

BAB III

METODE PENELITIAN

3.1. Jenis Metode Penelitian

Sesuai dengan permasalahan yang telah dikemukakan sebelumnya, maka jenis penelitian ini adalah penelitian eksperimen. Penelitian eksperimen adalah penelitian yang terdapat tindakan yang digunakan untuk mencari pengaruh dari tindakan tertentu terhadap yang lain dalam kondisi yang terkendalikan. [9]

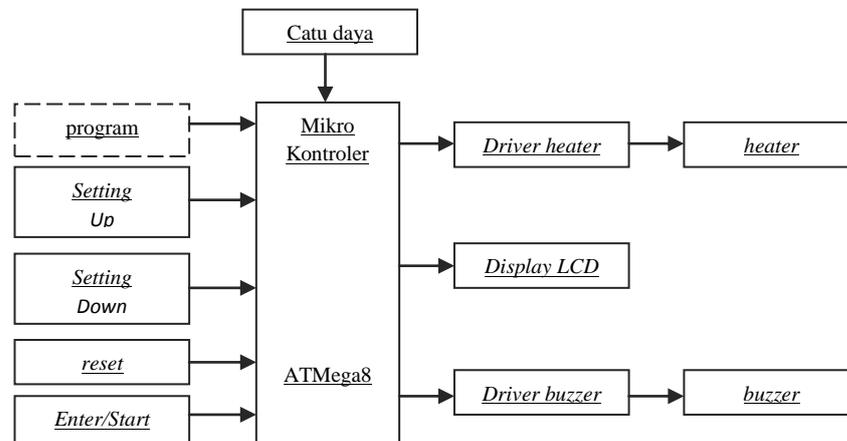
3.2. Tahap Pelaksanaan

Adapun tahap pelaksanaan yang ditempuh dalam pembuatan Tugas Akhir ini adalah sebagai berikut :

1. Perencanaan Blok Diagram.
2. Perencanaan Diagram Alir (*Flowchart*).
3. Daftar Komponen
4. Perencanaan desain alat.
5. Merancang rangkaian yang akan digunakan.
6. Mencari referensi sebagai sumber informasi.
7. Menyiapkan komponen dan peralatan yang dibutuhkan.
8. *Me-layout* rangkaian pada *PCB*.
9. Melakukan pengukuran dan pengujian.
10. Menganalisa hasil pengukuran untuk mendapatkan kesimpulan.

3.3. Perencanaan Diagram Blok

Di bawah ini adalah gambar blok diagram dari alat.



Gambar 3.19. Blok Diagram Penghangat ASIP

3.3.1. Fungsi Masing-Masing Blok

1. Catu Daya

Catu daya ini digunakan untuk memberi sumber tegangan *DC* yang dibutuhkan oleh *hardware* seperti mikrokontroler, *LCD*, *buzzer* Pada bagian catu daya ini dibutuhkan satu nilai tegangan yaitu *5 Volt*.

2. Program

Blok Program adalah jenis bahasa yang dimasukkan ke Mikrokontroler untuk diolah. Yang menghasilkan sinyal digital sebagai perintah dari program yang telah dibuat penulis.

3. Blok Reset

Reset berfungsi untuk me-*reset* ulang ketika pengoperasian sedang berlangsung.

4. Blok *Enter/Start*

Berfungsi untuk memulai pengoperasian pada alat.

5. Blok Mikrokontroler AVR ATmega8

Pada blok ini terdapat Mikro yang berfungsi sebagai otak atau pusat pengendali utama dari rangkaian keseluruhan. Mikrokontroler AVR ATmega8 akan diisi program yang telah diinginkan untuk menampilkan pada *display LCD*. Masukan dan perintah yang didapat oleh Mikrokontroler AVR ATmega8 berasal dari tombol *setting up* dan *setting down*, *Enter* atau *Start* dan *Reset*.

6. Blok *Driver Heater*

Rangkaian pengontrol *heater*.

7. Blok *Heater*

Komponen elektronik pemanas.

8. Blok *Setting Up*

Berfungsi untuk memberi perintah menaikkan lama waktu yang akan dipakai.

9. Blok *Setting Down*

Berfungsi untuk memberi perintah menurunkan lama waktu yang akan dipakai.

10. Blok *Driver Buzzer*

Sebagai rangkaian pengontrol pada *buzzer*.

11. Blok *Buzzer*

Untuk indikator bunyi bahwa pengoperasian alat sudah selesai.

12. Blok *Liquid Crystal Display (LCD)*

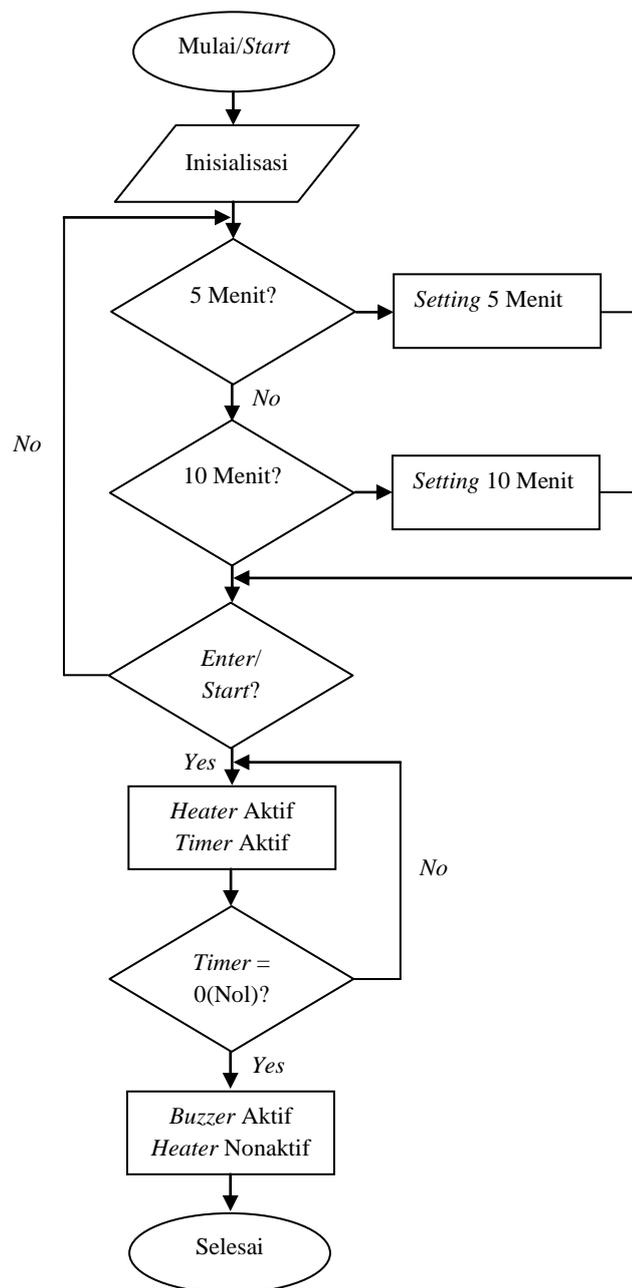
Pada blok *LCD* berfungsi untuk menampilkan perintah dari program yang sudah diolah oleh mikro. Dan menampilkan pada *display LCD* hitungan mundur seberapa lama waktu (*timer*) yang sudah ditentukan.

3.3.2. Cara Kerja Blok Diagram

Untuk penjelasan cara kerja blok diagram adalah, yang pertama catu daya memberi tegangan masukan pada mikro sebesar 5 volt, pilih waktu yang ditentukan 5 menit atau 10 menit dengan menggunakan *setting up* dan *setting down*. Setelah waktu terpilih, program yang sudah dimasukkan ke dalam mikro terbaca dan dari proses *setting* dan tampilan *timer* pada *LCD*. Setelah dipilih akan otomatis tampilan *timer* memulai penghitungan mundur, bersamaan dengan itu *heater* aktif. Untuk blok *reset* akan digunakan sebagai pengulangan program dari awal, jadi ketika alat sedang berlangsung, jika tombol *reset* ditekan, maka *timer* dan *heater* akan berhenti bekerja dan tampilan pada *LCD* kembali pada tampilan awal. Tetapi jika waktu yang terpilih sudah tercapai, bersamaan *timer* berhenti pada posisi 0 (nol) dan *heater* berhenti (*nonaktif*). Bersamaan dengan itu *buzzer* menyala sebagai pertanda penghangatan Air Susu Ibu Perah (ASIP) sudah selesai.

3.4. Perencanaan Diagram Alir (*Flowchart*)

Di bawah ini adalah susunan diagram alir (*flowchart*).



Gambar 3.20. *Flowchart* Rangkaian Penghangat ASIP

3.4.1. Cara Kerja *Flowchart*

Untuk pembacaan *flowchart*, ketika proses Mulai atau *Start* berlangsung proses pemasukan data inisialisasi langsung memberikan perintah untuk memilih waktu yang ditentukan, yaitu *setting* terlebih dahulu 5 Menit/10 Menit. Setelah itu, tentukan pilihan waktu 5 Menit/10 Menit? apakah waktu sudah ter-*setting* “5 menit?” jika belum “*Setting* 5 Menit”. kemudian jika *No* *setting* waktu 10 Menit. apakah sudah ter-*setting* “10 Menit?” jika belum “*Setting* 10 Menit”. selanjutnya jika waktu sudah ditentukan. Apakah memulai “*Enter/Start?*” Jika *No* proses kembali pada proses pemilihan data yang akan ditentukan. Jika *Yes* maka bersamaan proses “*heater* aktif dan tampilan *timer* hitungan mundur aktif” dan beroperasi sesuai perintah dari data yang diperintahkan. Setelah itu apakah *timer* sudah bekerja sesuai perintah data yang ditentukan? Atau apakah “*timer=0(Nol)?*”, jika *No* atau waktu yang ditentukan belum terpenuhi, maka *heater* dan *timer* akan tetap bekerja sesuai waktu yang ditentukan. Jika *Yes* akan memberi perintah proses *heater* dan *timer* akan berhenti bekerja bersamaan dengan itu proses *buzzer* menyala. Dan untuk alir yang terakhir adalah “Selesai” bahwa alir *flowchart* telah selesai.

3.5. Daftar Komponen

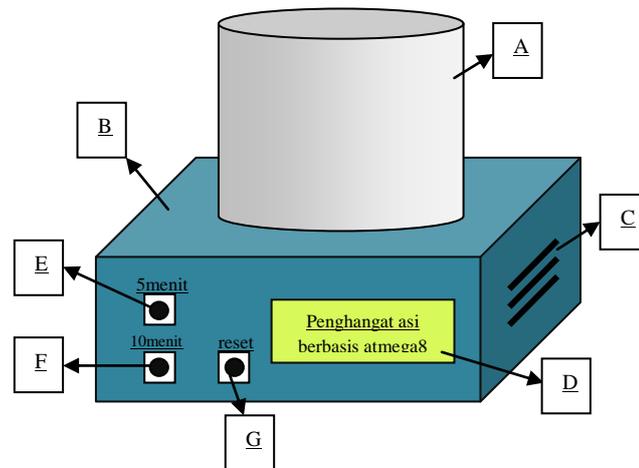
Di bawah ini adalah daftar tabel komponen yang dibutuhkan dalam pembuatan *project* alat Modifikasi Penghangat Air Susu Ibu Perah (ASIP).

Tabel 3.4. Daftar Tabel Komponen

No	Nama Komponen	Keterangan	Jumlah
1	Trafo	1 Ampere	1
2	Dioda <i>Bridge</i>	RC 102	1
3	Kapasitor	4700 uf/10 Volt	1
4	Mikrokontroler	Atmel ATmega8A	1
5	Resistor Variabel	1 Kilo Ohm	1
6	Resistor	1 Kilo Ohm	2
7	Transistor	BD139 JB414	1
8	<i>Crystal</i>	M16.000D4D	1
9	Kapasitor	Kramik 22 Pf	2
10	Kapasitor	Kramik 10.000 Pf	1
11	Dioda	1N4002	1
12	<i>Relay</i>	DC 5 Volt	1
13	<i>Liquid Crystal display (LCD)</i>	16x2	1
14	Regulator	IC7805 5 Volt	1

3.6. Perencanaan Desain Alat

Di bawah ini adalah bentuk fisik *box* yang akan di buat untuk alat Modifikasi Penghangat Air Susu Ibu Perah (ASIP).



Gambar 3.21. Desain Alat Penghangat ASIP

Keterangan Desain :

- a. Wadah botol, terbuat dari *stainless stell*
- b. *Box* alat,
- c. Lubang angin, untuk mengurangi panas yang dihasilkan *heater*
- d. *LCD* tampilan dari proses kerja alat
- e. Tombol pilih 5 menit,
- f. Tombol pilih 10 menit,
- g. Tombol *reset*

3.7. Perancangan Perangkat Keras (*Hardware*) dan Perangkat Lunak (*Software*)

3.7.1. Perancangan Skematik Menggunakan *Software Proteus*

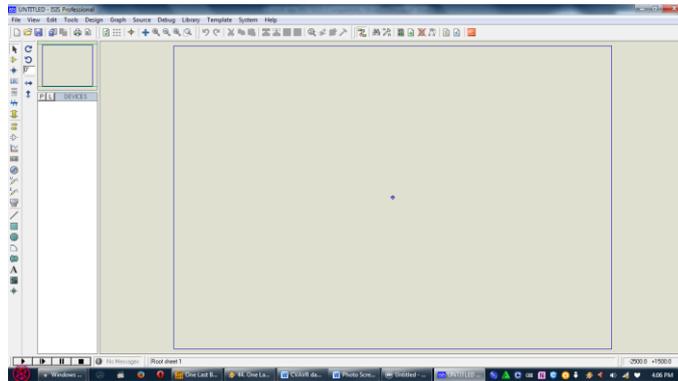
1. Pengenalan *Proteus*

Proteus adalah sebuah *software* yang memiliki program untuk mendesain *layout* yang akan dicetak pada *printed circuit board (PCB)* Dan Program *ISIS* untuk perangkaian skematik. Program *software* ini berguna sekali ketika rangkaian yang dibuat dapat dijalankan, dan dapat mengetahui kebenaran dari rangkaian yang akan dibuat. Jadi rogram *ISIS* ini untuk membuat skematik desain rangkaian.

Pengalaman saya menggunakan *Proteus* ini, *software* ini bagus digunakan untuk desain rangkaian mikrokontroler. *Proteus* juga bagus untuk belajar elektronika seperti dasar-dasar elektronika sampai pada aplikasi mikrokontroler. *Software* ini jika di-*install* menyediakan banyak contoh aplikasi desain yang disertakan, sehingga kita bisa belajar dari contoh-contoh skematik yang sudah ada.

2. Penggunaan *Software ISIS*

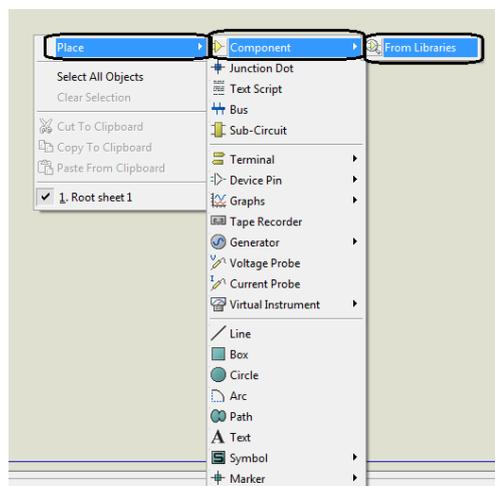
Seperti telah disinggung sebelumnya, untuk melakukan perancangan skematik, gunakan *ISIS*. Jalankan *ISIS* hingga muncul tampilan seperti pada Gambar 3.22..



Gambar 3.22. Tampilan *New Page* atau Halaman Perancangan Skematik

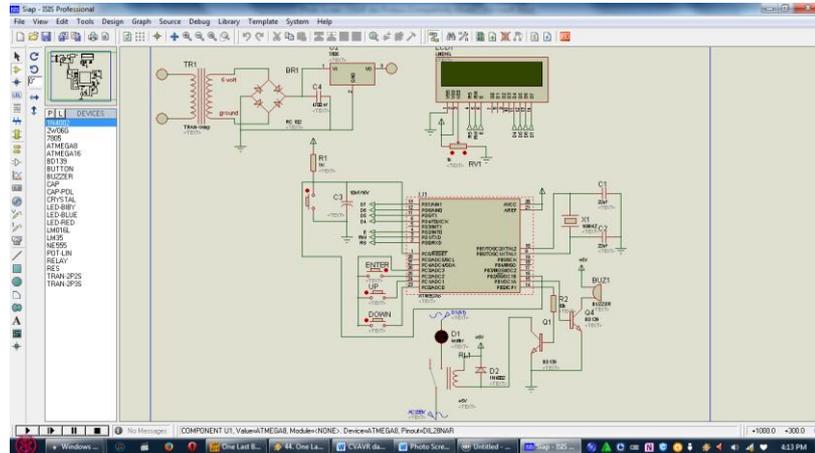
Perancangan skematik perangkat keras ini dibuat untuk mengetahui program yang akan digunakan atau dimasukkan dalam mikrokontroler dapat mengoperasikan *hardware* yang sudah disimulasikan.

Untuk langkah-langkah pembuatan skematik yaitu, masukkan beberapa komponen yang akan dibutuhkan untuk pembuatan skematik. Cara pengambilan komponen-komponen dapat dilihat pada Gambar 3.23. dan Gambar 3.24.



Gambar 3.23. Cara Pengambilan Komponen-Komponen

Untuk hasil skematik yang telah dirangkai, dapat dilihat pada Gambar.3.25..



Gambar 3.25. Skematik Modifikasi Penghangat Air Susu Ibu Perah (ASIP) Keseluruhan

Dapat dilihat pada Gambar 3.25., bahwa skematik sudah selesai dirangkai, tinggal program mana yang akan dimasukkan ke dalam mikrokontroler tersebut. Agar semua yang sudah terangkai dapat bekerja dan dapat disimulasikan.

3.7.2. Perancangan Perangkat Lunak (*Software*)

1. Pembuatan Program menggunakan *software CodevisionAVR*

CodevisionAVR merupakan sebuah *cross-compiler C*, *Integrated Development Environment (IDE)*, dan *Automatic Program Generator* yang didesain untuk mikrokontroler buatan *Atmel* seri *AVR*.

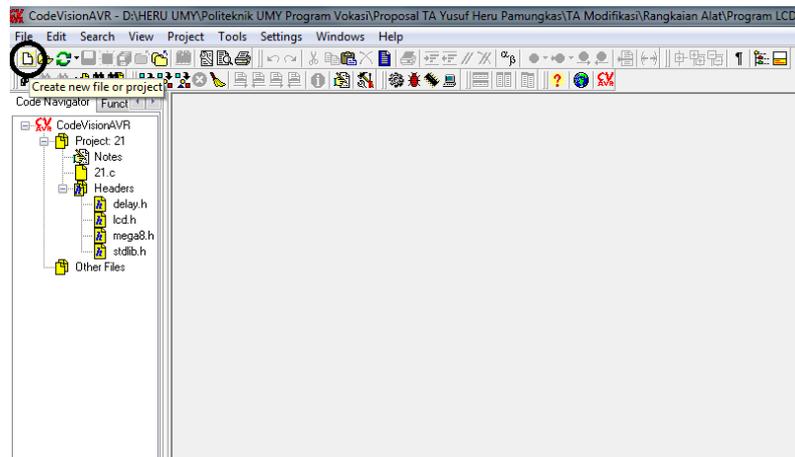
Untuk keperluan *debugging* sistem *embedded*, yang menggunakan komunikasi serial, *Integrated Development Environment (IDE)* mempunyai fasilitas internal berupa sebuah Terminal. Selain *library* standar C, *CodevisionAVR* juga mempunyai *library* tertentu.

2. Penggunaan *CodevisionAVR*

CodevisionAVR mempunyai suatu keunggulan dari *compiler* lain, yaitu adanya *codewizard*, fasilitas ini memudahkan kita dalam inialisasi mikrokontroler yang akan kita gunakan, *codevision* telah menyediakan konfigurasi yang bisa diatur pada masing-masing *chip* mikrokontroler yang akan kita gunakan, sehingga kita tidak perlu melihat *datasheet* untuk sekedar mengkonfigurasi mikrokontroler.

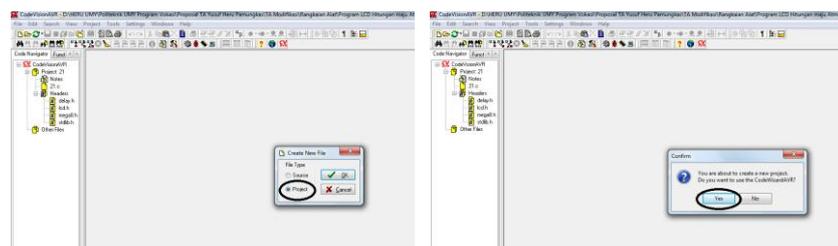
3. Langkah-langkah membuat *project* baru menggunakan *CodevisionAVR*

Untuk memulai membuat *project* baru, pada *menubar*, pilih *File* → *New*, selain itu kita juga dapat membuka dengan cara klik *icon* seperti ditunjukkan pada gambar Gambar.3.26. yang berada pada *Toolbar*.



Gambar 3.26. Tampilan Halaman Awal Pada *Software CodevisionAVR*

Setelah *icon New* di klik, akan muncul tampilan seperti yang diperlihatkan pada Gambar 3.27. (a) terdapat 2 pilihan, pilihlah *icon "project"* lalu klik "*ok*". Selanjutnya akan muncul tampilan pertanyaan seperti yang ditunjukkan pada Gambar 3.27. (b) jika ingin membuat *project* baru tekan "*yes*".



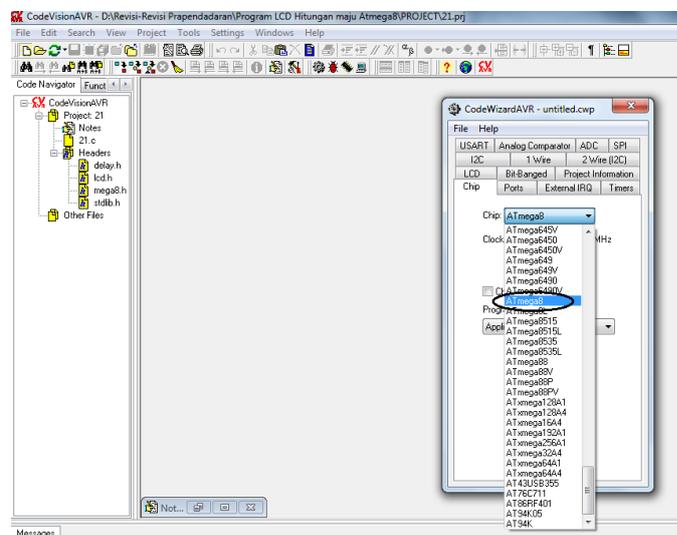
(a)

(b)

Gambar 3.27. Tampilan Memulai Pembuatan *Project* (Program)

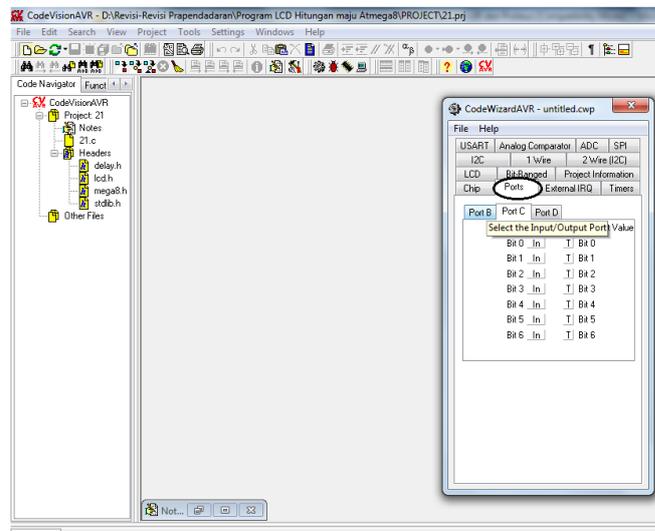
Untuk selanjutnya akan muncul tampilan pencocokan *Chip* dengan *IC* yang Anda gunakan.

Dapat dilihat pada Gambar 3.28.. *Tab-tab* pada *CodeWizardAVR* menunjukkan fasilitas yang dimiliki oleh *chip* yang Anda pilih. Cocokkan pula frekuensi *Crystal* yang Anda gunakan pada bagian *Clock*. Pengisian frekuensi *clock* digunakan oleh *software* untuk menghitung rutin-rutin seperti *delay* agar diperoleh perhitungan yang cukup akurat.

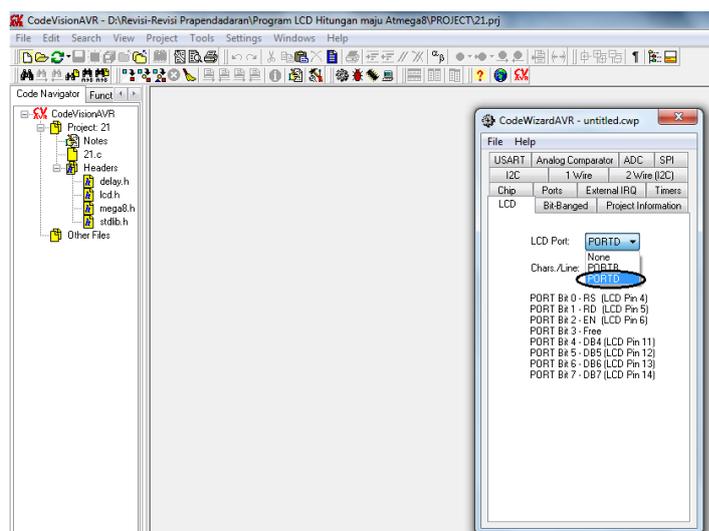


Gambar 3.28. Pencocokan Pada *Tab Chip*

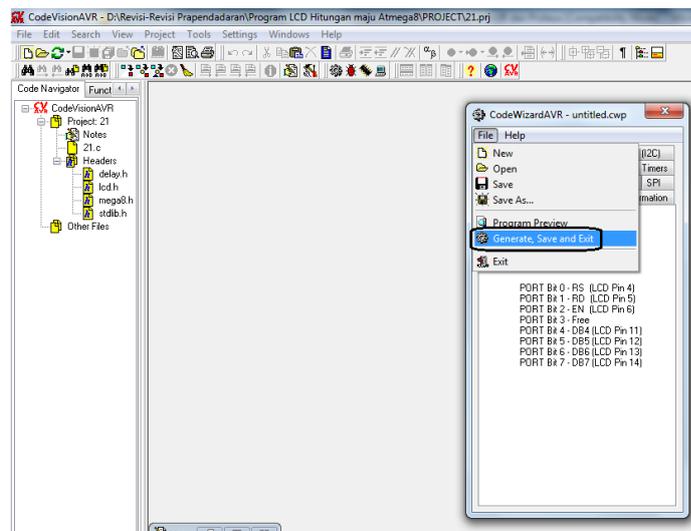
Selanjutnya merupakan inisialisasi dari *PORT* yang ditunjukkan Gambar 3.30. Terdapat 3 *PORT*, yaitu *PORT B*, *PORT C* dan *PORT D*. kita bisa menginisialisasi *port-port* ini sebagai *INPUT* ataupun sebagai *OUTPUT*. Selain itu kita dapat memilih aktif *HIGH* (1) atau aktif *LOW* (0) untuk *OUTPUT* ataupun *INPUT*.

Gambar 3.29. Inisialisasi *Port-Port*

Selanjutnya merupakan inisialisasi dari *PORT* yang akan digunakan untuk *display Liquid Crystal Display (LCD)*. Dapat ditunjukkan pada Gambar 3.30.. Ada 2 pilihan *PORT* yaitu, *PORT B* dan *PORT D*. dan pilih salah 1, *port* mana yang dipakai untuk inisialisasi ke *LCD*.

Gambar 3.30. Penginisialisasi *Port D* sebagai *LCD*

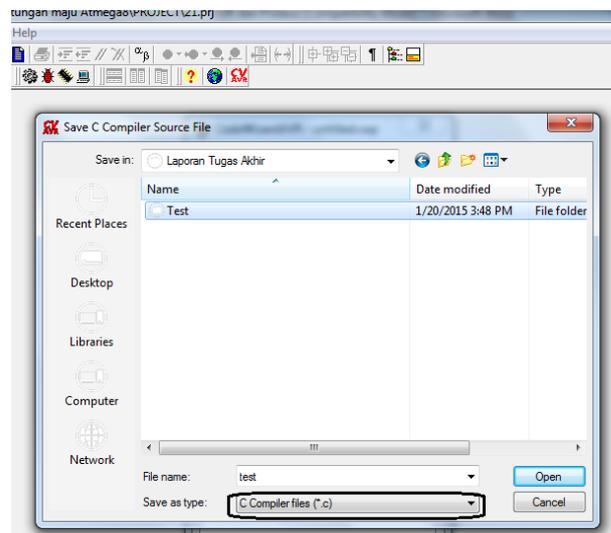
Setelah pencocokan yang kita inginkan dari mikrokontroler selesai, maka *setting CodewizardAVR* siap disimpan dalam *file*. Pada menu *CodewizardAVR*, pilih *File* → *Generate, Save and Exit*, seperti yang ditunjukkan pada Gambar 3.31..



Gambar 3.31. Penyimpanan *CodewizardAVR*

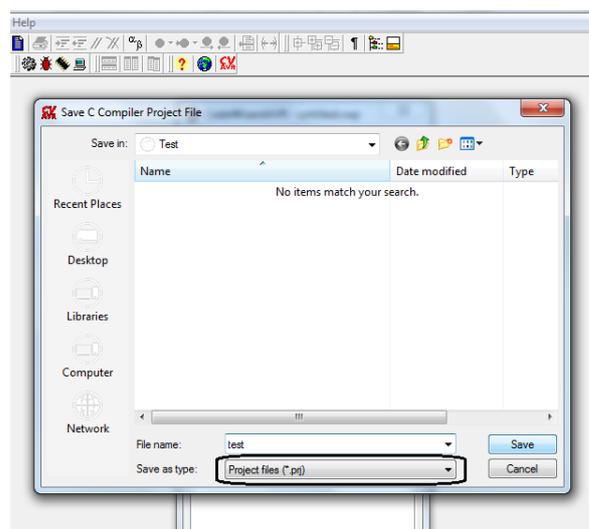
Agar file yang dihasilkan tidak berantakan, buatlah sebuah folder baru, untuk nama terserah dari anda.

Kemudian masuk ke dalam folder tersebut untuk menyimpan *file-file* yang dihasilkan oleh *CodewizardAVR*. Yang pertama Anda diminta untuk memberikan nama *file C* yang dihasilkan. Misalnya beri nama “test”, lalu klik tombol *Save*. Lebih jelas pada Gambar 3.32. *File* tersebut nantinya akan mempunyai akhiran “.C”.



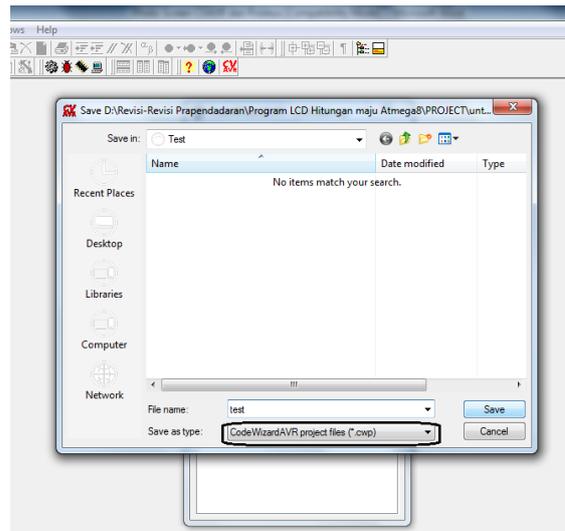
Gambar 3.32. Penyimpanan *Compiler Source File*

Yang kedua Anda diminta untuk memberikan nama file project yang dihasilkan. Misalnya beri nama “test”, lalu klik tombol *Save*. Lebih jelas pada Gambar 3.33.. *File* tersebut nantinya akan mempunyai akhiran *.prj*.



Gambar 3.33. Penyimpanan *Compiler Project File*

Yang terakhir Anda diminta untuk memberikan nama *file project Codewizard* yang dihasilkan. Misalnya beri nama “test”, lalu klik tombol *Save*. Lebih jelas pada Gambar 3.34.. *File* tersebut nantinya akan mempunyai akhiran *.cwp*.



Gambar 3.34. Penyimpanan *CodewizardAVR File*

Setelah ketiga file disimpan maka akan muncul lembar kerja dapat dilihat pada Gambar 3.35.. Pada bagian atas merupakan nama dari *project*. Selanjutnya pada bagian bawah dari nama *project* yaitu merupakan program bahasa C yang merupakan program yang telah di-*setting* pada langkah awal (pada form inialisasi).

Jadi dari program “`#include <mega8.h> sampai // LCD module initialization lcd_init(16);`” merupakan program yang telah di-*setting* pada langkah awal (pada *form* inialisasi).

Untuk program yang sudah tersusun dapat dilihat pada Gambar 3.35.

```

22 .....
23 unsigned int kali=0, detik=00, menit=6;
24 unsigned char cdetik[10], cmenit[10];
25
26
27 #include <mega8.h>
28 #include <stdlib.h>
29 #include <delay.h>
30 // Alphanumeric LCD Module functions
31 #asm
32 .equ __lcd_port=0x12 ;PORTD
33 #endasm
34 #include <lcd.h>
35 #define tombol_enter PINC.2
36 #define tombol_up PINC.1
37 #define tombol_down PINC.0
38 #define on 0
39 #define off 1
40 bit status_up=0,status_down=0, status_enter=0 ;
41 // Timer 2 output compare interrupt service routine
42 interrupt [TIM2_COMP] void timer2_comp_isr(void)
43 {
44 // Place your code here
45 TCNT2=0;
46 if(++kali==787)
47 {
48 //fungsi menjalankan penambahan jam;menit;detik untuk menca
49 if (--detik==1)
50 {
51 lcd_clear();
52
53 if (--menit==1)
54 {
55 menit=0;
56 // TIMSR=0x00;
57 lcd_clear();
58 PORTB=0x0E;
59 lcd_gotoxy(0,0);
60 lcd_putsf(" SELESAI ");
61
62 }
63 }
64 }
65 }

```

Gambar 3.35. Hasil Lembar Kerja Pada *CodevisionAVR*

Pastikan program yang dibuat sudah benar.

```

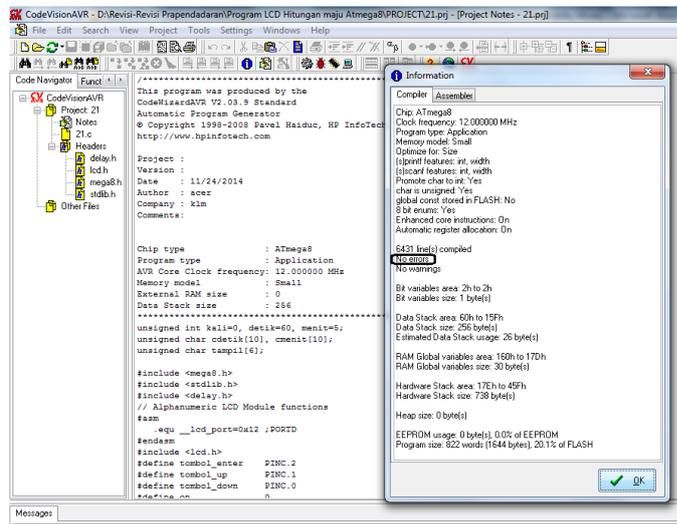
22 .....
23 unsigned int kali=0, detik=00, menit=5;
24 unsigned char cdetik[10], cmenit[10];
25
26
27 #include <mega8.h>
28 #include <stdlib.h>
29 #include <delay.h>
30 // Alphanumeric LCD Module functions
31 #asm
32 .equ __lcd_port=0x12 ;PORTD
33 #endasm
34 #include <lcd.h>
35 #define tombol_enter PINC.2

```

Gambar 3.36. *Icon* Untuk Mengkompilasi Program

Setelah program utama telah ditambahkan, kemudian pilih menu *Project* → *Compile* untuk melakukan kompilasi, bisa langsung klik *icon* pada menu *toolbar* seperti yang ditunjukkan

pada Gambar 3.36.. Lalu kotak *dialog* seperti ditunjukkan Gambar 3.37. akan muncul. Klik tombol “OK”.



Gambar 3.37. Tampilan Setelah Dilakukan kompilasi

Setelah kita kompilasi dan tidak ada informasi *error* (apabila terdapat informasi *warning*, program kita masih dapat dijalankan atau tidak bermasalah) pada program kita, selanjutnya klik tombol “OK” yang berada pada kanan bawah.

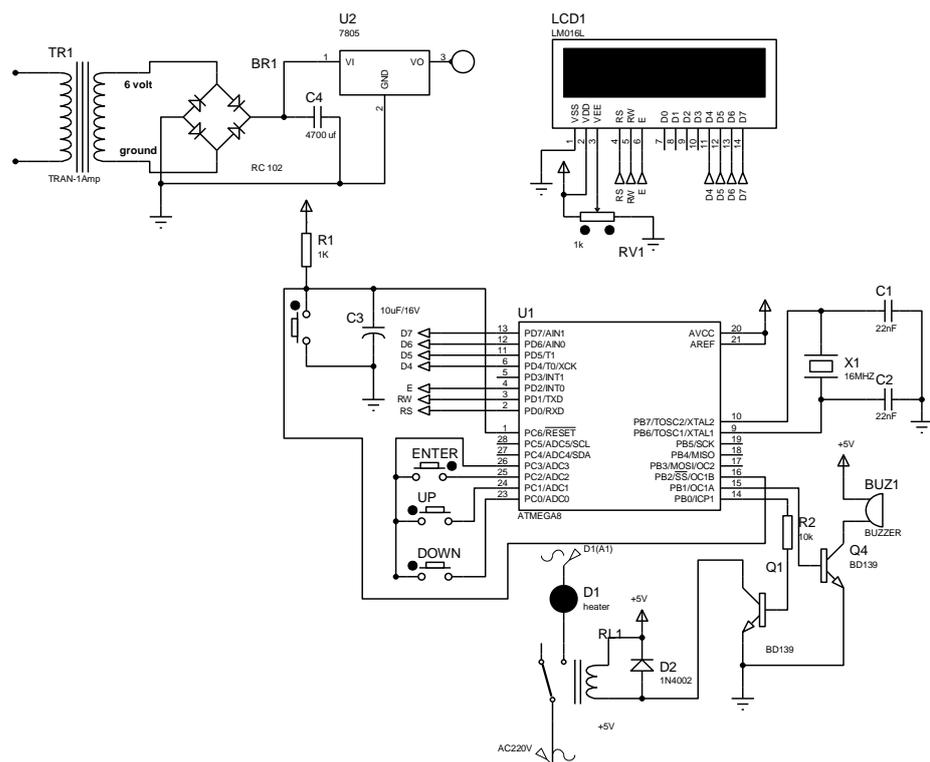
Setelah itu kita buka file tempat dimana kita menyimpan proyek kita tadi. Kemudian buka file folder bernama ‘Exe’ Setelah itu akan muncul file ber-ekstensi ‘hex’. File inilah yang nantinya akan kita downloadkan ke dalam mikrokontroler yang akan digunakan.

3.8. Perancangan Perangkat Keras (*Hardware*)

3.8.1. Rangkaian Keseluruhan

Hasil skematik perangkat keras ini dibuat untuk mengetahui program yang akan digunakan atau dimasukkan dalam mikrokontroler dapat mengoperasikan *hardware* yang sudah disimulasikan seperti pada Gambar 3.38..

Di bawah ini adalah gambar skematik keseluruhan dari skematik perangkat keras alat Modifikasi Penghangat Air Susu Ibu Perah (ASIP).

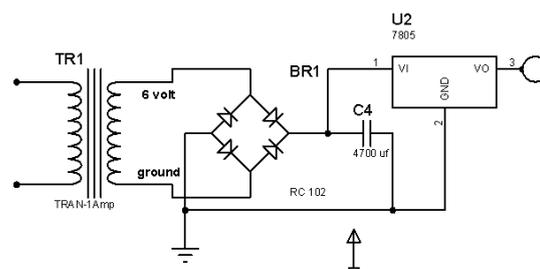


Gambar 3.38. Skematik Keseluruhan

3.8.2. Skematik dan Rangkaian Catu Daya

Catu daya berfungsi untuk memberikan *suplly* tegangan, terutama pada *mikrokontroler* ATmega8 yang hanya bisa bekerja pada tegangan 4,5Volt-5,5Volt. Dengan menggunakan trafo 1 Ampere dan memakai keluaran 6 Volt, dioda *bridge* digunakan untuk menyearahkan gelombang penuh dari AC ke DC, Sedangkan kapasitor berfungsi sebagai *filter* tegangan yang dihasilkan dioda *bridge*, maka untuk menstabilkan tegangan menjadi 5 volt digunakan IC LM7805 (regulator tegangan) agar tegangan yang dihasil kan tidak melebihi atau kurang dari yang dibutuhkan Mikrokontroler ATmega8. Untuk gambar Skematik dan rangkaian perangkat keras catu daya dapat dilihat pada Gambar 3.39. dan Gambar 3.40.

Di bawah ini adalah gambar skematik dari catu daya.



Gambar 3.39. Skematik Catu Daya 5 Volt

Di bawah ini adalah gambar dari rangkaian perangkat keras catu daya.



Gambar 3.40. Perangkat Keras Catu Daya

3.8.3. Skematik dan Rangkaian Perangkat Keras *Minimum System*

Rangkaian *minimum system* mikrokontroler ini adalah rangkaian utama yang digunakan untuk memfungsikan mikrokontroler ATmega8, dimana rangkaian ini merupakan pengontrol utama dalam alat Modifikasi Penghangat Air Susu Ibu Perah (ASIP). Rangkaian ini difungsikan juga untuk melakukan *programming*. Pada rangkaian *minimum system* ini menggunakan kristal 16MHz sebagai pembangkit sinyal *eksternal*.

Di bawah ini adalah gambar dari rangkaian *minimum system* beserta *driver buzzer driver heater*.

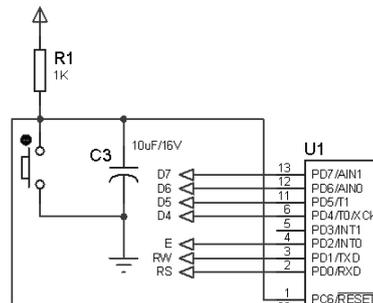


Gambar 3.41. Perangkat Keras *Minimum System*

3.8.4. Skematik Tombol-Tombol Pengontrol

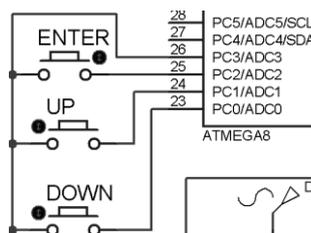
Rangkaian tombol pengontrol adalah sarana *input* data atau perintah bagi mikrokontroler. Sarana *input* tersebut berupa sinyal *low* untuk pengisian maupun pemilihan data. adanya masukan sinyal *low* ke *pin* mikrokontroler ketika salah satu tombol ditekan sehingga terhubung ke *ground*. Jika salah satu tombol tidak ditekan, maka kaki *mikrokontroler* dalam keadaan sinyal *high* karena memiliki *internal pull up resistance*. Skematik tombol-tombol, dapat dilihat pada Gambar 3.42. dan Gambar 3.43.

Untuk skematik dari tombol *reset* dapat dilihat pada gambar 3.42.



Gambar 3.42. Skematik Tombol *Reset*

Untuk skematik tombol *enter*, *Setting Up* dan *Setting Down* dapat dilihat pada Gambar 3.43..

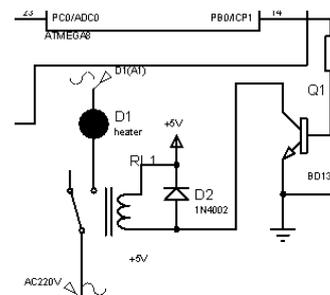


Gambar 3.43. Skematik Tombol *Enter*, *Setting Up* dan *Setting Down*

3.8.5. Skematik *Driver Heater*

Pada skematik yang telah dibuat seperti pada Gambar 3.44. *heater* di simulasikan menggunakan *LED* (D1). Dan pada rangkaian ini terdapat transistor, sebagai saklar. Diode untuk penyearah arus menuju *coil* pada *relay*. Dapat dilihat pada rangkaian skematik, *heater* berada pada posisi kontak *NO* (*Normaly Open*), jadi ketika mendapat tegangan, *heater* tersebut baru bisa aktif.

Di bawah ini adalah skematik dari *driver heater*.



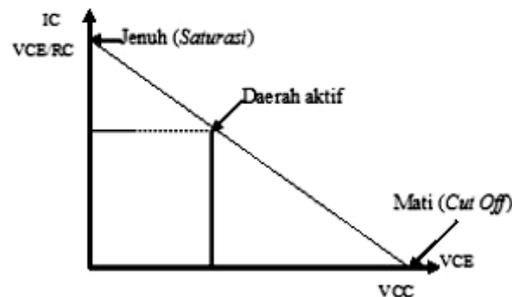
Gambar 3.44. Skematik *Driver Heater*

3.8.6. Skematik *Driver Buzzer*

transistor juga dapat digunakan sebagai sebuah saklar, dimana ketika transistor berada pada dua daerah kerjanya yaitu pada kondisi jenuh (*saturasi*) dan juga kondisi mati (*cut-off*). Transistor bekerja seperti saklar tertutup saat transistor pada kondisi *saturasi* dan transistor bekerja seperti saklar terbuka saat transistor pada kondisi *cut-off*. Titik kerja transistor pada kondisi *saturasi* atau daerah jenuh *buzzer* aktif, yaitu ketika nilai arus yang mengalir pada kolektor ke emitor sama dengan nilai *IC* saturasi atau arus maksimal

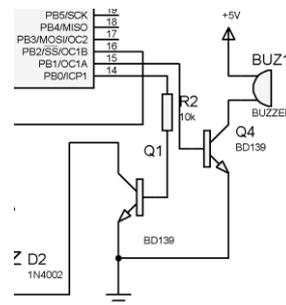
seolah-olah terjadi hubung singkat pada kolektor dengan emitor. Daerah mati transistor atau daerah *cut-off* merupakan daerah kerja dimana kolektor dengan emitor tidak terhubung karena tidak ada arus yang mengalir dari kolektor ke emitor.

Berikut ini merupakan Gambar kurva karakteristik transistor:



Gambar 3.45. Grafik Kurva Karakteristik Transistor

Di bawah ini adalah skematik dari *driver buzzer*.

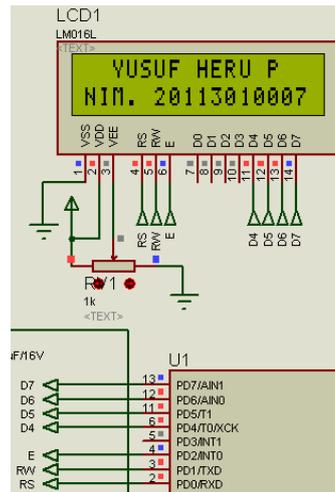


Gambar 3.46. Skematik *Driver Buzzer*

3.8.7. Skematik dan Perangkat Keras *Driver Liquid Crystal Display (LCD)*

Rangkaian *driver LCD* ini digunakan sebagai media penampil data program yang sudah diolah dari tombol-tombol pengontrol

mikrokontroler ditampilkan pada *LCD*. Dalam rangkaian ini menggunakan jalur data 4 bit sehingga D0-D3 tidak terhubung. Untuk gambar skematik dan Perangkat keras dar tampilan *LCD*, dapat dilihat pada Gambar 3.47. dan Gambar 3.48.



Gambar 3.47. Skematik *Driver Liquid Crystal Display (LCD)*

Di bawah ini adalah gambar fisik *driver LCD*.



Gambar 3.48. Perangkat Keras *Liquid Crystal Display (LCD)*

3.9. Program Pendukung

Pemrograman perangkat lunak pendukung ATmega8 dilakukan dengan menulis *source code* program pada aplikasi CVAVR seperti yang sudah dijelaskan sebelumnya. *Source code* program yang sudah disusun atau ditentukan selanjutnya untuk disimpan, kemudian di-*compile* sehingga ber-*ekstensi hex*. Dan untuk memasukan program atau proses *downloading* ke dalam mikrokontroler ATmega8 ini menggunakan program aplikasi *ProgISP*.

3.9.1. Listing Program

Berikut adalah *list* program yang digunakan dan di-*downloading*-kan ke dalam mikrokontroler ATmega8, sebagai pengontrol dan pengendali alat Modifikasi Penghangat Air Susu Ibu Perah (ASIP):

```

unsigned int kali=0, detik=00, menit=5;

unsigned char cdetik[10], cmenit[10];

#include <mega8.h>

#include <stdlib.h>

#include <delay.h>

#asm

    .equ __lcd_port=0x12 ;PORTD

#endasm

```

```
#include <lcd.h>

#define tombol_enter  PINC.2

#define tombol_up     PINC.1

#define tombol_down   PINC.0

#define on            0

#define off           1

bit status_up=0,status_down=0, status_enter=0 ;

{

TCNT2=0;

if(++kali==787)

{

if (--detik==1)

{detik=59;

lcd_clear();

if (--menit==1)

menit=0;

lcd_clear();

PORTB=0x0E;

lcd_gotoxy(0,0);

lcd_putsf(" SELESAI ");

lcd_gotoxy(0,1);

lcd_putsf("TEKAN POWER OFF ");

delay_ms(5000);
```

```
    lcd_clear();  
  
    lcd_gotoxy(0,0);  
  
    lcd_putsf(" TERIMAKASIH");  
  
    lcd_gotoxy(0,1);  
  
    lcd_putsf("  |^_^|");  
  
    delay_ms(500);  
  
    lcd_clear();  
  
    PORTB=0x00;  
  
    delay_ms(0);  
  
    }  
  
    }  
  
kali=0;  
  
itoa(detik,cdetik);  
  
itoa(menit,cmenit);  
  
lcd_gotoxy(0,0);  
  
lcd_puts(cmenit);  
  
lcd_putsf(" MENIT");  
  
lcd_putsf(":");  
  
lcd_puts(cdetik);  
  
lcd_putsf(" DETIK ");  
  
lcd_gotoxy(0,1);  
  
lcd_putsf(" SILAKAN TUNGGU");  
  
    }
```

```
}  
  
void main(void)  
{  
PORTB=0x04;  
DDRB=0xFF;  
PORTC=0x07;  
DDRC=0x08;  
  
    ASSR=0x00;  
  
    TCCR2=0x02;  
  
    TCNT2=0x00;  
  
    OCR2=0x96;  
  
    lcd_init(16);  
  
    lcd_clear();  
  
    lcd_gotoxy(0,0);  
  
    lcd_putsf("PENGHANGAT ASIP");  
  
    lcd_gotoxy(0,1);  
  
    lcd_putsf("BERBASIS ATMEGA8");  
  
    delay_ms(300);  
  
    lcd_clear();  
  
    lcd_gotoxy(0,0);  
  
    lcd_putsf(" YUSUF HERU P");  
  
    lcd_gotoxy(0,1);  
  
    lcd_putsf("NIM. 20113010007");
```

```
delay_ms(300);

lcd_clear();

lcd_gotoxy(0,0);

lcd_putsf("  SILAKAN");

lcd_gotoxy(0,1);

lcd_putsf(" SETTING WAKTU");

delay_ms(200);

lcd_clear();

lcd_gotoxy(0,0);

lcd_putsf(" ASIP DINGIN");

lcd_gotoxy(0,1);

lcd_putsf(" SETTING 5MENIT");

delay_ms(300);

lcd_clear();

lcd_gotoxy(0,0);

lcd_putsf(" ASIP BEKU");

lcd_gotoxy(0,1);

lcd_putsf("SETTING 10MENIT");

delay_ms(300);

lcd_clear();

#asm("sei")

while (1)

{
```

```

{ if (tombol_up==off) status_up=on;

if ((tombol_up==on)&(status_up==on))

    {menit++;

    if (menit>10){menit=5;detik=00; }

    status_up=off;}

if (tombol_down==off) status_down=on;

if ((tombol_down==on)&(status_down==on))

    {menit--;

    if (menit<5){menit=10; detik=00;}

    status_down=off;}

    }

itoa(detik,cdetik);

itoa(menit,cmenit);

lcd_gotoxy(0,0);

lcd_puts(cmenit);

lcd_putsf(" MENIT");

lcd_putsf(":");

lcd_puts(cdetik);

lcd_putsf(" DETIK ");

lcd_gotoxy(0,1);

lcd_putsf("LALU TEKAN ENTER");

{ if (tombol_enter==off) status_enter=on;

    if ((tombol_enter==on)&(status_enter==on))

```

```
{  
  lcd_clear();  
  PORTB=0x05;  
  while(1)  
  {  
    TIMSK=0x80;  
  }  
  status_enter=off; }  
}  
};  
}
```