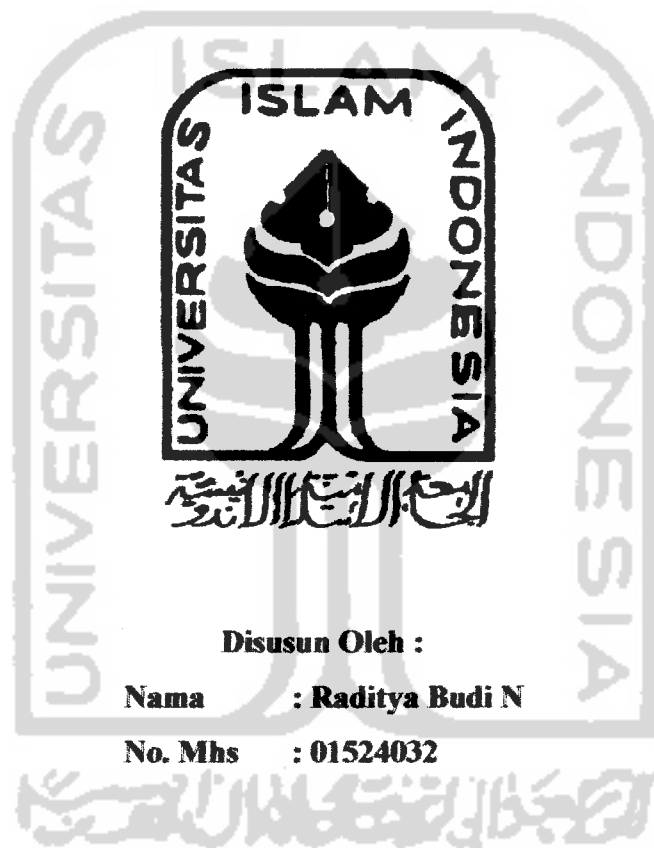


**PENCATAT SKOR DENGAN KENDALI REMOTE  
TUGAS AKHIR**

**Diajukan Sebagai Salah Satu Syarat Memperoleh Gelar Sarjana  
Teknik Elektro Fakultas Teknologi Industri  
Universitas Islam Indonesia**



**Disusun Oleh :**

**Nama : Raditya Budi N**

**No. Mhs : 01524032**

**JURUSAN TEKNIK ELEKTRO  
FAKULTAS TEKNOLOGI INDUSTRI  
UNIVERSITAS ISLAM INDONESIA  
JOGJAKARTA**

**2007**

**LEMBAR PENGESAHAN DOSEN PENGUJI**

**PENCATAT SKOR DENGAN KENDALI REMOTE**

**TUGAS AKHIR**

Disusun oleh :

Nama : Raditya Budi N

No. Mhs : 01 524 032

Telah Dipertahankan Di Depan Sidang Penguji, Sebagai Salah Satu  
Syarat Untuk Memperoleh Gelar Sarjana Teknik Elektro,  
Fakultas Teknologi Industri, Universitas Islam Indonesia

Yogyakarta, Juni 2007

Tim Penguji

Ir. Hj. Budi Astuti, MT  
Ketua

Wahyudi Budi Pramono, ST  
Anggota I

Yusuf Aziz Amrulloh, ST  
Anggota II



Mengetahui,

Ketua Jurusan Teknik Elektro

Fakultas Teknologi Industri

Universitas Islam Indonesia



(Uto Yuwono, ST, M.Sc)

## **HALAMAN PERSEMBAHAN**

**Dengan segala Rasa hormat, Kasih sayang dan kerendahan hati serta bakti Kupersembahkan karyaku ini untuk Ayahandaku, alm. Ibundaku dan adek-adeku, Terima kasih atas segala do'a dan kasih sayangnya serta kepercayaan yang telah diberikan untuk Radit selama ini.**



## MOTTO

*“Hai orang-orang yang beriman, mintalah pertolongan dari Allah dengan kesabaran dan salat. Sungguh Allah bersama orang-orang yang sabar.”*

*(QS. Al-Baqarah:153)*

*“...Dan aku serahkan segala urusanku kepada Allah. Sesungguhnya Allah Maha Melihat akan hamba-hambaNya”*

*(QS. Ghaafir : 44)*

*“Tuhanmu telah memerintahkan, “Mohonlah kepada-Ku, Aku pasti akan mengabulkan doamu””*

*(QS. Ghaafir : 60)*

*“Tiada suatu kepayahan, kesakitan, kesedihan, kesusahan, penderitaan dan kesukaran sampaipun duri yang menyakitkan itu menimpa kepada seorang mu'min melainkan dengan itu semua Allah akan menutupi dosa-dosanya”*

*(HR. Bukhari dan Muslim)*

*“ Sungguh bersama kesukaran pasti ada kemudahan”*

*(QS. Asy Syarh : 5)*

## KATA PENGANTAR



*Assalamu'alaikum Wr. Wb.*

Puji syukur kehadirat Allah SWT, atas nikmat iman, rahmat, hidayah dan pikiran yang diberikan. Sehingga penulis dapat menyelesaikan sebuah tugas akhir dengan judul “ Pencatat Skor Dengan Kendali Remote“. Tidak lupa shalawat serta salam selalu tercurah kepada Nabi Muhammad. SAW beserta keluarga dan para sahabatnya.

Adapun maksud dari penyusunan tugas akhir ini adalah untuk memenuhi kurikulum S-1 Jurusan Teknik Elektro, Fakultas Teknologi Industri, Universitas Islam Indonesia. Disamping itu juga untuk menambah pengetahuan atas disiplin ilmu yang telah dipelajari di bangku perkuliahan untuk diterapkan ke masyarakat.

Selama mengerjakan tugas akhir dan dalam penyusunan laporan, tidak lepas dari hambatan, namun berkat motivasi, informasi dan konsultasi dari berbagai pihak, semua masalah dapat diatasi. Untuk itu penyusun menyampaikan rasa hormat sebagai ungkapan terima kasih kepada :

1. Kedua orang tuaku yang senantiasa memberikan dukungan moril, materi dan do'a setiap saat.
2. Bpk Tito Yuwono, ST, M.Sc, selaku Ketua Jurusan Teknik Elektro Fakultas Teknologi Industri Universitas Islam Indonesia.

3. Bpk. Wahyudi Budi Pramono, ST sebagai dosen pembimbing I.
4. Bpk Yusuf Aziz Amrulloh, ST selaku Sekretaris Jurusan Teknik Elektro dan sebagai dosen pembimbing II.
5. Segenap dosen Teknik Elektro Fakultas Teknologi Industri Universitas Islam Indonesia yang telah memberikan ilmunya.
6. My luv nophe yang telah memberi suport jasmani dan rohani. Semoga kita punya anak sebanyak-banyaknya.
7. Seluruh mahasiswa jurusan Teknik Elektro UII dan teman-temanku khususnya Somadto, Rembol ST, Chickent, Gembel, Tukul ST, Enggar dan Ade terima kasih atas bantuannya yang telah diberikan.
8. .Mr Boshe... *Thanx for support*
9. *Xperience... thanx guys.. keep on rock*
10. And especially to Boshe vvip, Caesar, Liquid, Hugos, Embessy, alm TJ's The Club, dan Terrace *Thanx to spend my money.*

Penulis menyadari bahwa penyusunan tugas akhir ini masih terdapat kesalahan dan kekurangan. Oleh karena itu, kritik dan saran yang membangun akan senantiasa penulis terima dengan senang hati. Semoga laporan ini dapat bermanfaat kepada penulis pada khususnya dan kepada pembaca yang budiman pada umumnya, amin.

***Wassalamu'alaikum Wr.Wb***

Yogyakarta, Juni 2007

Raditya Budiharyanto Nugroho

## ABSTRAK

Perkembangan teknologi elektronika telah berkembang dengan pesat, sehingga manusia berusaha untuk membuat suatu sistem yang bisa mempermudah dan membuat efisien segala sesuatu yang dikerjakannya. Alat kendali papan skor di berbagai gedung olah raga masih menggunakan sistem tombol atau sistem manual. Oleh karena itu untuk mempermudah dalam pengoperasian maka digunakan kendali remote supaya lebih praktis dan efisien. Pada pembuatan alat ini digunakan mikrokontroler AVR yang memiliki arsitektur RISC 8 bit dimana semua instruksi dikemas dalam kode 16 bit (16-bit word) dan sebagian besar instruksinya dikemas dalam satu siklus clock. Alat pencatat skor dengan kendali remote ini menggunakan infra merah dari remote sebagai masukan dan IR receiver sebagai penerima infra merah. Mikrokontroler ATmega8 merupakan pengontrol yang mengatur proses kerja secara keseluruhan baik mulai dari masukan/input mengolah data yang masuk sampai ke output/keluaran display yang berupa led maupun dot matrik. Pembuatan seven segment ini menggunakan lampu LED yang disusun selayaknya sebuah seven segment. Untuk LED kecil membutuhkan arus 15 mA dan tegangan 1,5 V. Sedangkan LED besar membutuhkan arus 15mA dan tegangan 3 V. Ketika power On maka alat dapat disetting baik nama kedua team, babak pertandingan, waktu pertandingan. Setelah semua di-setting maka untuk menjalankannya tekan tombol display pada remote. Dan akan berhenti ketika waktu yang disetting berakhir. Saat waktu pertandingan berjalan, maka skor baru dapat dijalankan. Dengan remote merk sony alat ini dapat bekerja, dan remote inilah yang dapat mengendalikan papan skor. Dari hasil perancangan, alat ini dapat berfungsi seperti yang diinginkan, meskipun masih terdapat kelemahan pada remote, yakni kendala pada jarak atau jangkauan 1 meter.

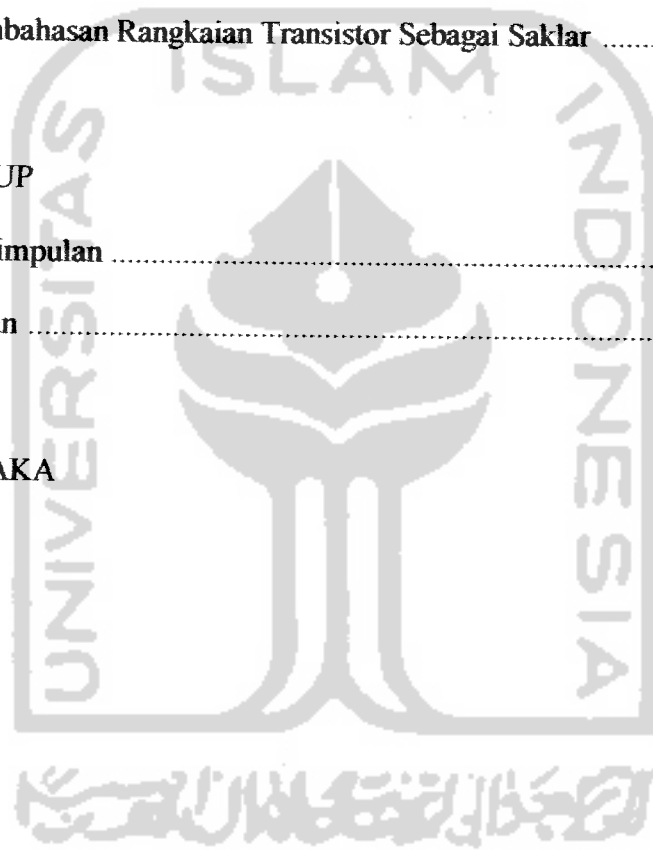
## DAFTAR ISI

HALAMAN JUDUL .....	i
HALAMAN LEMBAR PENGESAHAN PEMBIMBING .....	ii
HALAMAN PENGESAHAN PENGUJI .....	iii
HALAMAN PERSEMBAHAN .....	iv
MOTTO .....	v
KATA PENGANTAR .....	vi
ABSTRAK .....	viii
DAFTAR ISI .....	ix
DAFTAR GAMBAR .....	xii
<b>BAB I PENDAHULUAN</b>	
1.1 Latar Belakang Masalah .....	1
1.2 Perumusan Masalah .....	1
1.3 Batasan Masalah .....	2
1.4 Tujuan .....	2
1.5 Manfaat Pembuatan Alat .....	2
1.6 Sistematika Penulisan .....	3
<b>BAB II DASAR TEORI</b>	
2.1 Transistor .....	4
2.2 Mikrokontroler ATmega8 .....	7



2.2.1 Fitur ATmega8 .....	8
2.2.2 Konfigurasi Pin ATmega8 .....	9
2.2.3 PETA Memori .....	10
2.2.4 Timer/Counter .....	10
2.3 LED (Diode Penghasil Cahaya) .....	13
2.4 IC75HC595A .....	14
2.5 Remote Sony .....	16
2.6 <i>Infra Red Receiver</i> .....	17
 <b>BAB III PERANCANGAN SISTEM</b>	
3.1 Perancangan Sistem .....	22
3.2 Perancangan Perangkat Keras .....	23
3.2.1 <i>Power Supply</i> .....	23
3.2.2 Mikrokontroler ATmega8 .....	24
3.2.3 Dot Matrik .....	26
3.2.4 Seven Segment .....	27
3.3 Perancangan Perangkat Lunak .....	28
 <b>BAB IV ANALISA DAN PEMBAHASAN</b>	
4.1 Pengamatan Sensor Infra Merah .....	30
4.2 Pengamatan Dot Matrik Sebagai penampil Nama Team .....	32
4.3 Pengamatan Seven Segment sebagai Penampil Skor, Waktu dan	

Babak Pertandingan .....	34
4.3.1 Pengamatan Seven Segment Sebagai Penampil Skor .....	34
4.3.2 Pengamatan Seven segment Sebagai Penampil Waktu .....	34
4.3.3 Pengamatan Seven Segment Sebagai Penampil Babak .....	35
4.4 Bentuk Gelombang Sensor Infra Merah .....	36
4.5 Pembahasan Rangkaian Transistor Sebagai Saklar .....	39
BAB V PENUTUP	
5.1 Kesimpulan .....	41
5.2 Saran .....	41
DAFTAR PUSTAKA	
LAMPIRAN	



## DAFTAR GAMBAR

Gambar 2.1 Simbol transistor NPN dan PNP .....	4
Gambar 2.2 Garis beban DC .....	6
Gambar 2.3 Arsitektur Atmega8 .....	8
Gambar 2.4 Konfigurasi pin ATmega8 .....	9
Gambar 2.5 Inisialisasi Prescaler TCCR0 .....	11
Gambar 2.6 Register TCCRA .....	11
Gambar 2.7 Register TCCRB .....	12
Gambar 2.8 Simbol LED .....	13
Gambar 2.9 Logic diagram IC 74HC595A .....	14
Gambar 2.10 Teknik pengkodean pulsa .....	16
Gambar 2.11 Tampilan sensor infrared .....	18
Gambar 2.12 Tampilan rangkaian sensor infra merah .....	18
Gambar 2.13 Respon penerimaan sensor infrared .....	19
Gambar 3.1 Blok diagram pencatat skor dengan kendali remote .....	22
Gambar 3.2 Rangkaian <i>power supply</i> .....	24
Gambar 3.3 Mikrokontroler ATmega8 .....	25
Gambar 3.4 Dot matrik penampil nama team .....	26
Gambar 3.5 Seven segmen sebagai penampil skor .....	27
Gambar 3.6 Seven segmen sebagai penampil waktu dan babak .....	28
Gambar 3.7 Flowchart program ATmega8 .....	29

Gambar 4.1 Remote merk sony .....	31
Gambar 4.2 Tampilan papan skor .....	32
Gambar 4.3 Tampilan nama kedua team yang bertanding .....	33
Gambar 4.4 Tampilan skor kedua team .....	34
Gambar 4.5 Tampilan waktu pertandingan .....	35
Gambar 4.6 Tampilan babak pertandingan .....	36
Gambar 4.7 Output gelombang infra merah kondisi normal .....	36
Gambar 4.8 Output gelombang infra merah pada saat tombol ditekan 1 .....	37
Gambar 4.9 Output gelombang infra merah pada saat tombol ditekan 2 .....	37
Gambar 4.10 Output gelombang infra merah pada saat tombol ditekan 4 .....	37
Gambar 4.11 Output gelombang infra merah pada saat tombol ditekan 5 .....	38
Gambar 4.12 Output gelombang infra merah pada saat tombol ditekan 6 .....	38
Gambar 4.13 Output gelombang infra merah pada saat tombol ditekan 9 .....	38
Gambar 4.14 Output gelombang infra merah pada saat tombol ditekan 0 .....	39
Gambar 4.15 Rangkaian transistor sebagai saklar .....	39

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang Masalah**

Perkembangan teknologi elektronika telah berkembang dengan pesat, sehingga manusia berusaha untuk membuat suatu sistem yang bisa mempermudah dan membuat efisien segala sesuatu yang dikerjakannya. Manusia tidak henti-hentinya mengkaji dan meneliti teknologi elektronika untuk berbagai macam kegunaan atau manfaat yang akan dirasakan seperti alat pencatat skor dengan kendali remote. Kebanyakan alat kendali papan skor di berbagai gedung olah raga masih menggunakan sistem tombol atau sistem manual kecuali gedung olah raga yang bertaraf nasional dan internasional. Oleh karena itu untuk mempermudah dalam pengoperasian atau penggunaannya maka digunakan kendali remote supaya lebih praktis dan efisien. Dalam pembuatan alat ini digunakan sinyal *infra red* sebagai masukan dari remote, dan *out put* berupa dot matrik yang ditampilkan oleh LED. Sebagai penampil skor, babak dan waktu juga digunakan LED berukuran besar yang mana disusun layaknya sebuah seven segmen. Sebagai otak dari alat ini digunakan ATMega8.

### **1.2 Perumusan Masalah**

Untuk merancang dan membuat alat ini permasalahan lebih ditekankan pada :

1. Bagaimana merancang alat pencatat papan skor dengan kendali remote.

2. Bagaimana merancang dan membuat perangkat lunak mikrokontroler yang mampu mengolah data masukan dari remote dan keluaran menggunakan dot matrik yang ditampilkan berupa LED.

### 1.3 Batasan Masalah

Batasan masalah dari pembuatan alat pencatat skor dengan kendali remote ini adalah sebagai berikut :

1. Jangkauan yang mampu diterima sensor adalah 2m.
2. Remote hanya menggunakan 1 merk, yaitu sony.
3. *Output* ditampilkan melalui LED.

### 1.4 Tujuan

Adapun dalam rangka merancang dan membuat **Pencatat Skor Dengan Kendali Remote**, maka tujuan dari pembuatan alat ini adalah :

1. Merancang dan membuat alat pencatat skor yang praktis dan efisien.
2. Mampu diaplikasikan ke dalam gedung olah raga kampus maupun di gedung olah raga manapun.
- 3.

### 1.5 Manfaat Pembuatan Alat

Manfaat pembuatan alat ini yaitu membuat praktis dan efisien dengan cara mengganti skor secara langsung menggunakan remote sehingga dapat terlihat pada papan skor yang ditampilkan oleh LED.

## 1.6 **Sistematika Penulisan**

Dalam penulisan, sistematika penulisannya terdiri dari lima bab, yaitu :

### **BAB I : PENDAHULUAN**

Bab ini memuat latar belakang masalah, perumusan masalah, batasan masalah, tujuan, manfaat penelitian dan sistematika penulisan.

### **BAB II : DASAR TEORI**

Berisi tentang penjelasan komponen-komponen yang digunakan dalam pembuatan alat "Pencatat Skor Dengan Kendali Remote".

### **BAB III : PERANCANGAN SISTEM**

Berisi tentang perancangan sistem yang berupa perancangan perangkat keras dan perancangan perangkat lunak..

### **BAB IV : ANALISA DAN PEMBAHASAN**

Bab ini berisi hasil pengujian dari alat yang dibuat dan kemudian dianalisa.

### **BAB V : PENUTUP**

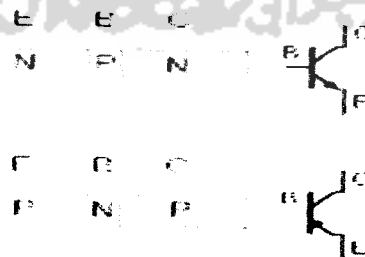
Pada bagian penutup memuat kesimpulan dari alat yang dibuat dan saran guna perbaikan serta pengembangan alat ini.

## BAB II

### DASAR TEORI

#### 2.1 Transistor

Transistor merupakan dioda dengan dua sambungan (*junction*). Sambungan itu membentuk transistor PNP maupun NPN. Ujung-ujung terminalnya berturut-turut disebut emitor, basis dan kolektor. Basis selalu berada di tengah, di antara emitor dan kolektor. Transistor ini disebut transistor bipolar, karena struktur dan prinsip kerjanya tergantung dari perpindahan elektron di kutub negatif mengisi kekurangan elektron (*hole*) di kutub positif. Transistor adalah komponen yang bekerja sebagai saklar (*switch on/off*) dan juga sebagai penguat (*amplifier*). Transistor bipolar memiliki 2 *junction* yang dapat disamakan dengan penggabungan 2 buah dioda. *Emitor-Base* adalah satu *junction* dan *Base-Kolektor junction* lainnya. Adapun simbol dari transistor NPN dan PNP ditunjukkan pada Gambar 2.1 berikut ini :



Gambar 2.1 Simbol transistor NPN dan PNP



Pada awalnya, diperkirakan bahwa transistor bekerja dalam salah satu arah yaitu dengan saling menghubungkan ujung-ujung kolektor dan emitter karena kedua ujung tersebut terbuat dari jenis bahan yang sama. Namun hal ini tidaklah mungkin karena kedua ujung tidak berukuran sama. Kolektor berukuran lebih besar dan kebanyakan dihubungkan secara langsung ke kotaknya untuk penyerapan panas. Sambungan basis *emitter* hanya mampu menahan tegangan rendah. Jika arus mengalir ke dalam basis dan melewati sambungan basis *emitter*, suatu suplai positif pada kolektor akan menyebabkan arus mengalir di antara kolektor dan emitter. Dua hal yang perlu diperhatikan pada arus kolektor, adalah :

1. Untuk arus basis nol, arus kolektor turun sampai pada tingkat arus kebocoran, yaitu  $<1 \mu\text{A}$  dalam kondisi normal (transistor silikon).
2. Untuk arus basis tertentu, arus kolektor yang mengalir akan jatuh lebih besar daripada arus basis itu.

Dalam kondisi normal, masukan kaki basis transistor tidak dibias sehingga titik kerjanya berada pada daerah *cut off* dan tidak ada arus yang mengalir melalui tahanan kolektor. Apabila masukan kaki basis transistor mendapat bias yang cukup untuk mengaktifkan transistor, maka titik kerjanya berubah dari *cut off* ke saturasi. Besarnya arus yang mengalir melalui tahanan basis adalah :

$$I_B = \frac{V_{BB} - V_{BE}}{R_B} \dots\dots\dots (2.1)$$

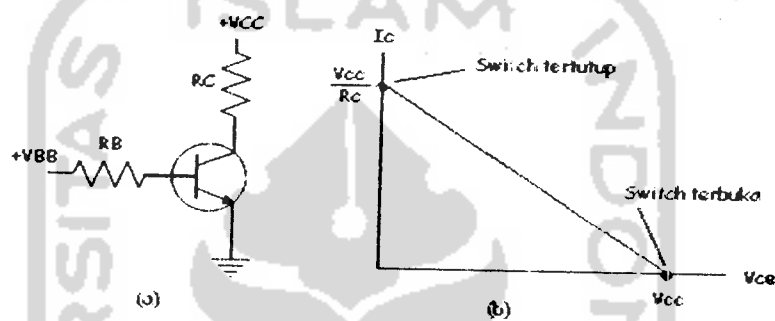
Keterangan :  $I_B$  : Arus basis (A)

$V_{BB}$  : Sumber tegangan basis (V)

$V_{BE}$  : Tegangan basis-emiter (V)

$R_B$  : Hambatan basis ( $\Omega$ )

Adapun garis beban transistor tampak seperti pada Gambar 2.2 dibawah ini :



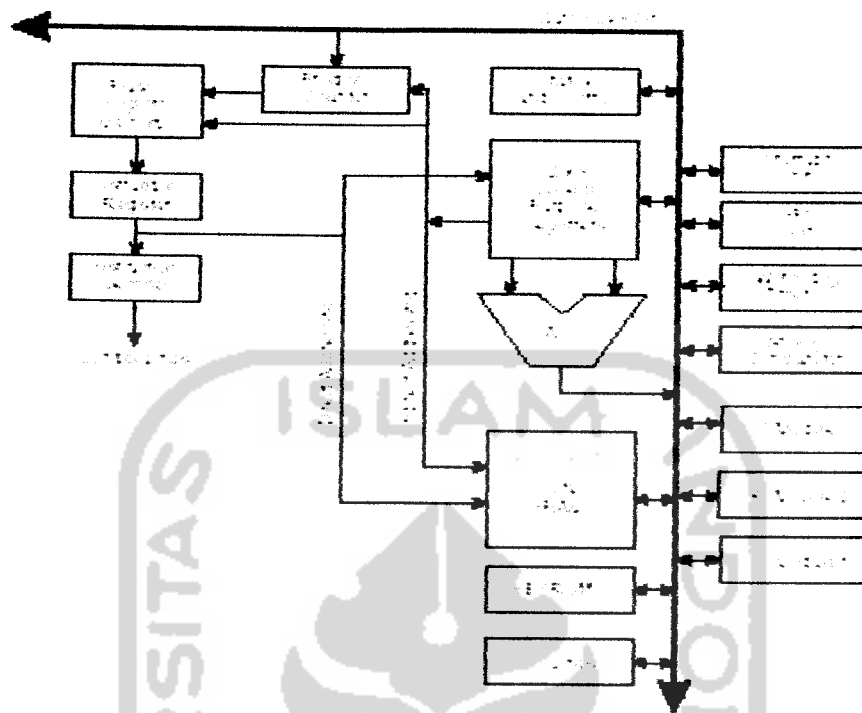
Gambar 2.2 Garis beban DC

Pada penelitian ini digunakan transistor TIP2955, transistor ini digunakan sebagai saklar dimana transistor memanfaatkan keadaan kerja penuh dan keadaan tidak bekerja sama sekali (*cut off*) sebagai saklar transistor hanya mempunyai dua keadaan, yaitu On dan Off. Keadaan On dicapai pada saat VCE mendekati nol, karena pada keadaan On (*saturasi*) VCE sangat rendah sedangkan IC sangat tinggi, sehingga transistor tersebut seperti sebuah saklar yang tertutup dari kolektor ke emitor. Keadaan OFF akan dicapai pada saat VCE mendekati VCC. Karena pada keadaan OFF (*cut off*) akan dicapai pada saat VCE mendekati VCC. Karena pada keadaan OFF (*cut off*) VCE sangat besar sedangkan arus yang mengalir sangat kecil, sehingga transistor seperti sebuah saklar terbuka.

## 2.2 Mikrokontroler ATmega8

Mikrokontroler merupakan salah satu sistem yang dapat digunakan sebagai sistem komputer yang dapat digunakan untuk mengendalikan atau sebagai otak dan sistem. Mikrokontroler AVR memiliki arsitektur RISC 8 bit dimana semua instruksi dikemas dalam kode 16 bit (16-bit *word*) dan sebagian besar instruksinya dikemas dalam 1 (satu) siklus clock yang berbeda dengan siklus MCS-51 yang membutuhkan 12 siklus clock. AVR berteknologi RISC (*Reduced Instruction Set Computing*). sedangkan MSC-51 berteknologi CISR (*Complex Instruction Computing*). Arsitektur dari ATmega8 dapat ditunjukkan seperti Gambar 2.3, dimana bagian-bagian dari ATmega8 itu sendiri adalah sebagai berikut :

1. Saluran I/O sebanyak 22 buah, yaitu Port B, Port C, Port D.
2. ADC 10 bit sebanyak 6 saluran.
3. Tiga buah Timer/ Counter.
4. CPU dengan 32 buah register.
5. Watchdog Timer.
6. 1 Kbyte internal SRAM.
7. Memori flash sebesar 8 Kbytes
8. EEPROM sebesar 512 yang dapat diprogram saat operasi.



Gambar 2.3 Arsitektur Atmega8

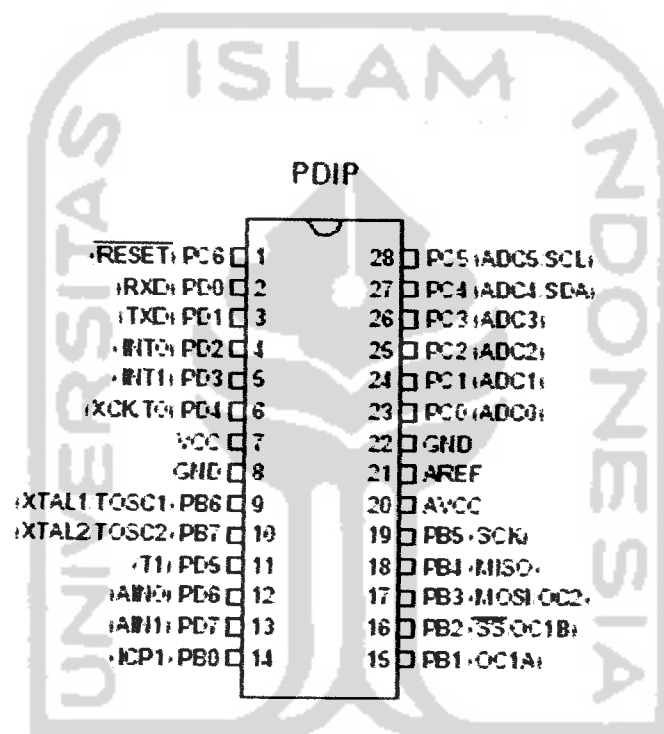
### 2.2.1 Fitur Atmega8

Kapabilitas detail dari ATmega8 adalah sebagai berikut :

1. Dua buah Timer/Counter 8-bit satu buah Timer/Counter 16-bit.
2. Tiga buah saluran PWM.
3. ADC 10-bit dengan 6 saluran.
4. USART untuk komunikasi serial.
5. Lima pilihan mode sleep untuk menghemat penggunaan energi listrik yaitu *Idle*, *ADC Noise Reduction*, *Power save*, *PowerDown* and *Standby*.

### 2.2.2 Konfigurasi Pin Mikrokontroler ATmega8

Karakteristik dari mikrokontroler ATmega8 memiliki 28 pin yang mempunyai fungsi yang berbeda, tergantung penggunaan port yang tersedia dan tentunya tergantung pada perancangan program yang dituliskan dan di-download pada mikrokontroler tersebut. Konfigurasi pin ditunjukkan seperti Gambar 2.4 berikut ini :



Gambar 2.4 Konfigurasi pin ATmega8

Konfigurasi pin ATmega8 dapat dijelaskan secara fungsional sebagai berikut :

1. VCC merupakan pin yang berfungsi sebagai masukan catu daya.
2. GND merupakan pin Ground.
3. Port B (PBO — PB7) merupakan pin I/O dua arah.
4. Port C (PC0 PC5) merupakan pin I/O dua arah pin masukan ADC.

5. Port D (PC0 — P07) merupakan pin I/O dua arah.
6. RESET merupakan pin yang digunakan untuk mereset mikrokontroler.
7. XTAL1 dan XTAL2 merupakan pin masukan clock eksternal.
8. AVCC merupakan pin masukan untuk ADC.
9. AREF merupakan pin masukan referensi ADC.

### 2.2.3 PETA Memori

ATMega8 memiliki pengalamatan memori data dan memori program yang terpisah. Memori data terbagi menjadi 3 bagian, yaitu 32 register umum, register I/O, dan 1 Kbyte SRAM Internal.

Register keperluan umum menempati *space* data pada alamat terbawah, yaitu \$00 sampai \$1F. Sementara itu, register khusus untuk menangani I/O dan kontrol terhadap mikrokontroler menempati alamat berikutnya yaitu mulai \$20 sampai \$5F. Register tersebut merupakan register khusus yang digunakan untuk mengatur fungsi terhadap sebagian periperal mikrokontroler, seperti kontrol register, timer/counter, fungsi-fungsi I/O.

### 2.2.4 Timer/Counter

AVR ATMega8 memiliki tiga buah timer, yaitu Timer/Counter 0 (8 bit), Timer Counter 1 (16 bit) dan Timer Counter 2 (8 bit). Ketiga bit tersebut digunakan untuk memilih sumber clock oleh timer/counter.

Timer/Counter 0 merupakan Timer/Counter 8 bit multifungsi, diskripsi untuk Timer/Counter 0 pada ATmega8 adalah sebagai berikut :

1. Timer dinolkan saat *match compare (auto reload)*.
2. Frekuensi generator.
3. Prescaler 10 bit untuk timer.
4. Instruksi timer yang disebabkan timer *overflow* dan *match compare*.

Pengaturan Timer/Counter 0 diatur oleh TCCRO (Timer Counter Control Register 0) yang dapat dilihat pada Gambar 2.5 berikut ini.



Gambar 2.5 Inisialisasi Prescaler TCCRO

Pengaturan Timer/Counter I diatur melalui register TCCRIA, yang dapat dilihat pada Gambar 2.6 berikut ini.



Gambar 2.6 Register TCCRIA

Penjelasan setiap bit :

1. Bit 7 : 6 \_ COMIA 1 : 0 : Compare output mode untuk channel A

Bit 5 : 4 \_ COMIB 1 : 0 : Compare output mode untuk channel B

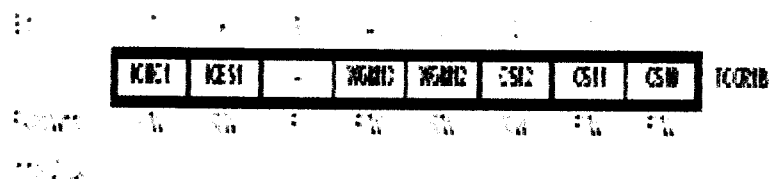
Register COM1A 1 : 0 dan COMIB 1 : 0 mengontrol kondisi pin output compare (OCIA dan OCIB). Jika salah satu atau kedua bit pada register COMIA 1 : 0 ditulis menjadi satu, maka kaki pin OCIA tidak berfungsi normal sebagai port I/O. Begitu juga pada register COMIB 1:0 ditulis menjadi satu, maka kaki pin OCIB juga tidak berfungsi normal sebagai port I/O. Fungsi pada pin OCIA dan OCIB tergantung pada setting bit pada register WGM1 3 : 0 diset sebagai mode atau non PWM.

2. Bit 3 \_ FOCIA : Force Output Compare untuk Channel A

Bit 2 \_ FOCIB : Force Output Compare untuk Channel B

4. Bit 1:0 \_ WGM1 1:0: Waveform Generation Mode

Pengaturan timer I Juga Diatur Oleh Register TCCRIB, yang dapat dilihat pada Gambar 2.7 berikut ini.



Gambar 2.7 Register TCCRIB



Penjelasan:

- a. Bit 7\_ICNCI : Input Capture Noise Conceler
- b. Bit 6\_ICES I : Input Capture Edge Select
- c. Bit 5 Reserved bit
- d. Bit 4 : 3 \_ WGM II: 3 : Waveform Generation Mode
- e. Bit 2 : 0 : Clock Select

### 2.3 LED (Diode Penghasil Cahaya)

LED merupakan emitter cahaya semikonduktor yang memiliki sambungan p n yang mengeluarkan cahaya jika dibiaskan dalam arah biasa, yaitu positif ke jenis p (*anode*). Simbol sirkuitnya diperlihatkan oleh Gambar 2.8. Karena LED merupakan diode dengan sambungan p n, karakteristik elektrisnya sama dengan diode normal, yaitu mengkonduksi arus bila dibiaskan dalam arah bias maju dan menyekat aliran arus jika dibiaskan dalam arah terbalik, tetapi juga menghasilkan energi cahaya secara efisien jika dibiaskan dalam arah bias maju.



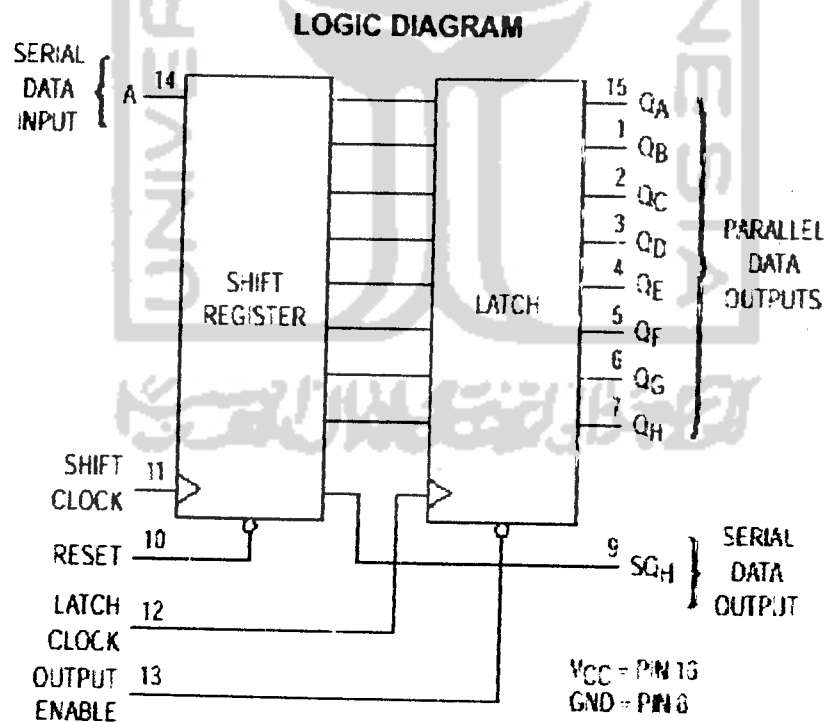
Gambar 2.8 Simbol untuk LED

Emitter cahaya semikonduktor dibuat dalam berbagai ukuran panjang gelombang, sehingga secara teoritis dapat diberi berbagai tanda warna. Warna

yang sangat sering digunakan adalah hijau, merah, kuning, dan jingga. Untuk menyalakan sebuah LED kecil diperlukan suatu R, dimana R terhubung dengan tegangan 1,5 Volt.

## 2.4 IC 74HC595A

Adalah sebuah IC dengan 8-bit masukan seri dan 8-bit keluaran paralel, yang terdiri dari sebuah 8-bit shift register dan 8-bit latch tipe D. Jadi IC ini menerima data serial dan mengeluarkan data paralel. Shift register dan latch mempunyai masukan clock yang terpisah serta memiliki reset tak serempak untuk shift register.



Gambar 2.9 Logic diagram IC 74HC595A

Konfigurasi pin IC 74HC595A :

1. Pin 14 (A)

Masukan data serial, data pada pin ini akan digeser ke dalam 8-bit register serial.

2. Pin 11 (Shift Clock)

Masukan clock untuk shift register, perubahan dari "low" ke "high" membuat data pada pin masukan data serial akan digeser ke shift register.

3. Pin 10 (Reset)

Masukan "low" akan membuat shift register reset.

4. Pin 12 (Latch Clock)

Masukan clock untuk latch, perubahan dari "low" ke "high" akan menyimpan data ke dalam register.

5. Pin 13 (Output Enable)

Masukan "low" akan membuat data yang tersimpan dalam latch akan ditampilkan melalui pin-pin output.

6. Pin 15,1,2,3,4,5,6,7 (QA – QH)

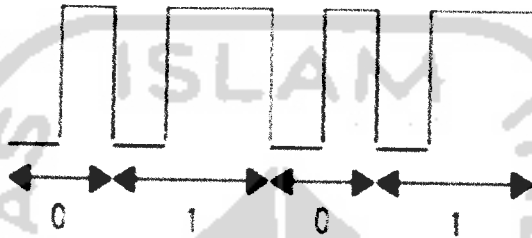
Keluaran data paralel.

7. Pin 9 (Serial Data Output)

Keluaran dari data yang tersimpan dalam shift register.

## 2.5 Remote Sony

Remote *control* adalah alternatif untuk memecahkan masalah ini. Remote *control* yang dibahas saat ini adalah remote *control* merk Sony, di mana remote *control* ini sudah mempunyai standard bentuk data yang mudah diperoleh.



Gambar 2.10 Teknik pengkodean pulsa

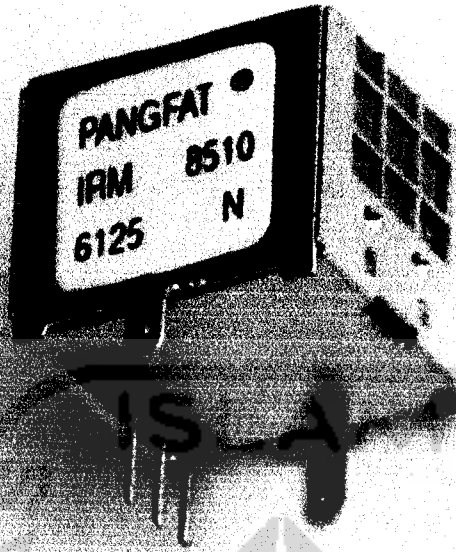
Gambar 2.10 menunjukkan teknik peng-kode-an data dari remote kontrol Sony, di mana dalam hal ini kode 0 diwakili oleh sebuah pulsa positif yang pendek dan kode 1 diwakili oleh sebuah pulsa positif yang panjang. Pada gambar tersebut, tampak data 5 (0101) ditransfer secara serial dengan teknik peng-kode-an secara pulsa.

Aplikasi ini sebetulnya tidak memerlukan pemancar infrared, karena bagian pemancar infrared sudah digantikan oleh remote kontrol Sony. Modulasi dilakukan dengan frekuensi *carrier* sebesar 30-40 KHz akan membentuk pulsa negatif dan tidak adanya frekuensi *carrier* akan membentuk pulsa positif.

## 2.6 *Infra Red Receiver*

Sinar infra merah yang dipancarkan oleh pemancar infra merah tentunya mempunyai aturan tertentu agar data yang dipancarkan dapat diterima dengan baik di *receiver*. Oleh karena itu baik di *transmitter* infra merah maupun *receiver* infra merah harus mempunyai aturan yang sama dalam mentransmisikan (bagian *transmitter*) dan menerima sinyal tersebut kemudian mendekodekannya kembali menjadi data biner (bagian *receiver*).

Komponen yang dapat menerima infra merah ini merupakan komponen yang peka cahaya yang dapat berupa dioda (photodiode) atau transistor (phototransistor). Komponen ini akan merubah energi cahaya, dalam hal ini energi cahaya infra merah, menjadi pulsa-pulsa sinyal listrik. Komponen ini harus mampu mengumpulkan sinyal infra merah sebanyak mungkin sehingga pulsa-pulsa sinyal listrik yang dihasilkan kualitasnya cukup baik. Semakin besar intensitas infra merah yang diterima maka sinyal pulsa listrik yang dihasilkan akan baik jika sinyal infra merah yang diterima intensitasnya lemah maka infra merah tersebut harus mempunyai pengumpul cahaya (*light collector*) yang cukup baik dan sinyal pulsa yang dihasilkan oleh sensor infra merah ini harus dikuatkan. Pada prakteknya sinyal infra merah yang diterima intensitasnya sangat kecil sehingga perlu dikuatkan. Selain itu agar tidak terganggu oleh sinyal cahaya lain maka sinyal listrik yang dihasilkan oleh sensor infra merah harus difilter pada frekuensi sinyal *carrier* yaitu pada 30KHz sampai 40KHz. Selanjutnya baik photodiode maupun phototransistor disebut sebagai photodetector.

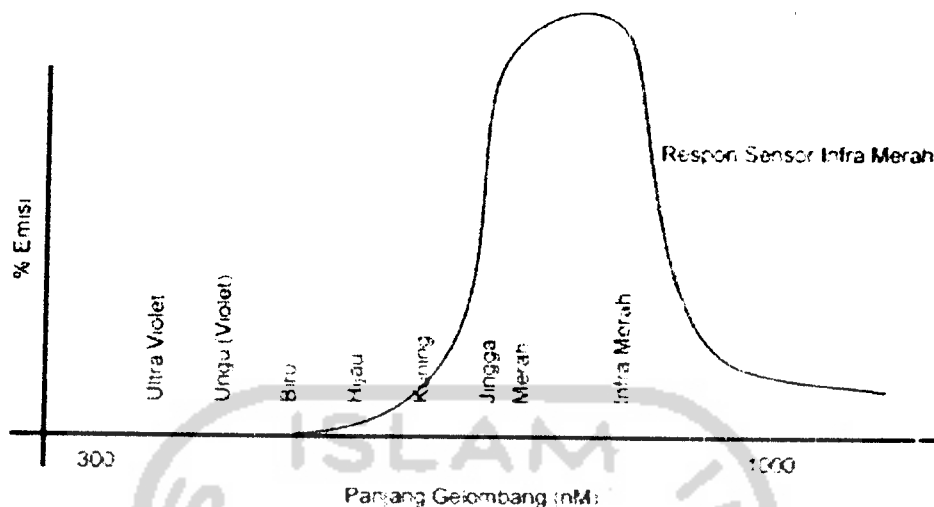


Gambar 2.11 Tampilan sensor infrared



Gambar 2.12 Tampilan rangkaian sensor infrared

Dalam penerimaan infra merah, sinyal ini merupakan sinyal infra merah yang termodulasi. Pemodulasian sinyal data dengan sinyal *carrier* dengan frekuensi tertentu akan dapat memperjauh transmisi data sinyal infra merah.



Gambar 2.13 Respon penerimaan sensor infrared

Komponen photodetector mempunyai karakteristik seperti komponen yang dinamakan 'solar cell', yang merubah energi cahaya menjadi energi listrik. Jika photo detector ini mendapat cahaya maka akan menghasilkan tegangan sekitar 0.5 volt dan arus yang dihasilkan tergantung dari intensitas cahaya yang masuk pada photo detector tersebut. Teknik ini biasa disebut sebagai '*unbiased current sourcing*' atau '*photovoltaic mode*'. Teknik ini jarang digunakan karena tidak efisien dan mempunyai respon yang lambat terhadap pulsa-pulsa cepat sinyal cahaya.

Konfigurasi photo detector yang umum dipakai adalah teknik yang dikenal sebagai '*reverse biased*' atau '*photoconductive mode*'. Pada mode reverse bias/bias terbalik, photo detector dibias dengan tegangan external mulai dari beberapa volt sampai sekitar 50 volt (tergantung karakteristik photo detector). Jika karakteristik photodetector tidak diketahui maka bias tegangan dapat diberi 12V agar tidak merusak photodetector tersebut. Photo detector ini mendapat cahaya, dalam hal ini cahaya infra

merah maka terdapat arus bocor yang relatif kecil. Besar-kecilnya arus bocor ini tergantung dari intensitas cahaya infra merah yang mengenai photodetector tersebut.

Sebuah photodiode, biasanya mempunyai karakteristik yang lebih baik daripada phototransistor dalam responya terhadap cahaya infra merah. Biasanya photodiode mempunyai respon 100 kali lebih cepat daripada phototransistor. Oleh sebab itulah para designer cenderung menggunakan photodiode daripada menggunakan phototransistor, tetapi sebuah phototransistor tetap mempunyai keunggulan yaitu mempunyai kemampuan untuk menguatkan arus bocor menjadi ratusan kali jika dibandingkan dengan photodiode.

Sebuah photodiode biasanya dikemas dengan plastik transparan yang juga berfungsi sebagai lensa fresnel. Lensa ini merupakan lensa cembung yang mempunyai sifat mengumpulkan cahaya. Lensa tersebut juga merupakan filter cahaya, lebih dikenal sebagai '*optical filter*', yang hanya melewatkan cahaya infra merah saja. Walaupun demikian cahaya yang nampakpun masih bisa mengganggu kerja dari diode infra merah karena tidak semua cahaya nampak bisa difilter dengan baik. Oleh karena itu sebuah penerima infra merah harus mempunyai filter kedua yaitu rangkaian filter yang berfungsi untuk memfilter sinyal 30KHz sampai 40KHz saja.

Faktor lain yang juga berpengaruh pada kemampuan penerima infra merah adalah '*active area*' dan '*respond time*'. Semakin besar area penerimaan suatu diode infra merah maka semakin besar pula intensitas cahaya yang dikumpulkannya sehingga arus bocor yang diharapkan pada teknik '*reserved bias*' semakin besar. Selain itu semakin besar area penerimaan maka sudut penerimaannya juga semakin



besar. Kelemahan area penerimaan yang semakin besar ini adalah noise yang dihasilkan juga semakin besar pula. Respon terhadap frekuensi, semakin besar area penerimaannya maka respon frekuensinya turun dan sebaliknya jika area penerimaannya kecil maka respon terhadap sinyal frekuensi tinggi cukup baik.

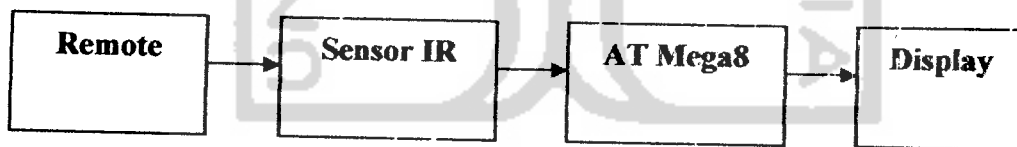
*Respond time* dari suatu dioda infra merah (penerima) mempunyai waktu respon yang biasanya dalam satuan nano detik. *Respond time* ini mendefinisikan lama agar dioda penerima infra merah merespon cahaya infra merah yang datang pada area penerima. Sebuah dioda penerima infra merah yang baik paling tidak mempunyai *respond time* sebesar 500 nano detik atau kurang. Jika *respond time* terlalu besar maka dioda infra merah ini tidak dapat merespon sinyal cahaya yang dimodulasi dengan sinyal carrier frekuensi tinggi dengan baik. Hal ini akan mengakibatkan adanya *data loss*.

## BAB III

### PERANCANGAN SISTEM

#### 3.1 Perancangan Sistem

Pada bab ini akan dijelaskan tentang perangkat keras dan perangkat lunak yang digunakan dalam perancangan pencatat skor dengan kendali remote. Pada perancangan perangkat keras, perancangan akan dimulai dari sistem minimum mikrokontroler yang merupakan pusat pengendali dari seluruh sistem, kemudian diikuti perancangan pada masing-masing bagiannya. Sedangkan pada perancangan perangkat lunak perancangan akan ditekankan pada bagian utama yang digunakan untuk menjalankan sistem yang dibuat. Secara umum diagram blok rancang bangun pencatat skor dengan kendali remote seperti pada Gambar 3.1.



Gambar 3.1 Blok diagram pencatat skor dengan kendali remote

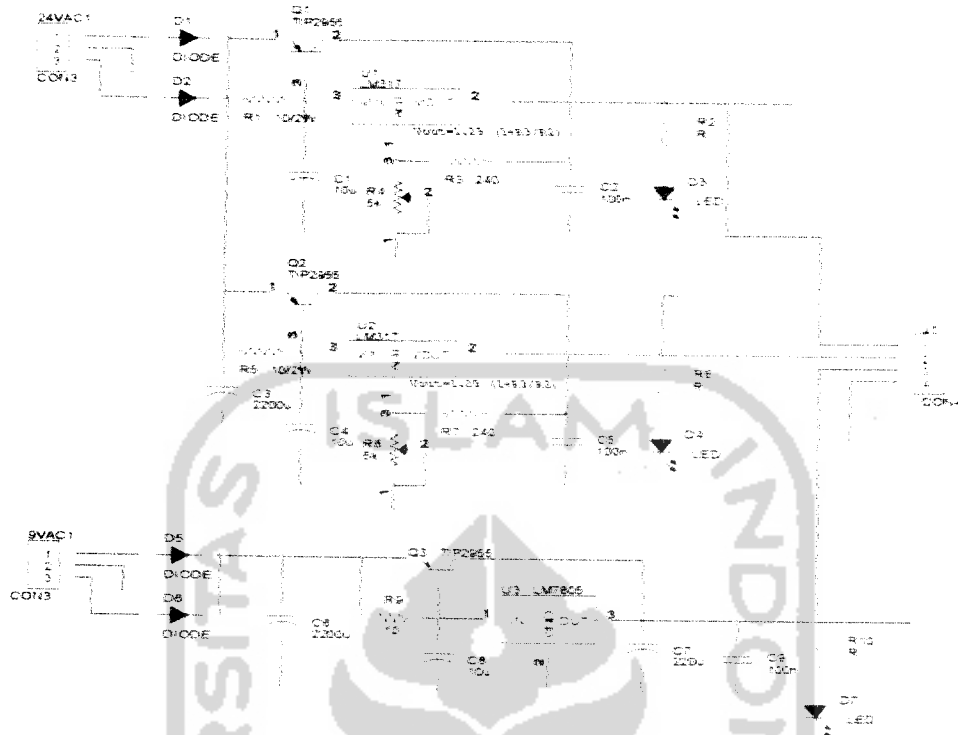
Alat pencatat skor dengan kendali remote ini menggunakan inframerah dari remote sebagai masukan dan IR *receiver* sebagai penerima inframerah. Mikrokontroler AT Mega8 merupakan pengontrol yang mengatur proses kerja secara keseluruhan baik mulai dari masukan/input mengolah data yang masuk sampai ke output/keluaran display yang berupa led maupun dot matrik.

## 3.2 Perancangan Perangkat Keras

### 3.2.1 Power Supply

Rangkaian catu daya merupakan bagian yang sangat penting pada rangkaian karena tanpa catu daya alat ini tidak dapat bekerja, catu daya digunakan sebagai penyedia sumber tegangan untuk keseluruhan sistem. Sehingga dapat mempertahankan suatu level tegangan yang konstan yang sangat diperlukan dalam rangkaian catu daya. Dengan demikian rangkaian catu daya pada tugas akhir ini menggunakan regulator tegangan (*voltage regulator*) yang berbentuk IC (*integrated circuit*) yang mengandung sejumlah rangkaian untuk tegangan referensi, alat pengontrol, komparator, dan pelindung tegangan berlebih (*overload protection*). Jenis regulasi yang di gunakan seri 78XX, yaitu 7805. Komponen ini biasanya sudah dilengkapi dengan pembatas arus (*current limiter*) dan juga pembatas suhu (*thermal shutdown*). Rangkaian catu daya yang digunakan pada alat ini tampak seperti pada Gambar 3.2 di bawah ini, dimana dioda berfungsi sebagai penyearah pada bagian *rectifier* yang kemudian arus diperkuat oleh TIP2955.

Rangkaian dalam sistem menggunakan tiga buah catu daya, dikarenakan pemakaian dalam perancangan membutuhkan tegangan 9 V untuk membangkitkan regulator LM7805 yang berfungsi untuk menghasilkan tegangan 5 volt sebagai masukan tegangan mikrokontroler, tegangan  $\pm 21$  V untuk membangkitkan led besar pada skor dan tegangan 10,5 V untuk membangkitkan led kecil pada dot matrix, babak, waktu.

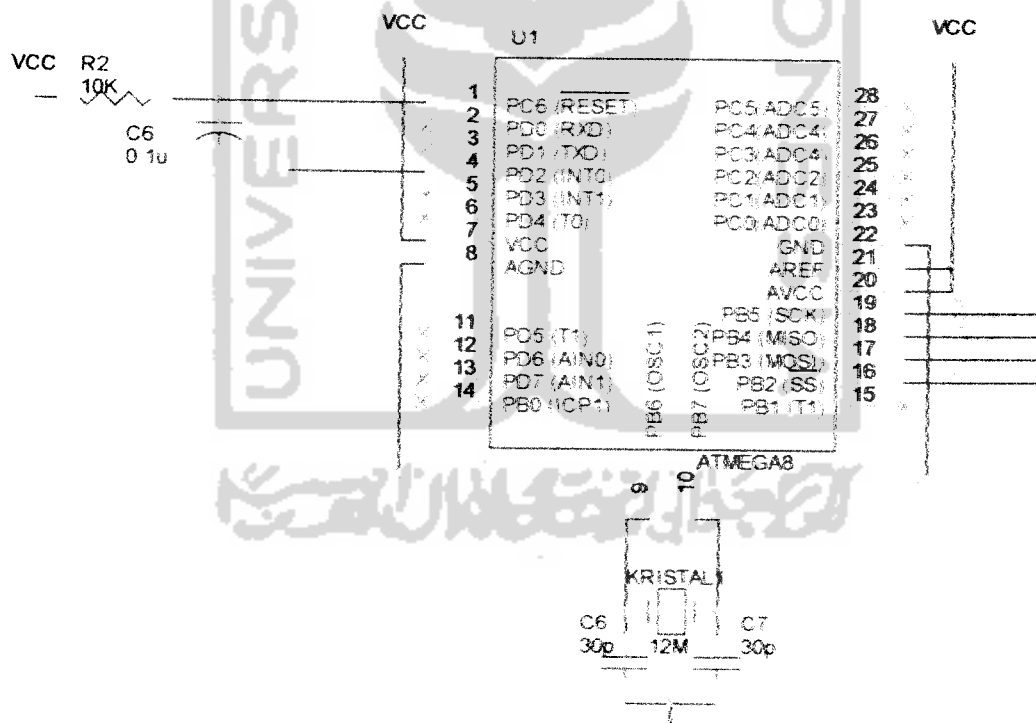


Gambar 3.2 Power Supply

### 3.2.2 Mikrokontroler AT Mega 8

Mikrokontroler AT Mega8 digunakan sebagai perangkat pengendali utama pada rangkaian ini. Mikrokontroler AT Mega8 ini membutuhkan dan memiliki 3 buah *port* yang dapat difungsikan sebagai *port-port* alternatif, sehingga perangkat-perangkat yang akan dikendalikan dapat langsung dihubungkan dengan *port-port* tersebut. Pada ke-3 *port* tersebut mempunyai fungsi yang berbeda-beda. Rangkaian osilator yang dipakai adalah kristal 12 MHz dan 2 buah kapasitor 30 pF dan rangkaian *reset* yang terdiri dari gabungan sebuah resistor dan kapasitor. Sedangkan rangkaian *reset* berfungsi untuk menjaga agar pin RST mikrokontroler selalu berlogika rendah saat

mikrokontroler mengeksekusi program. Mikrokontroler di *reset* pada transisi tegangan rendah ke tegangan tinggi, oleh karena itu pada pin RST dipasang kapasitor yang terhubung Vcc dan resistor ke *ground* yang menjaga RST bernilai 1 saat pengisian kapasitor dan bernilai 0 saat kapasitor penuh. Pada saat sumber tegangan diaktifkan kapasitor terhubung singkat sehingga arus mengalir dari Vcc langsung ke kaki RST sehingga *reset* berlogika 1, kemudian kapasitor terisi hingga tegangan pada kapasitor sama dengan Vcc pada saat kapasitor penuh. Dengan demikian tegangan reset akan turun menjadi 0 sehingga kaki RST berlogika 0.

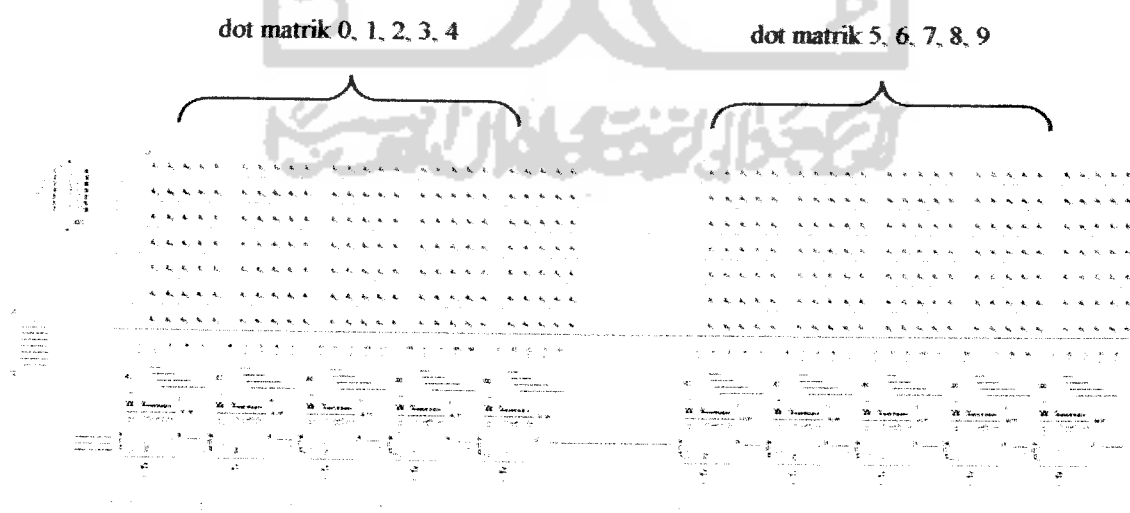


Gambar 3.3 Mikrokontroler AT Mega8

Seperti pada Gambar 3.3, VCC merupakan pin yang berfungsi sebagai masukan catu daya. GND yang merupakan pin ground. Port B (PB0 - PB7) merupakan pin I/O dua arah. Port C (PC0 - PC5) merupakan pin I/O dua arah pin masukan ADC. Port D (PD0 - P07) merupakan pin I/O dua arah. RESET merupakan pin yang digunakan untuk mereset mikrokontroler. XTAL1 dan XTAL2 merupakan pin masukan clock eksternal. AVCC merupakan pin masukan untuk ADC.

### 3.2.3 Dot Matrik

Pada dot matrik ini data masukan yang diterima dari mikrokontroler diterima oleh semua IC 74HC595A kemudian saat masukan data memilih diantara salah satu dari dot matrik (0 - 9) maka dot matrik yang dipilih akan menyala sesuai dengan perintah atau masukan yang diterima dari mikrokontroler dan IC ULN2003 sebagai *driver* untuk memilih segment. Gambar 3.4 ini merupakan gambar rangkaian dot matrik.

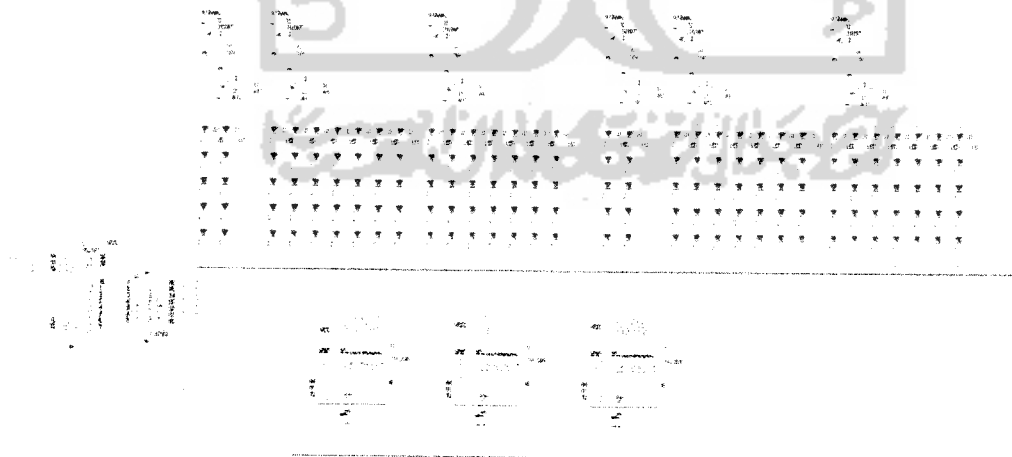


Gambar 3.4 Dot matrik penampil nama team

### 3.2.4 Seven Segment

Pembuatan seven segment ini menggunakan lampu LED yang disusun selayaknya sebuah seven segment. Untuk LED kecil membutuhkan arus 15 mA dan tegangan 1,5 V. Sedangkan LED besar membutuhkan arus 15mA dan tegangan 3 V.

Di dalam seven segment ini terdapat dua jenis IC diantaranya ULN2003 yang berfungsi sebagai driver led dan IC74HC595A menerima data serial dan mengeluarkan data paralel. Data masukan yang diterima dari mikrokontroler akan diterima oleh IC 74HC595A yang akan menampilkan skor dari dua tim pada saat waktu sudah berjalan. Pada rangkaian ini dipasang suatu rangkaian tambahan yang berfungsi untuk menyalakan setiap satu seven segment. Gambar 3.5 menunjukkan seven segment sebagai penampil skor. Untuk menampilkan waktu pertandingan gambar rangkaiannya dapat ditunjukkan pada Gambar 3.6. Pada Gambar 3.4 dan Gambar 3.6 IC74HC595A dipasang pada setiap seven segment yang berfungsi untuk merubah data serial ke paralel.



Gambar 3.5 Seven segment sebagai penampil skor



Gambar 3.6 Seven segment sebagai penampil waktu dan babak

### 3.3 Perancangan Perangkat Lunak

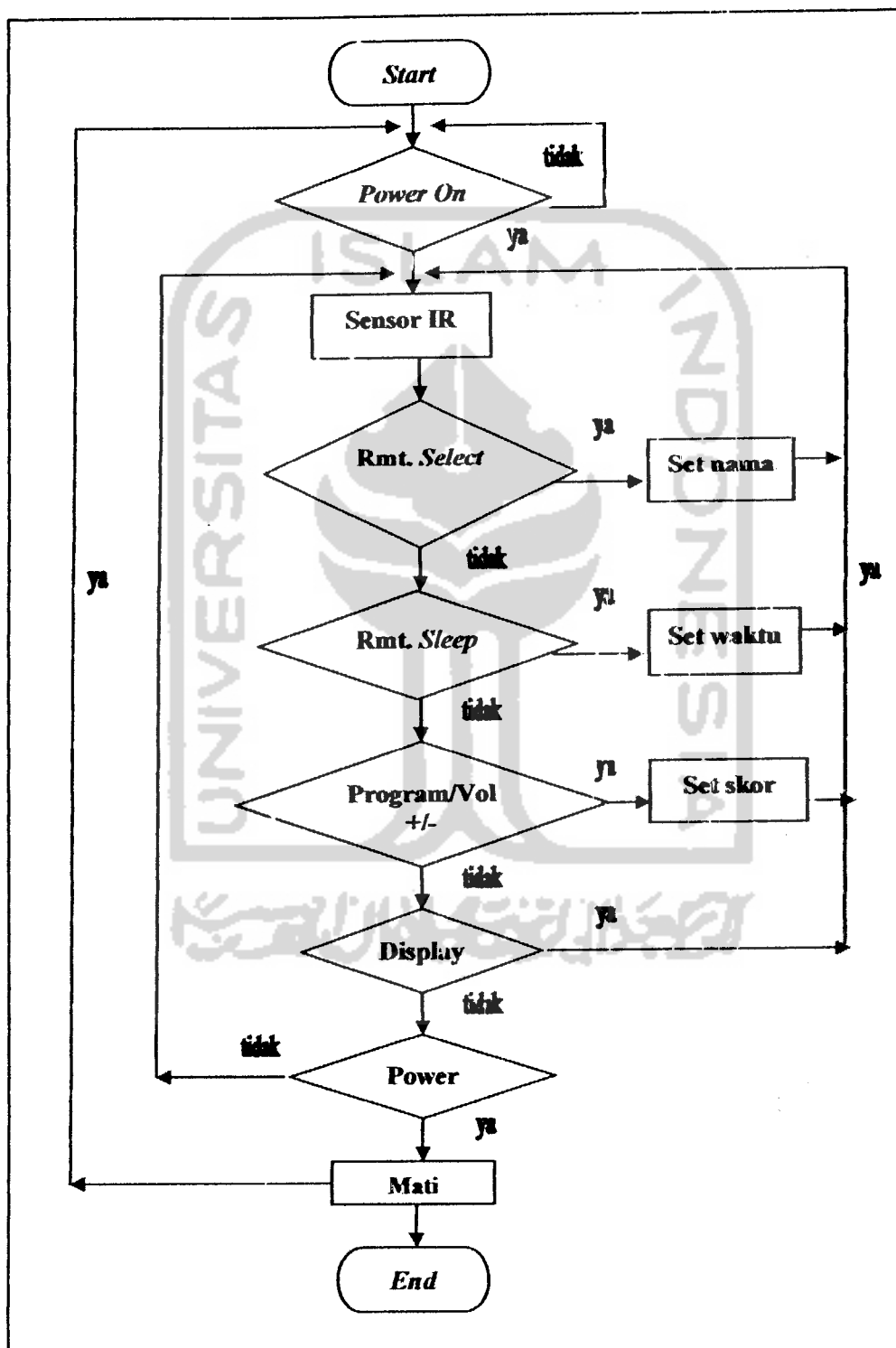
Dalam perancangan perangkat lunak dibahas tentang perancangan dan implementasi diagram alir yang meliputi inisialisasi program. Bahasa pemrograman yang biasa digunakan adalah bahasa C.

*Flowchart* program Mikrokontroler ATmega8 dapat dilihat pada Gambar 3.7, dan *listing program* dapat dilihat pada halaman lampiran.

Dari Gambar 3.7 dapat dijelaskan bahwa papan skor ketika terhubung dengan catu daya maka sensor IR siap menerima masukan data dari remote. Ketika power on dan jika remote memilih select berarti memilih untuk menyeting nama, apabila tidak akan diteruskan ke remote sleep jika ya maka akan memilih untuk menyeting waktu, apabila tidak akan diteruskan ke program atau volume +/-, jika ya berarti memilih untuk menyeting skor. Apabila tidak akan diteruskan ke display yang mana display akan menjalankan semua settingan yang telah dimasukkan remote. Apabila papan skor



masih digunakan dapat menyeting kembali seperti semula, jika diberi input power off maka papan skor tidak akan berfungsi atau mati.



Gambar 3.7 Flowchart program Mikrokontroler ATmega8

## **BAB IV**

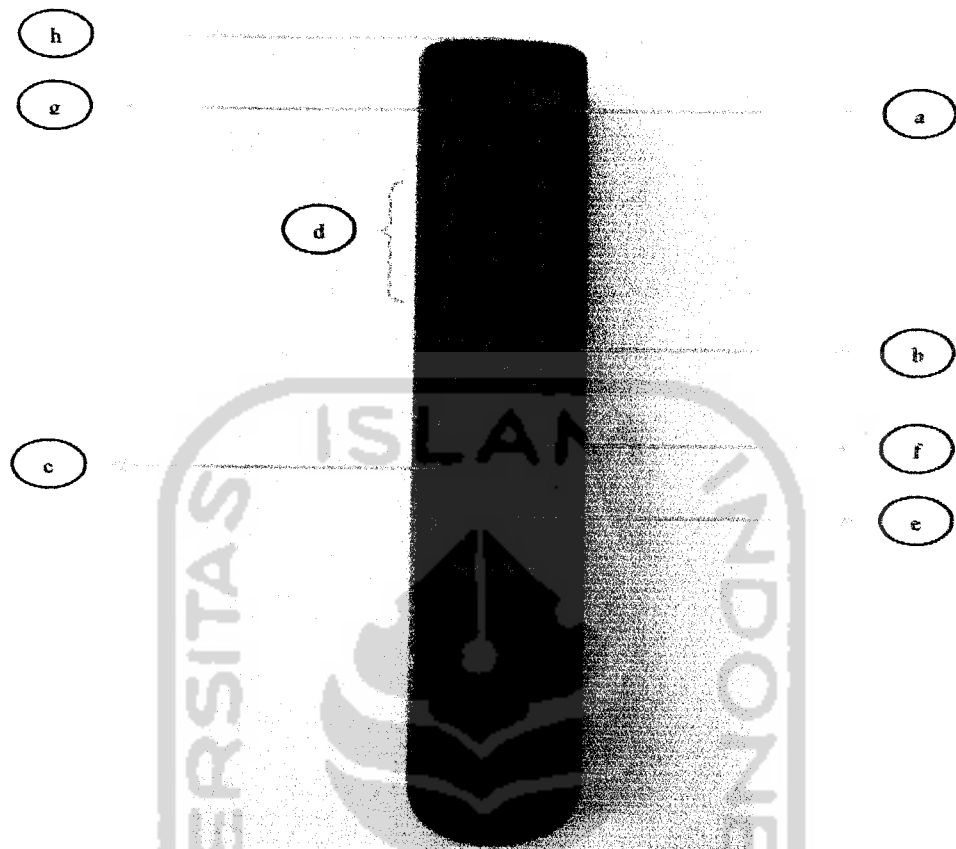
### **ANALISA DAN PEMBAHASAN**

Tugas akhir ini ditujukan untuk mencatat skor dengan kendali remote. Pengamatan dilakukan untuk menguji hasil perancangan dan implementasi alat sehingga dapat diketahui sejauh mana alat dapat bekerja selain itu dalam pengamatan juga akan dibahas mengenai beberapa kendala yang terjadi. Dengan mendapatkan parameter hasil pengujian tersebut dapat disimpulkan rangkaian secara keseluruhan dan cara kerja alat dapat diketahui. Pengamatan dan pengukuran pada bab ini dilakukan pada beberapa bagian sebagai berikut :

1. Pengamatan sensor infra merah.
2. Pengamatan dot matrik sebagai penampil nama *team* yang bertanding.
3. Pengamatan seven segment sebagai penampil skor, waktu pertandingan, dan babak.

#### **4.1 Pengamatan sensor infra merah**

Pencatat skor dengan kendali remote ini hanya dapat menggunakan remote merk sony, karena remote dengan merk sony merupakan remote yang paling mudah penkodeannya.



Gambar 4.1 Remote merk sony

Keterangan Gambar 4.1

a. Tombol power

Tombol ini berfungsi untuk menghidupkan dan mematikan papan skor.

b. Tombol select

Tombol ini berfungsi untuk menyeting nama team pada dot matrik

c. Tombol sleep

Tombol ini berfungsi untuk menyeting waktu.

d. Tombol angka

Tombol ini berfungsi untuk menyeting urutan dotmatrik mana yang akan

digunakan, menyeting lamanya pertandingan dan babak pertandingan.

e. Tombol volume

Tombol ini berfungsi untuk menyeting skor team 1 (sebelah kiri) dan pada setting waktu volume up digunakan sebagai enter dan volume down sebagai clear.

f. Tombol program

Tombol ini berfungsi untuk menyeting skor team 2 (sebelah kanan) dan pada setting dot matrik digunakan untuk mengganti huruf/angka.

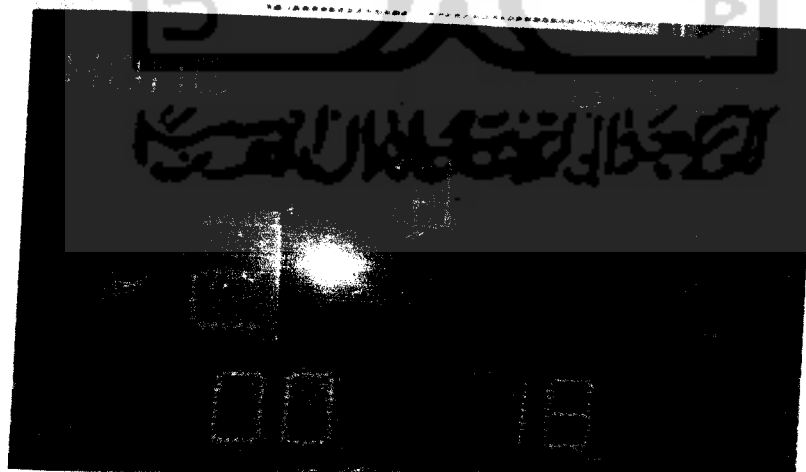
g. Tombol mute

Tombol ini berfungsi sebagai pause/menghentikan waktu pertandingan yang bersifat hanya sementara.

h. Tombol display

Tombol ini berfungsi sebagai tombol start.

#### 4.2 Pengamatan dot matrik sebagai penampil nama team yang bertanding



Gambar 4.2 Tampilan keseluruhan papan skor



Gambar 4.3 Tampilan nama kedua team yang bertanding

Gambar 4.3 diatas, 1) dan 2) Merupakan tampilan dari nama team yang bertanding, sedangkan cara untuk menampilkan nama team tersebut diatas adalah sebagai berikut ini :

1. Tekan tombol "power" untuk menyalakan papan skor.
2. Tekan tombol "select", lalu tekan tombol 0 s.d 9 untuk memilih salah satu dot matrik yang merupakan satu dot matrik mewakili satu huruf. Untuk merubah huruf atau angka pada dot matrik menggunakan "program up atau down". Misalkan pada Gambar 4.2 nama team **Home**, maka pada dot matrik 0, kita set dengan menekan tombol "program up atau down" hingga muncul huruf H. Kemudian tekan '1' untuk ke dot matrik berikutnya, tekan tombol "program up atau down" hingga muncul huruf O. Selanjutnya tekan '2' untuk ke dot matrik berikutnya, tekan tombol "program up atau down" hingga muncul huruf M. Selanjutnya tekan '3' untuk ke dot matrik berikutnya, tekan tombol "program up atau down" hingga muncul huruf E.
3. Untuk menampilkan nama team yang kedua, caranya sama seperti dengan menampilkan nama team pertama, hanya pada dot matrik nama team kedua dimulai dari tombol 5.

**4.3 Pengamatan seven segment sebagai penampil skor, waktu pertandingan, babak.**

**4.3.1 Pengamatan seven segment sebagai penampil skor**

Skor hanya dapat di set setelah waktu pertandingan berjalan”(*display*)”, pada Gambar 4.4, 4) merupakan skor dari kedua team yang bertanding. Cara untuk menampilkan skor adalah sebagai berikut :

1. Ketika waktu berjalan”(*display*)” yang hanya bisa dikendalikan oleh remote hanya tombol ”*program up down*” yang berfungsi sebagai pengendali skor team 1. Sedangkan untuk team kedua digunakan tombol ”*volume up down*” yang berfungsi sebagai pengendali skor. Pada Gambar 4.4 dibawah ini dicontohkan kedua team memiliki skor 2-2.



Gambar 4.4 Tampilan skor kedua team yang bertanding

**4.3.2 Pengamatan seven segment sebagai penampil waktu**

Penampil waktu dibagi menjadi dua, yaitu 6)menit dan 7)detik seperti pada Gambar 4.5. Cara menyeting waktu adalah sebagai berikut :

1. Tekan "sleep" hingga penampil menit berkedip-kedip.
2. Untuk memasukkan waktu menggunakan tombol 0 s.d 9, misalkan waktu diset 15 menit, maka tekan tombol 1 dan 5 lalu tekan "vol up" sebagai *enter*.
3. Setelah ditekan tombol *enter* secara otomatis tampilan waktu akan berpindah ke tampilan detik.
4. Apabila waktu akan diubah, tekan tombol "vol down" sebagai *clear*.

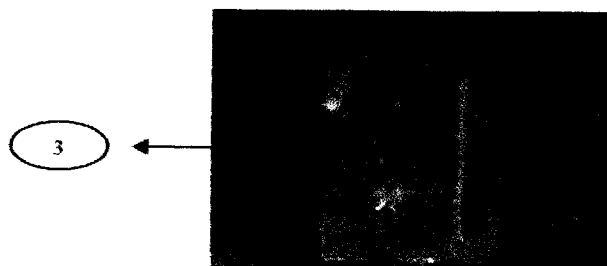


Gambar 4.5 Tampilan waktu pertandingan

#### 4.3.3 Pengamatan seven segment sebagai penampil babak

Pada Gambar 4.6 ditunjukkan pada nomor 3), untuk menyeting babak dapat dilakukan sebelum waktu berjalan "(display)", cara menyeting babak adalah sebagai berikut :

1. Set dulu lama waktu pertandingan yang diinginkan.
2. Untuk mengganti babak tekan tombol 1 atau 2 atau 3.
3. Ketika waktu pertandingan babak pertama sudah selesai babak pertandingan selanjutnya baru dapat diganti.

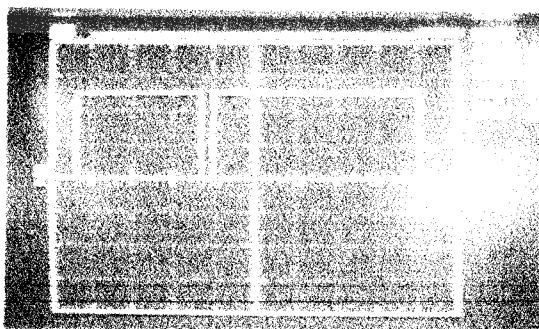


Gambar 4.6 Tampilan babak pertandingan

#### 4.4 Bentuk Gelombang Sensor Infra Merah

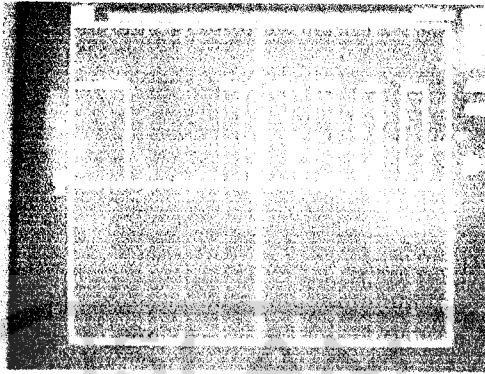
Pada penelitian kali ini akan dilihat *output* dari penerima infra merah. Pengamatan dilakukan dengan menggunakan *oscilloskop* karena gelombang yang akan diamati berupa gelombang digital. Cara mengamatinya adalah dengan menghubungkan tegangan sumber 5 Volt pada VCC di rangkaian infra merah, ground dihubungkan dengan ground pada rangkaian infra merah. Kemudian pada *oscilloskop* diatur nilai volt/div-nya sebesar 2 volt/div agar bentuk gelombang yang diamati dapat terlihat dengan jelas.

Dari penelitian dan pengamatan yang sudah dilakukan pada output infra merah akan diperoleh gambar gelombang sebagai berikut, lihat Gambar 4.7 sampai dengan Gambar 4.7.

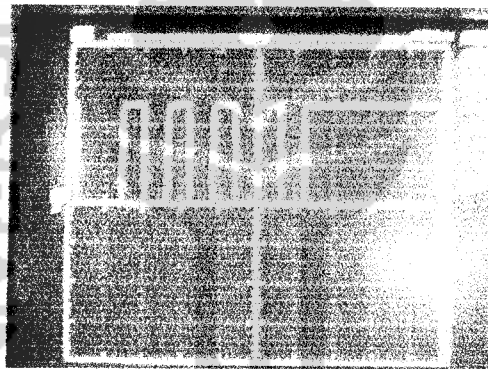


Gambar 4.7 Output gelombang infra merah pada saat kondisi normal

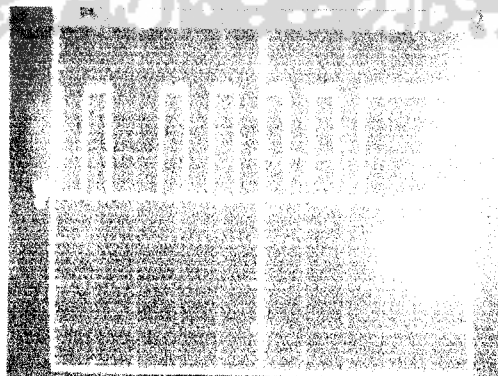




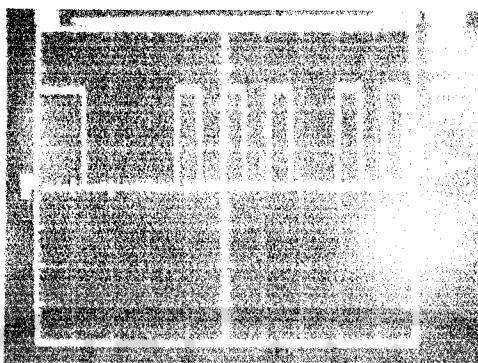
Gambar 4.8 Output gelombang infra merah pada saat tombol remote ditekan angka 1



Gambar 4.9 Output gelombang infra merah pada saat tombol remote ditekan angka 2



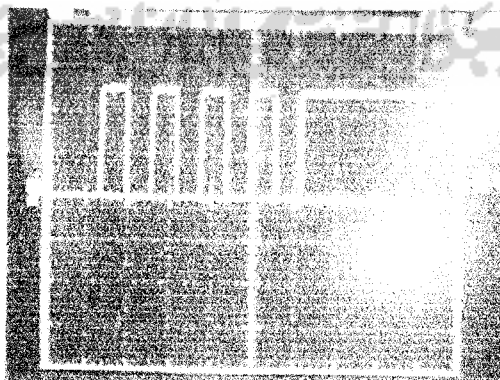
Gambar 4.10 Output gelombang infra merah pada saat tombol remote ditekan angka 4



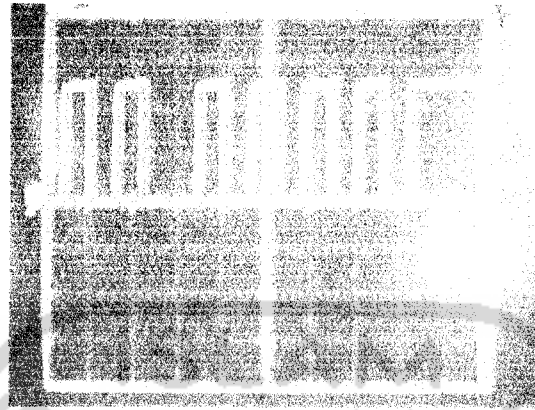
Gambar 4.11 Output gelombang infra merah pada saat tombol remote ditekan angka 5



Gambar 4.12 Output gelombang infra merah pada saat tombol remote ditekan angka 6

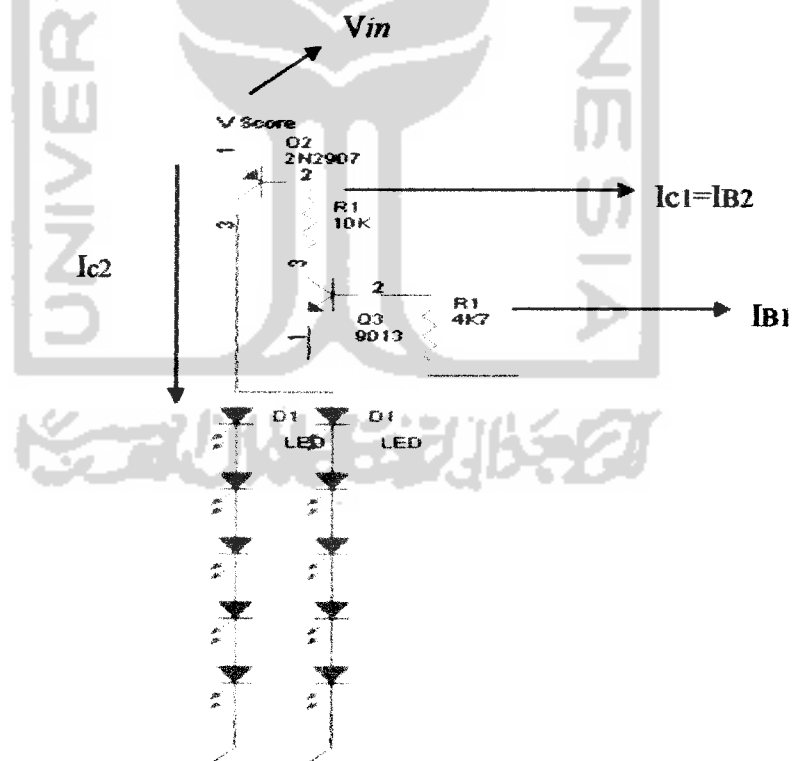


Gambar 4.13 Output gelombang infra merah pada saat tombol remote ditekan angka 9



Gambar 4.14 Output gelombang infra merah pada saat tombol remote ditekan angka 0

#### 4.5 Pembahasan Rangkaian Transistor Sebagai Saklar



Gambar 4.15 Rangkaian transistor sebagai saklar

Rangkaian diatas digunakan transistor tipe 2N907 dan tipe 9013 yang berfungsi sebagai saklar untuk menghidupkan atau memilih masing – masing digit. Transistor tipe 2N907 yang digunakan adalah jenis transistor pnp yang mana mampu dilewati arus maksimal 1A, sedangkan transistor tipe 9013 merupakan jenis npn. Untuk menghidupkan satu led besar membutuhkan 3V, jadi LED pada papan skor membutuhkan tegangan 21V untuk setiap digit. Untuk LED kecil satu led membutuhkan 1.5V jadi total tegangan yang dibutuhkan satu digit LED kecil sebesar 10.5V. Rangkaian LED yang tersusun seri memiliki arus yang sama namun tegangan yang dibutuhkan 2 kali lebih besar. Sedangkan untuk LED yang tersusun paralel memiliki tegangan yang sama namun membutuhkan arus 2 kali lebih besar.

Transistor pada rangkaian diatas masing-masing memiliki tegangan 0,7V, sedangkan tegangan pada  $V_{in}$  sebesar 5V. Arus yang masuk ke LED ( $I_{c2}$ ) sebesar 15 mA, sedangkan tegangan yang melewati Q2 sebesar 4,3V yang kemudian dilewatkan pada R1 sebesar 10K, maka :

$$I_{c1} = \frac{4,3}{10K} \dots\dots\dots (4.1)$$

$$I_{B1} = \frac{V_{in} - 0,7}{4K7} \dots\dots\dots (4.2)$$

## **BAB V**

### **PENUTUP**

Berdasarkan proses perancangan, pembuatan, pengamatan dan hasil uji alat yang dibuat, maka dapat diambil beberapa kesimpulan dan saran.

#### **5.1 Kesimpulan**

Dari proses perancangan, analisis dan pengujian dapat diambil beberapa kesimpulan sebagai berikut :

1. Remote yang dapat digunakan hanyalah remote ber-merk sony, karena pengkodeannya lebih mudah.
2. Kemampuan maksimal remote hanya berjarak 1-2m
3. Sensor pada remote terkadang menjadi kendala pada alat ini
4. Hasil dari penelitian ini berupa alat yang dapat menampilkan skor, waktu, nama team yang bertanding dan babak pertandingan yang dapat diimplementasikan pada suatau gedung olahraga.

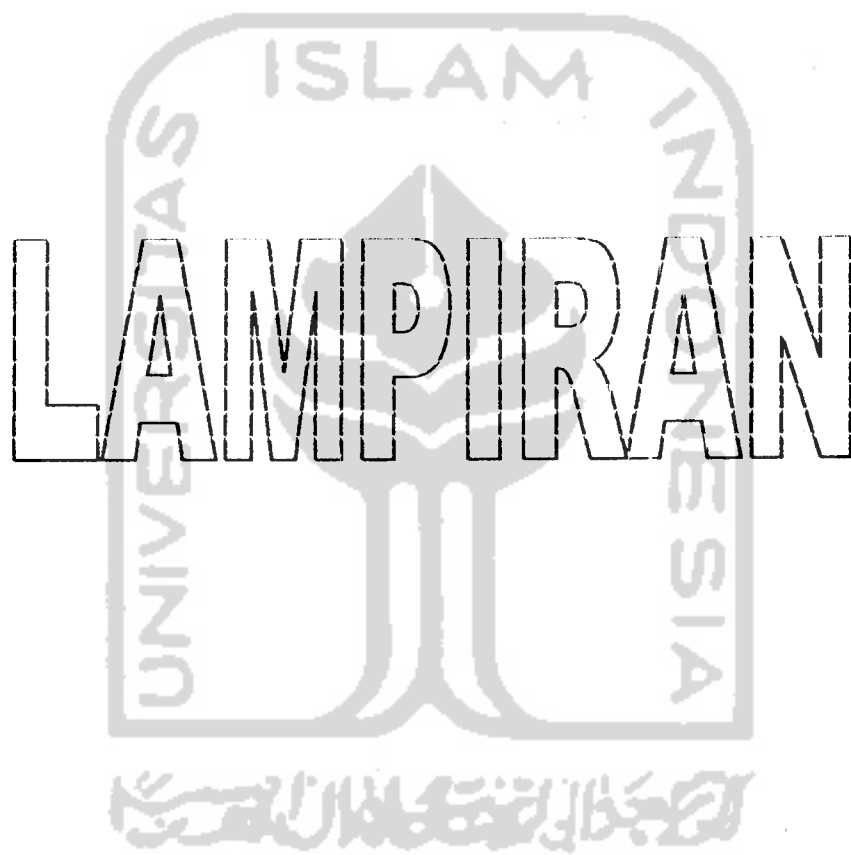
#### **5.2 Saran**

Adapun saran yang diberikan untuk menyempurnakan alat ini adalah sebagai berikut :

1. Kedepannya diharapkan alat ini dapat dikembangkan dengan menambahkan suara dan saat waktu pertandingan berakhir ada alarm yang berbunyi.

2. Untuk penggunaan remote sebagai kendalinya, diharapkan remote lebih simple penggunaannya.





```

/*****
This program was produced by the
CodewizardAVR V1.24.4 Standard
Automatic Program Generator
© Copyright 1998-2004 Pavel Haiduc, HP InfoTech s.r.l.
http://www.hpinfotech.com
e-mail:office@hpinfotech.com

```

```

Project :
Version :
Date    : 04/06/2007
Author  : Radit
Company : Teknik Elektro UII
Comments:

```

```

Chip type      : ATmega8
Program type   : Application
Clock frequency : 12,000000 MHz
Memory model   : Small
External SRAM size : 0
Data Stack size : 256
*****/

```

```

#include <mega8.h>
#include <delay.h>
#include <spi.h>

```

```

#define running      0
#define edit_group   1
#define edit_babak   2
#define edit_menit   3
#define edit_detik    4
#define edit_time     5
#define pause        6
#define standby      7
#define finish       8

```

```

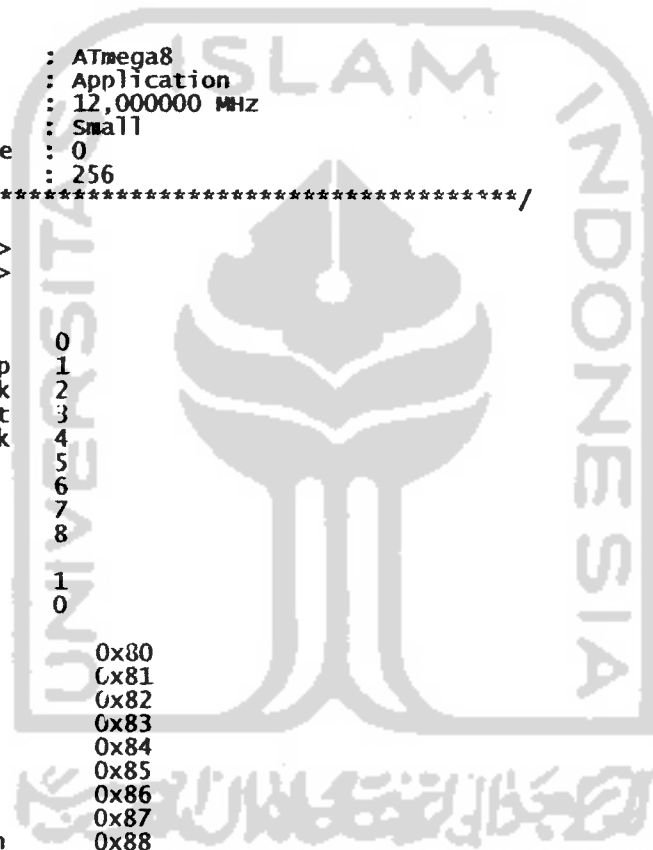
#define group_a      1
#define group_b      0

```

```

#define btSatu       0x80
#define btDua        0x81
#define btTiga       0x82
#define btEmpat      0x83
#define btLima       0x84
#define btEnam       0x85
#define btTujuh      0x86
#define btDelapan    0x87
#define btSembilan   0x88
#define btNo1        0x89
#define btBelasan    0x8c
#define btDuaPuluhan 0x8d
#define btProgUp     0x90
#define btProgDw     0x91
#define btVolUp      0x92
#define btVolDw      0x93
#define btMute       0x94
#define btPower      0x95
#define btPicMode    0x96
#define btAB         0x97
#define btTVvideo    0xa5
#define btSleep      0xb6
#define btDisplay    0xba
#define btTambah     0xf4
#define btKurang     0xf5
#define btSelect     0xfc

```





```

flash unsigned char
char_number[11]={0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7d,0x07,0x7f,0x6f,0x00};
//r
flash unsigned char seg_slc[8]={1,2,4,8,16,32,64,128};
//
0 1 2 3 4 5 6 7
8 9 A
//11 12 13 14 15 16 17 18 19 20 21 22 23 24
25 26 27 28 29 30 31 32 33 34 35 36 37
flash unsigned char char_1[38]={0, 14, 4, 14, 31, 8, 31, 12,
4, 31, 14, 14,
15, 14, 7, 31, 31, 14, 17, 14, 28, 17, 1, 17,
17, 17, 14, 15, 14, 15, 30, 31, 17, 17, 17, 17,
17, 31, 0};
flash unsigned char char_2[38]={0, 17, 6, 17, 8, 12, 1,
2, 17, 17, 17,
10, 17, 17, 17, 9, 1, 1, 17, 17, 4, 8, 9, 1,
17, 27, 17, 17, 17, 17, 17, 17, 1, 4, 17, 17, 17, 17,
17, 16, 0};
flash unsigned char char_3[38]={0, 25, 4, 16, 4, 10, 15, 1,
16, 17, 17,
17, 17, 1, 17, 1, 1, 1, 17, 4, 8, 5, 1,
21, 19, 17, 17, 17, 17, 1, 4, 17, 17, 17, 17, 10,
10, 8, 0};
flash unsigned char char_4[38]={0, 21, 4, 8, 8, 9, 16, 15, 8,
14, 30,
17, 15, 1, 17, 15, 15, 29, 31, 4, 8, 3, 1,
21, 21, 17, 15, 17, 15, 14, 4, 17, 17, 21,
4, 4, 0};
flash unsigned char char_5[38]={0, 19, 4, 4, 15, 31, 16, 17,
4, 17, 16,
31, 17, 1, 17, 1, 1, 17, 17, 4, 8, 5, 1,
17, 25, 17, 1, 21, 5, 16, 4, 17, 17, 21,
10, 4, 2, 0};
flash unsigned char char_6[38]={0, 17, 4, 2, 17, 8, 17, 17,
4, 17, 8,
17, 17, 17, 9, 1, 1, 17, 17, 4, 9, 9, 1,
17, 17, 17, 17, 1, 9, 9, 16, 4, 17, 10, 21,
17, 4, 1, 0};
flash unsigned char char_7[38]={0, 14, 14, 31, 14, 8, 14, 14, 4,
14, 6,
17, 15, 14, 7, 31, 1, 30, 17, 14, 6, 17, 31, 17,
17, 14, 1, 22, 17, 15, 4, 14, 4, 10, 17,
4, 31, 31};

unsigned char buff_seg[11]={10,0,0, 10,0,0, 1, 0,0,0,0};
unsigned char buff_char[11]={18,25,23,15,0, 17,31,15,29,30,0};
unsigned char seg_matrik[70];
unsigned char score[2]={0,0};
unsigned char game_menit,game_detik,menit,detik,remote;

unsigned char flag_remote=0,seg_aktif,mode,power=0;
unsigned char countUpDw=1;

```

```

void time_display(unsigned char val_menit,unsigned char val_detik);
void char_display(char seg,char val);

```

```

// Timer 1 overflow interrupt service routine
interrupt [TIM1_OVF] void timer1_ovf_isr(void)
{
    static char i,j,c;
    TCNT1H=0xe9;
    TCNT1L=0x1c;

    if(i)
    {
        if(mode==edit_group)char_display(seg_aktif,37);
        else if(mode==edit_menit)time_display(100,game_detik);
    }
}

```

```

        else if(mode==edit_detik)time_display(game_menit,100);
        i=0;
    }
    else
    {
        if(mode==edit_group)char_display(seg_aktif,buff_char[seg_aktif]);
        if(mode==edit_menit ||
mode==edit_detik)time_display(game_menit,game_detik);
        i=1;
    }
    if(mode==running)
    {
        j++;
        if(j>1)
        {
            j=0;
            time_display(menit,detik);
            if(menit==game_menit && detik==game_detik)
            {
                mode=finish;
                goto fin1;
            }
            detik++;
            if(detik==60)
            {
                detik=0;
                menit++;
                if(menit==99)
                    menit=0;
            }
        }
    }
    }
    fin1:
}

// Timer 2 overflow interrupt service routine
interrupt [TIM2_OVF] void timer2_ovf_isr(void)
{
    static char m,n;
    int i;
    TCNT2=0xb0;

    PORTB.1=1;
    for(i=9;i>=0;i--)spi(seg_matrik[m+i*7]);
    spi(seg_slc[m]);

    spi(char_number[buff_seg[n]]);

    if(n<8){spi(seg_slc[n]);spi(0);}
    else{spi(0);spi(seg_slc[n-8]);}

    m++;if(m>7)m=0;
    n++;if(n>11)n=0;

    PORTB.2=1;
    PORTB.2=0;

    PORTB.1=0;
}

void time_display(unsigned char val_menit,unsigned char val_detik)
{
    buff_seg[7]=val_menit/10;
    buff_seg[8]=val_menit%10;
    buff_seg[9]=val_detik/10;
    buff_seg[10]=val_detik%10;
    if(val_menit==100)buff_seg[7]=buff_seg[8]=10;
    if(val_detik==100)buff_seg[9]=buff_seg[10]=10;
}

```

```

}

void score_display(unsigned char group,unsigned int val)
{
    group=group*3;
    if(val>99)buff_seg[3]=1;else buff_seg[group]=10;
    if(val>9)buff_seg[(unsigned char)(group+1)]=val/10%10;else
buff_seg[(unsigned char)(group+1)]=10;
    buff_seg[(unsigned char)(group+2)]=val%10;
}

void babak_display(unsigned char val)
{
    buff_seg[6]=val;
}

void char_display(char seg,char val)
{
    seg=seg*7;
    seg_matrik[(char)(0+seg)]=char_1[val];
    seg_matrik[(char)(1+seg)]=char_2[val];
    seg_matrik[(char)(2+seg)]=char_3[val];
    seg_matrik[(char)(3+seg)]=char_4[val];
    seg_matrik[(char)(4+seg)]=char_5[val];
    seg_matrik[(char)(5+seg)]=char_6[val];
    seg_matrik[(char)(6+seg)]=char_7[val];
}

void clear_display()
{
    char i;
    for(i=0;i<11;i++)
    {
        buff_seg[i]=10;
        char_display(i,0);
    }
}

void baca_remote()
{
    unsigned char i;

    delay_ms(200);
    remote=0;
    while(PIND.2)
    TCNT0=0x00;while(!PIND.2);
    if(TCNT0>20 && TCNT0<35)
    {
        for(i=0;i<8;i++)
        {
            TCNT0=0x00; while(PIND.2);
            if(TCNT0>30)break;
            TCNT0=0x00; while(!PIND.2);
            if(TCNT0>8)remote+=seg_slc[i];
            if(TCNT0>20){remote=0; break;}
        }
    }
}

void enter_nilai_menit()
{
    unsigned char inkey;
    unsigned char smtr=0;

    while(1)
    {
        time_display(game_menit,game_detik);
        delay_ms(300);
    }
}

```

```

        baca_remote();
        inkey=remote;
        if(inkey>=btSatu && inkey<=btNo1)
        {
            if(inkey==0x89)inkey=0x7f;
            inkey=inkey-0x7f;
            smtr = game_menit;
            game_menit=game_menit*10+inkey;
            if(game_menit>99)game_menit=smtr;
        }
        else if(inkey==btVolDw)game_menit=0;
        else if(inkey==btVolUp)break;
        else game_menit=game_menit;
    }
    delay_ms(300);
}

void enter_nilai_detik()
{
    unsigned char inkey;
    unsigned char smtr=0;
    while(1)
    {
        time_display(game_menit,game_detik);
        delay_ms(300);
        baca_remote();
        inkey=remote;
        if(inkey>=btSatu && inkey<=btNo1)
        {
            if(inkey==0x89)inkey=0x7f;
            inkey=inkey-0x7f;
            smtr = game_detik;
            game_detik=game_detik*10+inkey;
            if(detik>59)game_detik=smtr;
        }
        else if(inkey==btVolDw)game_detik=0;
        else if(inkey==btVolUp)break;
        else game_detik=game_detik;
    }
    delay_ms(300);
}

void __init()
{
    PORTB=0x00;
    DDRB=0x2f;
    PORTC=0x00;
    DDRC=0x00;
    PORTD=0x04;
    DDRD=0x00;

    // Timer/Counter 0 initialization
    // Clock source: System Clock
    // Clock value: 11,719 khz
    TCCR0=0x05;
    TCNT0=0x00;

    // Timer/Counter 1 initialization
    // Clock source: System Clock
    // Clock value: 11,719 khz
    // Mode: Normal top=FFFFh
    // OC1A output: Discon.
    // OC1B output: Discon.
    // Noise Canceler: Off
    // Input Capture on Falling Edge
    TCCR1A=0x00;
    TCCR1B=0x05;
}

```

```

TCNT1H=0xd2;
TCNT1L=0x39;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;

// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: 46,875 kHz
// Mode: Normal top=FFh
// OC2 output: Disconnected
ASSR=0x00;
TCCR2=0x06;
TCNT2=0x80;
OCR2=0x00;

// Timer(s)/Counter(s) Interrupt(s) initialization
//TIMSK=0x40;
// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x44;

// SPI initialization
// SPI Type: Master
// SPI Clock Rate: 3000,000 kHz
// SPI Clock Phase: Cycle Half
// SPI Clock Polarity: Low
// SPI Data Order: MSB First
SPCR=0x50;
SPSR=0x00;

}

void main(void)
{
    unsigned char i;
    _init();

    start:
    clear_display();delay_ms(100);
    #asm("cli")
    baca_remote();
    if(remote!=btPower)goto start;

    #asm("sei")

    for(i=0;i<11;i++)char_display(i,buff_char[i]);

    power=1;
    seg_aktif=0;
    mode=standby;
    score[group_a]=score[group_b]=0;
    menit=detik=0;
    babak_display(1);
    game_menit=45;
    game_detik=0;
    time_display(game_menit,game_detik);
    while (1)
    {
        score_display(group_a,score[group_a]);
        score_display(group_b,score[group_b]);
        char_display(seg_aktif,buff_char[seg_aktif]);
    }
}

```

```

    baca_remote();
    switch (remote)
    {
//=====
        case btSleep: //edit time
            mode=edit_menit;
            enter_nilai_menit();
            mode=edit_detik;
            enter_nilai_detik();
            time_display(game_menit,game_detik);
            mode=standby;
            break;
        case btSelect: //edit group
            mode=edit_group;
            seg_aktif=0;
            break;
//=====
        case btVolUp:
            if(mode==running)
            {
                if(score[group_a]>198)score[group_a]=199;
                score[group_a]++;
            }
            break;
        case btVolDw:
            if(mode==running)
            {
                if(score[group_a]<1)score[group_a]=1;
                score[group_a]--;
            }
            break;
//=====
        case btProgUp: //button Program Up
            if(mode==running) //jika mode running score b
            increment
            {
                if(score[group_b]>198)score[group_b]=199;
                score[group_b]++;
            }
            else if(mode==edit_group) //jika mode edit group karakter
            increment
            {
                buff_char[seg_aktif]++;
                if(buff_char[seg_aktif]>37)buff_char[seg_aktif]=0;
            }
            break;
        case btProgDw: //button Program Down
            if(mode==running) //jika mode running score b
            decrement
            {
                if(score[group_b]<1)score[group_b]=1;
                score[group_b]--;
            }
            else if(mode==edit_group) //jika mode edit group karakter
            decrement
            {
                if(buff_char[seg_aktif]<1)buff_char[seg_aktif]=38;
                buff_char[seg_aktif]--;
            }
            break;
//=====
        case btDisplay: //mode running
            char_display(seg_aktif,buff_char[seg_aktif]);
            seg_aktif=0;
            mode=running;
            break;
        case btMute: //Mode standby

```

```

        mode=standby;
        break;
    case btPower:          //Matikan display
        power=0;
        break;
    }
    if(remote>=btSatu && remote<=btNo1)    //outton 0 s/d 9
    {
        if(remote==0x89)remote=0x7f;
        remote=remote-0x7f;

        if(mode==standby)babak_display(remote);    //jika mode standby
        else if(mode==edit_group)seg_aktif=remote; //jika mode edit
    }
    group
        remote=0;
        power=1;
    }
    if(!power)break;
    }
    goto start;
}

```

