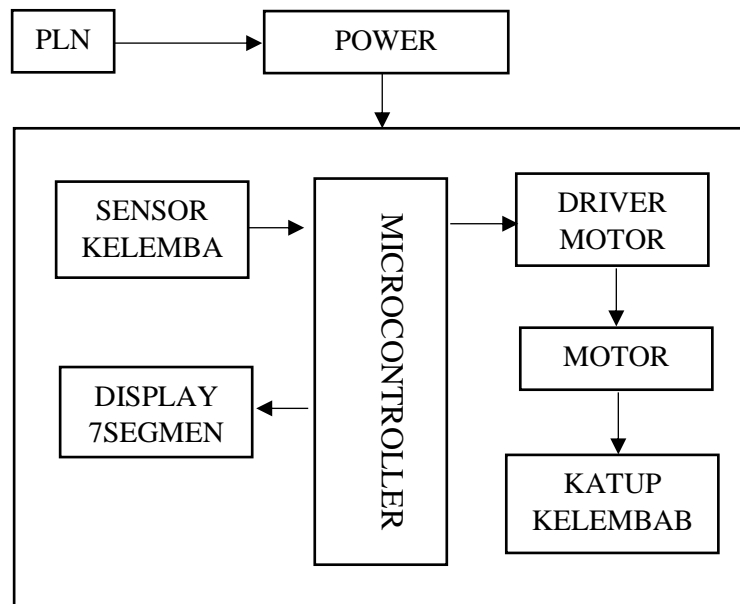


BAB III METODOLOGI PENELITIAN

3.1 Blok Diagram *Baby Incubator*

Dalam perancangan alat *baby incubator* penulis membuat blok diagram alat sebagai berikut:



Gambar 3.1 Blok diagram alat *baby incubator*

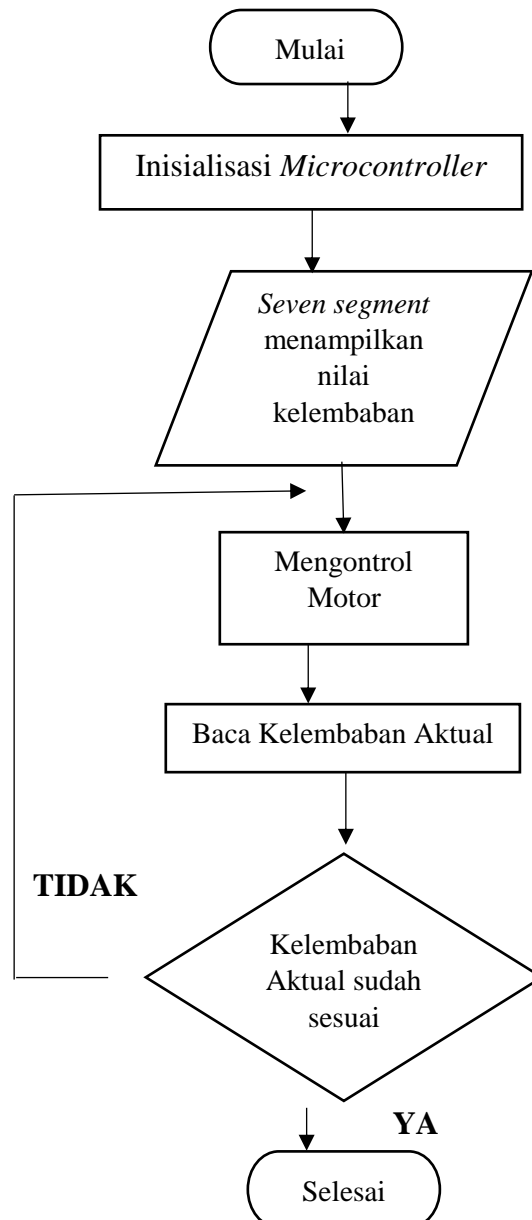
Tegangan dari PLN 220 V digunakan untuk *input* tegangan trafo *stepdown* yang kemudian oleh rangkaian *power supply* dirubah menjadi tegangan +12V DC dan +5V DC yang digunakan untuk memberi *input* tegangan ke blok rangkaian lainnya.

Microcontroller berfungsi untuk mengontrol semua rangkaian, antara lain: sensor kelembaban, *driver* motor dan *display seven segment*. Sensor kelembaban berfungsi untuk mensensor kelembaban udara di dalam alat *baby incubator*. Data dari sensor yang berupa nilai analog masuk pada blok ADC

pada *microcontroller*. Lalu *microcontroller* mengolah semua data untuk mengatur kerja keseluruhan modul *baby incubator*.

3.2 Diagram Alir *Baby Incubator*

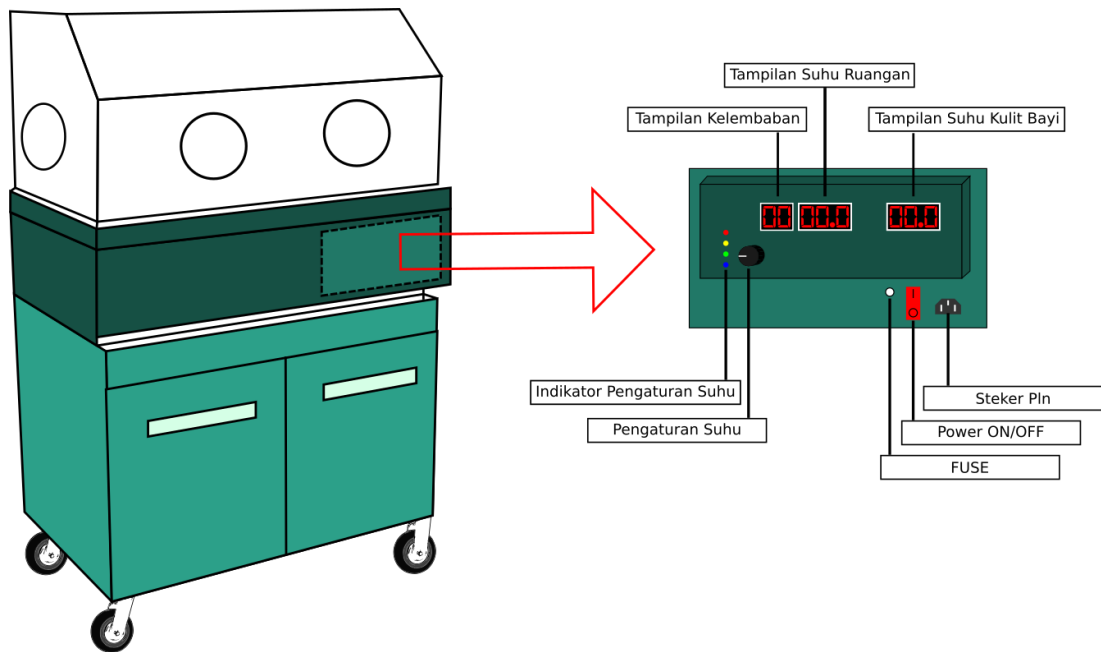
Dalam pembuatan alat *baby incubator* penulis membuat diagram alir alat *baby incubator* sebagai berikut:



Gambar 3.2 Diagram Alir *baby incubator*

Seven segment menampilkan nilai kelembaban di dalam *baby incubator*. Motor berfungsi untuk menggerakkan katup kelembaban, apabila kelembaban kurang dari *setting* maka katup akan membuka sedangkan ketika nilai kelembaban melebihi *setting* maka katup akan menutup.

3.3 Desain Alat

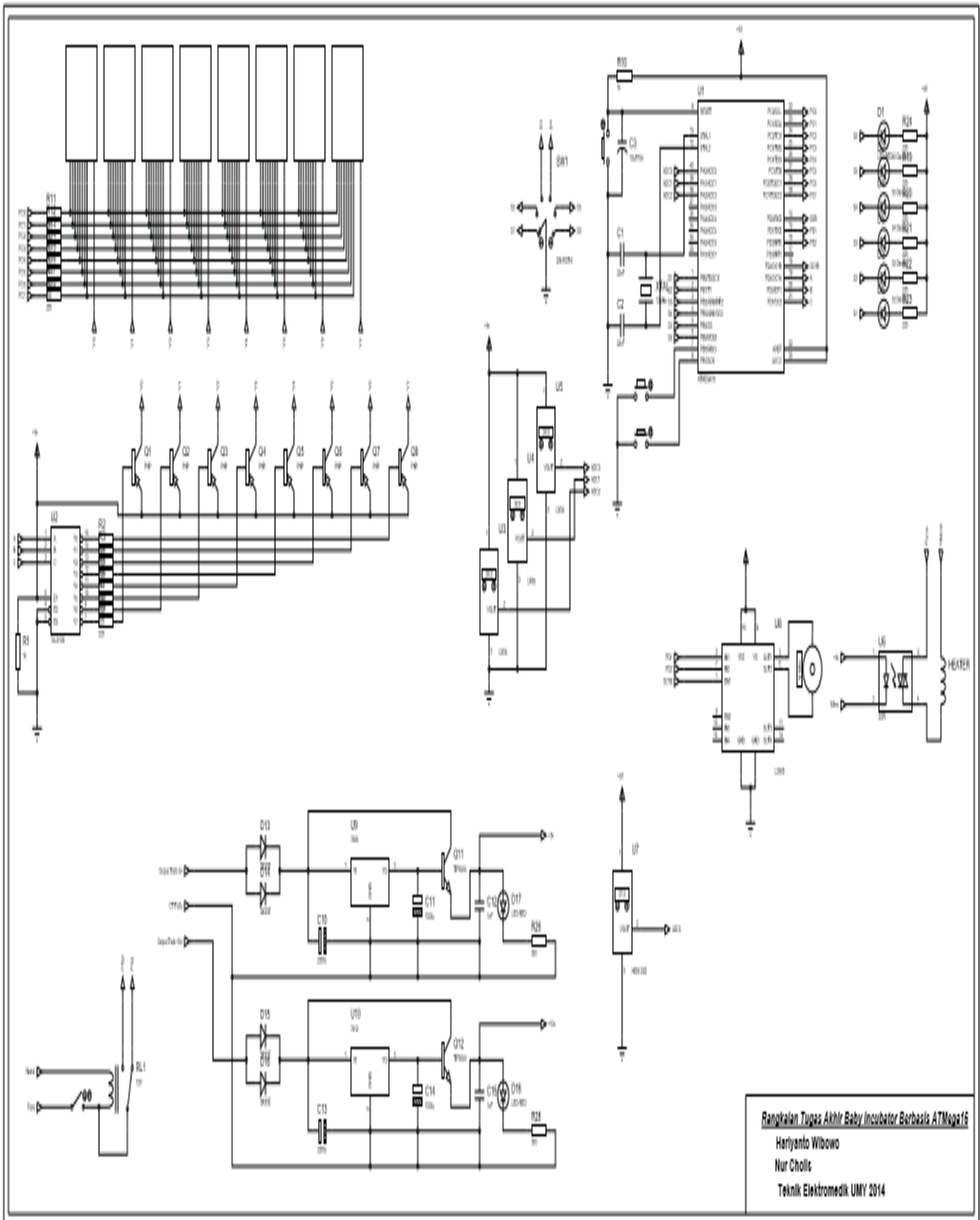


Gambar 3.3 Desain Alat

3.4 Perancangan Perangkat Keras

Perancangan perangkat keras dilakukan sebagai tata cara untuk menentukan program yang akan dimasukkan ke dalam *microcontroller* yang berfungsi sebagai pengontrol perangkat keras.

Adapun perangkat keras yang dibutuhkan pada pembuatan alat *Baby Incubator* ini adalah terdiri dari: Rangkaian catu daya, rangkaian *driver* motor DC, rangkaian *Microcontroller* dan rangkaian *driver seven segment*. Perancangan perangkat keras ditunjukkan pada Gambar 3.4 di bawah ini.

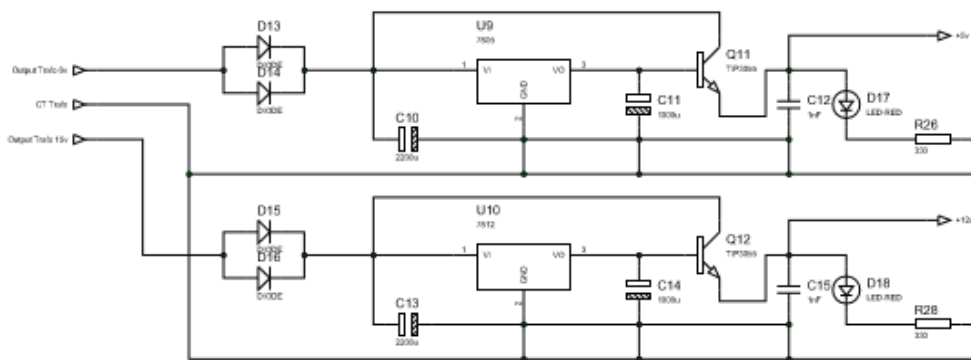


Rangkaian Tugas Akhir Baby Incubator Berbasis ATmega16
 Haryanto Wibowo
 Nur Cholic
 Teknik Elektromedik UMY 2014

Gambar 3.4 Rangkaian Keseluruhan

3.4.1 Rangkaian *Power Supply*

Catu daya berfungsi untuk memberikan *input* tegangan, khususnya ke IC *microcontroller ATmega16* dan *relay*. Catu daya yang digunakan untuk memberi tegangan pada *microcontroller* adalah 5 Volt DC. Untuk menurunkan tegangan trafo dari 9 V menjadi 5 V, maka digunakan IC *voltage regulator LM7805*. Pada rangkaian catu daya, dioda digunakan untuk menyearahkan gelombang penuh dari AC ke DC. Sedangkan kapasitor 4700 μ F berfungsi sebagai *filter* tegangan yang dihasilkan oleh dioda. Berikut adalah gambar perangkat keras rangkaian catu daya:



Gambar 3.5. *Schematic Power Supply*



Gambar 3.6. Perangkat Keras *Power Supply*

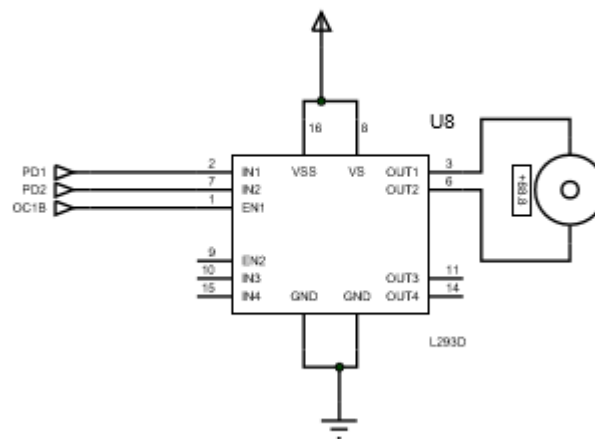
Prinsip kerja dari rangkaian catu daya (*power supply*) di atas yaitu tegangan jala-jala 220 *volt* dari listrik PLN diturunkan oleh transformator penurun tegangan (*step down*) yang menerapkan perbandingan lilitan. Dimana perbandingan lilitan dari suatu transformator akan mempengaruhi perbandingan tegangan yang dihasilkan. Tegangan yang dihasilkan oleh trafo masih berbentuk gelombang AC dan harus disearahkan dengan menggunakan penyearah yang telah dirancang untuk bisa meloloskan kedua siklus gelombang AC menjadi satu arah (*DC*).

Gelombang AC yang telah diubah menjadi gelombang DC keluaran dari dioda masih memiliki amplitudo tegangan yang tidak rata. Hal ini dikarenakan dioda hanya menghilangkan siklus negatif dan menjadikannya siklus positif tetapi tidak merubah bentuk gelombang sama sekali dimana masih memiliki lembah dan bukit. Untuk itu dimanfaatkan kapasitor yang mempunyai kapasitansi yang cukup besar untuk membuat rata gelombang tersebut. Hal ini dikarenakan lamanya proses pelepasan muatan oleh kapasitor sehingga seolah-olah amplitudo dari gelombang tersebut menjadi rata.

Tingkat kerataan dari gelombang yang dihasilkan masih dipengaruhi oleh impedensi beban yang nanti akan dihubungkan dengan rangkaian *power supply* tersebut. Semakin kecil impedensi beban maka akan menjadikan proses pelepasan muatan pada kapasitor akan semakin cepat, sehingga dengan begitu maka bisa dipastikan gelombang yang semula rata akan berubah kembali menjadi memiliki riak akibat proses pelepasan muatan yang begitu cepat. Kemudian tegangan tersebut diubah menjadi +5 *VDC* ketika melalui regulator LM7805.

3.4.2 Rangkaian *Driver Motor*

Rangkaian *driver* motor berfungsi untuk *driver* motor DC yang digunakan sebagai penggerak dari katup kelembaban. Rangkaian *driver* motor menggunakan IC L293D. Gambar rangkaian *driver* motor dapat dilihat pada gambar 3.7



Gambar 3.7 Schematic Driver Motor

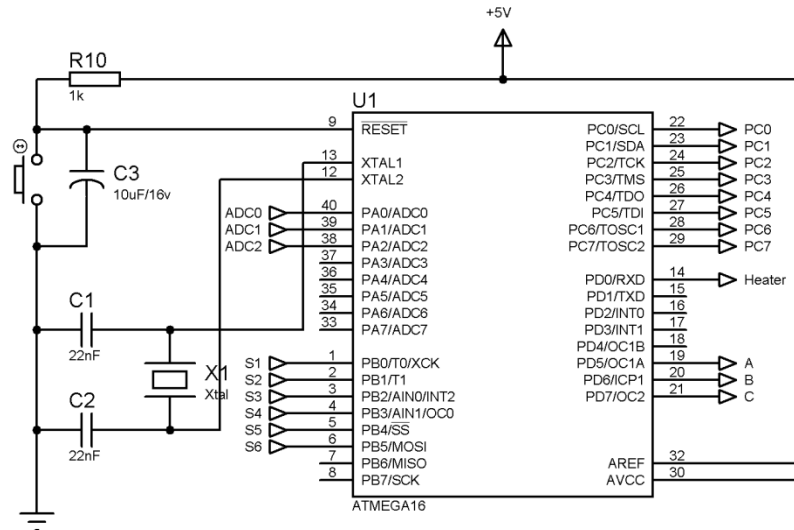


Gambar 3.8 Perangkat keras *Driver* Motor

3.4.3 Rangkaian Minimum Sistem (*Microcontroller*)

Rangkaian minimum sistem *microcontroller* ini adalah rangkaian utama yang digunakan untuk memfungsikan *microcontroller*, dimana merupakan pengontrol utama dalam *baby incubator*. Rangkaian ini difungsikan juga sebagai salah satu media untuk melakukan *programming*. Dalam rangkaian ini, menggunakan kristal

12MHz sebagai pembangkit sinyal *eksternal*. Skema rangkaian sistem minimum *microcontroller* dapat dilihat pada gambar 3.9 dibawah ini:



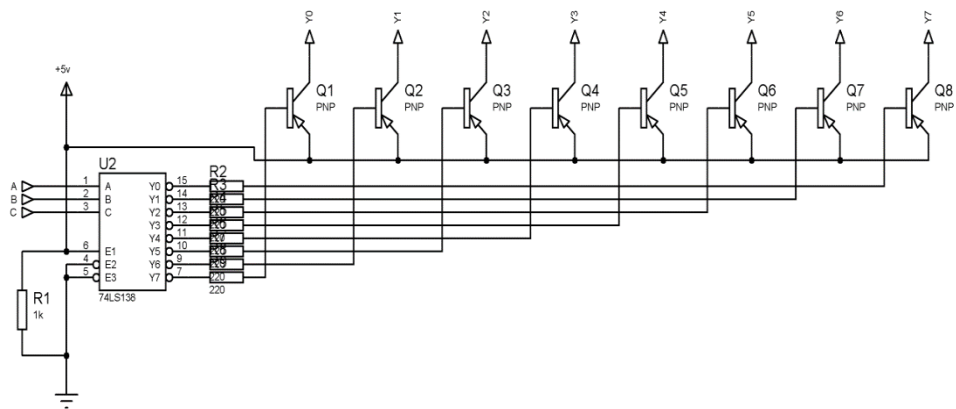
Gambar 3.9. Skema Rangkaian Minimum Sistem *ATMega16*



Gambar 3.10. Perangkat Keras Minimum Sistem

3.4.4 Rangkaian *Driver Seven Segment*

Rangkaian *driver seven segment* ini digunakan sebagai media untuk mengatur tampilan *seven segment*. Gambar rangkaian *driver seven segment* diperlihatkan pada gambar di bawah ini.



Gambar 3.11. Schematic Rangkaian Driver Seven Segment

Tabel 3.1. Tabel Kebenaran 74LS138

Selector			Enable			Output							
C	B	A1	G1	/G2A	/G2B	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
0	0	0	1	0	0	0	1	1	1	1	1	1	1
0	0	1	1	0	0	1	0	1	1	1	1	1	1
0	1	0	1	0	0	1	1	0	1	1	1	1	1
0	1	1	1	0	0	1	1	1	0	1	1	1	1
1	0	0	1	0	0	1	1	1	1	0	1	1	1
1	0	1	1	0	0	1	1	1	1	1	0	1	1
1	1	0	1	0	0	1	1	1	1	1	1	0	1
1	1	1	1	0	0	1	1	1	1	1	1	1	0

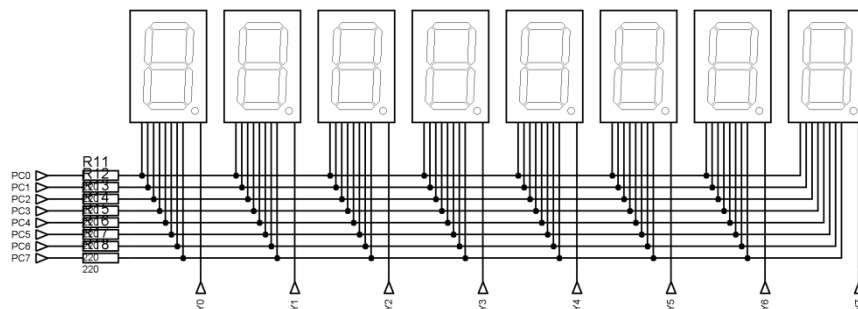
Pada tabel kebenaran tersebut tampak bahwa *seven segment* yang hidup tergantung pada *output* dari *decoder 74LS138* yang sedang mengeluarkan logika *low*, sehingga dari 8 buah *seven segment* tersebut, selalu hanya satu *seven segment* yang akan menyala. Agar ke-delapan *seven segment* tampak menyala secara bersamaan maka ke-delapan *seven segment* tersebut harus dinyalakan secara bergantian dengan waktu tunda tertentu.

Pada gambar tersebut *seven segment common anoda* dikendalikan dengan menggunakan transistor PNP melalui *decoder 74LS138*, apabila ada logika *low*

pada basis transistor, maka *seven segment* akan menyala dan sebaliknya apabila diberi logika *high* akan padam.

3.4.5 Rangkaian *Seven Segment*

Rangkaian *seven segment* ini digunakan sebagai media untuk menampilkan besaran suhu yang terukur. Gambar rangkaian *seven segment* diperlihatkan pada Gambar 3.12 dibawah ini.



Gambar 3.12. Schematic Rangkaian *Seven Segment*

Tabel 3.2. Tabel Data *Seven Segment*

Port C.6	Port C.5	Port C.4	Port C.3	Port C.2	Port C.1	Port C.0	Tampilan
G	F	E	D	C	B	A	
1	0	0	0	0	0	0	0
1	1	1	1	0	0	1	1
0	1	0	0	1	0	0	2
0	1	1	0	0	0	0	3
							:
							:
0	0	0	1	0	0	0	A
Port C.6	Port C.5	Port C.4	Port C.3	Port C.2	Port C.1	Port C.0	Tampilan
G	F	E	D	C	B	A	
0	0	0	0	0	1	1	B
1	0	0	0	1	1	0	C
							:

Pada tabel tersebut tampak bahwa untuk menghidupkan sebuah *segment*, harus diberikan data logika *low* dan sebaliknya untuk mematikan *segment*, harus diberikan data logika *high*.

3.5 Perancangan Perangkat Lunak

Perangkat lunak merupakan suatu program yang berhubungan dengan perangkat keras. Perangkat lunak sifatnya pun berbeda dengan *hardware* atau perangkat keras, jika perangkat keras adalah komponen yang nyata yang dapat dilihat dan disentuh secara langsung oleh manusia, maka *software* atau perangkat lunak tidak dapat disentuh dan dilihat secara fisik. *Software* memang tidak tampak secara fisik dan tidak berwujud benda namun dapat dioperasikan. Perangkat lunak bersifat tidak terpisah dengan *microcontroller*.

Perangkat keras yang sudah tertata dengan benar, tidak akan berfungsi dengan baik jika terdapat kesalahan pada perangkat lunak. Oleh karena itu, dibutuhkan ketelitian dan ketepatan dalam penyusunan perangkat lunak yang akan diprogram ke *microcontroller*.

3.5.1. Program Pendukung

Pemrograman perangkat lunak pendukung *ATMega16* dilakukan dengan menulis *source code* program pada aplikasi CV-AVR. *Source code* program yang sudah ditulis lalu disimpan dan di-*compile* sehingga berekstensi *hex*. Kemudian di *download* ke dalam *microcontroller* menggunakan program Progisp.

3.5.2. Listing Program

Berikut ini adalah listing program modul

```
void humidity() //fungsi pembacaan sensor RH
{
  datahum=read_adc(3); //membaca nilai dari adc 3
  hum=(float)datahum*5/1023; //merubah nilai adc ke RH
  sum=(hum*33)-15;

  pulhum=(unsigned int) sum/10; //mengambil nilai puluhan
  sathum=(unsigned int) sum%10; //mengambil nilai satuan

  ubah=pulhum;ubah_ke_format7segment();pulhum=ubah;
  ubah=sathum;ubah_ke_format7segment();sathum=ubah;
  tampil_7segment();
}
```

3.6 Cara Analisis Perhitungan Statistika

3.6.1 Rata-rata

Merupakan nilai atau hasil pembagian dari jumlah data yang diambil atau diukur dengan banyaknya pengambilan data atau banyaknya pengukuran.

Dirumuskan sebagai berikut:

$$\text{rata - rata } (\bar{X}) = \frac{\sum X_n}{n} \quad (3-1)$$

Keterangan:

X_n = jumlah nilai data

n = banyak data (1,2,3,4,5,.....n)

3.6.2 Simpangan

Simpangan adalah selisih dari rata-rata nilai harga yang dikehendaki dengan nilai yang diukur. Berikut rumus dari simpangan :

$$\text{Simpangan} = \bar{X}_{\text{Modul}} - \bar{X}_{\text{Alat ukur}} \quad (3-2)$$

3.6.3 Presentase Error

Persentase *Error* adalah nilai persen dari simpangan (*error*) terhadap nilai yang dikehendaki. Dirumuskan sebagai berikut:

$$\% \text{ error} = \frac{\bar{X}_{Modul} - \bar{X}_{Alat Ukur}}{\bar{X}_{Alat Ukur}} \times 100\% \quad (3-3)$$

