentuk data dari GPS *receiver* ke dalam format PDU sehingga siap dikirimkan melalui SMS *Gateway*. Dapat dijelaskan pada listing program berikut :

```
void get_PDU_data()
```

```
{
    data_2_hex();
    delay(100);
    hex_2_temp();
    delay(100);
    hex_2_PDU();
}
```

```
void data_2_hex()
```

```
{
    char Tmp1,Tmp2,Tmp3,n;
    int x,Ci,j,i;
    Tmp2=0x00;
    n=0x00;
    x=1;
    Ci=6;
    j=0;
    for(i=0; i<14; i++)
    {
        Tmp1 = _cror_((tbuffer[i]-Tmp2),6-Ci);
        Tmp2 = tbuffer[i+1] & x;
        Tmp3 = _crol_(Tmp2,i+1);
        delay(10);
        buff_PDU[j] = Tmp1+Tmp3;
        j++;n++;
        if(Ci==0)
        {
            i++;
            i++;
            Tmp2=0x00;
            x=1;
            Ci=6;
        }
        else
        {
            Ci--;
            x = x | _crol_(x,1);
        }
    }
}
```



```

}
void hex_2_temp()
{
    int i;
    for(i=0;i<14;i++)
    {
        the_PDU[((i*2)+0)] = buff_PDU[i] & 0xF0;
        delay(10);
        the_PDU[((i*2)+1)] = buff_PDU[i] & 0x0F;
        delay(10);
    }
}
void Hex_2_PDU()
{
    int i;
    for(i=0;i<28;i++)
    {
        if(the_PDU[i] == 0x00) {the_PDU[i] = '0';}
        else if((the_PDU[i] == 0x01) | (the_PDU[i] == 0x10))
            {the_PDU[i] = '1';}
        else if((the_PDU[i] == 0x02) | (the_PDU[i] == 0x20))
            {the_PDU[i] = '2';}
        else if((the_PDU[i] == 0x03) | (the_PDU[i] == 0x30))
            {the_PDU[i] = '3';}
        else if((the_PDU[i] == 0x04) | (the_PDU[i] == 0x40))
            {the_PDU[i] = '4';}
        else if((the_PDU[i] == 0x05) | (the_PDU[i] == 0x50))
            {the_PDU[i] = '5';}
        else if((the_PDU[i] == 0x06) | (the_PDU[i] == 0x60))
            {the_PDU[i] = '6';}
        else if((the_PDU[i] == 0x07) | (the_PDU[i] == 0x70))
            {the_PDU[i] = '7';}
        else if((the_PDU[i] == 0x08) | (the_PDU[i] == 0x80))
            {the_PDU[i] = '8';}
        else if((the_PDU[i] == 0x09) | (the_PDU[i] == 0x90))
            {the_PDU[i] = '9';}
        else if((the_PDU[i] == 0x0A) | (the_PDU[i] == 0xA0))
            {the_PDU[i] = 'A';}
        else if((the_PDU[i] == 0x0B) | (the_PDU[i] == 0xB0))

```

```

        {the_PDU[i] = 'B';}
    else if((the_PDU[i] == 0x0C) | (the_PDU[i] == 0xC0))
        {the_PDU[i] = 'C';}
    else if((the_PDU[i] == 0x0D) | (the_PDU[i] == 0xD0))
        {the_PDU[i] = 'D';}
    else if((the_PDU[i] == 0x0E) | (the_PDU[i] == 0xE0))
        {the_PDU[i] = 'E';}
    else {the_PDU[i] = 'F';}
    delay(10);
}
}

```

### 3.2.5. Program Kirim SMS

Program ini digunakan untuk mengirim pesan dari data berbentuk PDU melalui SMS Gateway. Dapat dijelaskan pada listing program berikut :

```

void SendSms()
{
    unsigned int j;

    for(j=0;j<42;j++)
    {
        TI = 0;
        SBUF = TEMPL[j];
        while(!TI);
    }
    for(j=0;j<28;j++)
    {
        TI = 0;
        SBUF = the_PDU[j];
        while(!TI);
    }
    TI = 0;
    SBUF = 0x1A;
    while(!TI);
    delay(100);
}

```

```

void send_ATCMGS()
{
    unsigned int j;
    for(j=0;j<10;j++)
    {
        TI = 0;
        SBUF = cmd_ATCMGS[j];
        while (!TI);
    }
    TI = 0;
    SBUF = 0x0D;
    while (!TI);
}

```

### 3.2.6. Program Delay

Dalam perancangan ini diperlukan prosedur *Delay* hal ini dikarenakan agar terdapat penundaan antara satu perintah dengan perintah lainnya. *Delay* yang dimaksud dapat dilakukan dengan memanfaatkan *Timer 0*. Pada *register* ini diberi nilai FC17 hexa dimana TH0 = FC hexa dan TL0 = 17 hexa. ini dijelaskan bagaimana cara mendapatkan *delay*

$$Delay = (65535 - TH0TL0) * 1/1000000 \quad (3.2)$$

Sesuai dengan persamaan diatas maka kita akan mencari *delay* dengan nilai TH0 dan TL0 yang kita dapatkan melalui software yang dibuat, sehingga :

$$Delay = (65535 - FC17) * 1/1000000$$

$$Delay = 1000 * 1/1000000$$

Sehingga kita dapatkan *delay* sebesar 0,001 atau 1 *mili second*.

Sehingga apabila dalam program apabila kita menginginkan *delay* waktu sebesar 1 detik maka prosedur *delay* harus dilakukan sebanyak 1000 kali (1 *Second* = 1000 *miliSecond*). Dapat dijelaskan pada listing program berikut :

```

void delay(unsigned int lama)
{
    int i;
    for (i=0;i<lama;i++)
    {
        TH0=0Xfc;           // satuan dalam 1 milisecon
        TL0=0X17;
        TR0=1;             // Start timer
        while(TF0==0);     // Flag Timer 0 overload
        TR0=0;             // Stop timer
        TF0=0;             // reset Flag
    }
}

```

### 3.2.7. Program Serial PC dengan SMS Gateway

Program ini ditulis dengan bahasa pemrograman Visual Basic, dimana program ini digunakan untuk merancang proses komunikasi antara SMS Gateway dengan PC dan keseluruhan *Mapping System*. Sebenarnya logika yang digunakan sama sama dengan pemrograman yang dilakukan pada *minimum system*. Untuk komunikasi serial Visual basic memerlukan file OCX yaitu MSComm6.OCX. Dapat dijelaskan pada listing program berikut :

```

Private sub in Serial
Dim MSComm1 As Object
Set MSComm1 = CreateObject("MSCOMMLIB.MSCOMM")
On Error GoTo err_handler
With MSComm1
    .CommPort = 1           'Comm Port yang digunakan
    .Settings = "19200,N,8,1" 'konfigurasi Baudrate
    .InputLen = 0
End With
err_handler:
If Err.number = 8002 Or Err.number = 8005 Then Set MSComm1 = Nothing
End Sub

```

### 3.2.8. Program Kirim Pesan Request Posisi

Program ini digunakan untuk mengirimkan pesan berupa *key code* yang dikenali oleh *minimum system* sebagai permintaan (*request*) untuk mengirimkan posisi berdasarkan data dari GPS Receiver. Dilakukan dengan prosedur :

#### Private Sub KirimSMS()

```

Dim MSComm1 As Object
Set MSComm1 = CreateObject("MSCOMMLIB.MSCOMM")
On Error GoTo err_handler
If CbxPonsel.Text = "" Then
    MsgBox "Masukan jenis ponsel yang digunakan sebagai device!", vbInformation
    CbxPonsel.SetFocus
ElseIf CbxPort.Text = "" Then
    MsgBox "masukan port komunikasi serial!", vbInformation
    CbxPort.SetFocus
End If
With MSComm1
    .CommPort = IIf(CbxPort.Text = "COM1", "1", "2")
    .Settings = IIf(CbxPonsel.Text = "C45", "115200", "57600") & ",N,8,1"
    .InputLen = 0
    .PortOpen = True
End With
Dim waktu As Date
waktu = Now()
buffer0$ = ""
buffer1$ = ""
buffer2$ = ""
buffer3$ = ""
Do
    MSComm1.Output = "AT" & Chr$(13)
    Do
        MSComm1.Events
        buffer0$ = buffer0$ & MSComm1.Input
    Loop Until InStr(buffer0$, "OK") Or InStr(buffer0$, "ERROR") Or Now > _
        DateAdd("s", 10, waktu)
    Loop Until InStr(buffer0$, "OK") Or Now > DateAdd("s", 10, waktu)
Do

```

```

        MSComm1.Output = "AT+CMGD=1" & Chr$(13)
    Do
    DoEvents
        buffer1$ = buffer1$ & MSComm1.Input
    Loop Until InStr(buffer1$, "OK") Or InStr(buffer1$, "ERROR") Or Now > _
    DateAdd("s", 10, waktu)
    Loop Until InStr(buffer1$, "OK") Or Now > DateAdd("s", 10, waktu)
    Do
        MSComm1.Output = "ATE1" & Chr$(13)
    Do
    DoEvents
        buffer$ = buffer$ & MSComm1.Input
    Loop Until InStr(buffer$, "OK") Or InStr(buffer$, "ERROR") Or Now > _
    DateAdd("s", 10, waktu)
    Loop Until InStr(buffer$, "OK") Or Now > DateAdd("s", 10, waktu)
    Do
        MSComm1.Output = "AT+CMGS=" & HideCMGS.Text & Chr$(13)
    Do
    DoEvents
        buffer2$ = buffer2$ & MSComm1.Input
    Loop Until InStr(buffer2$, ">") Or InStr(buffer2$, "ERROR")
    Loop Until InStr(buffer2$, ">")
    Do
        MSComm1.Output = HidePLN.Text & Chr$(26)
    Do
    DoEvents
        buffer3$ = buffer3$ & MSComm1.Input
    Loop Until InStr(buffer3$, "OK") Or InStr(buffer3$, "ERROR")
    Loop Until InStr(buffer3$, "OK")
MSComm1.PortOpen = False
Set MSComm1 = Nothing
If InStr(buffer$, "OK") Then
    MsgBox "Message Send, And Wait for Minimum system Response"
    FrmStatus.Picture = LoadResPicture(102, vbResBitmap)
    DtaTanggal.Text = Format(Date, "Short Date")
    DtaJamKirim.Text = Format(Time, "h:m:s")
    CmdRequestPositioning.Enabled = False
    FrmMonitoring.SSTab1.Tab = 3
Else

```

```

        MsgBox "Requesting Error"
    End If

    err_handler:
        If Err.number = 8002 Or Err.number = 8005 Then Set MSComm1 = Nothing
End Sub

```

### 3.2.9. Program *Check SMS* dari *minimum system*

Program ini digunakan untuk melihat isi pesan dan mengambil data dari pesan tersebut yang berupa posisi yang dikirim oleh *minimum system* kemudian disimpan dalam database dan ditampilkan kedalam Map. Dapat dijelaskan pada listing program berikut :

```

Private Sub CmdRequestPositioning_Click()
    ClearDta
    ImgStatus.Picture = LoadResPicture(101, vbResPicture)
    InisialPDU
    KirimSMS
    Timer1.Enabled = True
    Timer1.Interval = 1000
End Sub

Private Sub CheckSMS()
    Dim pjsmskosong
    Dim MSComm1 As Object
    Set MSComm1 = CreateObject("MSCOMMLIB.MSCOMM")
    On Error GoTo err_handler
    With MSComm1
        .CommPort = IIf(CbxPort.Text = "Com1", "1", "2")
        .Settings = IIf(CbxPonsel.Text = "C45", "19200", "57600") & ",N,8,1"
        .InputLen = 0
        .PortOpen = True
    End With
    Dim waktu As Date
    waktu = Now()
    buffer5$ = ""

```

```

buffer4$ = ""
Do
MSComm1.Output = "ATE1" & Chr$(13)
Do
DoEvents
    buffer5$ = buffer5$ & MSComm1.Input
Loop Until InStr(buffer5$, "OK") Or InStr(buffer5$, "ERROR") Or Now > _
DateAdd("s", 10, waktu)
Loop Until InStr(buffer5$, "OK") Or Now > DateAdd("s", 10, waktu)
Do
    MSComm1.Output = "AT+CMGL=0" & Chr$(13)
Do
DoEvents
    buffer4$ = buffer4$ & MSComm1.Input
Loop Until InStr(buffer4$, "OK") Or InStr(buffer4$, "ERROR")
Loop Until InStr(buffer4$, "OK")
TxtLenCMGL2.Text = buffer4$
pjsmskosong = Len(buffer4$)
TxtLenCMGL.Text = pjsmskosong
MSComm1.PortOpen = False
err_handler:
    If Err.number = 8002 Or Err.number = 8005 Then Set MSComm1 = Nothing
End Sub

Private Sub Timer1_Timer                                'Check SMS tiap 1 second
CheckSMS
End Sub

Private Sub TxtLenCMGL_Change()                          'Stop apabila terdapat SMS baru
If TxtLenCMGL.Text < 17 Then
    Timer1.Enabled = False
    MsgBox "Message Received, Location Defined"
    TxtKonvertPduToString.Text = ""
    ClearDtaconv
    TxtLenCMGL3.Text = Mid(TxtLenCMGL2.Text, 85, 28)
    ImgStatus.Picture = LoadResPicture(103, vbResBitmap)
    DtaJamTerima.Text = Format(Time, "h:m:s")
    KonvertPduToString
    DtaLintangSelatan.Text = Mid(TxtKonvertPduToString.Text, 1, 2) & Chr$(176) &
    Mid(TxtKonvertPduToString.Text, 3, 2) & Chr$(46) & Mid(TxtKonvertPduToString.Text, 5, 3)

```



```

DtaBujurTimur.Text = Mid(TxtKonvertPduToString.Text, 8, 3) & Chr$(176) &
Mid(TxtKonvertPduToString.Text, 11, 2) & Chr$(46) & Mid(TxtKonvertPduToString.Text, 13,
3)
DtaPointS.Text = Mid(TxtKonvertPduToString, 1, 7)
DtaPointE.Text = Mid(TxtKonvertPduToString, 8, 8)
CmdRequestPositioning.Enabled = True
CmdMAP.Enabled = True
SimpanDtaRequest
Timer2.Enabled = True
Timer2.Interval = 5000
TxtStartStop.Text = "start"
End If
End Sub

```

### 3.2.10. Program Menampilkan data kedalaman Map

Program ini digunakan untuk menampilkan data berupa posisi yang diterima dari *minimum system* kedalam *Map* yang dibangun. Dapat dijelaskan pada listing program berikut :

```

Private Sub Go2Map()
  Dim Counter As Integer
  Dim Workarea(3000) As String

  ProgressBar1.Min = 0
  ProgressBar1.Max = Round(Workarea)
  ProgressBar1.Visible = True

  'Set the Progress's Value to Min.
  ProgressBar1.Value = ProgressBar1.Min

  'Loop through the array.
  For Counter = LBound(Workarea) To UBound(Workarea)
    'Set initial values for each item in the array.
    Workarea(Counter) = "Initial value" & Counter
    ProgressBar1.Value = Counter
  Next Counter

```

```

If (DtaPointE.Text = "00000000") And (DtaPointS.Text = "00000000") Then
    MsgBox "Position Error, Communication Signal with GPS Satelite may be lost. Send more
Message to define more location"
    FrmMonitoring.SSTab1.Tab = 2
ElseIf (IsNumeric(DtaPointE.Text)) And (IsNumeric(DtaPointS.Text)) = True Then
    Call Calculated_Map
Else: MsgBox "Position Error, Communication Signal with GPS Satelite may be lost. Send more
Message to define more location"
    FrmMonitoring.SSTab1.Tab = 2
End If

```

**End Sub**

**Private Sub Calculated\_Map()**

```

Dim PixPointX, PixPointY As Integer
PixPointX = (DtaPointE.Text - 11246275) / 1.66
PixPointY = (DtaPointS.Text - 717829) / 1.66
'-- Set X Location to the center of the map --
If PixPointX < 325 Then
    HScroll1.Value = 0
ElseIf PixPointX > 1045 Then
    HScroll1.Value = 100
Else: HScroll1.Value = ((PixPointX - 325) / 720) * 100
End If
'-- Set Y Location to the center of the map --
If PixPointY < 270 Then
    VScroll1.Value = 0
ElseIf PixPointY > 1770 Then
    VScroll1.Value = 100
Else: VScroll1.Value = ((PixPointY - 270) / 1450) * 100
End If

ProgressBar1.Value = ProgressBar1.Min
Timer3.Interval = 100
DrawCircle.Left = (PixPointX - 17)
DrawCircle.Top = (PixPointY - 17)

```

**End Sub**

### 3.2.11. Perhitungan dan Cara Memperoleh Data Awal Koordinat Lokasi

Data awal diperoleh dengan menggunakan metode sampling *Cluster* yang ini dimungkinkan mengingat populasi yang diambil sifatnya global dan sangat besar yaitu dunia. Metode *cluster* ini dimungkinkan untuk mengurangi biaya penarikan sampel populasi yang tersebar pada areal geografis yang luas serta mengingat beberapa keterbatasan :

- Tidak memungkinkan untuk mengambil sampel diluar Surabaya ataupun di luar negeri dikarenakan keterbatasan waktu, biaya.
- Mendapatkan peta yang benar-benar akurat yang digunakan sebagai bahan dasar untuk melakukan pengukuran nilai perbandingan untuk pemetaan posisi.

Sehingga sampel yang didapat dilakukan dengan jalan survei secara langsung dengan berdasar pada peta yang ditetapkan sebagai bahan perhitungan (Peta SURABAYA dan PERKEMBANGANNYA yang diterbitkan dan dicetak oleh P.T. Karya Pembina Swajaya, Jl. Urip Sumartono 11 Surabaya, dengan skala 1 : 26.000, tahun produksi 2004) menggunakan GPS *Receiver* dari Garmin Rino 120.

Tabel 3.1 Data Awal Lokasi

Lokasi	Bujur Timur	Lintang Selatan
A	112°47.005	07°18.085
B	112°47.012	07°18.565
C	112°46.563	07°19.112
D	112°46.848	07°19.115
E	112°46.637	07°19.112
F	112°46.704	07°18.612
G	112°46.748	07°18.049
H	112°46.840	07°18.676
I	112°46.334	07°19.867
J	112°47.218	07°19.909

Dari data awal tersebut, 10 lokasi dianggap sebagai sampel mewakili seluruh populasi, sehingga dapat dilakukan perhitungan untuk mencari nilai perbandingan antara keadaan sesungguhnya dengan peta yang discan dalam format JPG. Dasar dari perhitungan ini adalah dengan menyampling beberapa titik pada keadaan sesungguhnya berdasarkan pada peta yang ditetapkan sebagai dasar pengukuran. Kemudian beberapa titik tersebut dicari jarak dengan satuan *pixel* jarak *pixel* dengan data koordinat sesungguhnya dibandingkan sehingga kita memperoleh nilai faktor. Nilai faktor inilah yang nantinya akan dijadikan sebagai referensi sistem dalam proses pemetaan. Berikut ini tabel nilai perbandingan antara keadaan sesungguhnya dengan peta JPG dengan ukuran 1368 X 1991 *pixel*.

Tabel 3.2 Nilai perbandingan

No	Koordinat Lokasi			Selisi <i>Pixel</i>	Selisi aktual	Perbandingan	Faktor
1	A	E	112°47.005	Width :	0	7	~
		S	07°18.085	X			
	B	E	112°47.012	Height :	290	480	480 / 280
		S	07°18.56	Y			
2	A	E	112°47.005	Width :	257	442	442 / 257
		S	07°18.085	X			
	C	E	112°47.565	Height :	625	1027	1027 / 625
		S	07°19.17	Y			
3	A	E	112°47.005	Width :	175	257	257 / 175
		S	07°18.085	X			
	G	E	112°46.748	Height :	45	36	36 / 45
		S	07°18.049	Y			
4	A	E	112°47.005	Width :	164	301	301 / 164
		S	07°18.085	X			
	F	E	112°46.704	Height :	313	527	527 / 313
		S	07°18.612	Y			
5	A	E	112°47.005	Width :	101	165	165 / 101
		S	07°18.085	X			
	H	E	112°46.840	Height :	361	591	591 / 361
		S	07°18.676	Y			
6	H	E	112°46.840	Width :	159	277	277 / 159
		S	07°18.676	X			
	C	E	112°46.563	Height :	264	436	436 / 264
		S	07°19.112	Y			

No	Koordinat Lokasi		Selisih Pixel	Selisih aktual	Perbandingan	Faktor		
7	C	E	112°46.563	Width : X	148	285	285 / 148	1.92
		S	07°19.112					
	D	E	112°46.848	Height : Y	0	3	~	
		S	07°19.115					
8	C	E	112°46.563	Width : X	109	229	229 / 109	2.10
		S	07°19.112					
	I	E	112°46.334	Height : Y	460	755	755 / 460	
		S	07°19.867					
9	I	E	112°46.334	Width : X	533	884	884 / 533	1.66
		S	07°19.867					
	J	E	112°47.218	Height : Y	11	42	42 / 11	
		S	07°19.909					
10	H	E	112°46.840	Width : X	255	501	501 / 255	1.97
		S	07°18.676					
	I	E	112°46.334	Height : Y	723	1191	1191 / 723	
		S	07°19.867					

Dari perhitungan tabel 3.2 dengan 10 sampel yang dilakukan 10 kali pengukuran nilai perbandingan, maka nilai faktor dapat dirata-rata dengan perhitungan matematik sebagai berikut, untuk nilai faktor X :

$$X = \frac{\sum X_{faktor}}{n} \quad (3.3)$$

Sedangkan untuk nilai faktor Y didapat dengan :

$$Y = \frac{\sum Y_{faktor}}{n} \quad (3.4)$$

Keterangan : n = jumlah sampel data.

Sesuai dengan persamaan 3.3, maka untuk nilai faktor dari X adalah :

$$X = \frac{0 + 1.71 + 1.46 + 1.83 + 1.63 + 1.74 + 1.93 + 2.10 + 1.66 + 1.97}{10} = 1,603$$

Sedangkan untuk nilai faktor Y diperoleh sesuai dengan persamaan 3.4 :

$$Y = \frac{1.71 + 1.64 + 0.8 + 1.68 + 1.64 + 1.65 + 0 + 1.64 + 3.8 + 1.65}{10} = 1.621$$

Hasil rata-rata dari X dan Y tersebut merupakan faktor pengali dalam pixel untuk menentukan posisi sesungguhnya yang digambarkan dalam Map. Dengan kata lain bahwa 1 pixel mewakili sekitar 1.6 point jarak sesungguhnya. Dapat diasumsikan juga bahwa 1 pixel mewakili sekitar 41 meter jarak sesungguhnya ( $1.6 * 26 \text{ m}$ ).

STIKOMMP SURABAYA