

## **BAB IV**

### **PEMBAHASAN**

#### **4.1 Proses Kerja *Safety Board***

Papan Penampil Keselamatan Kerja atau *Safety Board* adalah sebuah papan yang digunakan untuk menampilkan data-data terkait informasi keselamatan kerja. Adapun informasi yang dicatat diantaranya adalah :

1. Tanggal (*dd-mm-yyyy*) : Dimana datanya diperoleh dari GPS dengan memperhatikan jam sebagai pengganti hari. Dikarenakan format waktu yang dikirim oleh GPS adalah GMT (*Greenwich Mean Time*), maka data dari GPS harus diolah agar sesuai dengan format waktu Indonesia (WIB) yaitu GMT+7. Pengambilan data dilakukan setiap pukul 07.00 WIB, dimana data jam pada GPS adalah 00.00 GMT serta tanggal bertambah (ganti hari).
2. Tanpa Kecelakaan Kerja : yang dimaksud tanpa kecelakaan kerja adalah jumlah hari tanpa kecelakaan kerja. Jika pada hari tersebut tidak terjadi kecelakaan dan tanggal sudah bertambah, maka nilai tanpa kecelakaan kerja juga bertambah 1. Namun jika pada hari itu terjadi kecelakaan kerja, maka nilai tanpa kecelakaan kerja akan di-*reset* sehingga kembali nol.
3. Jumlah Kecelakaan Kerja : adalah jumlah kecelakaan kerja yang terjadi. Jika *operator* menginputkan tanggal terakhir kecelakaan, maka secara otomatis nilai pada jumlah kecelakaan kerja akan bertambah.

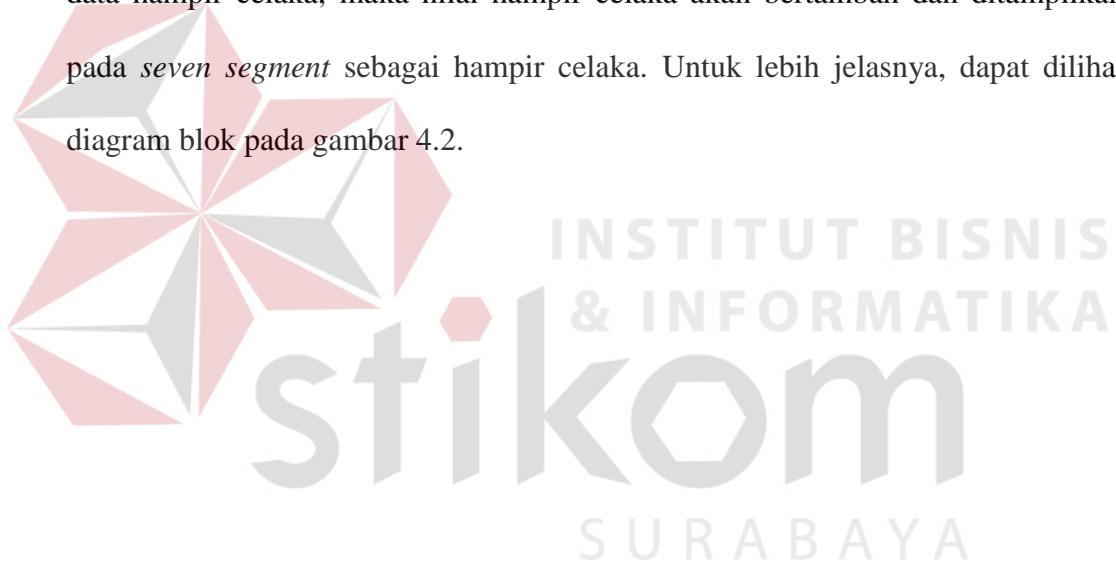
4. Hampir Celaka (*Near Miss*) : dilakukan pencatatan juga untuk kondisi hampir celaka. Dalam hal ini hampir celaka adalah kecelakaan kecil seperti terkena solder, teriris pisau atau kecelakaan kecil lainnya, yang berarti bukan kecelakaan namun mendekati kecelakaan.
5. Kecelakaan Terakhir (*Last accident*) : mencatat tanggal terakhir terjadinya kecelakaan. Kecelakaan dapat berupa kulit terbakar radiasi, tersiram bahan kimia, atau bahkan mungkin meninggal.

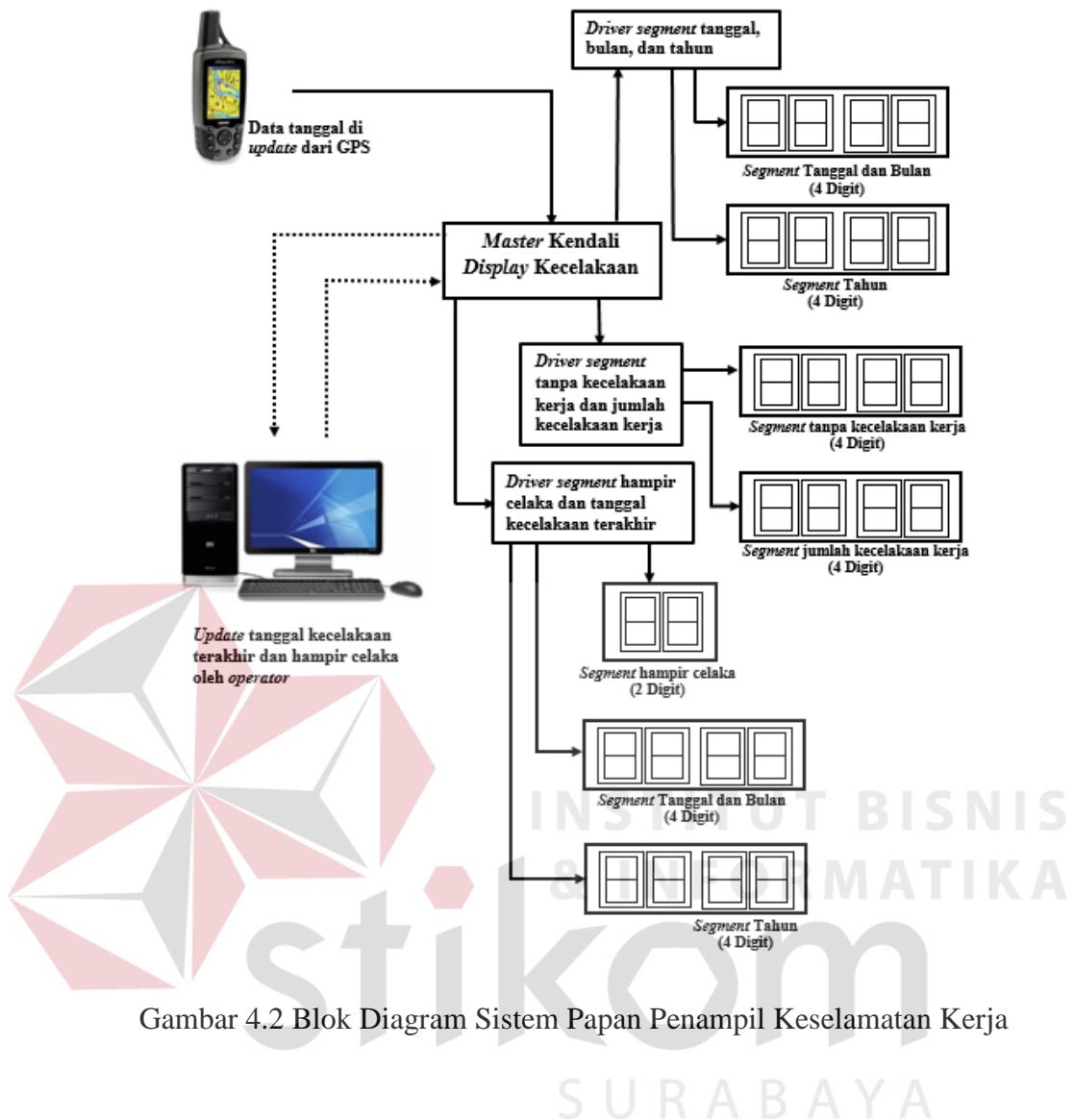


Gambar 4.1 Papan Keselamatan Kerja (*Safety Board*)

*Microcontroller Atmega 8* sebagai pemroses utama, mendapatkan masukan data dari GPS dan PC *operator*. Data yang dikirim dari GPS berupa blok data dengan berbagai tipe data NMEA. Data tersebut akan diolah oleh *microcontroller*, kemudian tanggal dan jam dari data tersebut disimpan dan selanjutnya ditampilkan pada *seven segment* sebagai tanggal, bulan, dan tahun

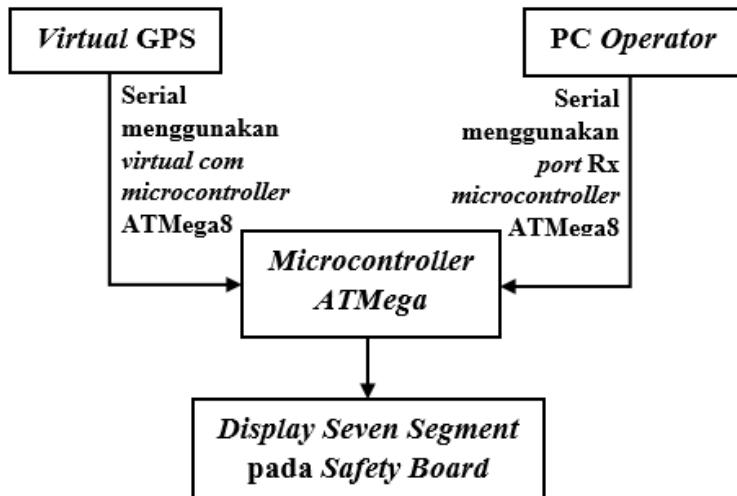
hari ini. Jika dalam satu hari tidak terjadi kecelakaan, kemudian tanggal telah bertambah, maka nilai tanpa kecelakaan kerja akan bertambah secara otomatis dan ditampilkan pada *seven segment* sebagai tanpa kecelakaan kerja. Sedangkan dari PC *Operator* dapat mengirimkan tanggal kecelakaan terakhir serta hampir celaka. Ketika *operator* mengirim tanggal kecelakaan terakhir, maka secara otomatis nilai jumlah kecelakaan kerja bertambah dan nilai tanpa kecelakaan kerja menjadi nol. Data tersebut juga akan ditampilkan pada *seven segment* sebagai jumlah kecelakaan kerja dan tanpa kecelakaan kerja. Kemudian jika *operator* mengirim data hampir celaka, maka nilai hampir celaka akan bertambah dan ditampilkan pada *seven segment* sebagai hampir celaka. Untuk lebih jelasnya, dapat dilihat diagram blok pada gambar 4.2.





Gambar 4.2 Blok Diagram Sistem Papan Penampil Keselamatan Kerja

## 4.2 Perancangan Rangkaian Simulasi



Gambar 4.3 Blok Diagram Perancangan Rangkaian Simulasi *Safety Board*

Dalam perancangan rangkaian simulasi, dibutuhkan beberapa komponen agar sistem dapat berjalan sesuai yang diinginkan. Diantaranya adalah :

1. *Microcontroller Atmega 8* : berfungsi sebagai pemroses utama, dimana akan mengolah data dari GPS maupun PC *Operator*.
2. GPS dan Protokol NMEA 0183 : digunakan sebagai penyedia data tanggal yang akan ditampilkan pada *seven segment* sebagai tanggal hari ini,
3. Compim : compim merupakan wujud *virtual* dari DB-9 yang digunakan sebagai komponen komunikasi serial antara GPS dengan mikro, maupun PC *Operator* dengan mikro.
4. *Seven Segment* : digunakan untuk menampilkan data yang diinformasikan.

5. IC *Shift Register* 4094: digunakan sebagai *decoder seven segment*, dimana mendapatkan masukan serial dari mikro dan menghasilkan keluaran pararel yang ditampilkan pada *seven segment*.

#### **4.2.1 Microcontroller Atmega 8**

*Microcontroller* Atmega 8 adalah *microcontroller* keluarga AVR 8 bit. Atmega 8 memiliki ukuran *flash* memori sebesar 8KB, SRAM sebesar 1KB, dan memori EEPROM sebesar 512 Bytes. Atmega 8 memiliki 3 buah PORT utama yaitu PORTB, PORTC, dan PORTD dengan total pin *input/output* sebanyak 23 pin. Dalam sistem ini, Atmega digunakan sebagai kendali utama dari seluruh sistem. Dimana akan memproses 2 blok data, diantaranya adalah data dari GPS dan data dari PC *Operator*.

#### **4.2.2 GPS (*Global Positioning System*) dan Protokol NMEA 0183**

GPS dalam sistem ini digunakan sebagai penyedia data tanggal dan jam. Format data GPS mengikuti standar NMEA (*National Marine Electronics Association*), standar ini mengijinkan peralatan elektronik untuk mengirim informasi kepada komputer atau peralatan yang lain. Tipe data ini terdiri dari PVT (*position, velocity, time*) yang dikirimkan oleh GPS *transmitter* melalui komunikasi serial *port* menggunakan *protocol* RS232 dengan *baudrate* 4800b/s, 8 bit data, tanpa *parity* dan *stop* bit. Setiap data dimulai dengan ‘\$’, data dikirim dalam satu baris yang terdiri dari berbagai data yang dipisahkan dengan koma (“,”) dan dikirimkan setiap 2 detik oleh *transmitter*.

Dari beberapa jenis data yang dikirim oleh GPS, yang akan diolah dalam sistem ini adalah format data RMC.

Berikut contoh format RMC :

\$GPRMC,123519,A,4807.038,N,01131.000,E,022.4,084.4,2303  
94,003.1,W\*6A

Dimana :

RMC	<i>Recommended Minimum sentence C</i>
123519	<i>Fix taken at 12:35:19 UTC</i>
A	<i>Status A=active or V=Void.</i>
4807.038,N	<i>Latitude 48 deg 07.038' N</i>
01131.000,E	<i>Longitude 11 deg 31.000' E</i>
022.4	<i>Speed over the ground in knots</i>
084.4	<i>Track angle in degrees True</i>
230394	<i>Date - 23rd of March 1994</i>
003.1,W	<i>Magnetic Variation</i>
*6A	<i>The checksum data, always begins with *</i>

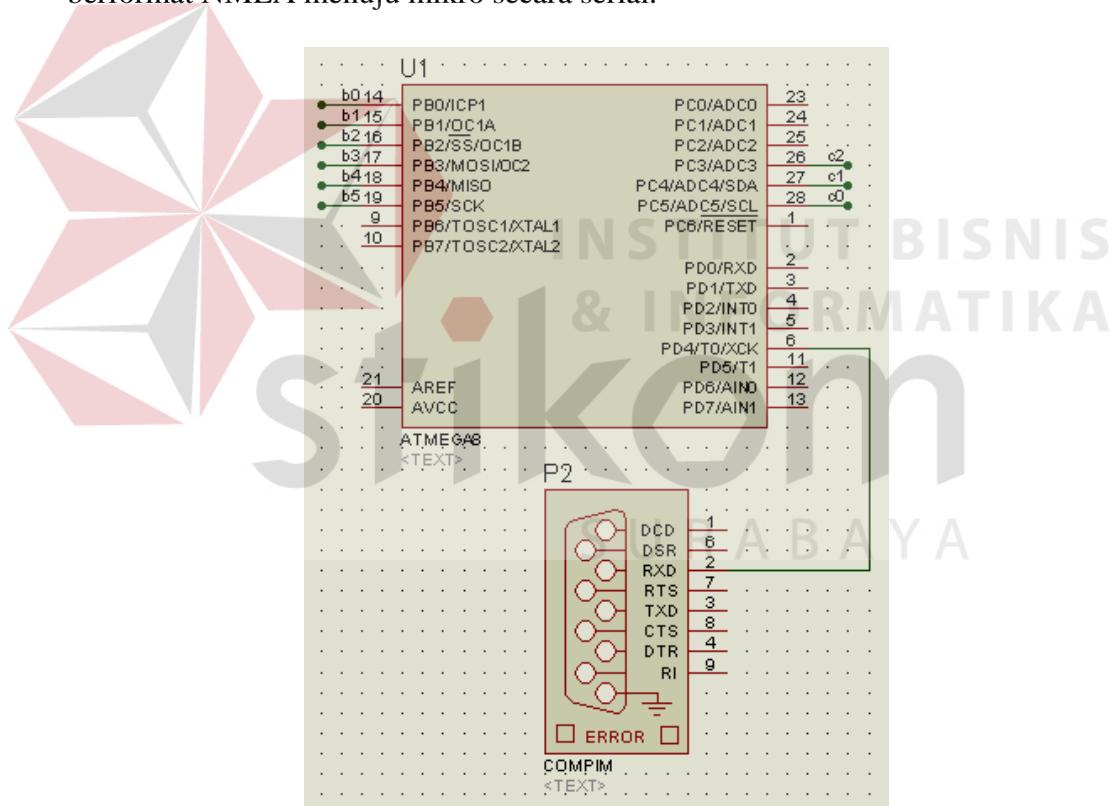
Tipe data RMC memiliki versi data sendiri yang sangat penting dalam penentuan posisi dan waktu yaitu PVT (*position, velocity, time*) sehingga tipe data ini merupakan rekomendasi minimum data untuk GPS. Pada format RMC, terdapat tanggal dan jam pengambilan data. Sehingga sesuai dengan yang dibutuhkan untuk pengolahan.

### 4.2.3 Compim

Compim adalah sebuah komponen yang disediakan Proteus untuk dapat mengkomunikasikan mikro dengan PC secara serial. Compim juga merupakan wujud *virtual* dari sebuah DB-9.

#### 1. Komunikasi Antara *Virtual GPS* dengan Mikrokontroler Atmega 8

Pada simulasi ini, GPS yang digunakan berupa *virtual*. Yaitu dengan membuat sebuah program dari Visual *Basic* yang mengirimkan blok data berformat NMEA menuju mikro secara serial.

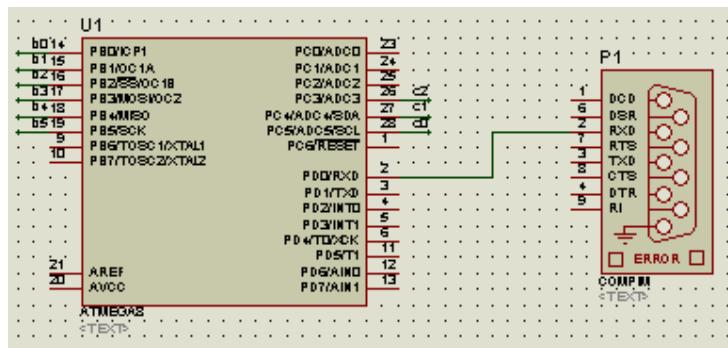


Gambar 4.4 Rangkaian komunikasi antara Atmega 8 dengan PC (*Virtual GPS*) menggunakan compim

Untuk dapat mengkomunikasikan *virtual GPS* dengan mikro, maka ada beberapa hal yang perlu diperhatikan. Diantaranya adalah :

1. Dikarenakan pada simulasi ini mikro hanya sebagai penerima data, maka kaki Rx (*Receiver*) pada compim dihubungkan dengan PORTD.4 pada mikro. Sehingga tidak perlu menghubungkan kaki Tx pada compim dengan mikro. PORTD.4 adalah *port* biasa yang dijadikan *virtual com*, dimana PORTD.4 akan menjalankan fungsi yang sama seperti fungsi Rx pada mikro. Cara menjadikannya menjadi *virtual com* hanya dengan menyambungkan kaki Rx pada COMPIM dengan PORTD.4.
2. Pengaturan *Baud Rate* pada COMPIM harus sesuai dengan *Baud Rate* pada mikro, agar data yang dikirim GPS bisa diterima dan terbaca oleh mikro.
3. Pastikan telah menghubungkan COM yang akan dipakai oleh GPS dan mikro. Disini, kami menggunakan aplikasi VSPE untuk menghubungkannya. VSPE adalah *software* simulasi *port* serial yang memungkinkan kita membuat *port* serial *virtual* dan menghubungkannya tanpa menggunakan kabel serial.

## 2. Komunikasi Antara PC *Operator* dengan Mikrokontroler Atmega 8



Gambar 4.5 Rangkaian komunikasi antara Atmega 8 dengan PC *Operator* menggunakan compim

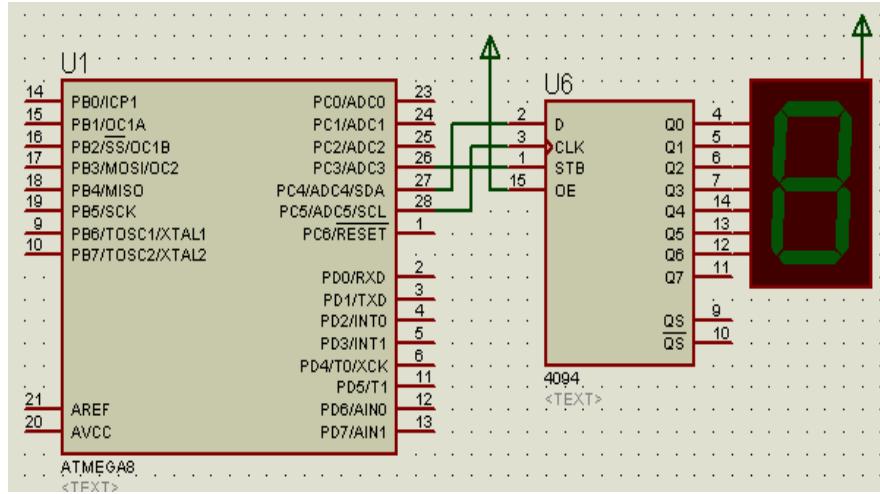
Hal-hal yang perlu diperhatikan dalam mengkomunikasikan mikro dengan PC *Operator* sama dengan komunikasi antara mikro dengan GPS diatas. Namun pada komunikai ini tidak menggunakan *virtual com* sebagai Rx, karena mikro memiliki *port Rx* sebagai penerima dalam komunikasi serial.

### 3. **7 Segment Display**

*7 Segment Display* digunakan untuk menampilkan data-data yang akan diinformasikan. Dalam sistem ini ada 26 buah *7 Segment*, diantaranya untuk menampilkan tanggal pada hari tersebut sebanyak 8 digit, jumlah hari tanpa kecelakaan sebanyak 4 digit, jumlah hari kecelakaan kerja sebanyak 4 digit, jumlah hari hampir celaka sebanyak 2 digit, dan tanggal kecelakaan terakhir sebanyak 8 digit.

### 4. **IC Shift Register 4094**

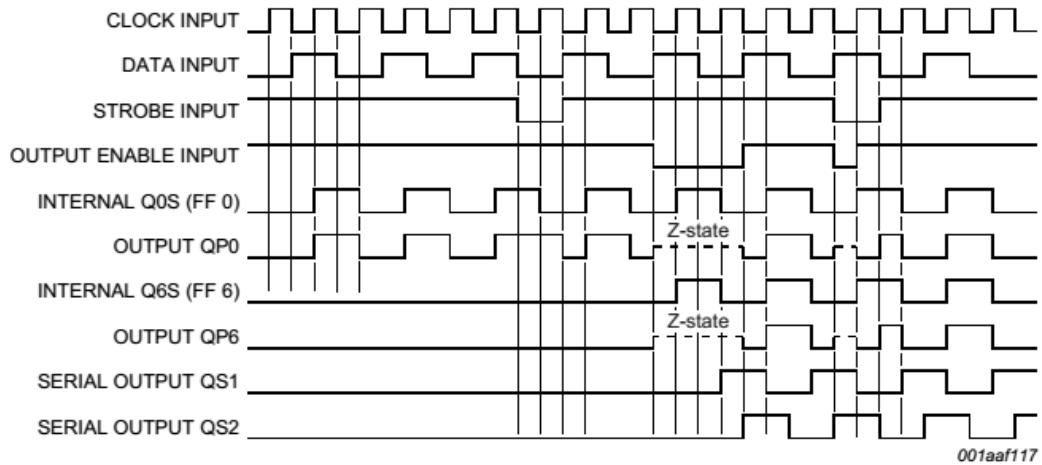
IC *Shift Register* 4094 disini digunakan untuk memasukkan data secara serial dan mengeluarkan data secara paralel atau biasa disebut *Serial Input Pararel Output* (SIPO). Yaitu sebagai *driver* untuk *seven segment*.



Gambar 4.6 Rangkaian *seven segment* dan IC 4094

Pada gambar 4.5, ditunjukkan data masuk secara serial melalui pin D pada IC 4094. Pada IC *Shift Register* ini, data yang masuk akan disimpan setelah terjadi *clock*. Jadi cara memasukkan data pada *shift register* ini adalah *strobe* harus berlogika *high*. Kemudian data masuk-*clock*-data masuk-*clock*- data masuk-*clock*, begitu seterusnya. Setelah satu blok data dikirim, *strobe* harus dibuat berlogika *low* untuk mengakhiri pengiriman. Pin OE atau *Output Enable* digunakan untuk mengaktifkan *output* serial maupun *output* paralel. Logika 1 untuk *enable* dan logika 0 untuk *disable*. Q0 - Q7 adalah *output parallel* dari *shift register* ini sedangkan QS dan  $\overline{QS}$  adalah *output* serial dari *shift register* ini. Jika menggunakan lebih dari satu IC *Shift Register* maka pin data dari IC *Shift Register*

Berikut adalah *timing diagram* IC 4094 :

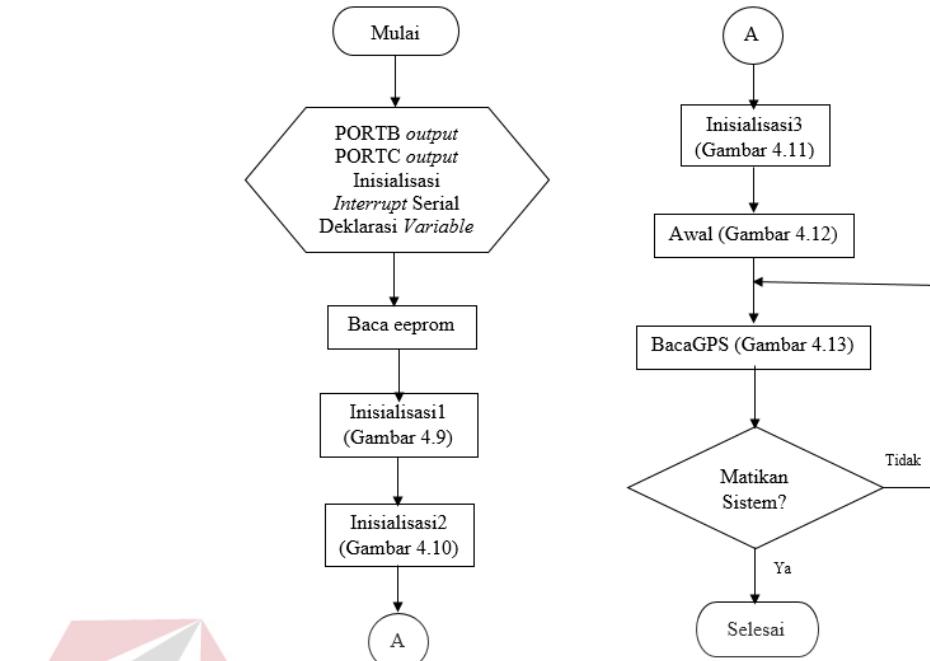


Gambar 4.7 Timing Diagram IC Shift Register 4094

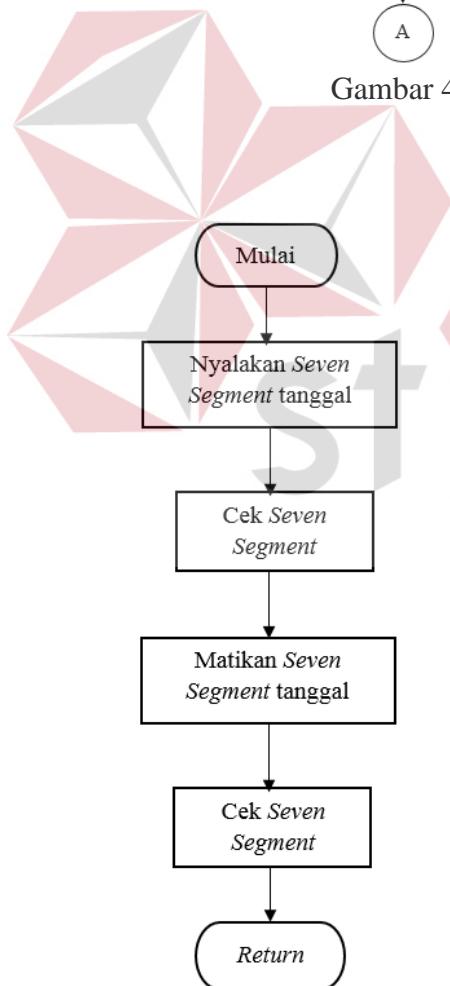
### 4.3 Perancangan Program

#### 4.3.1 Program Utama pada Microcontroller

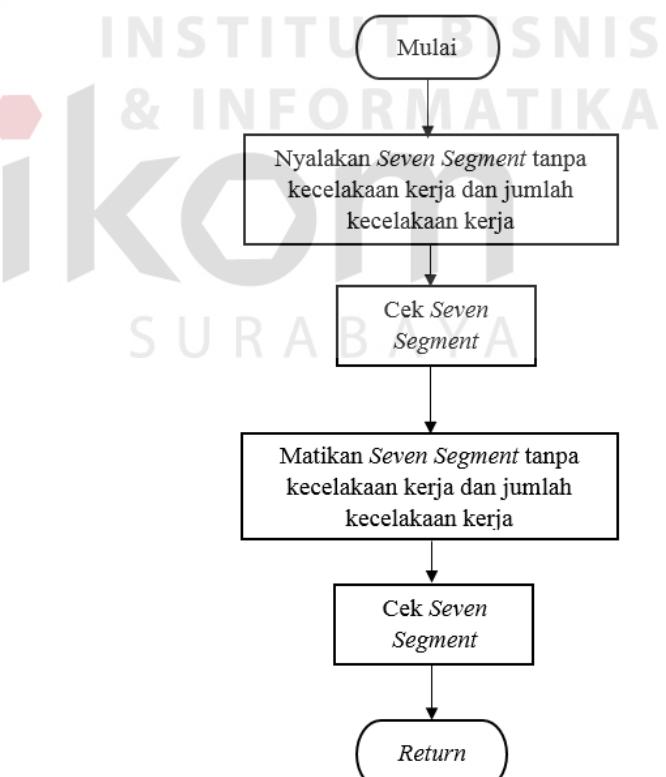
Pembuatan program pada simulasi ini menggunakan bahasa pemrograman *BASIC* dengan *BASCOM AVR* sebagai *compiler*-nya. Program ini dibuat berdasarkan diagram alir berikut :



Gambar 4.8 Diagram Alir Main Program

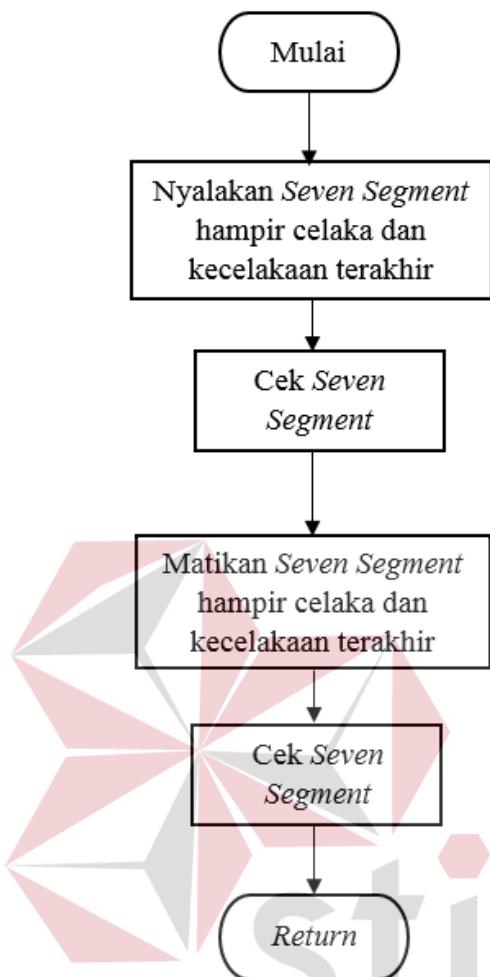


Gambar 4.9 Diagram Alir



Gambar 4.10 Diangram Alir

Inisialisasi1



Inisialisasi2

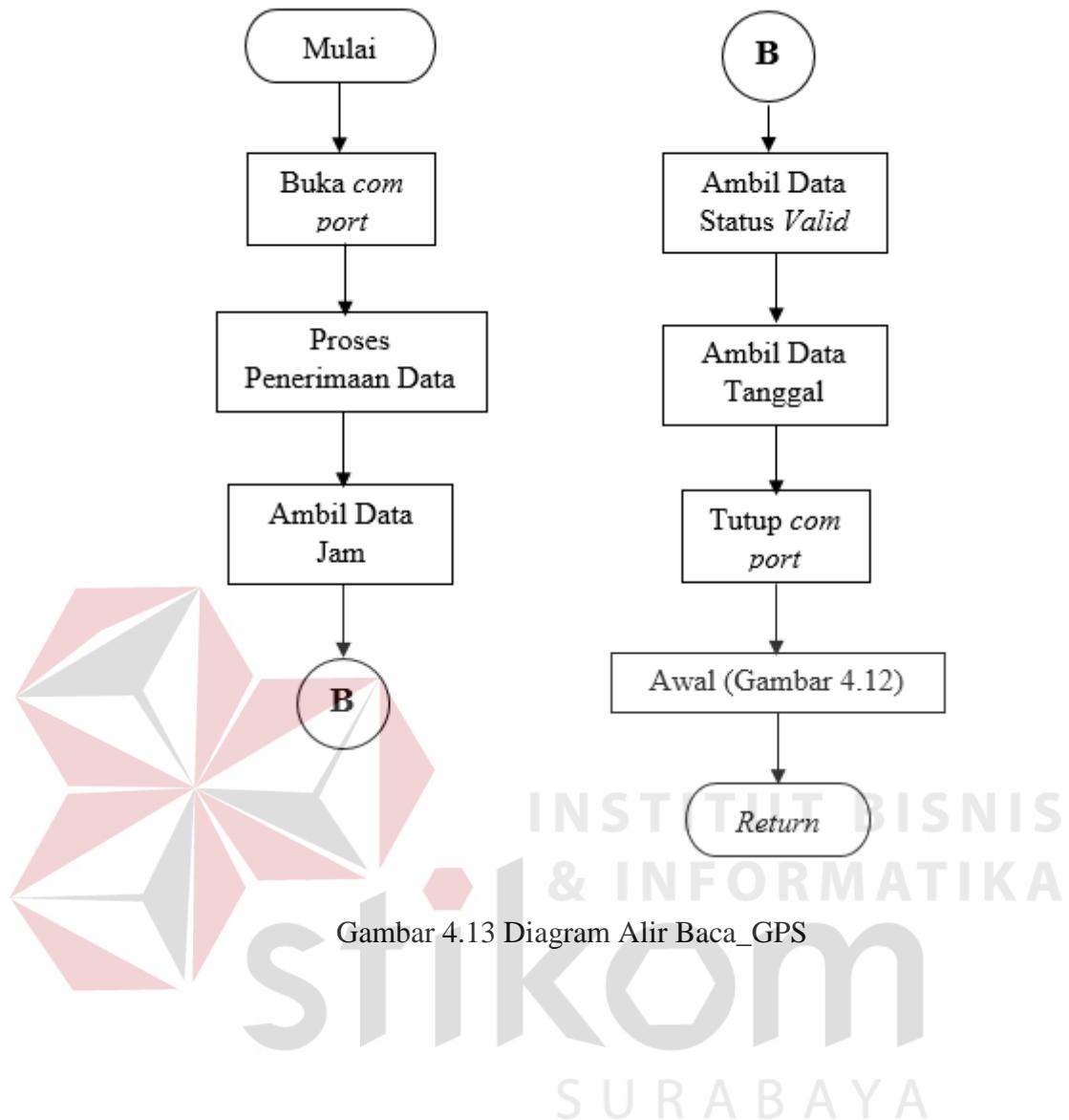


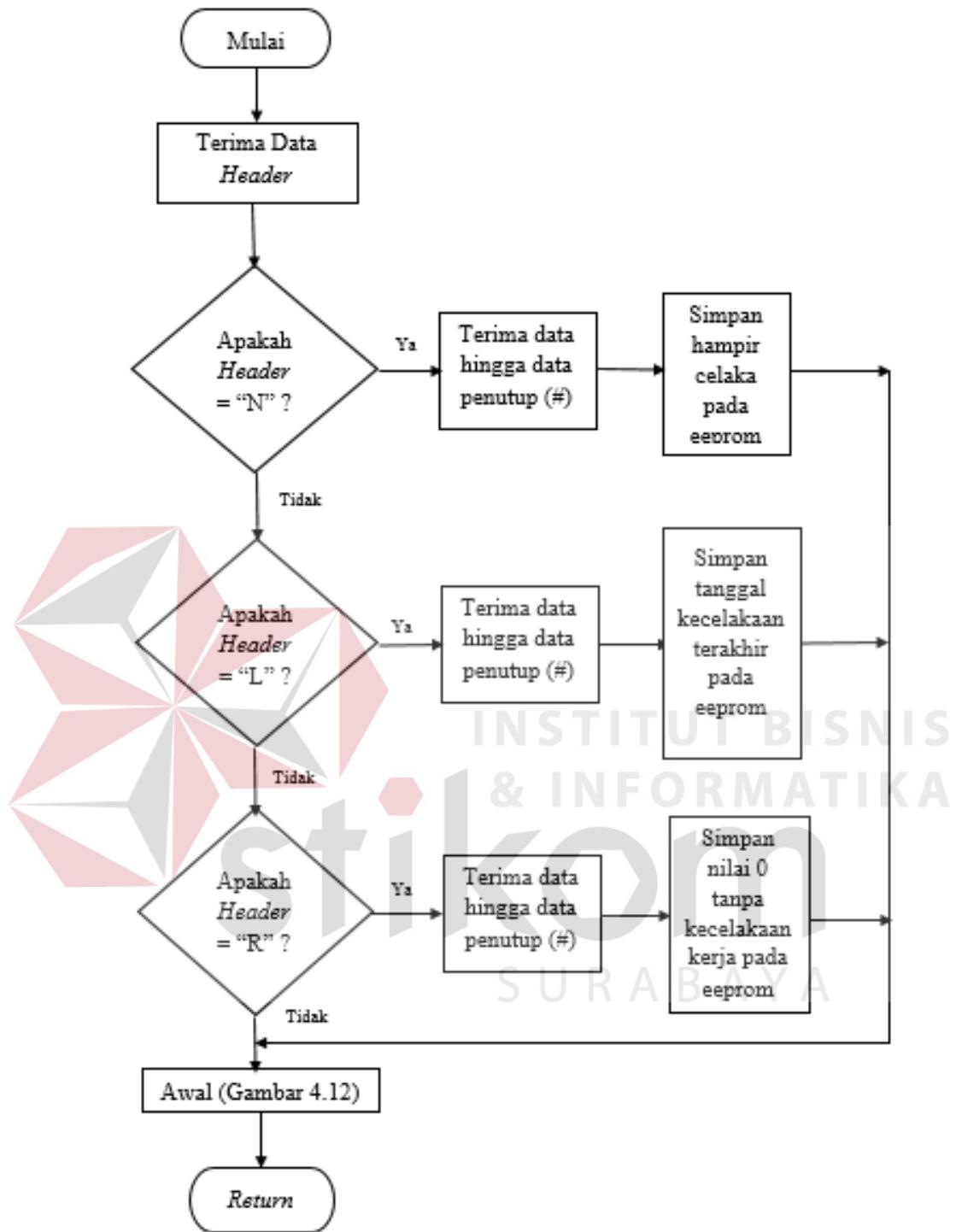
Gambar 4.11 Diagram Alir

Inisialisasi 3

Gambar 4.12 Diagram Alir

Awal

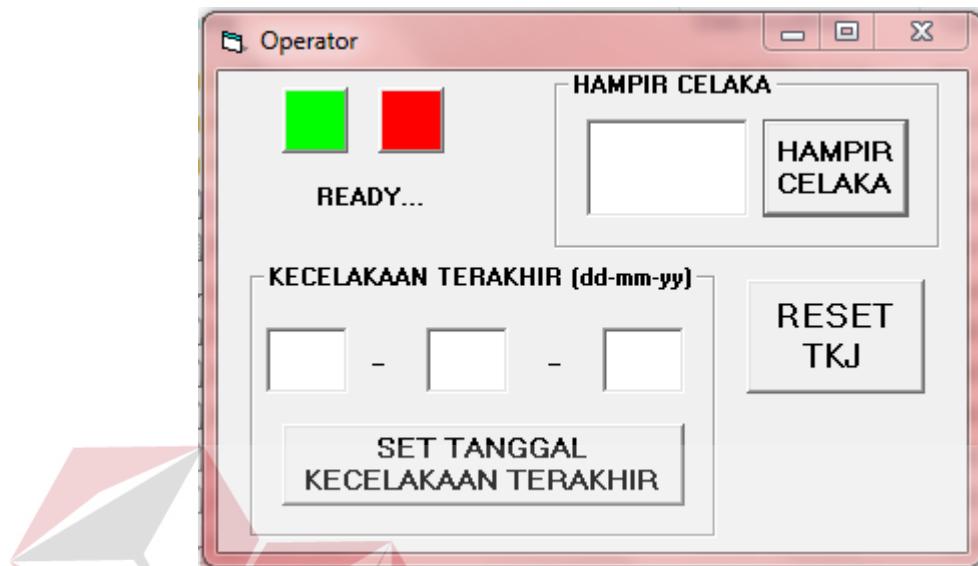




Gambar 4.14 Diagram Alir Int\_PC

### 4.3.2 Program Pendukung

#### 1. Program untuk PC *Operator*



Gambar 4.15 Tampilan Program untuk PC *Operator*

Program untuk PC *Operator* berisi 3 perintah untuk mengontrol *Safety Board*. Diantaranya adalah tanggal terakhir kecelakaan, hampir celaka, dan *reset* tanpa kecelakaan kerja. Perintah yang dikirim dari PC *Operator* berisi *header*, nilai perintah, dan penutup. Berikut ini contoh format pengiriman data dari PC *Operator*:

N1# → N : *Header* untuk mengatur nilai hampir celaka

1 : Jumlah yang akan dikirim untuk nilai hampir celaka

# : Penutup yang menandakan akhir pengiriman data

L010112# → L : *Header* untuk mengatur tanggal kecelakaan terakhir

010112 : Nilai tanggal yang akan mengisi tanggal kecelakaan terakhir

# : Penutup yang menandakan akhir pengiriman data

R0# → R : *Header* untuk *reset* nilai tanpa kecelakaan kerja

0 : nilai *reset* untuk tanpa kecelakaan kerja

# : Penutup yang menandakan akhir pengiriman data

## 2. Program untuk *Virtual GPS*



Gambar 4.16 Tampilan Program untuk *Virtual GPS*

Program untuk *virtual GPS* hanya berisi *timer* yang mengirim data dengan *interval* 2 detik. Data yang dikirim berupa sebuah blok data dengan berbagai jenis tipe data NMEA.