

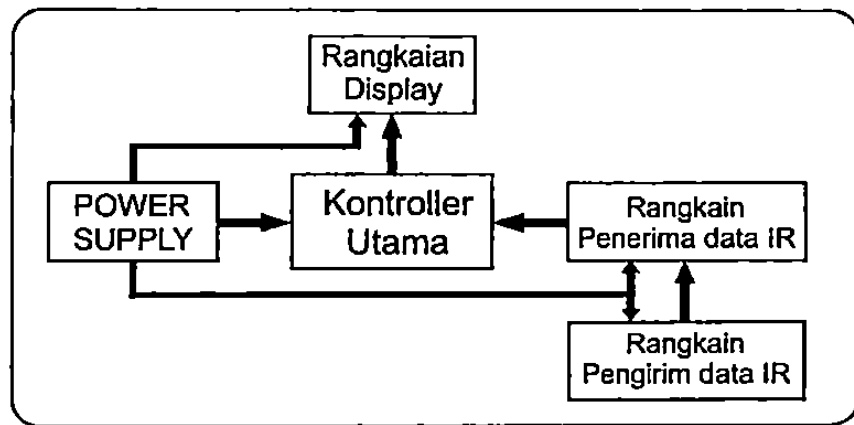
## BAB 3

### PERANCANGAN, PEMBUATAN DAN PENGUJIAN

#### 3.1 Rancangan 1

Rancangan 1 ini berupa perancangan awal perangkat keras dari rangkaian alat pewaktu iqomah berdasarkan konsep yang telah direncanakan. Rangkaian ini terdiri dari beberapa blok rangkaian, yaitu:

- Rangkaian kontroler utama yang di dalamnya berisikan mikrokontroler ATmega16, sensor suhu LM35, *real time clock* DS1307 dan komponen-komponen pelengkap lainnya.
- Rangkaian *power supply* yang berfungsi sebagai sumber daya dari rangkaian utama maupun rangkaian yang lain.
- Rangkaian *Multiplexing 7 Segment* sebagai penampil data.
- Rangkaian pengirim dan penerima data infra merah dari *remote control (IR Transmitter and Receiver)*.



Gambar 3.1 Hubungan antar blok rangkaian

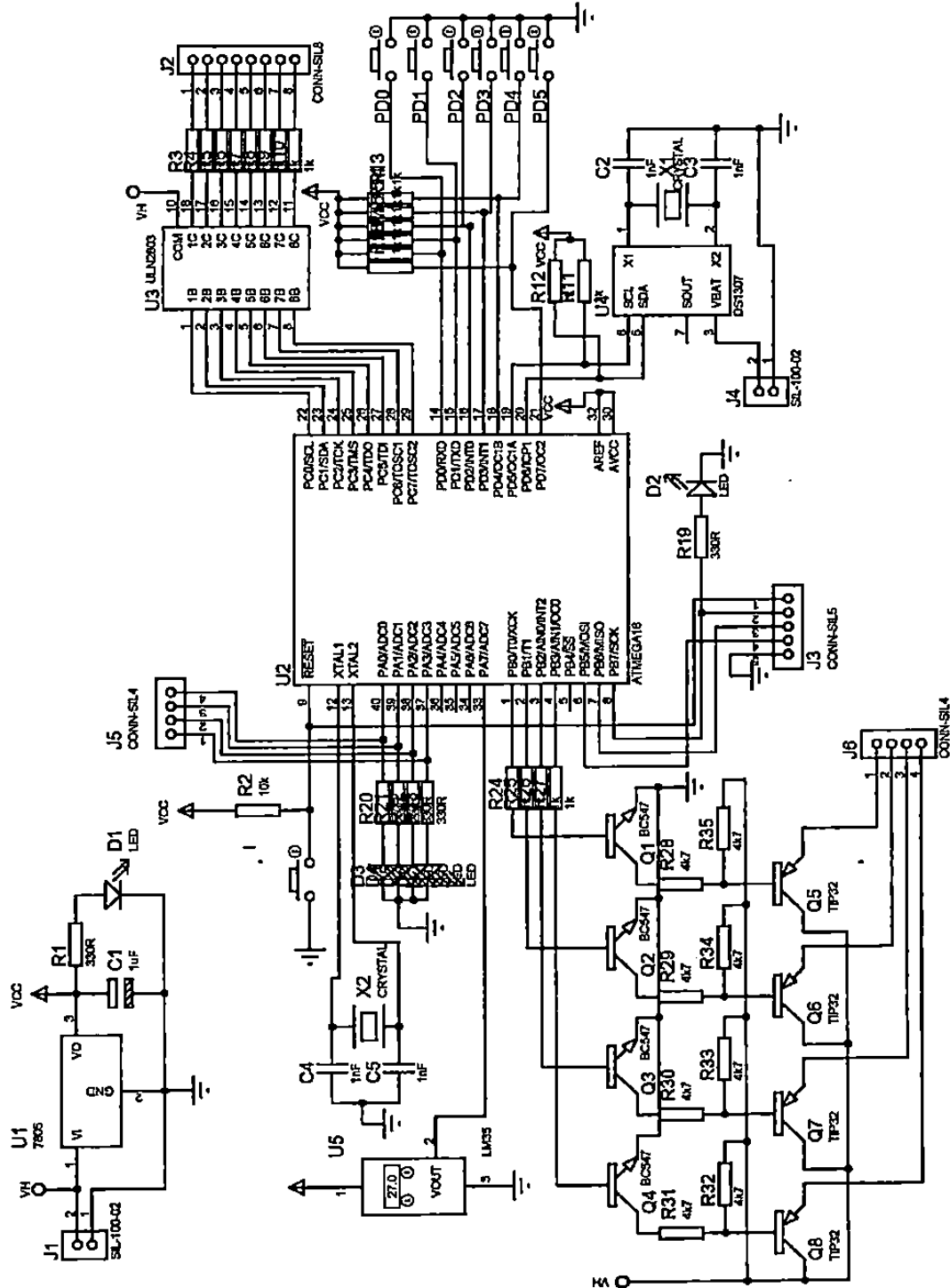
### 3.1.1 Rangkaian Kontroler Utama

Konsep awal rangkaian kontroler utama ini dirancang sedemikian rupa agar dapat memiliki kemampuan untuk menghitung waktu mundur yang akan dijadikan pewaktu iqomah menggunakan fasilitas *timer* internal yang ada di dalam mikrokontroler (menggunakan *timer0*), menerima data analog yang dikirimkan oleh sensor suhu LM35 dan kemudian mengolahnya dalam bentuk digital (PORT ADC 7), berkomunikasi dengan IC RTC DS1307 menggunakan komunikasi I2C, menerima dan mengolah data yang dikirimkan oleh sensor infra merah TSOP 1738 menggunakan metode interupsi eksternal (menggunakan PORT INT0) serta menampilkan keseluruhan data yang telah diolah kedalam tampilan 7 segment yang berukuran 4 inch.

Langkah-langkah perancangan yang dilakukan adalah sebagai berikut:

- Perancangan diagram skematik dengan *software* PROTEUS ISIS.

### 3.1.1.1 Perancangan Rangkaian Kontroler



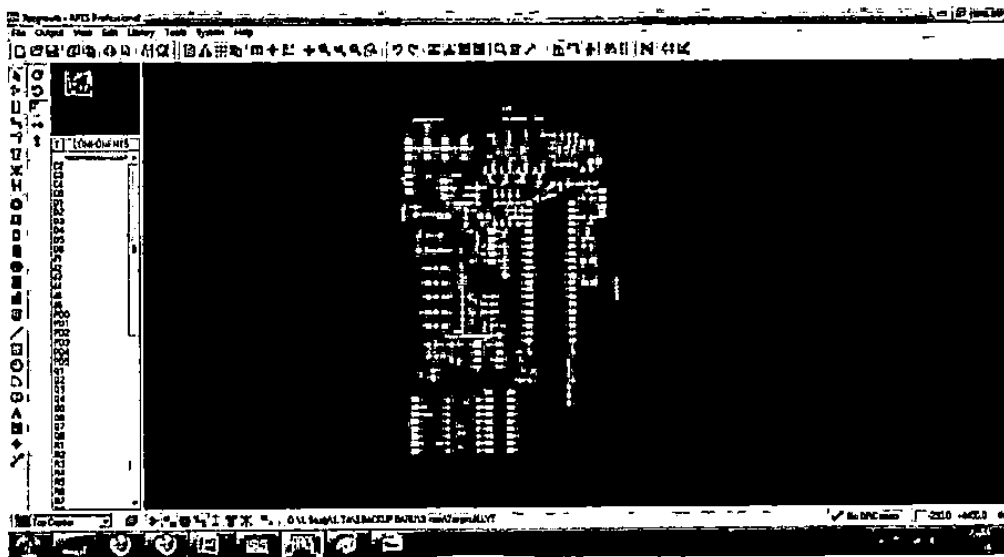
Rangkaian kontrol utama ini merupakan pusat dari segala aktifitas kontrol yang terdapat pada alat pewaktu iqomah. Rangkaian kontrol ini berisikan komponen-komponen sebagai berikut:

**Tabel 3.1** Komponen Rangkaian Kontroler Utama

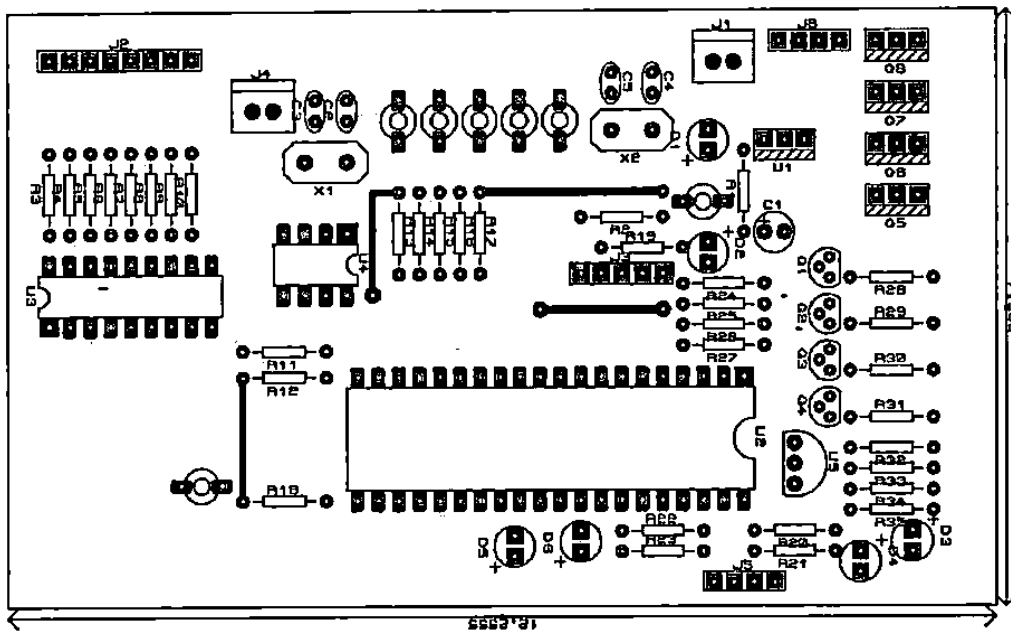
No	Jumlah	Kode Komponen	Nilai
<b>Resistor</b>			
1	6	R1, R19-R23	330R
2	1	R2	10k
3	20	R3-R18, R24-R27	1k
4	8	R28-R35	4k7
<b>Capasitor</b>			
5	1	C1	220uF
6	4	C2-C5	22nF
<b>Integrated Circuits</b>			
7	1	U1	7805
8	1	U2	ATMEGA16
9	1	U3	ULN2803
10	1	U4	DS1307
11	1	U5	LM35
<b>Transistor</b>			

12	4	Q1-Q4	BC547
13	4	Q5-Q8	TIP32
14	6	D1-D6	LED
<b>Komponen Lain</b>			
15	2	J1, J4	SIL-100-02
16	1	J2	CONN-SIL8
17	1	J3	CONN-SIL5
18	2	J5, J6	CONN-SIL4
19	6	PD0-PD5	SAKLAR
20	1	X1	11,059 MHz
21	1	X2	32,768 MHz

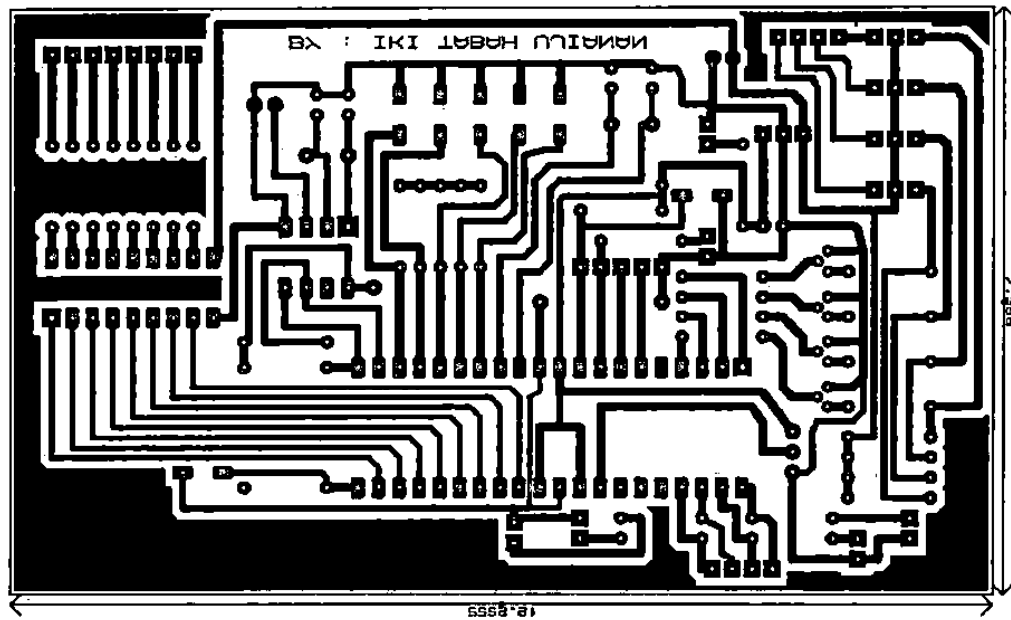
### 3.1.1.2 Layout PCB Rangkaian Kontroler utama

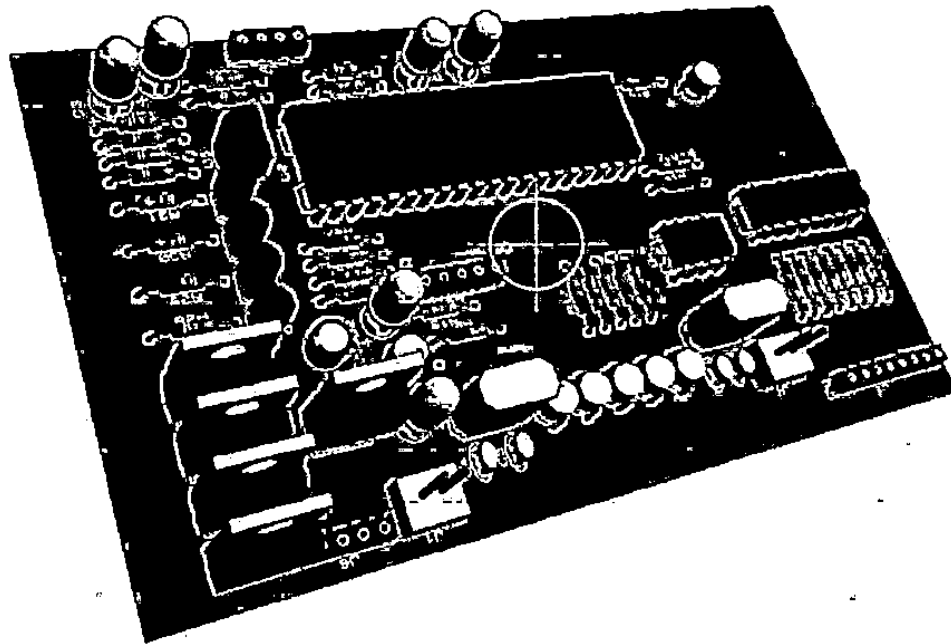


Gambar 3.3 Layout PCB pada PROTEUS ARES



Gambar 3.4 Hasil layout PCB tampilan bawah



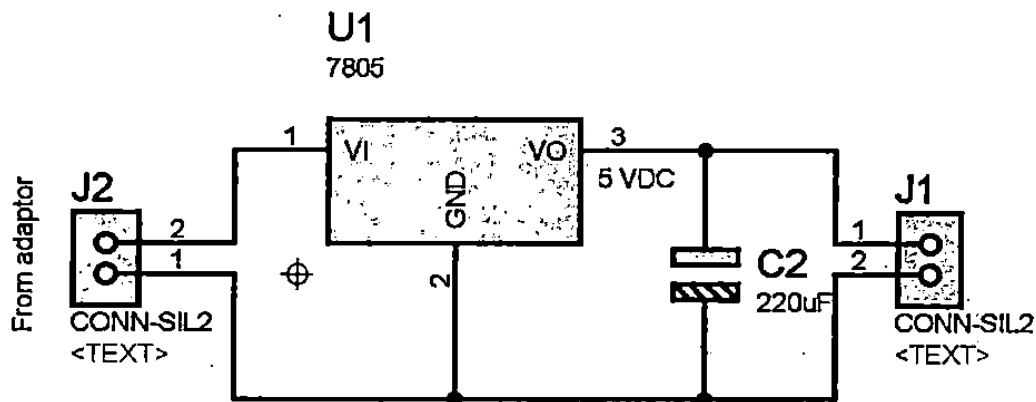


**Gambar 3.6** Tata letak komponen dalam 3 dimensi

### **3.1.2 Rangkaian *Power Supply***

Rangkaian power supply ini berasal dari adaptor (*transformator step down*) yang memiliki keluaran 18 volt, lalu keluaran tersebut dihubungkan ke IC 7805 dengan tujuan mendapatkan tegangan 5 volt. Output 18 volt nantinya akan digunakan sebagai penguat pada transistor untuk dapat menyalakan tampilan 7 segment. Sedangkan

5 volt digunakan sebagai input pada rangkaian utama dan input pada

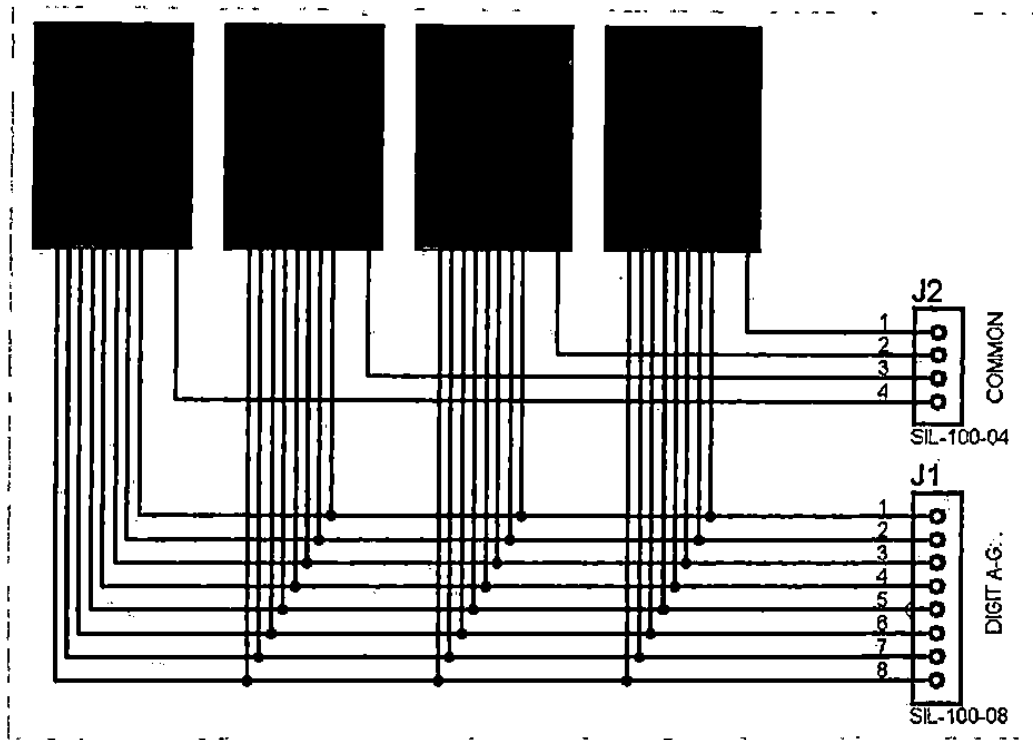


**Gambar 3.7** Rangkaian *Power Supply*

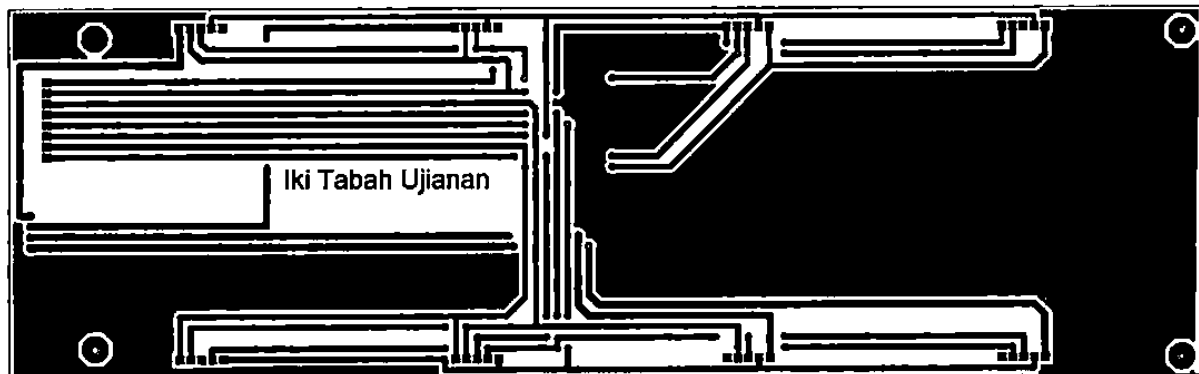
### 3.1.3 Rangkaian *Multiplexing* 7 Segment

Rangkaian *Multiplexing* 7 Segment ini berfungsi sebagai penampil data yang telah diolah rangkaian kontroler utama. Metode yang digunakan adalah *multiplexing* yaitu seluruh port digit pada tiap-tiap 7 segment tersebut dihubungkan paralel. Sedangkan untuk *port common* tidak dirangkai paralel, melainkan berdiri sendiri-sendiri. Rangkaian *multiplexing* ini akan memungkinkan kita menyalakan segment satu persatu namun dalam tempo waktu yang sangat cepat sehingga nantinya-kita dapat melihatnya seolah-olah keempat segment tersebut menyala secara bersamaan

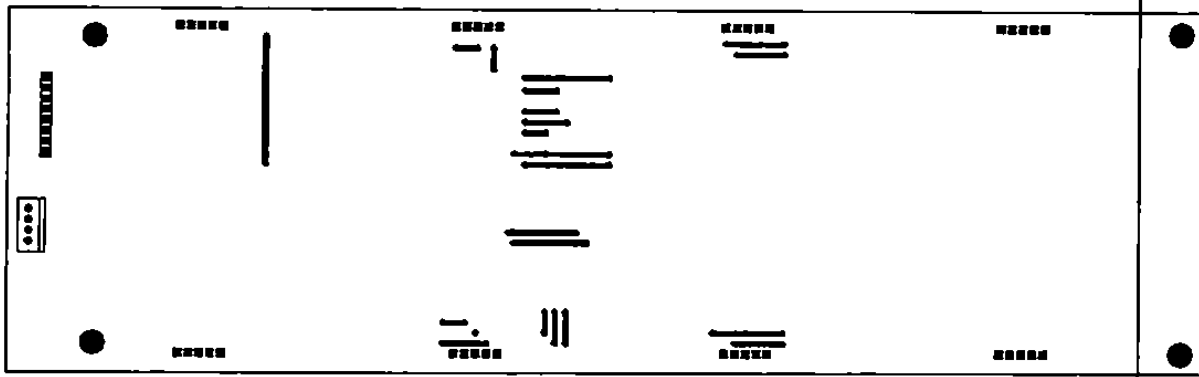




Gambar 3.8 Skema rangkaian *Multiplexing 7 Segment*



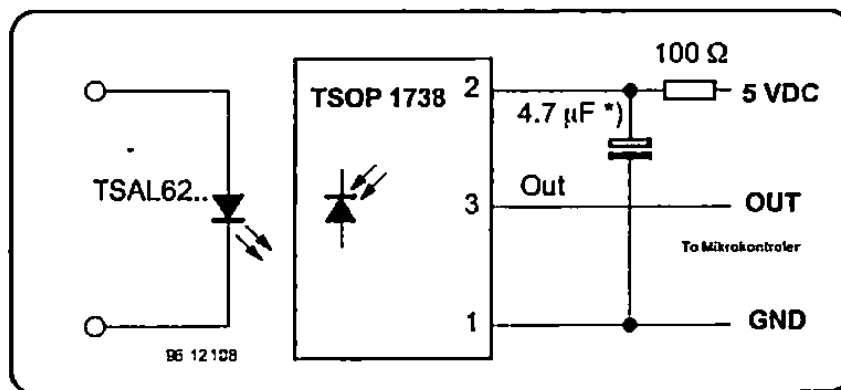
Gambar 3.9 Layout rangkaian *Multiplexing 7 Segment* bagian bawah



Gambar 3.10 Layout rangkaian *Multiplexing 7 Segment* bagian atas

### 3.1.4 Rangkaian Pengirim dan Penerima Data *Remote*

Rangkaian pengirim data *remote* menggunakan *remote control* tipe Sony. *Remote control* tipe ini dipilih karena banyak ditemukan di pasaran dan harganya relatif murah. Rangkaian penerima sinyal infra merah adalah sebagai berikut.



Gambar 3.11 Rangkaian Penerima Infra Merah

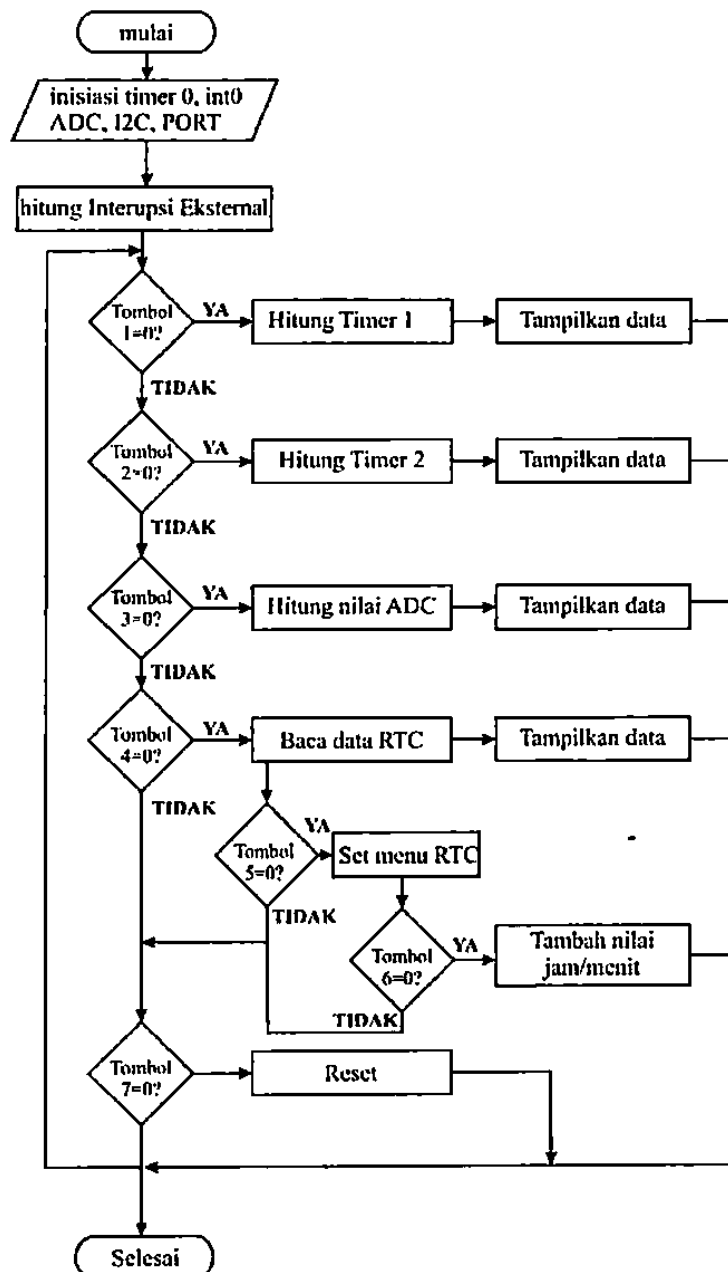
Saat tombol pada *remote control* ditekan, LED akan berkedip pada frekuensi  $\pm 38,5$

KHz. Frekuensi ini adalah frekuensi dari data pembawa (*carrier*) dan frekuensi dari



CodeVisionAVR dalam bahasa C, dan kemudian dimasukkan ke mikrokontroler dengan alat K-125R yang diproduksi oleh Innovative Electronics.

### 3.2.1 Perancangan *Software* Kontroler utama



Gambar 3.13 Diagram Alir Program Kontroler Utama

Perancangan *software* ini digunakan untuk mengolah data *timer* untuk *counter down* pewaktu iqomah, menghitung nilai ADC yang dikirimkan oleh IC LM35, berkomunikasi dengan DS1307 agar dapat mengambil data jam, mengolah data infra merah yang dikirimkan oleh TSOP 1738 yang kemudian dapat digunakan untuk memanggil program serta dapat menampilkan data yang diolah tersebut dalam rangkaian 7 segment yang telah dibuat.

Gambar 3.12 menunjukkan bahwa *flowchart* program dimulai dengan inisiasi timer0, interupsi eksternal, ADC, jalur komunikasi I2C, dan port yang digunakan sebagai input dan output. Timer0 dan interupsi eksternal selalu berjalan di awal program. Timer0 digunakan untuk menghitung lamanya waktu selama 10ms setiap detak yang dihasilkan. Apabila detak yang dihasilkan sudah seratus kali, maka detik akan berkurang satu dan seterusnya sampai nilai detik menjadi nol. Nilai detik yang sudah mencapai 0 akan mengurangi menit satu dari nilai menit yang sudah diatur di awal dan seterusnya sampai dengan menit menjadi 0.

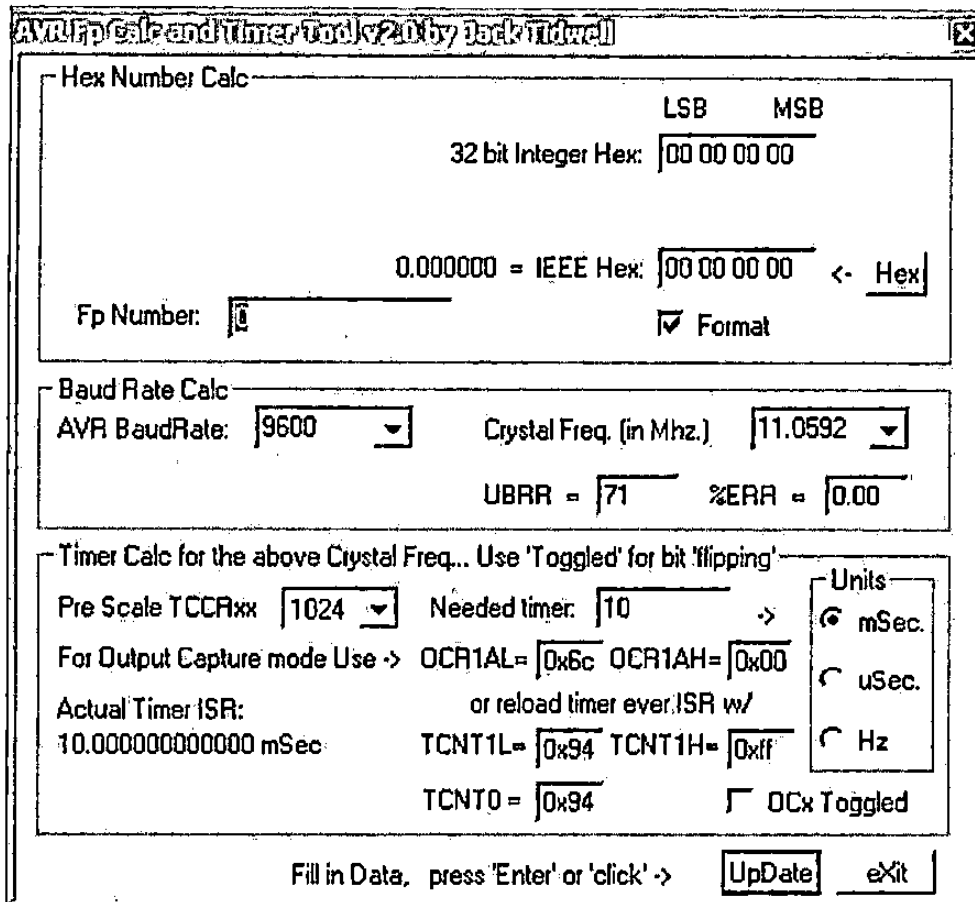
Adapun program untuk menentukan banyaknya pilihan menit dan detik yang

```

        if(PIND.0==1)
            { men=10; det=0; exit=0;}
        }
    if(PIND.1==1)
        {delay_ms(100);
        if(PIND.1==1)
            {men=10; det=0; exit=0; }
        }
    }
}

```

Nilai TCNT0=0x94 adalah suatu nilai yang memerintahkan *timer* tersebut untuk dapat menghitung waktu 10ms dalam setiap detaknya. Nilai inisialisasi timer0 sebesar 0x94 ditentukan dengan menggunakan bantuan *software* AVRcalc. *Software* AVRcalc adalah *software* untuk menentukan nilai *timer* pada mikrokontroler AVR berdasarkan unit milidetik, mikrodetik dan frekuensi

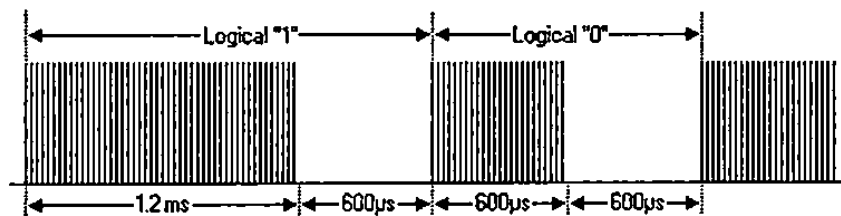


**Gambar 3.13** setting pada AVR Calc

Gambar di atas adalah *screenshot* AVRcalc untuk menghitung nilai *timer* yang dibutuhkan sebesar 10msec, Pre Scale sebesar 1024 dan menggunakan kristal 11.0592Mhz, dengan memasukkan data ini maka akan menghasilkan nilai TCNT0 sebesar 0x94. Hasil interrupt timer0 akan menambah nilai cacah dengan nilai satu setiap 10 milidetik, setiap nilai cacah mencapai 100 maka variabel *count* akan direset ke 0 karena saat inilah waktu satu detik tercapai.

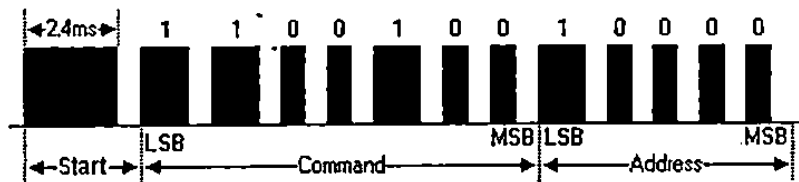
Interupsi eksternal yang digunakan adalah PORTD.2 yang terdapat di dalamnya fasilitas untuk interupsi eksternal 0 (INT0). Input data pada port int0 akan

memicu mikrokontroler untuk menghitung banyaknya interupsi yang masuk. *Remote control* Sony memiliki karakteristik mengirimkan data dengan format *5 bit address* dan *7 bit command*. Untuk modulasi lebar pulsa saat “high” atau “low” adalah:



**Gambar 3.14** Modulasi lebar pulsa saat “high” atau “low

Dari gambar di atas, terlihat bahwa yang membedakan logika high dan low adalah lebar pulsa. Lebar pulsa *low* adalah 600 us dan pulsa *high* bernilai lebih dari itu (1,2 ms). Dapat dilihat protokol *Remote control* Sony berikut ini.



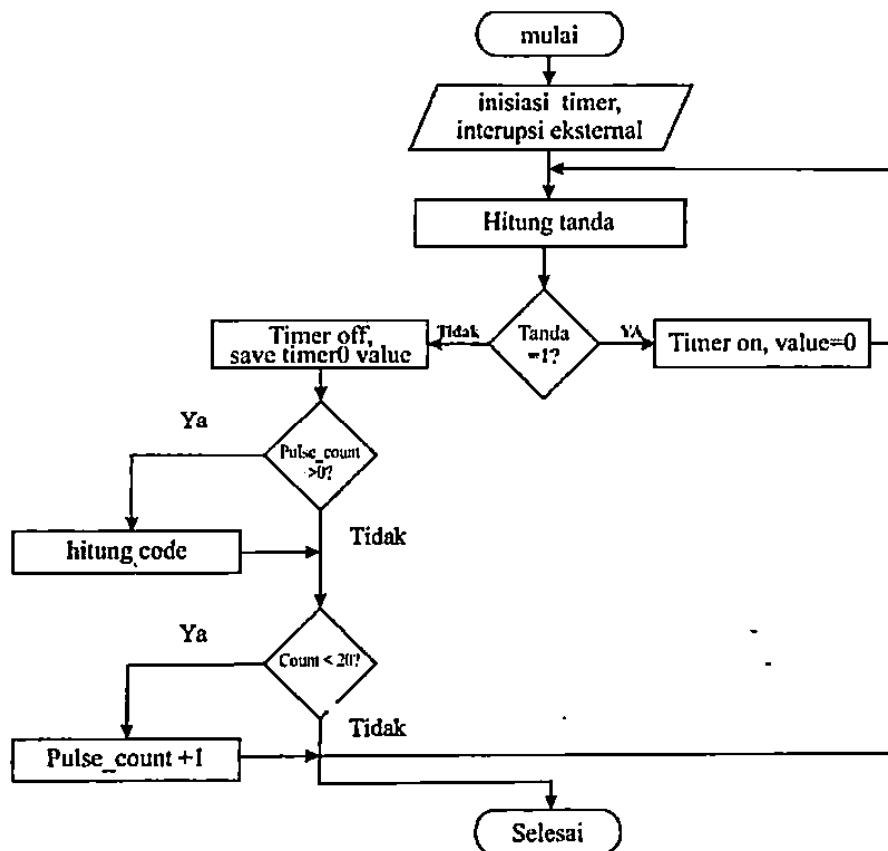
**Gambar 3.15** Contoh data pada suatu tombol

Dengan protokol di atas, data LSB dikirimkan pertama kali, pulsa diawali dengan



command dan 5-bit address. Berikut adalah tabel *output* data yang dikirim *remote control*, dengan penekanan tombol (*Device/Function*).

Setiap kali ada data yang masuk, maka interupsi eksternal ini akan menghitung data address dan data command menggunakan *timer*. Diagram alir di bawah ini akan menunjukkan bagaimana langkah kerja dari program pengolah data *remote control* ini.



**Gambar 3.16** Diagram alir program pengolah data *remote control*

Dari diagram alir tersebut, dapat dibuat source code untuk program pengolah data

```

interrupt [EXT_INT0] void ext_int0_isr(void)
{
tanda=~tanda;
if (tanda==1)
    { MCUCR=0x03; timer0_init(); return; }
else
    { TCCR0=0;count=TCNT0;MCUCR=0x02;
      if(pulse_count>0)goto skip;
      if(count<20)
          {delay_ms(20); return;};
      pulse_count=pulse_count+1;
      return;

      skip:
      data [pulse_count]=count;
      if(data[pulse_count]>9)
          { code=code|(1<<(pulse_count-1));}
      else
          {code=code&~(1<<(pulse_count-1));};
      pulse_count=pulse_count+1;
      if (pulse_count==8)
          { nilai=code;code=0; pulse_count=0;};
    };
}

void timer0_init()
{TCNT0=0;TCCR0=0x05;}

```

```

void int0_init()

```

```

{MCUCR=0x02;GICR|=0x10;}

```

Dari program tersebut, kita akan mendapatkan perhitungan nilai dari interupsi eksternal tersebut. Nilai-nilai itulah yang nantinya akan kita gunakan untuk mempersamakan dengan digit tombol yang ditekan. Persamaan nilai dengan tombol terlihat pada tabel di bawah ini.

**Tabel 3.2** Persamaan nilai dengan tombol pada *remote control* Sony

<b>Nilai</b>	<b>Tombol yang di tekan</b>
0	Digit key 1
1	Digit key 2
2	Digit key 3
3	Digit key 4
4	Digit key 5
5	Digit key 6
6	Digit key 7
7	Digit key 8
8	Digit key 9
9	Digit key 0
16	Channel +
17	Channel -
18	Volume +

19	Volume -
20	Mute
21	Power
22	Reset
23	Audio Mode
24	Contrast +
25	Contrast -
26	Colour +
27	Colour -
30	Brightness +
31	Brightness -
38	Balance Left
39	Balance Right
47	Standby

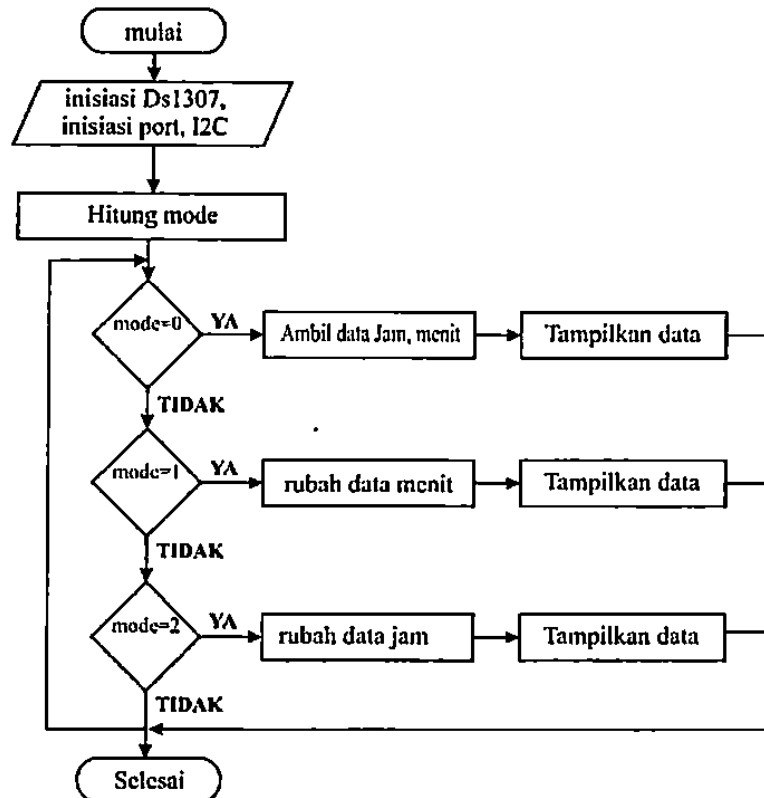
Untuk memproses data analog dari sensor suhu LM35, maka kita dapat menggunakan salah satu port ADC yang ada pada mikrokontroler. Pada program ini port ADC yang digunakan adalah ADC 7 atau PORTA.7. Berikut adalah *source code* inisiasi dan pemrosesan data pada sensor suhu.

```
#define sensor_read_adc(7)
unsigned char read_adc(unsigned char adc_input)
{
    ADMUX=adc_input/(ADC_VREF_TYPE & 0xff);
```

```
PORTA.2=0;  
}
```

Output dari LM 35 adalah sepuluh kali dari nilai suhu yang sebenarnya. Jadi apabila suhu sebenarnya, misalnya,  $30^{\circ}\text{C}$  maka tegangan dari output IC tersebut adalah 300mV. Sehingga perhitungan dalam program adalah nilai tegangan input dari output LM35 dibagi 10.

Dalam melakukan komunikasi antara mikrokontroler ATmega16 sebagai pusat kontrol dan RTC DS 1307, maka metode yang digunakan adalah metode komunikasi I2C. port SDA pada DS 1307 harus sesuai dengan PORT mikrokontroler yang sudah kita set sebagai SDA. Demikian juga dengan port SCL pada DS1307 harus sesuai dengan PORT mikrokontroler yang sudah kita set sebagai SCL. Diagram alir untuk program mengambil data waktu dari RTC DS1307 adalah sebagai berikut



**Gambar 3.17** Diagram alir program pengambil data waktu DS1307

Inisialisasi PORT I2C pada mikrokontroler terlihat seperti di bawah ini.

```

#asm
.equ __i2c_port=0x12 ;PORTD
.equ __sda_bit=6
.equ __scl_bit=5
#endasm
#include <i2c.h>

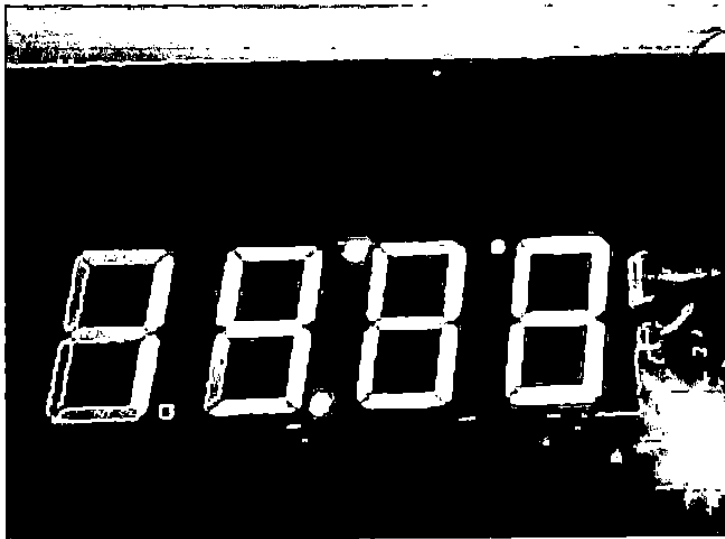
```

Dengan menggunakan komunikasi seperti di atas, maka data yang ada pada RTC

dapat diambil oleh mikrokontroler ATmega16 yang selanjutnya dapat diproses lebih



**Gambar 3.18** Bagian kontrol pada alat



**Gambar 3.19** Bagian display pada alat

### **3.3.4 Pengujian**

#### **3.3.4.1 Pengujian *Software***

Pengujian *software* dilakukan pada tingkat simulasi dengan menggunakan *software* PROTEUS. Pengujian ini dilakukan untuk meyakinkan jika *software* yang dibuat



lanjut dan ditampilkan pada 7 segment. Program pengambilan data dan cara mengubah data waktu (jam dan menit) pada RTC DS1307 adalah sebagai berikut:

```

    delay_ms(200);
    menit = menit + 1;
    if (menit >= 60)
    menit = 0;
    rtc_set_time(jam,menit,detik);
    }
}
else if(mode == 2). //setting jam
{
    display_jam();delay_ms(20); konversi();
    if(SET==0)
    {
        delay_ms(200);

        jam = jam + 1;
        if (jam >= 24)
        jam = 0;
        rtc_set_time(jam,menit,detik);
    }
} -
else;}

```

Pada awalnya program tersebut akan mendeklarasikan DS1307 yang memang di dalam CodeVisionAVR sudah ada. Lalu fungsi mode dan set didefinisikan ke dalam beberapa tombol (PORTD.3 dan PORTD.7). void set\_mode() digunakan untuk memilih mode perubahan yang akan dilakukan. Jika mode perubahan sudah terpilih, maka eksekusi berikutnya adalah pada waktu itu sendiri. apabila mode=0 maka akan

ditampilkan data jam dan menit. Apabila mode=1 maka akan menuju menu untuk mengatur menit. Apabila mode=2 maka akan menuju menu untuk mengatur jam.

Proses menampilkan data pada 7 segment adalah menggunakan array di tiap mode penampilan data. Array adalah sekumpulan data dengan tipe yang sama yang dideklarasikan dalam satu nama variable (*Agus Bejo, 2008*). Pewaktu iqomah, jam dan suhu masing-masing memiliki nama variable array yang berbeda-beda. Walaupun pada intinya array tersebut juga memiliki karakteristik yang sama, namun hanya nama pemanggilannya saja yang berbeda. Hal ini dilakukan agar mempermudah dan tidak membingungkan saat terjadinya proses pemanggilan data. Di bawah ini akan ditunjukkan nama dan deklarasi dari ketiga array tersebut.

```
unsigned int huruf[10] = {0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7d,0x07,0x7f,0x6f};  
unsigned char Angka[10] = {0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7d,0x07,0x7f,0x6f};  
unsigned char bin7seg(unsigned char c)  
{  
    char d7seg_dt_off[10] = {0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7d,0x07,0x7f,0x6f};  
    return d7seg_dt_off[c];  
}
```

Element yang terdapat di dalam array tersebut merupakan angka heksadesimal dimana elemen-elemen tersebut masing-masing mewakili angka pada 7 segment dari angka nol sampai dengan angka Sembilan. Untuk dapat memahaminya lebih mudah

**Tabel 3.3** Array untuk menampilkan digit data

No	Heksadesimal	Biner	Angka pada 7 Segmen
1	3f	0111111	0
2	06	0000110	1
3	5b	1011011	2
4	4f	1001111	3
5	66	1100110	4
6	6d	1101101	5
7	7d	1111101	6
8	07	0000111	7
9	7f	1111111	8
10	6f	1101111	9

Metode untuk menampilkan angka-angka tersebut pada setiap menu yang dipilih adalah menggunakan metode multiplexing. Metode multiplexing akan memungkinkan kita untuk dapat mengontrol beberapa 7 segment hanya dengan menggunakan sedikit port. Setiap *common* 7 segment akan di hubungkan ke port yang berbeda. Cara penampilannya adalah dengan memberikan tegangan pada *common* 7 segment secara bergantian dengan waktu tunda yang diatur sedemikian rupa hingga seolah-olah nyala bersamaan. Untuk metode multiplexing penampil waktu gambar terlihat seperti di bawah ini

```
void display_det()
{
    PORTB.3 = 1;
    PORTC = huruf[satuan_det];
    delay_ms(5);
    PORTB.3 = 0;
    PORTB.2 = 1;
    PORTC = huruf[puluhan_det];
    delay_ms(5);
    PORTB.2 = 0;
}
```

```
void display_men()
{
    PORTB.1 = 1;
    PORTC = huruf[satuan_men];
    delay_ms(5);
    PORTB.1 = 0;
    PORTB.0 = 1;
    PORTC = huruf[puluhan_men];
    delay_ms(5);
    PORTB.0 = 0;
}
```

```
void konver()
{
    puluhan_det = det / 10;
```

```
puluhan_men = men / 10;  
satuan_men = men - (10*puluhan_men);  
}
```

Uraikan source code program di atas untuk menghitung jam dan menit terlewat seperti di bawah ini

```

    delay_ms(1);
    PORTB.0 = 0;
}

void konversi()
{
    puluhan_menit = menit / 10;
    satuan_menit = menit - (10*puluhan_menit);
    puluhan_jam = jam / 10;
    satuan_jam = jam - (10*puluhan_jam)
}

```

*Source code* keseluruhan dari program untuk rangkaian kontroler utama ini dapat dilihat pada Lampiran.

### **3.3 Pembuatan**

Pembuatan ini meliputi realisasi rancangan seluruh rangkaian di atas. Dimulai dari pengadaan bahan, persiapan alat, pengerjaan, dan pengujian.

#### **3.3.1 Pengadaan Bahan**

Dalam pembuatan rangkaian pada tahap pembuatan adalah sebagai berikut:

- Spacer PCB
- *Heatsink*
- Terminal Blok
- Komponen Elektronika seperti table 3.1
- 7 segment 4 buat
- TSOP 1738
- Buzzer
- Komponen pendukung
- Kabel

### **3.3.2 Persiapan Alat**

Alat-alat yang digunakan dalam pembuatan ini antara lain:

- Solder dan Timah



- Penggaris

### 3.3.3 Pengerjaan

Pengerjaan dimulai dengan pembuatan PCB seluruh rangkaian yang telah dirancang sebelumnya. Teknik pembuatan PCB yang diterapkan adalah teknik *transfer paper*. Teknik *transfer paper* adalah suatu teknik pembuatan PCB yang murah tetapi tidak mengesampingkan kualitas. Mula-mula layout PCB di cetak menggunakan printer. Kemudian *printout* tersebut di fotokopi menggunakan kertas Artpaper 120gsm. Hasil fotokopi tersebut dipanaskan dan ditekan pada permukaan PCB menggunakan setrika listrik. Setelah yakin semua tinta berpindah tempat dari transparansi ke PCB, lalu rendam dalam air dan mulai menggosok untuk mengilangkan bekas kertas yang menempel pada permukaan PCB. Selanjutnya PCB dapat dilarutkan dalam larutan  $\text{FeCl}_3$  agar jalurnya dapat tercetak. Untuk mempercepat proses pelarutan maka wadah tempat pelarutan dapat digoyang-goyang. Apabila jalur telah tercetak maka tahap selanjutnya adalah pengeboran lubang-lubang komponen dan pembersihan jalur tembaga pada PCB. Langkah berikutnya yaitu memasang komponen sesuai dengan

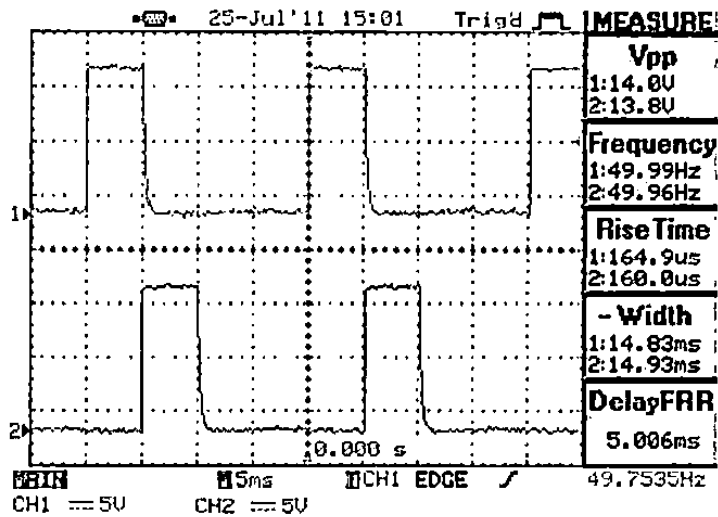
masih terdapat kesalahan dapat langsung diketahui dan dapat segera dilakukan perubahan pada *software*.

#### **3.3.4.2 Pengujian Rangkaian**

Untuk pengujian rangkaian mula-mula adalah apakah dalam rangkaian tersebut terdapat hubung singkat atau tidak. Rangkaian diberikan masukan tegangan sesuai referensi lalu cek apakah ada panas pada komponen tertentu atukah tidak. Jika ada panas, maka akan diperbaiki sampai pada tidak ada komponen yang panas di luar kenormalan.

Berikutnya adalah dengan memasukkan program ke mikrokontroler. Apabila program telah masuk, maka akan dilakukan cek terhadap eksekusi perintah yang dijalankan. Apabila ada kesalahan, maka dilakukan perbaikan pada bagian *software* dan *hardware* yang bersesuaian.

Pengujian rangkaian juga meliputi pengujian terhadap tegangan pada display 7 segment. Pengujian tersebut dilakukan menggunakan oscilloscope digital. Penggunaan oscilloscope digital ini dilakukan karena nilai delay yang ada pada masing-masing segment berganti sangat cepat. Sehingga dalam pengukurannya juga membutuhkan alat ukur yang dapat mengukur dengan cepat



**Gambar 3.20** Pengujian menggunakan oscilloscope

Gambar pengukuran diatas menunjukkan bahwa antara channel 1( berwarna kuning ; 7segment pertama ) dan channel 2 (berwarna biru; 7segment kedua) memiliki jeda waktu yang berbeda pada saat masing-masing 7segment tersebut menyala. Jeda waktu nyala masing-masing segment adalah 5,006 ms. Pada posisi delay tersebut, mata manusia sangat susah untuk dapat melihat pergantian masing-masing segment. Sehingga seolah-olah pada akhirnya keempat segment tersebut menyala dalam waktu yang bersamaan. Tegangan puncak puncak (Vpp) pada common 7 segment adalah antara 13,8 - 14VDC. Sedangkan frekuensi pada common tersebut adalah sekitar 50Hz.

### 3.3.4.3 Pengujian Jarak dan Sudut *Remote Control*

Pengujian ini dilakukan untuk mengetahui seberapa jauh jarak dan sudut yang dapat dikirimkan oleh *Remote control* dan diterima oleh sensor infra merah. Dari hasil pengukuran didapatkan data sebagai berikut:

**Tabel 3.4** Pengaruh penerimaan sensor infra merah terhadap jarak dan sudut (sudut  $0^{\circ}$ ; Tegangan High 4,78 VDC)

No	Jarak	Satuan	Tegangan Output	Satuan	Keadaan
1	0	Cm	3,55	VDC	merespon
2	50	Cm	3,52	VDC	merespon
3	150	Cm	3,6	VDC	merespon
4	250	Cm	3,55	VDC	merespon
5	350	Cm	3,52	VDC	merespon
6	450	Cm	3,62	VDC	merespon
7	550	Cm	4,59	VDC	merespon
8	650	Cm	3,61	VDC	merespon
9	750	Cm	3,57	VDC	merespon
10	850	Cm	3,67	VDC	merespon
11	950	Cm	4,78	VDC	tidak merespon

**Tabel 3.5** Pengaruh penerimaan sensor infra merah terhadap jarak dan sudut (sudut  $45^{\circ}$ ; Tegangan High 4,78 VDC)

No	Jarak	Satuan	Tegangan Output	Satuan	Keadaan
----	-------	--------	-----------------	--------	---------

1	0	cm	3,55	VDC	merespon
2	50	cm	3,5	VDC	merespon
3	150	cm	3,6	VDC	merespon
4	250	cm	3,8	VDC	merespon
5	350	cm	3,53	VDC	merespon
6	450	cm	3,58	VDC	merespon
7	550	cm	4,59	VDC	merespon
8	650	cm	3,56	VDC	merespon
9	750	cm	4,77	VDC	tidak merespon
10	850	cm	4,77	VDC	tidak merespon
11	950	cm	4,78	VDC	tidak merespon

**Tabel 3.6** Pengaruh penerimaan sensor infra merah terhadap jarak dan sudut (sudut  $90^0$ ; Tegangan High 4,78 VDC)

No	Jarak	Satuan	Tegangan Output	Satuan	Kedaaan
1	0	cm	3,55	VDC	merespon
2	50	cm	3,52	VDC	merespon
3	150	cm	3,62	VDC	merespon
4	250	cm	3,5	VDC	merespon
5	350	cm	3,23	VDC	merespon

6	450	cm	4,78	VDC	tidak merespon
7	550	cm	4,78	VDC	tidak merespon
8	650	cm	4,78	VDC	tidak merespon
9	750	cm	4,78	VDC	tidak merespon
10	850	cm	4,78	VDC	tidak merespon
11	950	cm	4,78	VDC	tidak merespon

**Tabel 3.7** Pengaruh penerimaan sensor infra merah terhadap jarak dan sudut (sudut 180<sup>0</sup>; Tegangan High 4,78 VDC)

No	Jarak	Satuan	Tegangan Output	Satuan	Keadaan
1	0	cm	3,55	VDC	merespon
2	50	cm	3,62	VDC	merespon
3	150	cm	3,52	VDC	merespon
4	250	cm	4,77	VDC	tidak merespon
5	350	cm	4,77	VDC	tidak merespon
6	450	cm	4,77	VDC	tidak merespon
7	550	cm	4,77	VDC	tidak merespon
8	650	cm	4,77	VDC	tidak merespon
9	750	cm	4,77	VDC	tidak merespon
10	850	cm	4,77	VDC	tidak merespon

11	950	cm	4,78	VDC	tidak merespon
----	-----	----	------	-----	----------------

Dari data tersebut kita dapat melihat bahwa saat sensor penerima infra merah tidak menerima data, maka tegangan pada output adalah sekitar 4,77 VDC (*High*). Sedangkan pada saat penerima infra merah mendapatkan data, tegangan output tercatat sekitar 3,6 VDC (*Low*). Ada perbedaan tegangan sekitar 1,2 V antara logika *High* dan *Low*.

Pada sudut  $0^{\circ}$ , atau dalam kondisi lurus, jangkauan pengiriman data dapat mencapai 8,5 meter. Apabila sudut pengiriman diubah menjadi  $45^{\circ}$  maka jarak penerimaan maksimal menjadi lebih pendek, yaitu 6,5 meter. Pada sudut  $90^{\circ}$  jarak penerimaan maksimal hanya 3,5 meter. Sudut yang terakhir dicoba adalah sudut  $180^{\circ}$  atau pengiriman dilakukan dari belakang sensor penerima infra merah. Ternyata data masih dapat diterima, namun penerimaan maksimumnya sangat pendek yaitu hanya 1,5 meter.

#### 3.3.4.4 Pengujian Sensor Suhu LM 35

Pengujian ini dilakukan untuk mengetahui apakah data analog yang dikeluarkan oleh sensor suhu LM35 nilainya sama dengan data digital yang ditampilkan oleh 7 segment. Hasil dari pengukuran dan pengujian tersebut dapat dilihat pada tabel di

**Tabel 3.8** Perbandingan antara nilai analog yang terukur oleh LM35 dan nilai yang terbaca pada mikrokontroler.

No	Pengukuran data analog	Satuan	Perhitungan nilai suhu	Pembacaan nilai suhu	Satuan
1	296	mVolt	29,6	30	Celcius
2	312	mVolt	31,2	31	Celcius
3	316	mVolt	31,6	32	Celcius
4	329	mVolt	32,9	33	Celcius
5	335	mVolt	33,5	34	Celcius
6	343	mVolt	34,3	35	Celcius
7	357	mVolt	35,7	36	Celcius
8	364	mVolt	36,4	37	Celcius
9	373	mVolt	37,3	38	Celcius
10	383	mVolt	38,3	39	Celcius
11	394	mVolt	39,4	40	Celcius
12	404	mVolt	40,4	41	Celcius
13	414	mVolt	41,4	42	Celcius
14	424	mVolt	42,4	43	Celcius
15	433	mVolt	43,3	44	Celcius



Dari data tersebut kita dapat melihat bahwa nilai pembacaan sudah sangat mendekati nilai perhitungan pembacaan. Hal tersebut menandakan bahwa setting yang dilakukan pada mikrokontroler dan perhitungan yang ada di dalamnya telah dieksekusi dengan baik. Dari semua bagian pengukuran tidak ada satupun yang memperlihatkan pembacaan data suhu melebihi data suhu berdasarkan perhitungan lebih dari 1°C. Proses pengambilan data tersebut dilakukan dengan bertahap. Sensor suhu mula-mula didekatkan pada sebuah pemanas, secara otomatis data suhu akan naik perlahan dan terus didekatkan sampai dengan batas yang kita inginkan.

#### **3.3.4.5 Kendala pada Saat Pengujian**

Saat pengujian dilakukan terdapat kendala sebagai berikut. Ketika data dari *remote control* memanggil sub program pewaktu iqomah, maka pewaktu iqomah mula-mula (misal dimulai pada 10:00) dapat berjalan seperti yang diharapkan. Namun pada suatu saat (misal pada 09:36) pewaktu iqomah tersebut berhenti. Setelah diberi masukan dari *remote control* lagi, baru perhitungan mundur berjalan lagi. Namun lagi-lagi berhenti pada waktu tertentu (misal pada 09.06).

Setelah dilakukan analisis, disimpulkan bahwa masalah tersebut timbul karena di dalam alat tersebut terdapat dua buah timer yang bekerja bersamaan, sehingga

ATMega16, jika terdapat beberapa timer yang bekerja bersamaan maka interupsi eksternallah yang selalu diprioritaskan. Dengan demikian wajarlah seandainya di tengah jalan timer untuk waktu iqomah berhenti dengan sendirinya, karena penggunaan interupsi eksternal dengan internal timer pada suatu mikrokontroler dapat mengakibatkan eror pada nilai variabel yang dieksekusi.

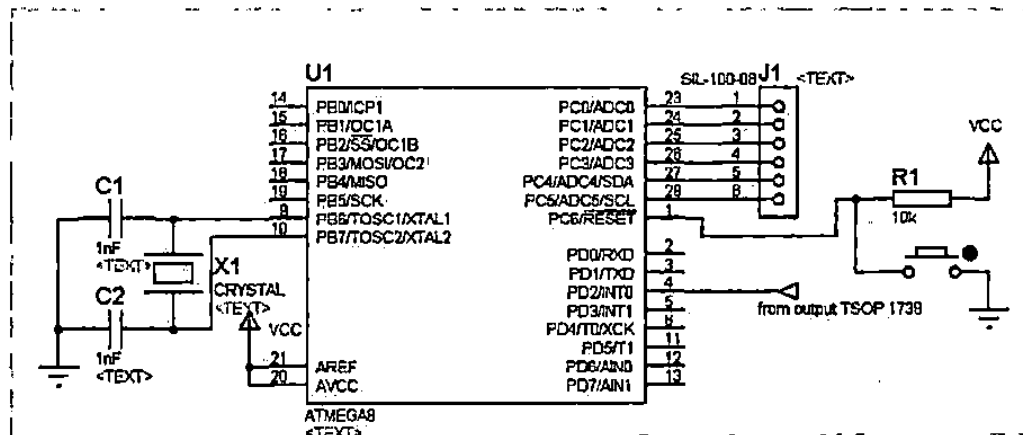
#### **3.3.4.6 Perubahan yang dilakukan pada rangkaian software dan Hardware**

Melihat kendala yang dialami saat dilakukan pengujian dan analisis yang dilakukan terhadap masalah tersebut, maka dilakukan pemecahan masalah sebagai berikut.

- Perlu adanya pemisahan 2 buah timer yang pada saat itu bekerja bersamaan.
- Pemisahan dapat dilakukan dengan cara menambahkan satu buah mikrokontroler tersendiri yang khusus menangani pemrosesan data dari *remote control*. Mikrokontroler yang dipilih untuk menangani data *remote control* adalah ATMega8. Mikrokontroler ini dipilih karena di dalamnya terdapat fasilitas interupsi eksternal (INT0), berukuran cukup kecil serta secara kebetulan penulis memikinya. Sehingga tidak perlu untuk mencari mikrokontroler yang lain.
- Output TSOP1738 yang tadinya masuk port INT0 ATMega16, dipindahkan ke PORTD.2 pada ATMega8. PORTD.2 adalah port yang

- Program pengolah data yang tadinya berada pada ATmega16, dipindahkan ke dalam ATmega8.
- Hubungan komunikasi antara Mikrokontroler ATmega16 dan ATmega8 dilakukan secara parallel. Komunikasi ini dipilih karena bukan data yang dikirimkan oleh ATmega8, melainkan hanya logika *High* (1) dan *Low* (0) saja. Output port pada ATmega8 digunakan untuk memilih menu yang berada pada ATmega 16.

Detail program yang terdapat pada ATmega16 dan ATmega 8 dapat dilihat pada Lampiran.



Gambar 3.21 Rangkaian pengolah data infra merah