

BAB III

PERENCANAAN PERANGKAT LUNAK

3.1 Umum

Sebuah computer tidak seperti beberapa mesin yang lain, karena sebelumnya dapat berbentuk suatu kesamaan yang sederhana dari perintahnya yang harus diungkapkan dalam bentuk program. Secara sederhana mikrokomputer harus diberi suatu program tertentu sebelum di switch-on. Program ini harus diinisialisasi dengan berbagai macam bagian komponen pada mikrokontroller didalam suatu system yang kompleks. Inisialisasi program dan selanjutnya di running.

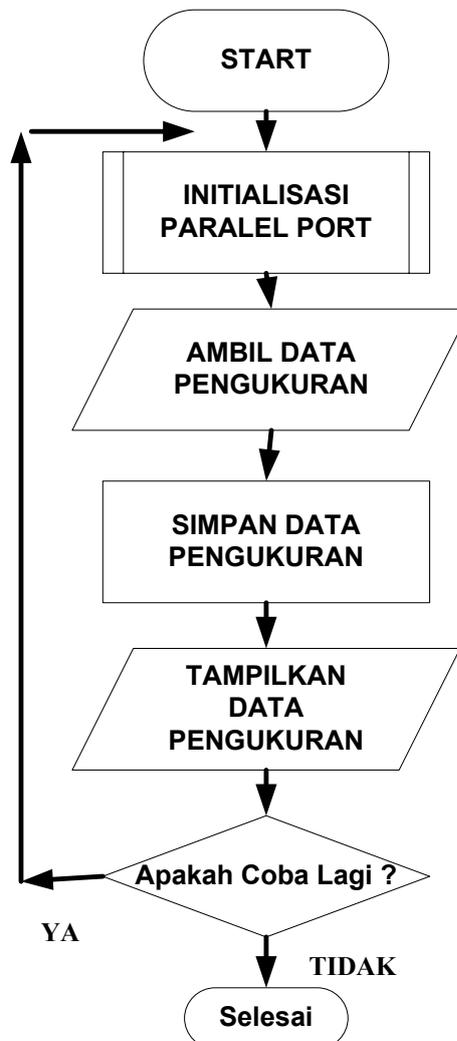
Dalam merancang suatu software, maka harus dipertimbangkan pemanfaatan dari fungsi yang akan dilakukan oleh hardware. Disini pemanfaatan hardware juga dipertimbangkan penggunaan komponen penyusunannya yang mempunyai karakteristik sesuai dengan kecepatan proses eksekusi yang dilakukan program.

Pertimbangan yang dilakukan dalam pembuatan software, haruslah meliputi data input yang harus diterima dan diproses oleh program dalam mikrokontroller. Demikian halnya dengan output yang diharapkan terdiri dari layer peraga LCD.

3.2 Perencanaan Perangkat Lunak

Untuk membentuk suatu system dengan hasil yang optimal perlu kiranya dilakukan perencanaan system tersebut sebelum merealisasikan yang sebenarnya. Hal ini sama juga dilakukan terhadap perangkat lunak.

Perencanaan perangkat lunak dalam test uji lampu ini meliputi perencanaan memori, alamat input/output juga yang juga berfungsi sebagai port input/output. Selain itu juga dibuat perencanaan fungsi dari port yang dihubungkan ke saklar lampu dan perencanaan hasil pembacaan oleh saklar lampu.



Gambar 3.1 Flowchart Secara Umum

3.2.1 Perencanaan Alamat Memori

Sebelum merencanakan alamat yang digunakan untuk menyimpan data, baik data sementara pada RAM maupun data permanent yang terletak pada ROM. Maka terlebih dahulu dibahas terlebih dahulu perencanaan pemetaan memori. Sehingga nantinya dapat dihindari kesalahan penempatan data.

Pada pemakaian mikrokontroler AVR ATmega8535 ini dikenal adanya internal RAM karena dalam penggunaan perangkat lunak tes uji coba bola lampu ini membutuhkan alamat yang panjang untuk menyimpan data hasil pembacaan saklar lampu dalam pembuatan hardware yang ditambahkan eksternal RAM. RAM yang ditambahkan dalam system mikrokontroler ini mempunyai kemampuan memori data 128 K byte, sehingga alamat yang dapat digunakan adalah 0000H – FFFFH.

Pada mikrokontroler AVR ATmega8535 tidak dilengkapi dengan internal ROM. Jadi harus ditambahkan eksternal ROM. Untuk penambahan eksternal ROM digunakan memori yang sudah ada dalam mikrokontroler tersebut, dengan kemampuan memori program 128 K byte. Alamat yang digunakan adalah 0000H-FFFF H. Untuk alamat dibawah 100H tidak digunakan karena alamat ini merupakan alamat dari interrupt yang ada pada mikrokontroler.

3.2.2 Perencanaan Input / Output

Unit input / output yang digunakan adalah pin yang ada pada mikrokontroler. Unit input yang digunakan untuk pengambilan data dari pembacaan saklar lampu. Pembacaan data dari saklar lampu tersebut dihubungkan ke pin T1 pada mikrokontroler.

Unit output adalah unit yang digunakan untuk menampilkan data hasil pembacaan saklar lampu. Dalam Tugas akhir ini digunakan LCD sebagai tampilan data yang sudah diproses dalam mikrokontroller dalam bentuk numerik.

3.2.3 Program Utama

Program utama adalah program yang dilaksanakan setelah alat diaktifkan atau di reset secara hardware.

Cara kerja dari program utama dapat digambarkan sebagai berikut :

a.1 Persiapan

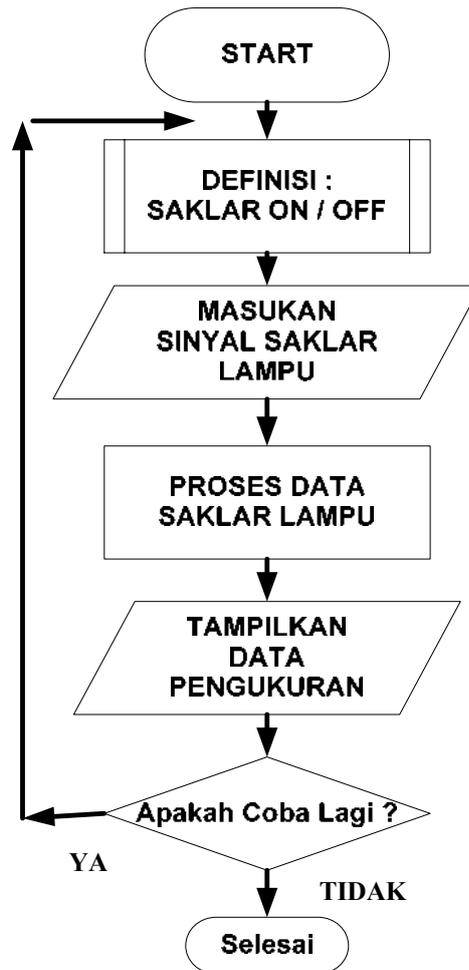
Yaitu mendefinisikan semua variabel yang digunakan yang meliputi address port, stack pointer, dan inialisasi

b.1 Pembacaan Data

Melakukan pembacaan data tegangan dari saklar lampu yang sudah diubah oleh sinyal analog menjadi digital.

c.1 Tampilan Data

Menampilkan data yang sudah terlebih dahulu diproses dalam mikrokontroller pada display LCD.



Gambar 3.2 Flowchart Program Utama

3.2.4 Inisialisasi

Pada program ini dilakukan inisialisasi stack pointer, mode port paralel dan semua variable yang digunakan dalam seluruh program. Pada mikrokontroller AVR ATmega8535 dan keluarganya alamat stack pointer akan naik pada operasi push dan setengah reset hardware stack pointer akan menuju alamat 09H. Sebab itu lokasi tersebut harus dipindahkan karena nantinya alamat tersebut akan dipergunakan untuk menyimpan data. Maka stack pointer dialamatkan pada alamat 80H.

Pada mikrokontroler AVR ATmega8535 perlu diinisialisasi untuk mode operasi dan fungsi dari masing-masing port. Direncanakan mikrokontroler AVR ATmega8535 tersebut beroperasi pada pin 0 sampai pin 7 dari port A. sebagai output dan port B sebagai input. Sehingga dari ketentuan tersebut control word akan diisi dengan kode 95H. Dari ketentuan tersebut diatas dapat dibuat contoh program inisialisasi sebagai berikut :

```
.include"C:\Appnotes\m8535def.inc"

.org 0x000
rjmp main

main:
Ldi r16, low(RAMEND)
out SPL, r16
Ldi r16, high(RAMEND)
out SPH, r16

Ldi r16,0xff

out ddra,r16
out PORTA,r16

cbi PORTA,0
cbi PORTA,1

henti :
rjmp henti
```

Dari program di atas diketahui bahwa porta akan berlogika high dan pada bit ke 0 dan bit ke 1 akan low, seperti terlihat pada jendela workspace yang terdapat pada lampiran.

Pada workspace, kotak yang berwarna putih menandakan bit logika low, sedangkan kotak yang berwarna hitam menandakan bit logika high. Pada program diatas terdapat kode `.include"C:\Appnotes\m8535def.inc"`. Kode tersebut berguna untuk meng-include-kan file m8535def.inc kedalam program .asm. File

m8535def.inc. yang berisi definisi dari mikrokontroler ATmega8535. Dengan menyertakan file tersebut, program assembly yang kita buat lebih dapat dimengerti dan dipahami.

3.2.5 Pengambilan Data

Pengambilan data yang dilakukan adalah pengambilan data dari ADC yang sudah ada pada mikrokontroller AVR ATmega8535. Setelah mikrokontroler AVR ATmega8535 memproses data dari saklar lampu. Data dari proses mikrokontroler AVR ATmega8535 ini dilakukan pengambilan melalui pin dari port mikrokontroller tersebut. Adapun instruksi pada saat pengambilan data tegangan yang nantinya disimpan dalam RAM pada alamat 75H. Instruksi tersebut adalah :

```
Mov  dptr, #0000
      Mov  a, #40h
      Movx @dptr, aa    ; start konversi
      Mov  dptr, #0001
      Movx a, @dptr    ; ambil data tegangan
      Mov  nyala_data, a ;
      ret
```

3.2.6 Subrutin Display

Subrutin ini menampilkan data tampilan pada peraga LCD. Penampilan yang dilakukan dengan cara multipleks yaitu pada suatu waktu hanya satu dari

digit display yang dinyalakan. Karena proses penyalaan lampu diatur dengan saklar untuk tiap digit adalah 10 ms, sehingga untuk menampilkan empat digit display dari LCD diperlukan waktu 4×10 ms atau sama dengan 40 ms dengan frekuensi 50 hz, sehingga akan tampak menyala bersama-sama.

Berikut adalah contoh program untuk menyalakan display LCD.

Display :

```
Mov  a,disbuf1    ; digit 1
Icall outa        ; kirim data ke port A
Mov  a,#1h
Icall outc        ; kirim data ke port C
Icall delay 10    ; delay 10 mdt
```

Instruksi tersebut diatas diulang untuk disbuf2, disbuf3 dan disbuf4 untuk digit kedua, ketiga dan digit ke empat.