

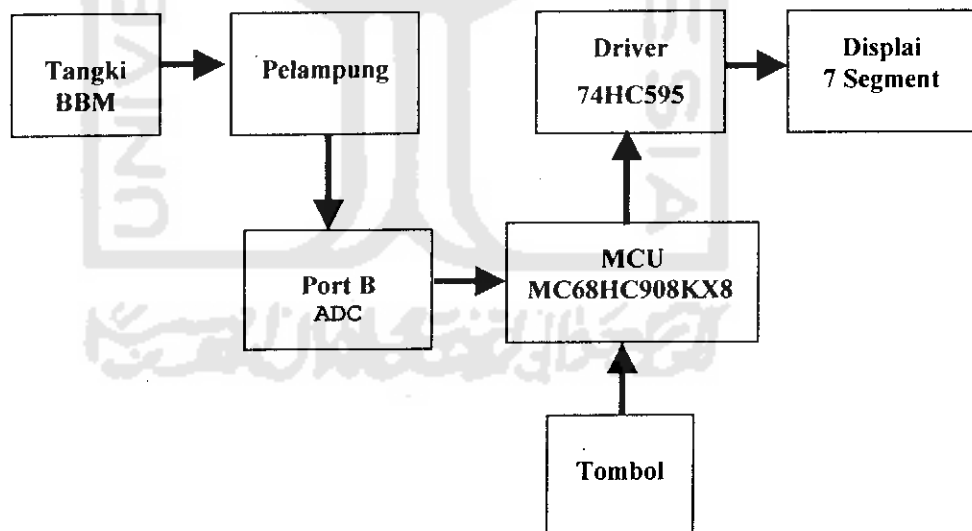
BAB III

PERANCANGAN SISTEM PENGUKUR VOLUME BAHAN BAKAR SEPEDA MOTOR HONDA ASTREA GRAND

3.1 Pendahuluan

Dalam bab ini akan dibahas perencanaan sistem pengukur volume bahan bakar sepeda motor Honda astrea grand berdasarkan teori-teori yang telah dibahas dalam bab sebelumnya.

Cara kerja dari sistem ini dapat dilihat dari diagram blok di bawah ini :



Gambar 3.1 Diagram Blok Sistem Pengukur volume bahan bakar sepeda motor Honda astrea grand

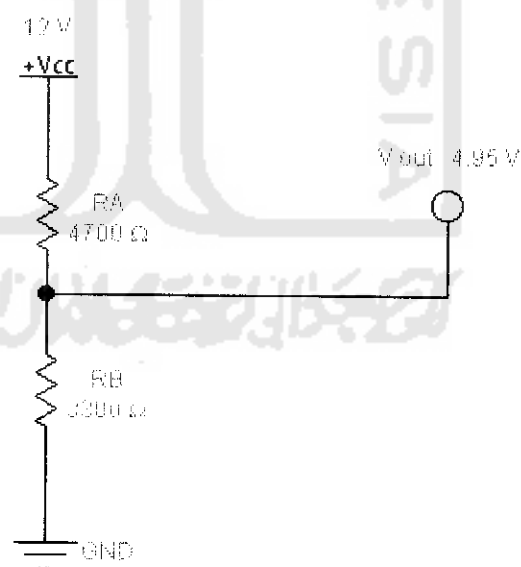
Fungsi dari masing-masing bagian dari diagram blok di atas adalah :

1. Tangki Bahan Bakar Sepeda Motor Honda Astrea Grand

Bagian Tangki bahan bakar sepeda motor berfungsi tempat penyimpanan bahan bakar dimana tangki tersebut mempunyai kapasitas bahan bakar sekitar 4,4 liter.

2. Pelampung pada tangki.

Bagian pelampung berfungsi sebagai pengukur volume bahan bakar, pelampung ini berbentuk resistansi yang terbuat dari karbon dimana resistansi tersebut dapat mengubah energi mekanik menjadi energi listrik. Pemrosesan yang dihasilkan menurut hukum pembagian tegangan.



Gambar 3.2 Rangkaian pembagi tegangan

$$V_{out} = \frac{R_B}{R_B + R_A} \times V_{input} \quad (3.1)$$

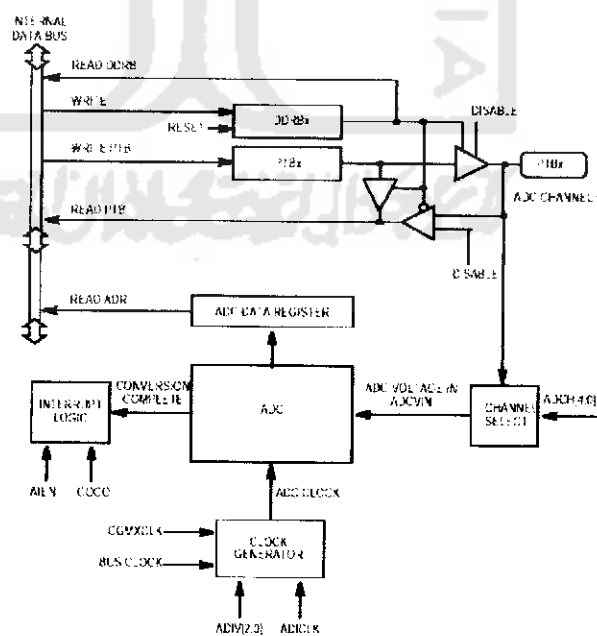
Indikator output adalah voltmeter yang dikalibrasi sebagai petunjuk tangki bahan bakar.

Penurunan hambatan pada pelampung menyebabkan penambahan tegangan, penambahan tegangan menyebabkan voltmeter (petunjuk tangki bahan bakar) naik ke pembacaan yang lebih tinggi.

Dengan berubahnya volume bahan bakar pada tangki maka tegangan listrik yang keluar dari pelampung dan masuk ke bagian Pengubah Analog ke Digital akan berubah.

3. ADC (*Analog to Digital Converter*)

Bagian ini berfungsi untuk merubah tegangan listrik menjadi nilai digital. Makin tinggi tegangan yang masuk maka makin tinggi nilai digital yang didapat.

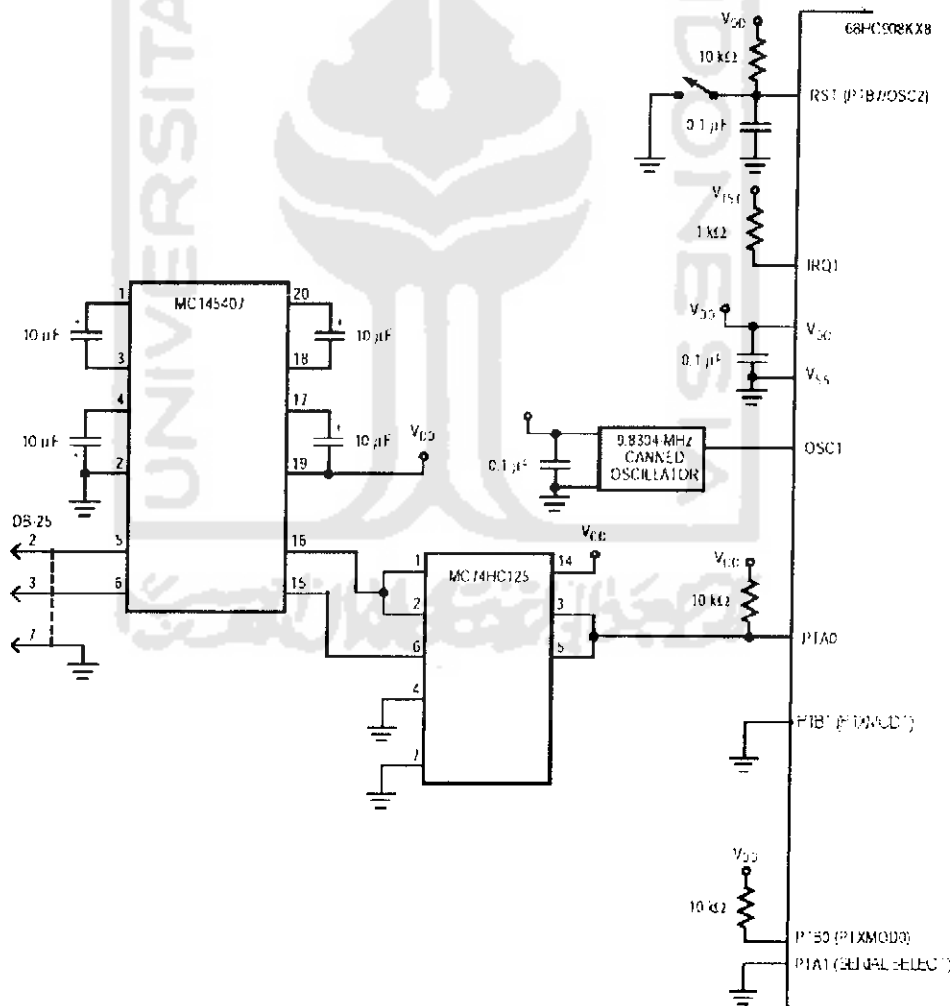


Gambar 3.3 *Analog to Digital Converter* dalam MC68HC908KX8

4. Mikrokontroler MC68HC908KX8

Mikrokontroler berfungsi sebagai kendali yang mengatur semua kerja dari sistem pengukur volume bahan bakar.

Untuk dapat mensimulasikan dan memasukkan program ke dalam mikrokontroler MC68HC908KX8 digunakan sebuah ICS (*In-Circuit Simulator*). Rangkaian ICS ini diperoleh dari *Data-Sheet* Motorola.



Gambar 3.4 Rangkaian *Down Loader* IC MC68HC908KX8

5. Keluaran (Port A & B)

Sebagai keluaran untuk mengirimkan data masukan ke dalam *driver* 74HC595 digunakan 3 buah jalur dari Port B yaitu PB2 sebagai *shift clock*, PB3 sebagai *shift clock*, PB5 sebagai data.

6. Tombol

Bagian ini berfungsi untuk menentukan tampilan yang ada pada penampil 7-segmen. Tampilan ini dapat berupa jumlah bahan bakar yang masih ada dalam tangki, jumlah bahan bakar hingga tangki penuh dan jumlah rupiah yang dibutuhkan untuk membeli bahan bakar hingga tangki penuh.

7. *Driver 7 segment* 74HC595

Driver 74HC595 adalah merupakan IC jenis *shift register* terdiri dari 16 pin, yang digunakan untuk mengubah data serial menjadi data paralel dan menggeser data serial ke 'data serial output' bila 'data serial input' diberi data serial lagi. Kaki – kaki yang digunakan adalah pin kaki 1 – 7 dan 15 digunakan sebagai paralel data output, pin kaki 9 untuk serial data output, pin kaki 10 untuk reset, pin kaki 11 untuk *shift clock*, pin kaki 12 untuk *latch clock*, pin kaki 13 untuk *output enable*, dan pin kaki 14 digunakan untuk serial data input. Tegangan yang dibutuhkan +5V yang dihubungkan dengan kaki 16 dan *ground* pada kaki 8.

8. Displai 7 Segment

Bagian ini merupakan penampil dari beberapa fungsi yang telah dibuat yaitu jumlah volume bahan bakar yang tersisa dalam tangki, jumlah

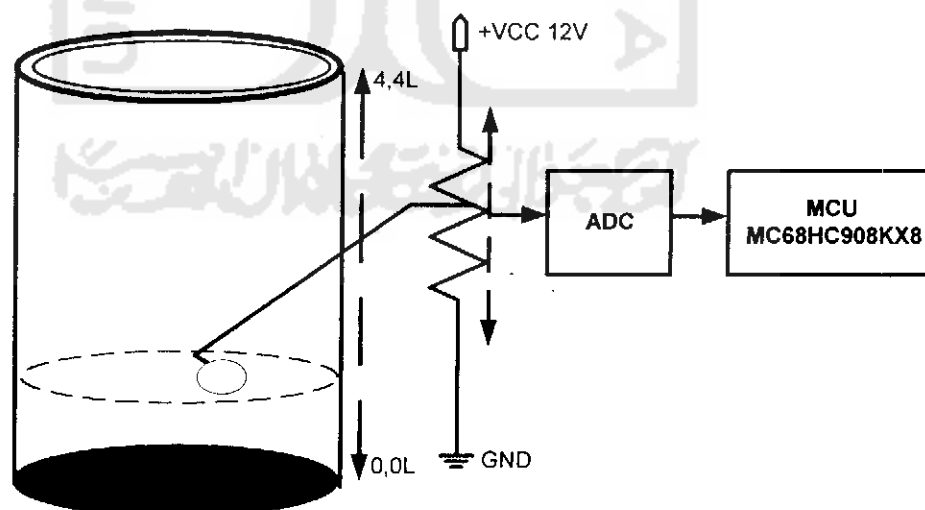
volume bahan bakar yang di butuhkan agar tangki penuh, harga bahan bakar yang harus diisi agar tangki penuh.

Untuk displai 7 segment menggunakan jenis *common anoda* dimana pada kaki *commonnya* diberi masukan tegangan positif dan sebelum diberi tegangan untuk membatasi arus yang masuk pada displai dipasang sebuah resistor dengan besar hambatannya sekitar 300 ohm.

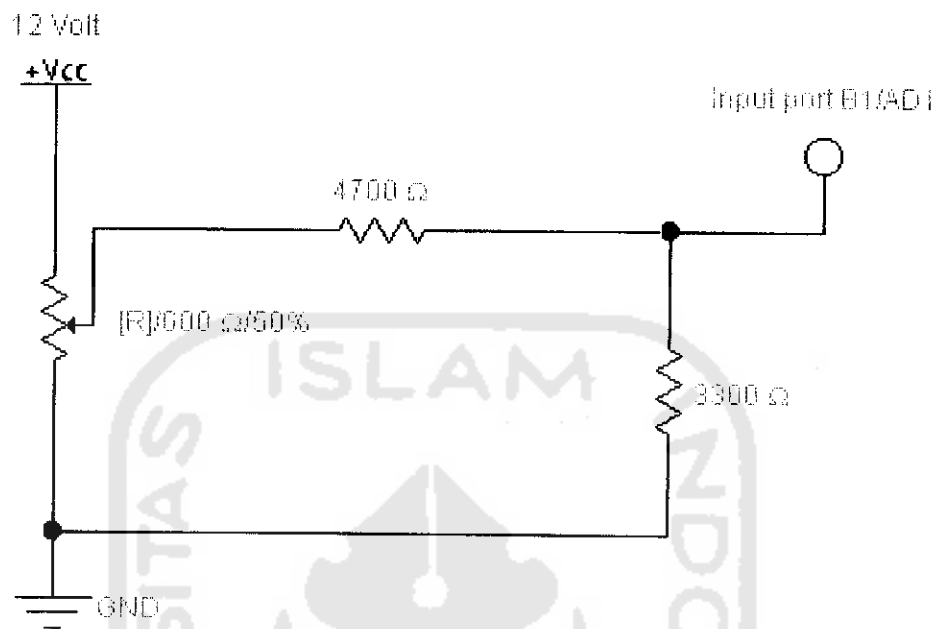
3.2 Perancangan Perangkat Keras

3.2.1 Pelampung pada tangki bahan bakar

Untuk dapat mengukur volume bahan bakar pada tangki digunakan sebuah pelampung dimana pelampung tersebut berupa resistansi yang mempunyai nilai resistansi sekitar 600Ω



Gambar 3.5 Tampak samping pelampung dalam tangki bahan bakar



Gambar 3.6 Rangkaian Input ADC

Resistansi pada pelampung dihubung paralel dengan 2 buah hambatan dengan nilai RA (4700 ohm) dan RB (3300 ohm) hal ini untuk menurunkan tegangan dari 12 volt menjadi 5 volt.

3.2.2 Mikrokontroler MC68HC908KX8

Sebagai kendali dari sistem, mikrokontroler akan menerima masukan dari rangkaian ADC dan tombol serta memberikan data tampilan pada driver 74HC595.

Sebagai masukan dari rangkaian resistansi, digunakan satu jalur Port B yang dapat berfungsi sebagai pengubah analog ke digital yaitu Port B-1. Sedangkan sebagai keluaran untuk memberikan data pada *driver* 74HC595 digunakan 3 buah jalu port B yaitu Port B-2, Port B-3, Port B-5

3 jalur Port A yang dipakai untuk tombol yaitu Port A-2 , Port A-3, dan Port A-4 dari tombol tersebut terhubung ke *ground*.

Bagian pengubah dari analog ke digital yang terdapat di dalam mikrokontroler akan merubah tegangan masukan dari rangkaian resistansi yang terdapat pada pelampung menjadi nilai digital 8-bit. Analog Digital Converter 8-bit (2^8-1) dalam mikrokontroler dapat mengubah tegangan ke dalam nilai digital dengan nilai dari 0 hingga 255 dengan tegangan referensi 0 volt dan tegangan maksimum 5 volt sehingga memiliki keakuratan 0,0196 volt.

$$\frac{V \text{ max}}{255} = \frac{5 \text{ volt}}{255} = 0,0196 \text{ volt} \quad (3.2)$$

Perubahan tegangan 0,0196 volt pada rangkaian resistansi dapat diketahui oleh mikrokontroler.

Dengan membandingkan nilai tegangan keluaran dari rangkaian resistansi pada pelampung akan diketahui berapa besar volume bahan bakar pada tangki sepeda motor.

3.2.3 Switch (tombol)

Tombol tersebut terdiri 3 bagian Sw1, Sw2, dan Sw3. Jjika Sw1 ditekan akan keluar tampilan jumlah volume bahan bakar yang tersisa dalam tangki, Sw2 ditekan akan keluar tampilan jumlah volume bahan bakar yang terpakai oleh sepeda motor, jika menekan Sw3 maka akan keluar jumlah harga bahan bakar yang harus diisi supaya tangki penuh.

3.2.4 Display 7 segment

Displai 7 segment ini menggunakan *Driver 74HC595* dimana untuk led 7 segmentnya menggunakan jenis common anoda.

3.3 Perangkat Lunak

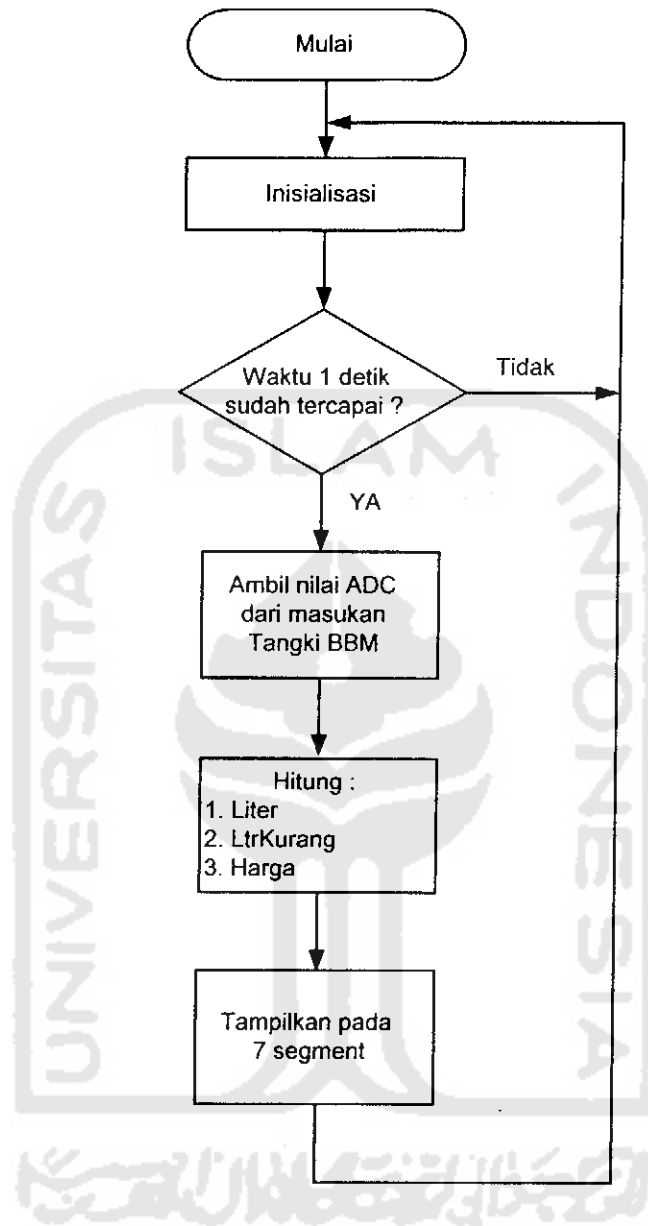
3.3.1 Diagram Alir Proses Pengendalian Sistem

Proses pengendalian sistem yang dikerjakan mikrokontroler dapat dibagi menjadi empat bagian utama yaitu :

- Perubahan analog ke digital
- Penghitungan data
- Penampilan data
- Pembacaan tombol

Keempat bagian utama di atas merupakan bagian yang dikerjakan oleh sistem secara terus menerus sesuai dengan keadaan yang telah ditetapkan. Walaupun dikerjakan secara bergantian, keempat bagian tersebut dapat terlihat bersamaan karena pembagian waktu yang tepat sehingga satu bagian akan dikerjakan lebih dahulu hingga selesai baru kemudian bagian lain dikerjakan.

Dari semua kegiatan di atas dapat dirangkum dalam diagram alir di bawah ini :



Gambar 3.7 Diagram Alir Proses pengendalian sistem

3.3.2 Bagian Pengubah Analog ke Digital

Pengubahan analog ke digital dilakukan terus menerus dan data hasil dari pengubahan tersebut akan diolah setiap satu detik.

Untuk mengaktifkan kaki 6 (Port B-1) sebagai masukan ADC, maka register ADSCR (*Analog to Digital Status and Control Register*) harus diatur terlebih dahulu. Register ADSCR harus diberi nilai \$21 (00100001) agar Port B-1 berfungsi sebagai masukan bagi ADC.

Selain register ADSCR, terdapat satu buah register yang harus diatur untuk menentukan masukan pembangkit pulsa bagi rangkaian ADC. Register tersebut adalah ADICLK (*ADC Input Clock Register*) yang diberi nilai \$10 (10000000) agar pembangkit pulsa sama dengan pulsa dari sistem mikrokontroler.

3.3.3 Pewaktuan

Agar dapat melaksanakan semua kegiatannya, maka mikrokontroler memerlukan referensi waktu yang baik dan tepat. Dengan referensi waktu ini maka mikrokontroler dapat membagi waktu kapan melakukan suatu kegiatan sehingga dapat melakukan berbagai macam kegiatan sekaligus.

Mikrokontroler menghasilkan sendiri pulsa yang berfungsi untuk menjalankan perintah dan proses pewaktuan. Pulsa ini dihasilkan dari bagian *Internal Clock Generator*. Frekuensi dari pulsa ini dapat diatur melalui program, yaitu dengan memberikan nilai pada register ICGCR, ICGMR dan ICGTR.

ICGCR (*Internal Clock Generator Control Register*) tidak perlu diberi nilai, karena saat reset sudah berisi \$08 yaitu pengaturan internal clock sudah diaktifkan.

ICGTR (*Internal Clock Generator Trimming Register*) juga tidak perlu diatur lagi karena dalam sistem ini tidak dibutuhkan pewaktuan yang sangat tepat.

ICGMR (*Internal Clock Generator Multiplier Register*) harus diberi nilai 26 (00011010) sehingga clock yang dihasilkan mempunyai frekuensi 8 MHz. ICGMR merupakan faktor pengali dari frekuensi dasar mikrokontroler yaitu 307,2 KHz. Sehingga dengan memberikan nilai 26 pada ICGMR maka frekuensi yang dihasilkan adalah $26 \times 307,2 \text{ KHz} = 7,9872 \text{ MHz}$ (hampir 8 MHz).

Referensi waktu menggunakan sebuah modul timebase (*TimeBase Module*). Modul ini dapat memberikan satu interupsi pada mikrokontroler jika telah mendapatkan masukan pulsa dalam jumlah tertentu yang diatur dalam register TBCR (*Timebase Control Register*).

Tabel pengaturan masukan pulsa bagi modul dapat dilihat pada tabel berikut ini.

Tabel 3.1 Masukan Pulsa bagi Modul *Timebase*

TBR2	TBR1	TBR0	Divider Tap
0	0	0	32768
0	0	1	8192
0	1	0	2048
0	1	1	128
1	0	0	64
1	0	1	32
1	1	0	16
1	1	1	8

Dapat dilihat bahwa modul *timebase* dapat diatur untuk membuat interupsi setelah mendapatkan 8, 16, 32, 64, 128, 2.048, 8.192 atau 32.768 *clock*. Periode 1 *clock* dapat dihitung dari frekuensi *bus clock* (frekuensi *clock* sistem / 4). Dengan frekuensi *clock* sistem 8 MHz, maka frekuensi bus = 8 MHz / 4 = 2MHz. Dengan demikian, periode 1 *clock* = 1 / 2MHz = 0.5 μ s.

Pembacaan dan perhitungan bahan bakar dilakukan setiap 1 detik. Untuk menghasilkan interupsi setiap detik, maka TBCR harus diberi nilai \$0E yaitu *Timebase* modul memberikan interupsi setiap 32.768 kali mendapatkan *clock*. 32.768 kali *clock* = 32.768 x 0,5 μ s = 16.384 μ s. Sehingga *Timebase* modul memberikan interupsi setiap 16.384 μ s. Satu detik diperoleh dari 61 x 16.384 μ s = 999.424 μ s (hampir 1 S).

3.3.4 Pembacaan ADC

Dengan menggunakan pewaktuan dari *Timebase* modul, maka jumlah bahan bakar akan dibaca setiap 1 detik sekali. Setelah dibaca dan

diubah ke dalam digital, jumlah bahan bakar ini akan dijadikan bahan perhitungan untuk tiga variabel yaitu banyaknya bahan bakar dalam tangki, banyaknya bahan bakar hingga tangki penuh dan jumlah rupiah yang diperlukan untuk membeli bahan bakar hingga tangki penuh.

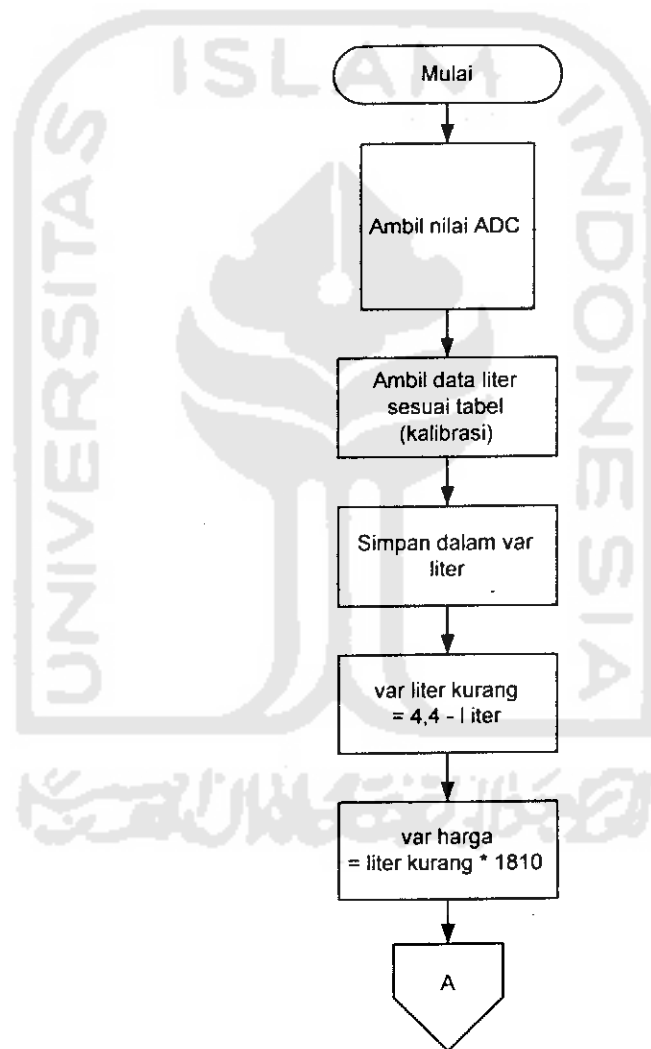
Pengubah analog ke digital dilakukan terus menerus dan dibaca tiap satu detik, sehingga saat dibaca nilai dari register ADC (hasil dari ADC) telah siap.

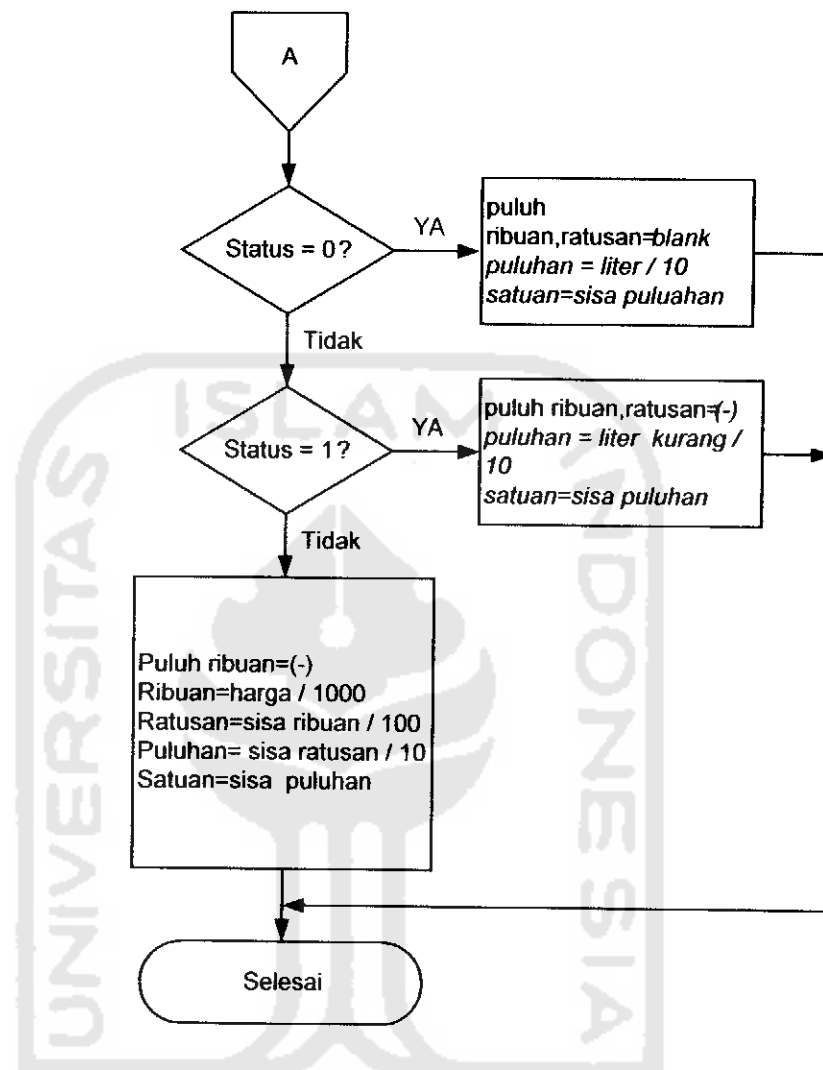
Setelah dibaca, hasil ADC tersebut dihitung dan hasilnya dimasukkan ke dalam tiga variabel (Liter, LtrKurang dan Harga_H, Harga_L). Liter adalah jumlah bahan bakar yang ada dalam tangki didapat dari pengukuran tegangan masukan ADC yang telah dirubah ke dalam digital. LtrKurang adalah jumlah bahan bakar hingga tangki penuh didapat dari 4,4 liter (kapasitas maksimum tangki) dikurangi variabel Liter (jumlah bahan bakar yang ada dalam tangki). Harga_H dan Harga_L adalah variabel 16-bit (karena nilainya besar) yang didapat dari variabel LtrKurang dikalikan dengan harga dasar bahan bakar (Premium = Rp. 1.810).

Dari ketiga variabel yang ada ini hanya satu yang akan ditampilkan pada 7-segment dan ini tergantung pada tombol mana yang ditekan. Jika tombol pertama (kiri) ditekan akan ditampilkan jumlah bahan bakar yang ada dalam tangki, tombol kedua (tengah) ditekan akan ditampilkan jumlah bahan bakar hingga tangki penuh dan tombol ketiga (kanan) ditekan akan

ditampilkan jumlah harga untuk membeli bahan bakar hingga tangki penuh.

Dibawah ini diagram alir dari proses membaca dan menghitung data :





Gambar 3.8 Diagram alir proses membaca dan menghitung data

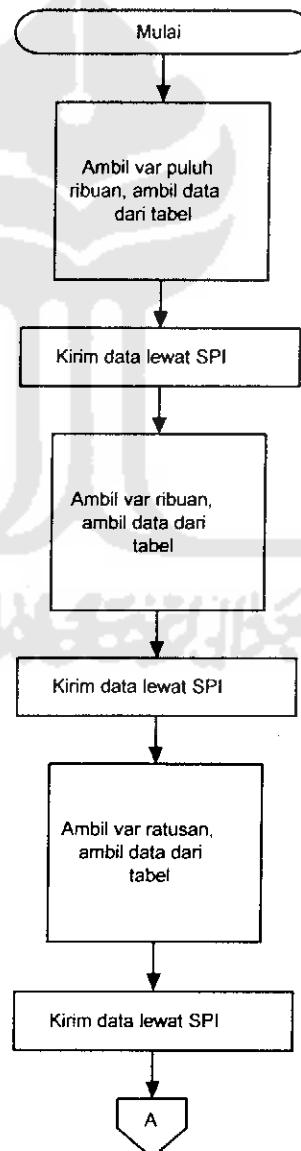
3.3.5 Tampilan 7-segment

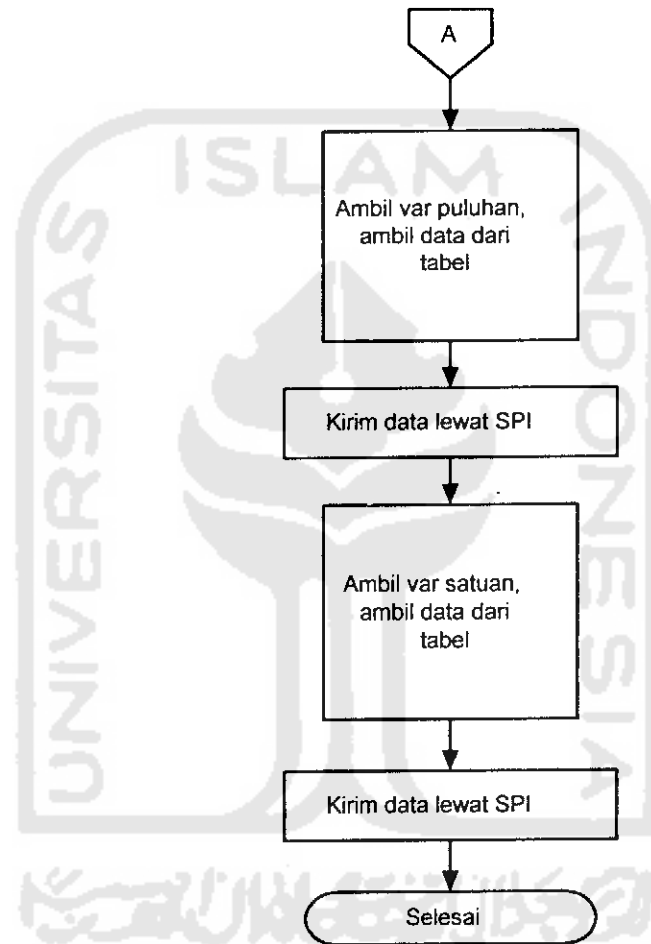
Untuk menampilkan data yang ada pada 6 buah 7-segment digunakan IC driver 74HC595 (serial in, paralel out) untuk masing-masing 7-segment. IC *driver* ini mendapat data masukan serial, *clock* dan *latch* dari mikrokontroler. Data tampilan dikirim satu persatu dari kiri oleh 7-segment.

Dengan menggunakan teknik ini hanya diperlukan 3 buah kaki dari mikrokontroler untuk menampilkan data pada 6 buah 7-segment.

Tetapi teknik ini mempunyai kelemahan, yaitu arus pada 6-segment menjadi cukup besar karena seluruh 7-segment menyala terus menerus.

Berikut adalah diagram alir dari proses tampilan data pada 7 segment :





Gambar 3.9 Diagram alir proses tampilan data pada 7 segment