

**PAPAN NILAI YANG DAPAT DIPROGRAM
BERBASIS MIKROKONTROLER AT 89C51**

TUGAS AKHIR

Diajukan sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana Pada Jurusan Teknik Elektro
Fakultas Teknologi Industri Universitas Islam Indonesia



Disusun Oleh :

Nama : Yudi Mulyana

No. Mahasiswa : 98524100

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

2007

LEMBAR PENGESAHAN PEMBIMBING
PAPAN NILAI YANG DAPAT DIPROGRAM
BERBASIS MIKROKONTROLER AT 89C51



Nama

Disusun oleh:
: Yudi Mulyana

No.Mahasiswa

: 98524100

Yogyakarta, 5 juli 2007

Pembimbing I,

Pembimbing II,

A handwritten signature in black ink, consisting of a stylized 'T' followed by a 'Y' and some loops.

Tito Yuwono,ST,MSc

A handwritten signature in black ink, appearing as a series of connected, flowing lines.

Wahyudi Budi Pramono,ST

LEMBAR PENGESAHAN PENGUJI
PAPAN NILAI YANG DAPAT DIPROGRAM BERBASIS
MIKROKONTROLER AT89C51

TUGAS AKHIR

Disusun Oleh :

Nama : Yudi Mulyana

No.Mahasiswa : 98 524 100

Telah Dipertahankan di Dalam Sidang Penguji sebagai Salah Satu Syarat untuk
Memperoleh Gelar Sarjana pada Jurusan Teknik Elektro Fakultas Teknologi Industri

Universitas Islam Indonesia

Yogyakarta, juli 2007

Tim Penguji,

Tito Yuwono, ST, MSc
Ketua

Wahyudi Budi Pramono, ST
Anggota I

Ir. H. Suyamto
Anggota II



Mengetahui,

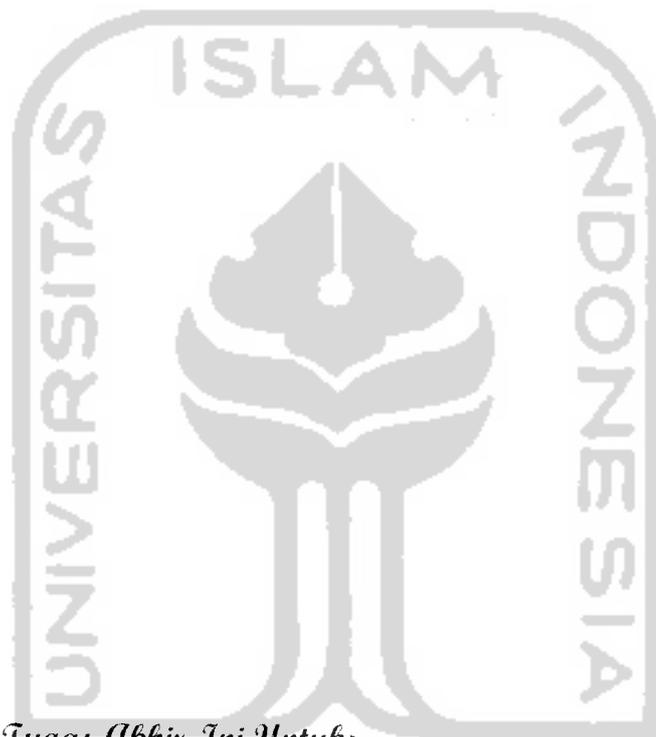
Ketua Jurusan Teknik Elektro

Fakultas Teknologi Industri Universitas Islam Indonesia



Tito Yuwono, ST, MSc

HALAMAN PERSEMBAHAN



“ Kutujukan Tugas Akhir Ini Untuk:

Allah SWT, serta Nabi Muhammad SAW Dan Para Pengikutnya.

*Ayahanda Moch. Abdul Rauf Dan Ibunda Amah Rohamah Yang
Tercinta, Atas Do'a Dan Kasih Sayang Serta Kepercayaannya Yang
Telah Diberikan Kepadaku. Serta Adik Saya yang selalu Memberikan
Dorongan Kpada saya.*

MOTTO

*“Jadikanlah Sabar Dan Shalat Sebagai Penolongmu. Dan Sesungguhnya Yang
Demikian Itu Sungguh Berat, Kecuali Bagi Orang-orang Yang Khusyu”*

(QS.AL-Baqarah : 45)

“Cukuplah Allah Menjadi Penolong Kami Dan Allah Terbaik-baik Pelindung”

(QS.AL-Imran : 173)

*“Sungguh, Allah Tidak Akan Mengubah Nasib Suatu Kaum Sampai Mereka
Sendiri Mengubah Dirinya”*

(QS.Ar-Ra'd : 11)

*“Allah Akan Meninggikan Orang-orang Yang Beriman Diantara Kamu Dan Orang-
orang Yang Diberi Ilmu Pengetahuan Beberapa Derajat. Dan Allah Mengetahui
Atas Apa Yang Kamu Kerjakan”*

(QS.AL-Mujadalah : 11)

*“Barang Siapa Melalui Jalan Mencapai Ilmu, Maka Allah Melewatkannya Pada
Jalan Menuju Sorga”*

(Sabda Rasulullah SAW)

KATA PENGANTAR

Ass, Wr, Wb.

Alhamdulillah, kami panjatkan puji dan syukur kehadirat Allah SWT yang telah memberikan hidayahnya sehingga penulis dapat menyelesaikan laporan Tugas Akhir dengan judul “**PAPAN NILAI YANG DAPAT DIPROGRAM BERBASIS MIKROKONTROLER AT 89C51**” yang merupakan salah satu syarat guna memperoleh gelar sarjana S-I dikalangan Universitas Islam Indonesia Fakultas Teknologi Industri Jurusan Teknik Elektro.

Penulisan dan penyelesaian tugas akhir ini bisa selesai tidak lepas dari saran-saran, bimbingan, dukungan, serta bantuan dari berbagai pihak. Untuk itu pada kesempatan ini penulis ingin menyampaikan ucapan terima kasih yang sebesar-besarnya kepada:

1. **Bapak Tito Yuwono, ST,MSc** selaku Ketua Jurusan Teknik Elektro Fakultas Teknologi Industri Universitas Islam Indonesia, sekaligus selaku Dosen Pembimbing I, yang telah berkenan memberikan bimbingan selama berlangsungnya pembuatan tugas akhir ini.
2. **Bapak Wahyudi Budi Pramono, ST** selaku Dosen Pembimbing II, yang telah memberikan pengarahan, motivasi dan nasehat kepada penulis dalam menyelesaikan tugas akhir ini.

3. **Bapak Ir. H. Suyamto**, selaku penguji yang banyak membantu dan memberikan masukan kepada penulis.
4. Bapak dan Ibu Dosen Teknik Elektro atas ilmu yang telah diberikan kepada penulis.
5. Orang tua tercinta yang selalu memberikan do'a, dukungan, baik berupa materi maupun spiritual.
6. Teman-teman elektro semuanya.
7. Semua pihak yang telah membantu penulis dalam menyelesaikan tugas akhir ini.

Laporan Tugas Akhir ini meskipun telah dikerjakan dengan sebaik-baiknya tetapi masih banyak kekurangan dan keterbatasannya, maka penulis mengharapkan saran dan kritik yang membangun.

Akhir kata, semoga laporan Tugas Akhir ini dapat bermanfaat bagi yang membutuhkan sebagaimana yang diharapkan penulis.

Wassalamu'alaikum Wr, Wb.

Yogyakarta, Juli 2007

Yudi Mulyana

ABSTRAKSI

Dalam perlombaan seperti kuis biasanya dibutuhkan suatu alat yang dapat mencatat nilai yang didapat oleh peserta. Alat ini dapat dibuat dengan menggunakan metode *konvensional* yaitu berupa papan tulis dan alat tulisnya hingga dapat pula dibuat dengan menggunakan sistem yang canggih yaitu komputerisasi. Dalam tugas akhir ini dibuat suatu pencatat nilai perlombaan tersebut dengan menggunakan komponen utama sebagai alat kontrolnya yaitu *mikrokontroler* AT 89C51. Nilai penambah dan nilai pengurang untuk peserta pada alat ini dapat diprogram sehingga penggunaan alat ini sebagai alat pencatat nilai menjadi fleksibel untuk beberapa macam jenis perlombaan. Cara kerja dari alat ini adalah, *mikrokontroler* sebagai pengatur utama alat ini berfungsi sebagai pengolah data masukan yang bersumber dari penekanan tombol oleh operator, memproses masukan tersebut secara matematis dan menyimpannya dalam *memori internalnya* dan juga menampilkan pada saluran keluarannya yang dihubungkan ke rangkaian displai *seven segment*. Kemudahan dari alat ini adalah, operator hanya tinggal menekan tombol penambah bila suatu peserta berhak mendapat nilai tambah dan tombol pengurang bila nilainya memang harus dikurangkan. Dan dari hasil uji coba kinerja alat yang telah dilakukan terhadap alat ini, didapat hasil yang cukup memuaskan dimana alat yang dibuat telah dapat menambah dan mengurangi nilai peserta dengan hanya menekan satu tombol dan disisi lain nilai penambah dan nilai pengurang juga dapat diseting sesuai dengan keinginan dari operator.

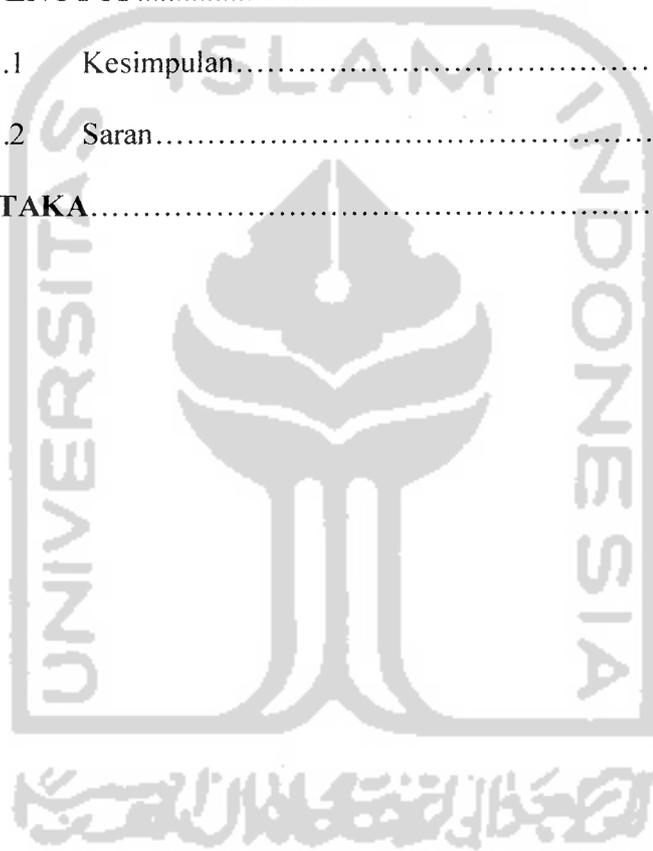
DAFTAR ISI

HALAMAN JUDUL

LEMBAR PENGESAHAN PEMBIMBING.....	i
LEMBAR PENGESAHAN PENGUJI.....	ii
HALAMAN PERSEMBAHAN.....	iii
MOTTO.....	iv
KATA PENGANTAR.....	v
ABSTRAKSI.....	vii
DAFTAR ISI.....	viii
DAFTAR TABEL.....	xi
DAFTAR GAMBAR.....	xii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang Masalah.....	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah.....	2
1.4 Tujuan.....	3
1.5 Sistematika Penulisan.....	3
BAB II LANDASAN TEORI.....	5
2.1 Mikrokontroler AT89C51.....	5

2.2	Transistor Sebagai Saklar.....	15
2.3	Seven Segment.....	18
BAB III	PERANCANGAN SISTEM.....	21
3.1	Bahan dan instrumen yang digunakan.....	21
3.1.1	Bahan/komponen utama yang digunakan.	21
3.1.2	Instrument yang digunakan.....	22
3.2	Perancangan Perangkat Keras.....	22
3.3	Perancangan Perangkat Lunak.....	28
BAB IV	PENGUJIAN SISTEM.....	38
4.1	Hardware.....	38
4.1.1	Pengujian Rangkaian System Minimum Mikrokontroler AT89C51.....	38
4.1.2	Pengujian Rangkaian Displey Seven Seg Ment.....	40
4.1.3	Pengujian Rangkaian catu daya.....	41
4.2	Software.....	42
4.2.1	Pengujian Penambah dan Pengurang Nilai i peserta.....	42
4.2.2	Pengujian Penyetingan Nilai Penambah...	43

	4.2.3	Pengujian Penyetingan Nilai Pengurang..	44
	4.2.4	Pengujian Penambah dan Pengurang Nilai peserta Seteleh Nilai Penambah dan Pengurang Diubah Menjadi 20 dan 10.....	45
BAB V		PENUTUP	47
	5.1	Kesimpulan.....	47
	5.2	Saran.....	48
		DAFTAR PUSTAKA	49
LAMPIRAN			



DAFTAR TABEL

Tabel 2.1 Fungsi khusus port 3.....	9
Tabel 2.2 Nilai register setelah direset.....	11
Tabel 4.1. Data pengujian rangkaian sistem minimum AT 89C51.....	39
Tabel 4.2. Pengujian Penambah dan Pengurang.....	43
Tabel 4.3. Data pengujian penyetingan nilai penambah.....	44
Tabel 4.4. Data pengujian penyetingan nilai pengurang.....	45
Tabel 4.5. Pengujian dan Pengurang Nilai Peserta Setelah Nilai Penambah dan Pengurang Diubah Menjadi 20 dan 10.....	45

BAB I PENDAHULUAN

I.1. Latar Belakang Masalah

Dalam suatu acara perlombaan seperti kuis, cepat tepat dan lain sebagainya biasanya dibutuhkan papan nilai untuk mencatat skor/nilai yang diperoleh oleh peserta acara tersebut. Jenis papan nilai telah banyak dibuat dari yang konvensional, yaitu dengan menggunakan papan tulis dan ditugaskan seseorang untuk mencatat bila terjadi perubahan perolehan nilai peserta lomba, hingga papan nilai yang canggih dengan menggunakan komputer.

Semakin canggih alat pencatat nilai ini (dengan sistem komputerisasi) akan semakin mudah penggunaannya akan tetapi sayangnya hal ini diimbangi dengan biaya yang besar untuk pengadaannya.

Pada tugas akhir ini penulis mencoba membuat suatu papan nilai secara elektronik dengan menggunakan komponen utama mikrokontroler. Mikrokontroler prinsip kerjanya sangat mirip dengan sebuah komputer hanya saja dalam cakupan kerja yang sangat terbatas. Saat ini penggunaan mikrokontroler untuk mengontrol sistem kerja sederhana banyak menjadi pilihan dikarenakan dengan menggunakan mikrokontroler ini dapat menghemat penggunaan komponen digital dan cara kerjanya dapat menyerupai sistem komputer, ditambah pula dengan murahnya harga mikrokontroler ini.

Dengan dibuatnya sistem papan nilai ini dengan menggunakan mikrokontroler, diharapkan didapat papan nilai yang mudah dalam meengoperasikannya serta murah dalam pengadaannya.

I.2. Rumusan Masalah

Dari ulasan mengenai permasalahan diatas, penulis merumuskan permasalahan tersebut sebagai berikut :

1. Bagaimana *mikrokontroler* dapat mengatur tampilan nilai pada display *seven segment*.
2. Bagaimana proses penambahan dan pengurangan nilai pada masing-masing peserta oleh *mikrokontroler*.
3. Bagaimana *mikrokontroler* mengatur pemrograman nilai tambah dan nilai kurang dan menyimpannya dalam memori internalnya.

I.3. Batasan Masalah

Untuk tidak meluasnya lingkup pembahasan masalah dalam memberikan keterangan dalam tugas akhir ini, penulis akan memberikan batasan-batasan masalah yang akan dibahas. Adapun masalah-masalah yang akan diterangkan dalam tugas akhir ini adalah masalah teori yang menunjang tersusunnya rangkaian ini, fungsi/kegunaan dari komponen yang akan dipakai, perencanaan dan cara kerja rangkaian, pembuatan/perakitan serta pengujian rangkaian.

Sementara itu batasan dari alat yang dibuat adalah membuat alat papan nilai dari suatu perlombaan dengan jumlah peserta sebanyak 3 dengan maksimal nilai pada masing-masing peserta sebanyak 3 digit atau 999.

I.4. Tujuan

Merancang dan membuat suatu rangkaian penghitung nilai peserta suatu perlombaan dengan menggunakan komponen utama mikrokontroler AT 89C51.

I.6. Sistematika Penulisan

Pada penyusunan laporan tugas akhir ini, untuk memudahkan penuangan ide kedalam penulisan, maka akan dibagi menjadi beberapa bab yang keseluruhannya meliputi lima bab, yang terbagi sebagai berikut :

BAB I : PENDAHULUAN

Sebagai bab pertama, bab ini akan membahas latar belakang masalah, identifikasi masalah, rumusan masalah, tujuan penulisan, kontribusi dan sistematika penulisan.

BAB II : LANDASAN TEORI

Dalam bab ini akan dibahas tentang teori-teori yang mendukung pada rangkaian alat ini

BAB III : PERANCANGAN SISTEM ELEKTRONIS

Pada bab ini akan dibahas perencanaan pembuatan alat dimana pembahasannya berisi tentang komponen yang dipakai dan penjelasannya serta konstruksi-konstruksi rangkaiannya, juga pada bab ini akan dibahas cara kerja rangkaian dimana dalam penjelasannya akan diterangkan blok perblok menurut diagram blok yang telah digambarkan diatas.

BAB IV : PENGUJIAN SISTEM ELEKTRONIS

Di dalam bab ke empat ini, akan dijelaskan proses perakitan rangkaian yang akan dibuat dan dalam, bab ini juga akan dijelaskan hasil pengukuran dan pengujian dari rangkaian alat tersebut.

BAB V : PENUTUP

Sebagai penutup penulisan dari tugas akhir ini akan dilengkapi dengan kesimpulan-kesimpulan yang di tarik dari alat yang telah dibuat, juga memuat saran-saran bagi para pembaca yang ingin membuat alat serupa yang akan diterapkan untuk digabungkan dengan rangkaian lain atau dimodifikasi sesuai dengan kreatifitas yang diinginkan.

BAB II

LANDASAN TEORI

II.1. Mikrokontroler AT89C51

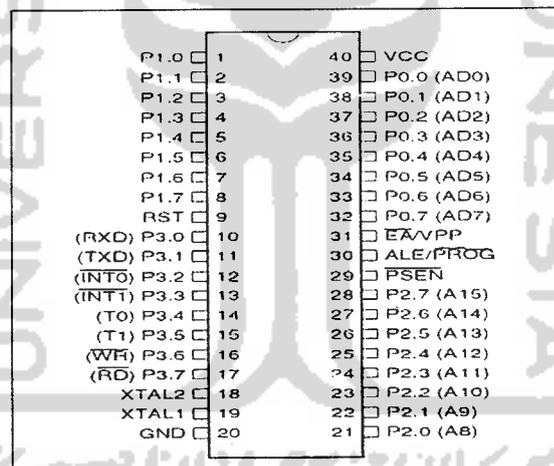
Mikrokontroler adalah suatu rangkaian terintegrasi yang tersusun atas beberapa komponen, antara lain: *Central Processing Unit (CPU)*, *Read Only Memory (ROM)*, *Random Acces Memory (RAM)*, *Timer* dan *Input/Output (I/O)* yang dikemas dalam satu keping tunggal (*Chip*). Jadi sebenarnya mikrokontroler merupakan sebuah piranti pengembangan mikroprosesor dengan teknik fabrikasi dan konsep pemrograman yang sama memungkinkan pembuatan mikroprosesor multiguna juga menghasilkan mikrokontroler.

Mikrokontroler merupakan sistem mikroprosesor yang dirancang secara khusus untuk aplikasi dengan kendali sekuensial, yaitu digunakan untuk mengatur dan memonitor suatu sistem dengan urutan kerja tertentu. Mikrokontroler merupakan gabungan antara mikroprosesor dengan *Read Only Memory (ROM)*, *Random Acces Memory (RAM)*, *Paralel I/O*, *Serial I/O*, *Counter*, *Timer* dan pembangkit isyarat pulsa detak. Sedangkan suatu mikroprosesor sendiri atau CPU terdiri atas beberapa komponen, antara lain: *Aritmathic Logic Unit (ALU)*, *Program Counter (PC)*, *Stack Pointer (SP)* dan register-register.

Dalam sejarahnya mikrokontroler MCS-51 merupakan jenis mikrokontroler yang termasuk tua, keluarga mikrokontroler MCS-51 adalah merupakan

mikrokontroler yang paling populer saat ini. Keluarga ini diawali oleh Intel yang mengenalkan IC mikrokontroler type 8051 pada awal tahun 1980 an. Sampai saat ini sudah ada lebih 100 macam mikrokontroler turunan 8051, sehingga terbentuklah keluarga besar mikrokontroler dan biasa disebut MCS-51. Belakangan ini, pabrik IC Atmel ikut menambah anggota keluarga MCS-51. Produksi mikrokontroler MCS-51 Atmel dibagi dua macam, yang pertama yaitu mikrokontroler dengan jumlah pin 40 setara dengan jumlah pin 8051 yang asli, dan yang kedua adalah mikrokontroler dengan jumlah pin 20 yang merupakan mikrokontroler MCS-51 yang disederhanakan. Perbedaan keduanya adalah dalam hal kapasitas *Flash* PEROM (*Programmable and Erasable Read Only Memory*) nya. AT89C51 mempunyai *Flash* PEROM dengan kapasitas 4 Kbyte, AT89C52 dengan kapasitas 8 Kbyte, AT89C53 dengan kapasitas 12 Kbyte, AT89C55 dengan kapasitas 20 Kbyte dan AT 89C8252 berisikan 8 Kbyte *Flash* PEROM dan 2 Kbyte EEPROM (*Electrical Erasable and Programable ROM*). Penyederhanaan dilakukan pula pada mikrokontroler ukuran kecil ini, yaitu dengan cara mengurangi jalur I/O paralel, kemampuan yang lain sama sekali tidak mengalami pengurangan, penyederhanaan ini dimaksudkan untuk membentuk mikrokontroler yang ukurannya fisiknya kecil akan tetapi dengan kemampuan yang sama. Atmel memproduksi 3 buah mikrokontroler 'mini' ini masing-masing adalah AT89C1051 dengan kapasitas *Flash* PEROM 1 Kbyte, AT89C2051 dengan kapasitas 2 Kbyte dan AT89C4051 dengan kapasitas 4 Kbyte.

AT89C51 adalah mikrokontroler 8 bit keluaran Atmel dengan 4 Kbyte *Flash* PEROM yang merupakan memori dengan teknologi *high density nonvolatile memory* dan kompatibel dengan mikrokontroler standar industri MCS-51, isi memori tersebut dapat diisi ulang atau dihapus berkali-kali sampai batas 1000 kali, mikrokontroler ini merupakan *high performance* teknologi CMOS (*Complementary Metal Oxide Semiconductor*) dan dikemas dalam paket 40 pin dengan catu daya tunggal. Susunan kaki dan simbol logika dari Mikrokontroler AT89C51 dalam bentuk PDIP (*Plastic Dual In Line Package*) ditunjukkan pada gambar 2.1.



Gambar 2.1. Konfigurasi pin dari Chip Mikrokontroler AT89C51.

Masing-masing pin pada gambar tersebut memiliki fungsi tersendiri. Satu kumpulan pin memiliki fungsi sama dan diwakili oleh sebuah register atau alamat tersendiri pada internal CPUnya disebut juga *port*. Fungsi dari pin-pin tersebut adalah sebagai berikut:

1. *Port 0*

Port 0 dapat berfungsi sebagai I/O biasa, *low order multiplex address/ data* atau menerima kode byte pada saat *Flash Programing*. Pada fungsi sebagai I/O biasa *port* ini dapat memberikan *output sink* ke delapan buah TTL input atau dapat diubah sebagai input dengan memberikan logika 1 pada *port* tersebut. Pada fungsi sebagai *low order multiplex address/ data port* ini akan mempunyai *internal pull up*. Pada saat *Flash Programing* diperlukan *external pull up* terutama saat verifikasi program. *Port 0* terdapat pada pin no. 32-39.

2. *Port 1*

Port 1 berfungsi sebagai I/O biasa atau menerima *low order address bytes* pada saat *Flash Programing*. *Port* ini mempunyai *internal pull up* dan berfungsi sebagai input dengan memberikan logika 1. Sebagai output, *port* ini dapat memberikan *output sink* ke empat buah input TTL. *Port 1* terdapat pada pin no. 1-8.

3. *Port 2*

Port 2 berfungsi sebagai I/O biasa atau menerima *high order address bytes* pada saat mengakses memori secara 16 bit (*Movx @DPTR*). Pada saat mengakses memori secara 8 bit, (*Mov @Rn*) *port* ini akan mengeluarkan isi dari *P2 Special Function Register (SFR)*. *Port* ini mempunyai *internal pull up* dan berfungsi sebagai input dengan memberikan logika 1. Sebagai output,

port ini dapat memberikan *output sink* ke empat buah input TTL. *Port 1* terdapat pada pin no. 21-28.

4. *Port 3*

Port 3 yang terdapat pada pin 10-17 berfungsi sebagai *input/output (I/O)* yang mempunyai sifat sama dengan *port 1* maupun *port 2*, sedangkan sebagai fungsi spesial, *port-port* ini mempunyai keterangan yang ditunjukkan pada tabel 2.1.

Tabel 2.1 Fungsi khusus port 3

Port Pin	No Pin	Fungsi Khusus
P3.0	10	RXD (masukan data port serial)
P3.1	11	TXD (keluaran data port serial)
P3.2	12	INT0' (masukan interupsi 0 dari luar)
P3.3	13	INT1' (masukan interupsi 1 dari luar)
P3.4	14	T0 (masukan ke pencacah 0)
P3.5	15	T1 (masukan ke pencacah 1)
P3.6	16	WR' (sinyal tulis untuk memori luar)
P3.7	17	RD' (sinyal baca untuk memori luar)

5. PSEN (*Program Strobe Enable*)

PSEN adalah kontrol sinyal yang memungkinkan untuk mengakses program (code) memori *eksternal*. Pin ini dihubungkan ke pin OE (*output enable*) dari EPROM. Sinyal PSEN akan 0 pada tahap *fetch* (penjemputan) instruksi. PSEN akan selalu bernilai 0 pada pembacaan program memori *internal*. PSEN terdapat pada pin 29.

6. ALE (*Address Latch Enable*)

Pin ini dapat berfungsi sebagai *Address Latch Enable* (ALE) yang me-*latch low byte address* pada saat mengakses memori *eksternal*. Sedangkan pada saat

Flash Programing berfungsi sebagai *pulse input*, pada operasi normal ALE akan mengeluarkan sinyal clock sebesar 1/16 frekuensi oscillator kecuali pada saat mengakses memori *eksternal*, sinyal clock pada pin ini dapat pula di *disable* dengan men-set bit 0 dari SFR di alamat 8Eh. ALE hanya akan aktif pada saat mengakses memori *eksternal* (Movx dan Movc). ALE terdapat pada pin 30.

7. EA (*External Access*)

Pada kondisi logika rendah, pin ini akan berfungsi sebagai EA yaitu mikrokontroler akan menjalankan program yang ada pada memori *eksternal* setelah sistem direset. Jika berkondisi logika tinggi, pin ini akan berfungsi untuk menjalankan program yang ada pada memori *internal*. Pada saat *Flash Programing* pin ini akan mendapat tegangan 12 Volt (VP), EA terdapat pada pin 31.

8. *On-Chip Oscillator*

AT89C51 telah memiliki *on-chip oscillator* yang dapat bekerja dengan menggunakan kristal *eksternal* yang dihubungkan ke pin XTAL1 dan XTAL2. Tambahan kapasitor untuk menstabilkan oscillator tersebut. Nilai kristal yang biasa dipakai oleh keluarga MCS-51 adalah 12 MHz. *On-chip oscillator* juga dapat menggunakan isyarat pulsa detak dari luar, misalnya AFG (*eksternal oscillator*) yang cukup dihubungkan pada pin XTAL1.

9. RST (*Reset*)

RST pada pin 9 merupakan reset dari AT89C51. Jika pada pin ini diberi masukan logika tinggi selama 2 *machine cycle* maka register-register *internal* pada AT89C51 akan berisi nilai *default* setelah sistem di reset seperti ditunjukkan pada tabel 2.2.

10. Koneksi Catu Daya

Beroperasi pada tegangan 5 volt. Pin Vcc terdapat pada pin 40 sedangkan Vss (*ground*) terdapat pada pin 20.

Tabel 2.2 Nilai register setelah direset

Register	Isi
Program counter	0000H
Accumulator	00H
B register	00H
PSW	00H
SP	07H
DPTR	0000H
Port 0-3	FFH
IP (8031/8051)	XXX00000B
IP (8032/8052)	XX000000B
IE (8031/8051)	0XX00000B
IE (8032/8052)	0X000000B
Timer register	00H
SCON	00H
SBUF	00H
PCON (HMOS)	0XXXXXXXB
PCON (CMOS)	0XXX0000B

Untuk merancang suatu sistem yang menggunakan mikrokontroler AT89C51 sebagai basis utamanya, diperlukan pemahaman terhadap konstruksi, instruksi dan

pendukung operasi yang dimiliki oleh mikrokontroler tersebut. Berikut ini adalah sekilas gambaran dari mikrokontroler AT89C51:

a. Konstruksi dasar mikrokontroler AT89C51

Mikrokontroler ini dilengkapi dengan berbagai fasilitas diantaranya adalah:

1. *Central Processing Unit (CPU)*
2. Memori data (RAM) didalam chip 128 byte, *Flash PEROM* didalam chip sebesar 4 Kbyte yang dapat diisi ulang sebanyak 1000 kali.
3. Pengendali interupsi (*Interrupt Control*)
4. Bus Kendali (*Bus Control*)
5. UART (*Universal Asynchronous Receiver/Transmitter*) yang digunakan untuk komunikasi data secara serial (Jalur seri berada pada pin RXD dan TXD)
6. Rangkaian oscilator didalam chip dengan frekuensi makasimal 24 MHz.
7. Empat buah *port* yang masing-masing berisi 8 bit, sifatnya 2 arah (I/O) dan setiap bitnya dapat dialamati, salah satu portnya yaitu *port* 3 (P3) juga dapat berfungsi untuk komunikasi data secara serial. Interupsi, masukan untuk pencacah dan masukan isyarat perintah baca dan tulis (R/W).
8. Pewaktu (*Timer*)/ pencacah (*Counter*) 16 bit sebanyak 2 buah.

b. RAM, ROM dan register dalam AT89C51

Dalam pengertian MCS-51, *Flash* PEROM merupakan memori penampung program pengendali AT89C51, dikenal sebagai memori program (nomor \$0000 sampai \$07FF, lihat gambar 4). Sedangkan RAM dalam chip AT89C51 adalah memori data (dialamati dari alamat \$00 sampai \$FF), yaitu memori yang dipakai untuk menyimpan data.

c. *Timer* pada AT89C51

Mikrokontroler AT89C51 dibuat dengan dibekali 2 buah *timer*, keduanya dapat di kendalikan, diset, dibaca dan dikonfigurasi sendiri-sendiri. *Timer* AT89C51 memiliki 3 fungsi umum, yaitu:

1. Menghitung waktu antara 2 kejadian (*event*)
2. Menghitung jumlah kejadian itu sendiri
3. Membangkitkan *baud rate* untuk *port* serial

d. Dasar kerja program

Program untuk mengendalikan kerja dari mikrokontroler disimpan dalam memori program, program pengendali tersebut merupakan kumpulan dari instruksi kerja mikrokontroler, suatu instruksi MCS-51 merupakan kode yang panjangnya antara 1 sampai 4 Byte.

Selama mikrokontroler bekerja, instruksi kerja tersebut byte demi byte diambil ke CPU dan selanjutnya digunakan untuk mengatur kerja mikrokontroler. Proses pengambilan instruksi dari memori program disebut '*fetch cycles*' dan saat-saat CPU melaksanakan instruksi disebut dengan '*execute cycles*'.

Semua mikrokontroler maupun mikroprosesor dilengkapi dengan sebuah register yang berfungsi khusus untuk mengatur *fetch cycles*, register tersebut dinamakan sebagai *Program Counter (PC)*. Nilai PC secara otomatis bertambah satu setiap kali selesai mengambil 1 byte isi memori program, dengan demikian isi memori program dapat berurutan diumpankan ke CPU.

Saat MCS-51 di reset, isi *program counter* di-reset menjadi 0000. artinya sesaat setelah reset, isi dari memori program alamat 0 dan seterusnya akan diambil ke CPU dan diperlakukan sebagai instruksi yang akan mengatur kerja mikrokontroler. Dengan demikian, awal dari program pengendali MCS-51 harus ditempatkan di memori alamat 0, setelah reset MCS-51 menjalankan program mulai dari memori program alamat 0000, dengan melakukan proses *fetch cycles* dan *execute cycles* terus menerus tanpa henti.

Jika sarana interupsi diaktifkan dan tegangan di pin INT0 (pin nomor 6) berubah dari '1' menjadi '0', maka proses menjalankan program diatas, akan dihentikan sebentar, mikrokontroler melayani dulu permintaan interupsi, CPU akan melanjutkan mengerjakan program utama lagi.

Untuk melaksanakan hal diatas, pertama-tama CPU menyimpan nilai *Program Counter* ke *Stack (Stack* merupakan satu bagian kecil dari data memori RAM) kemudian mengganti isi PC dengan 0003. artinya MCS-51 akan melaksanakan program yang ditempatkan di memori program mulai byte ke-3 untuk melayani interupsi yang diterima dari pin INT0. adalah tugas *programmer* untuk mengatur

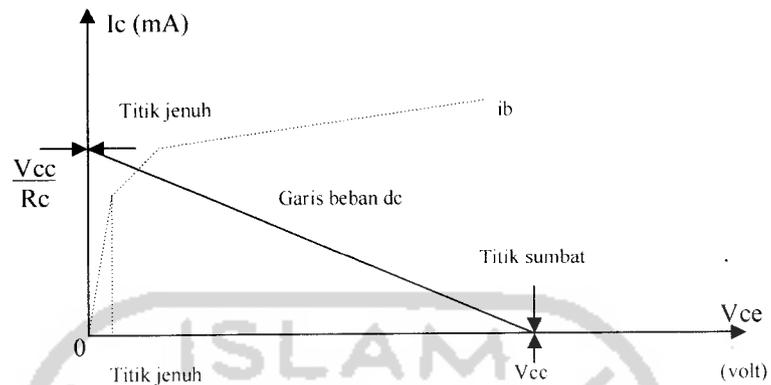
dimana agar program yang dipakai untuk melayani interupsi lewat INT0 harus ditempatkan.

Selesai melayani interupsi, nilai PC yang semula disimpan di dalam *stack* akan dikembalikan ke PC, dengan demikian CPU bisa melanjutkan '*execute cycles*'-nya di program utama.

Selain INT0, AT89C51 dapat menerima interupsi dari INT1, dari UART dan dari *Timer*. Agar permintaan interupsi itu dapat dilayani dengan program yang berlainan, maka masing-masing sumber interupsi itu mempunyai nomor awal program untuk layanan interupsi yang berlainan.

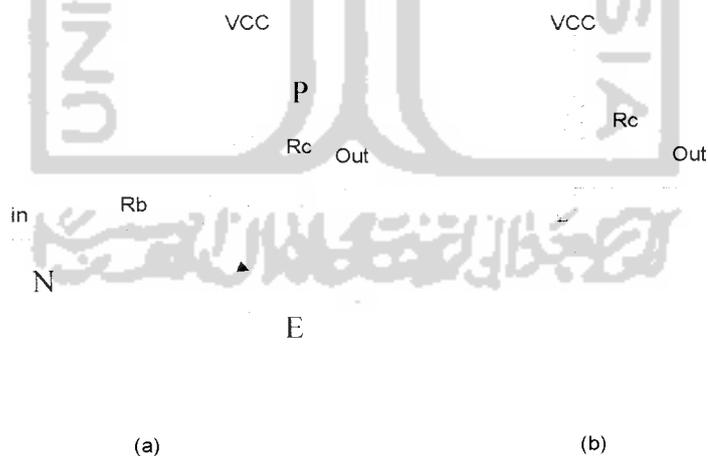
II.2. Transistor Sebagai Saklar

Transistor mempunyai yang beberapa fungsi, salah satu fungsi yang banyak diterapkan pada rangkaian digital adalah sebagai sebuah saklar elektronik yang bekerja hanya dalam dua kondisi *on* dan *off*. Dalam memfungsikan transistor sebagai saklar maka transistor hanya dioperasikan pada daerah jenuh (*saturation*) dan daerah sumbat (*cut-off*).



Gambar 2.2. Daerah operasi transistor sebagai saklar

Bila transistor dioperasikan pada daerah jenuh, maka transistor menjadi sebuah saklar yang tertutup (ON). Tapi jika transistor dioperasikan pada daerah sumbat, maka transistor menjadi sebuah saklar terbuka (OFF). Gambar 2.3. memperlihatkan fungsi transistor sebagai sakelar.



Gambar 2.3. (a) Bias transistor yang berfungsi sebagai saklar
(b) Ekuivalen transistor sebagai saklar *on*

Saat transistor bekerja pada daerah jenuh maka besarnya arus akan mengalir tanpa halangan dari terminal kolektor menuju terminal emitor, karena besarnya $V_{ce} = 0$ dan arus jenuh pada kolektor atau $I_{C_{sat}} = V_{cc} / R_c$. Kondisi ini menyerupai sebuah saklar mekanik dalam kondisi tertutup (ON). Untuk membuat transistor *on* diperlukan arus basis yang besarnya $I_b > I_c / \beta$.

Saat transistor dalam keadaan *cut-off* berlaku ketentuan $V_{ce} = V_{cc}$ dimana $I_c = 0$. Dalam hal ini transistor menyerupai sebuah saklar mekanik yang terbuka (OFF). Kondisi demikian dapat direalisasikan dengan memberikan bias basis $I_b = 0$, atau pada basis diberi tegangan mundur (*revers*) terhadap emitor [1].

Analisis perhitungan untuk kondisi transistor secara teori adalah sebagai berikut :

1. Kondisi *Saturation*

$$V_{ce} = V_{cc} - I_c \cdot R_c \dots\dots\dots (2.1)$$

Karena $V_{ce} = 0$ maka,

$$I_c = V_{cc} / R_c \dots\dots\dots (2.2)$$

2. Kondisi *Cut-off*

$$V_{ce} = V_{cc} - I_c \cdot R_c \dots\dots\dots (2.3)$$

Karena $I_c = 0$ maka :

$$V_{ce} = V_{cc} \dots\dots\dots (2.4)$$

Besarnya tahanan basis R_b untuk mendapatkan arus basis I_b saat transistor dalam keadaan saturasi adalah:

$$I_b \text{ sat} = I_c / \beta \dots\dots\dots (2.5)$$

$$\text{Jadi } R_b = (V_{cc} - V_{be}) / I_b \text{ sat} \dots\dots\dots (2.6)$$

Keterangan:

V_{ce} = Tegangan Kolektor Emiter

V_{cc} = Tegangan Masukan

I_c = Arus Kolektor

R_c = Resistansi Pada Kaki Kolektor

$I_b \text{ sat}$ = Arus Basis pada Kondisi Saturasi

R_b = Resistansi pada Kaki Basis

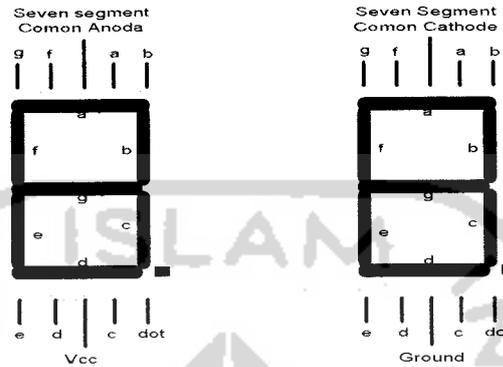
V_{be} = Tegangan Basis ke Emiter

II.3. Seven Segment

Ada dua macam *seven segment* yang dipakai, yaitu dioda pemancar cahaya (*light emitting diode* atau **LED**) dan peraga kristal cair (*liquid crystal display* atau **LCD**). Keduanya merupakan peralatan zat padat, **LED** mengandalkan pada pemancaran cahaya waktu pertemuan p-n semikonduktor dibias maju, dan **LCD** mengandalkan pada transmisi atau absorpsi kristal-kristal tertentu yang ditempatkan dalam medan listrik. Keuntungan utama **LCD** adalah konsumsi dayanya yang rendah, tapi kerugiannya dia tidak dapat dilihat di tempat yang gelap. Dalam sebuah *seven*

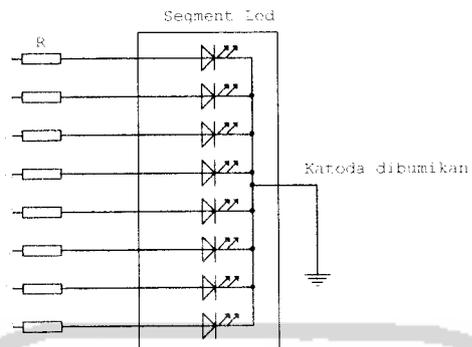
segment LED, *LED* diatur dalam suatu pola seperti yang ditunjukkan pada gambar

2.4. berikut :

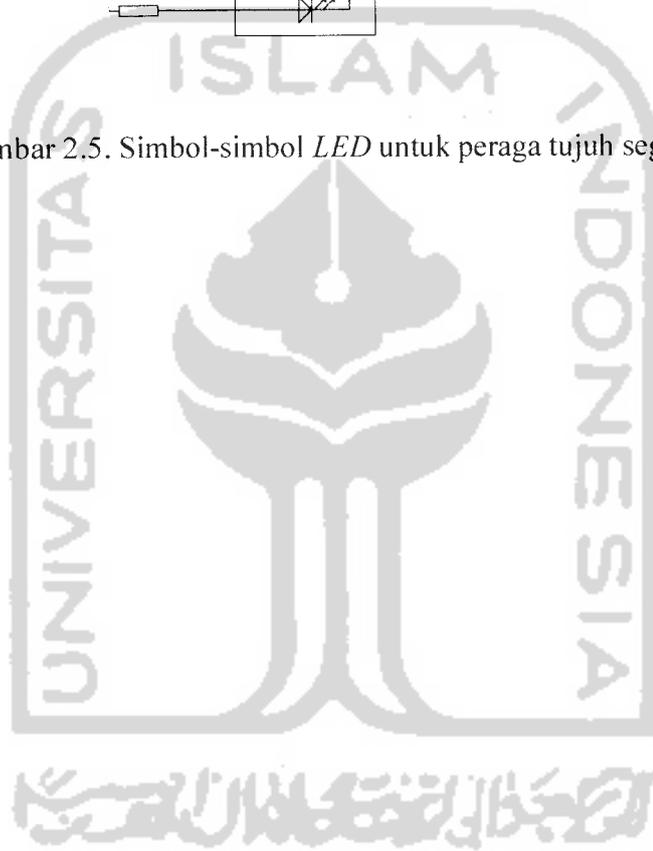


Gambar 2.4. Segmen-segmen pada peraga *LED Seven Segment*

Dengan menyalakan segmen E dan F diperoleh bilangan desimal 1. bilangan 5 diperoleh dengan menyinari segmen-segmen A, F, G, C, dan D, dan seterusnya. Hanya bilangan desimal dari 0 sampai 9 dan beberapa simbol khusus (seperti tanda -) serta beberapa huruf abjad (seperti C dan F) dapat diperagakan dengan *seven segment* ini. Peraga mempunyai sebuah titik desimal di sebelah kiri atau sebelah kanan digit. Peraga *LED* dapat diperoleh dalam berbagai warna, umumnya merah, tapi jingga, hijau, dan kuning juga dapat diperoleh.



Gambar 2.5. Simbol-simbol *LED* untuk peraga tujuh segmen



BAB III

PERANCANGAN SISTEM

Dalam bab ini akan dibahas mengenai perancangan alat, dikarenakan pada alat ini digunakan mikrokontroler yang membutuhkan program (*software*) yang harus diisikan (*download*) kedalam *chip* mikrokontroler tersebut, maka mengenai perancangan alat ini dibagi menjadi dua bagian yaitu perancangan perangkat keras (*hardware*) dan perancangan perangkat lunak (*software*).

3.1. Bahan dan Instrumen yang digunakan

3.1.1. Bahan/komponen utama yang digunakan

Komponen yang dipergunakan pada alat ini adalah :

- IC Mikrokontroler AT89C51
- Tansistor FCS 9012.
- *Seven segment*
- IC 7805, regulator tegangan untuk kestabilan tegangan keluaran sebesar 5 volt.
- Lampu LED (*light emitting diode*)
- Dioda *bridge* 1 Ampere
- Transformator
- *R-Pack* 10 k Ω , 9 pin.

3.1.2. Instrument yang digunakan

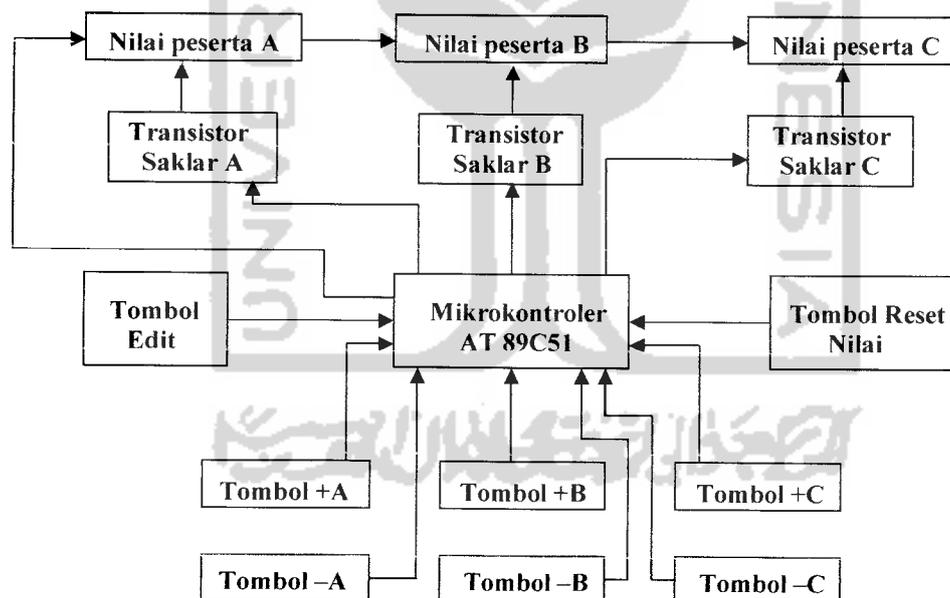
Untuk mewujudkan rancangan pembuatan papan nilai yang dapat diprogram berbasis mikrokontroler AT 89C51, diperlukan alat bantu utama dan alat-alat pengujian antara lain :

1. *PC (Personal Computer)*. Digunakan untuk menulis dan memperbaiki kode-kode program serta mengirim (*downloading*) kode-kode program ke *flash PEROM* mikrokontroler AT89C51.
2. *Universal Atmel Writer* Mikrokontroler dari *BERIN*. Untuk mengisikan dan merealisasikan program yang telah ditulis dalam *Text Editor* dan disimpan dalam ekstensi *HEX*, kedalam mikrokontroler.
3. Program *Assembler* untuk *MCS-51* yaitu *ASM51.EXE* untuk mengubah program *Assembler* yang telah diketik menjadi file *OBJ*, serta program *OH.EXE* untuk mengubah file *OBJ* ini menjadi file dengan ekstensi *HEX*.
4. Program *TS Emulator 8051* yang digunakan untuk menguji potongan-potongan program yang telah dibuat.
5. *Multimeter* yang digunakan untuk mengukur tegangan catu daya, arus yang mengalir, mengecek jalur *PCB (Printed Circuit Board)*
6. Catu Daya *DC 15V/ 1A*, yang digunakan untuk mensuplai mikrokontroler AT89C51 pada saat *download* program ke *flash PEROM*.

3.2. Perancangan Perangkat Keras

Papan nilai yang dapat diprogram ini dirancang dengan menggunakan komponen-komponen yang beberapa jenisnya telah disebutkan sebelumnya, komponen-komponen tersebut kemudian dirakit sesuai dengan rancangan skematik yang telah didesain. Untuk memudahkan penjelasan mengenai rancangan perangkat keras, digambarkan diagram blok alat yang dibuat pada gambar 3.1.

Dari diagram blok tersebut penggambaran detailnya bagaimana komponen-komponen tersebut terkoneksi satu dengan yang lainnya diuraikan dibawah ini yang juga disertakan skemanya bagian perbagian.



Gambar 3.1. Diagram Blok Alat

Jantung utama dari alat papan nilai ini adalah mikrokontroler AT89C51, dimana mikrokontroler ini mengontrol semua masukan dan keluaran sekaligus melakukan kalkulasi matematikanya. Tombol + dan – untuk masing masing peserta digunakan untuk menambah dan mengurangi nilai peserta tersebut. Tombol edit digunakan apabila juri hendak mengedit nilai masukan untuk poin tambah dan kurang dan sedangkan tombol reset adalah untuk mereset semua nilai peserta yang sedang berlangsung kembali menjadi 000.

Sistem yang digunakan oleh mikrokontroler untuk menampilkan nilai dari ketiga peserta itu adalah dengan menggunakan sistem *scanning* secara cepat dimana dengan metoda ini penggunaan kaki-kaki dari mikrokontroler menjadi lebih hemat dan dapat menjalankan 9 buah *seven segment* yang akan tampak menyala secara bersamaan.

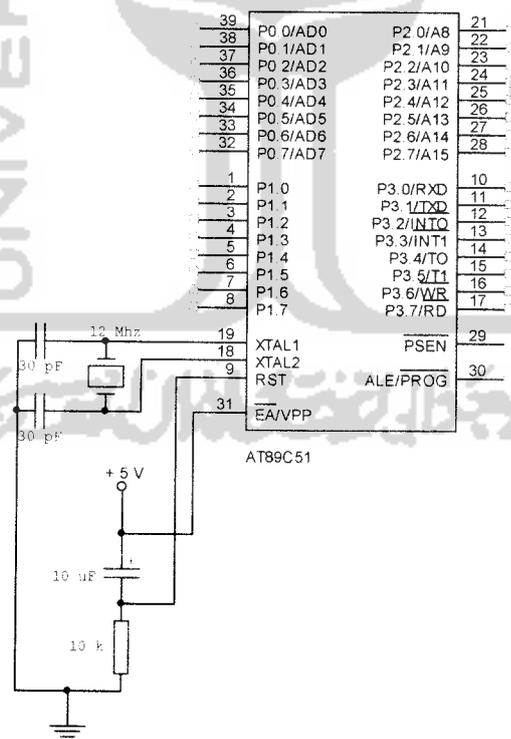
Transistor pada alat ini digunakan sebagai sakelar elektronik yang akan memutus sambung tegangan positif ke masing-masing *seven* segmen. Jenis *seven* segmen yang akan digunakan dalam alat ini adalah *seven* segmen dengan *common* katoda atau dengan masukan positifnya yang dijadikan satu.

Rangkaian sistem minimum AT 89C51

Sistem minimum mikrokontroler AT89C51 adalah sangat sederhana, hanya dibangun dengan menggunakan beberapa buah komponen tambahan saja yaitu sebuah kristal sebagai pembangkit getaran *oscilator*, dua buah kondensator 30 *pF*, satu buah

resistor $10\text{ k}\Omega$, dan satu buah kondensator $10\text{ }\mu\text{F}$. Mikrokontroler AT89C51 ini beroperasi pada tegangan 5 volt .

Pada saat mikrokontroler dihidupkan secara otomatis semua kaki pin pada *port* mikrokontroler tersebut diberi logika 1, dengan demikian untuk merubahnya menjadi logika 0, harus dirubah melalui pemrograman. Kaki pin mikrokontroler ini memiliki transistor keluaran dengan kondisi *open* kolektor jenis *NPN*, dengan demikian kaki pin mikrokontroler akan sangat baik pemberian arusnya pada saat ia berlogika 0. Gambar skema dari sistem minimum mikrokontroler AT89C51 ini dapat dilihat pada Gambar 3.2.



Gambar 3.2. Rangkaian sistem minimum mikrokontroler AT89C51

Isyarat Pulsa Detak

Isyarat pulsa detak digunakan untuk menentukan kecepatan operasi pada mikrokontroler. Isyarat pulsa detak dibentuk oleh rangkaian pembangkit pulsa dengan menggunakan osilator kristal sebagai pembangkit osilasi. Pin yang digunakan untuk pewaktuan adalah pin 18 dan pin 19 pada *chip* mikrokontroler AT89C51.

Osilator yang digunakan adalah osilator kristal dengan frekuensi 12 MHz, dengan nilai kapasitor C1 dan C2 sebesar 30pF (untuk penggunaan osilator kristal C1 dan C2 = 30 pF ± 10 pF), periode per siklusnya dapat dihitung dengan rumus:

$$T \text{ per siklus} = \frac{12}{\text{Frekuensi.Xtal}} \dots\dots\dots(3.1)$$

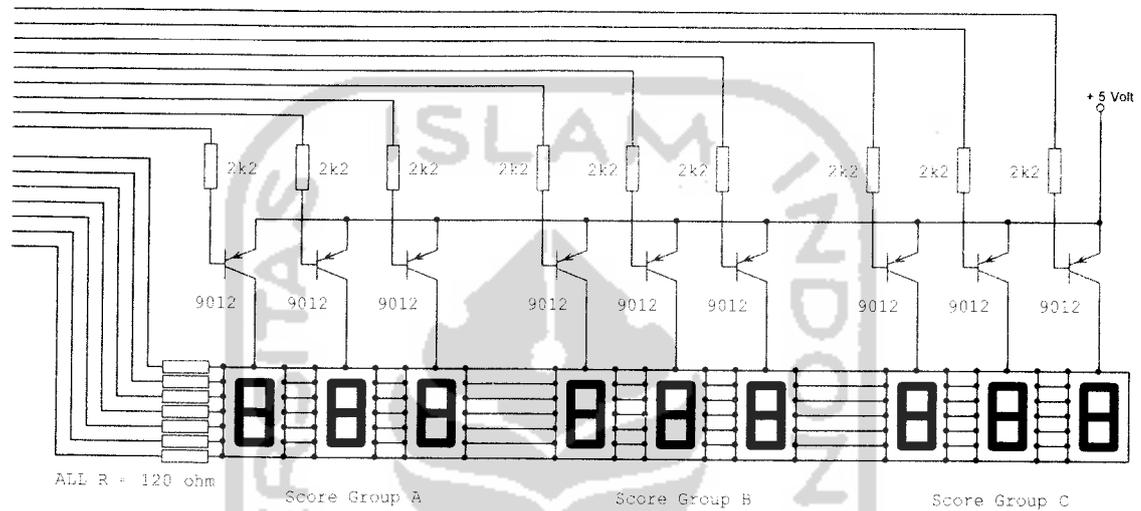
Kecepatan eksekusi suatu instruksi tergantung pada nilai perioda per siklus instruksi tersebut, pada frekuensi kristal = 12 MHz, perioda per siklusnya = 1 us

b. Rangkaian *display*

Seven segment yang digunakan pada alat ini berjumlah 9 buah karena nantinya *seven segment* ini direncanakan akan menampilkan nilai masing-masing yang masing-masingnya ada 3 digit. *Seven segmen* yang digunakan berjenis *common* anoda, atau masukan kutub positif dari catu daya yang dijadikan satu. Dipilihnya jenis *common* anoda adalah karena keluaran dari mikrokontroler memiliki arus yang kuat adalah pada saat ia memberikan logika 0. Sementara untuk pemberian arus positif dari catu daya yang juga dikontrol mikrokontroler digunakan transistor sebagai *driver*. Karena kondisi keluaran dari pin mikrokontroler yang digunakan adalah pada

logika 0, maka transistor yang digunakan adalah transistor dengan jenis *PNP* yang memiliki catu aktif untuk basisnya bernilai 0 (titik negatif catu daya).

Skema dari rangkaian display tersebut adalah sebagai berikut;



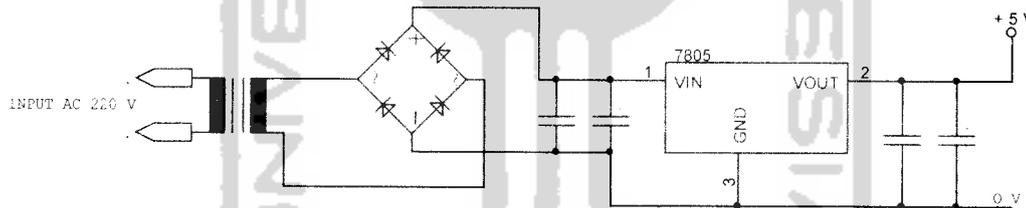
Gambar 3.3. Rangkaian *display*

Pada kondisi yang sebenarnya untuk mengaktifkan ketiga penampil *seven segmen* tersebut dibutuhkan hubungan ke *seven segmen* sebanyak $7 \times 9 = 63$ koneksi, akan tetapi pada gambar diatas hanya dibutuhkan koneksi sebanyak $7 + 9 = 16$, hal ini dapat dimungkinkan adalah karena mikrokontroler untuk menyalakan kesembilan display tersebut menggunakan sistem *scanning* atau pengulangan berulang-ulang. Oleh mata, proses pengulangan yang berulang-ulang tersebut tidaklah terlihat karena prosesnya sangat cepat sekali sehingga oleh mata, kesembilan *display* tersebut terlihat hidup secara bersamaan.

c. Rangkaian catu daya

Catu daya yang dibutuhkan untuk rangkaian papan nilai yang dapat diprogram ini adalah sebesar 5 Volt, karena komponen kontrol pada alat ini hanya menggunakan satu IC yaitu mikrokontroler AT89C51, maka trafo yang digunakan untuk mencatu rangkaian cukup digunakan trafo dengan daya suplai arus yang kecil, yang pada alat ini penulis menggunakan trafo 500 mA (mili Amper).

Untuk menstabilkan tegangan pada keluaran 5 Volt, pada catu daya ini digunakan IC regulator 7805. Rangkaian catu daya tersebut adalah seperti nampak pada gambar 3.5.



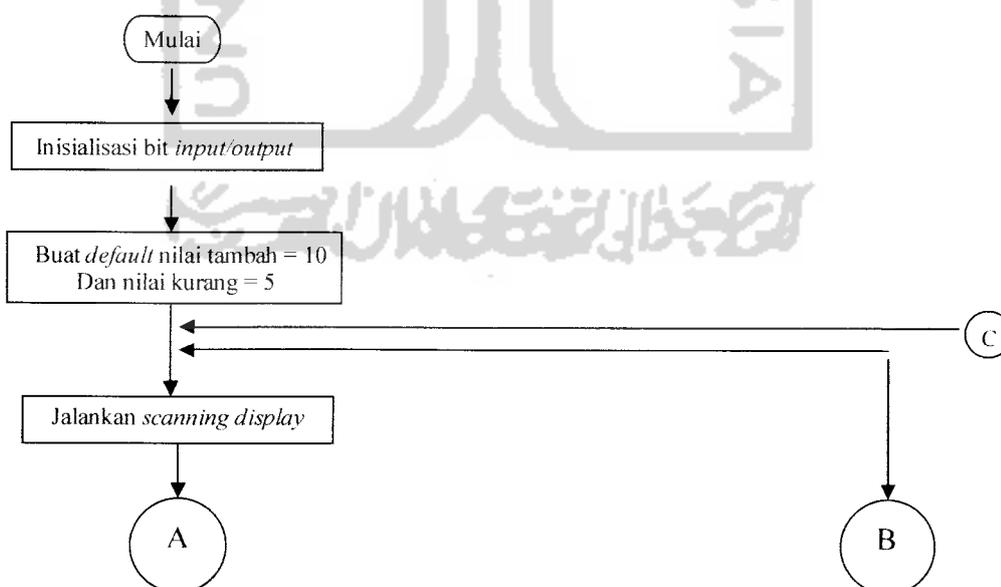
Gambar 3.4. Rangkaian catu daya

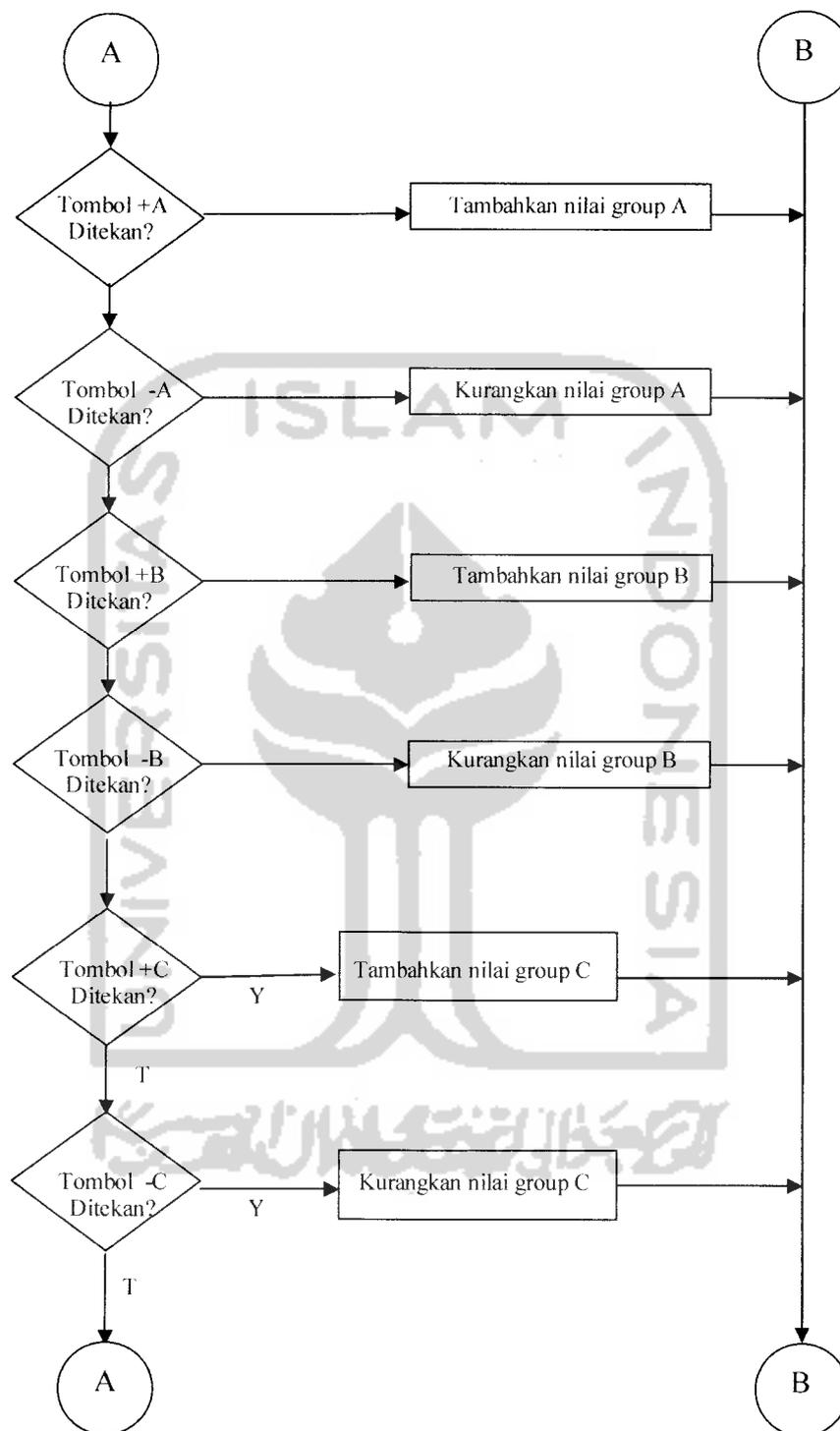
3.3. Perancangan perangkat lunak

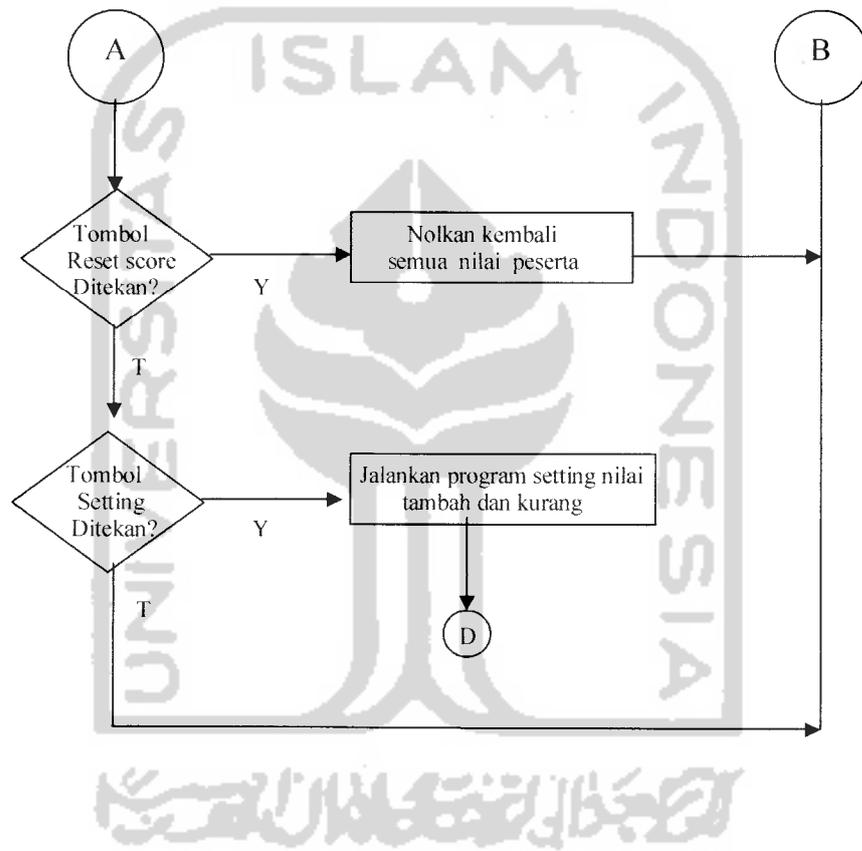
Program untuk mengatur kerja dari mikrokontroler dibuat dengan menggunakan bahasa pemrograman assembler untuk keluarga MCS-51. Program dalam bahasa assembler ini kemudian diubah menjadi format HEX agar dapat diisikan kedalam mikrokontroler. Proses format program menjadi HEX adalah, susunan

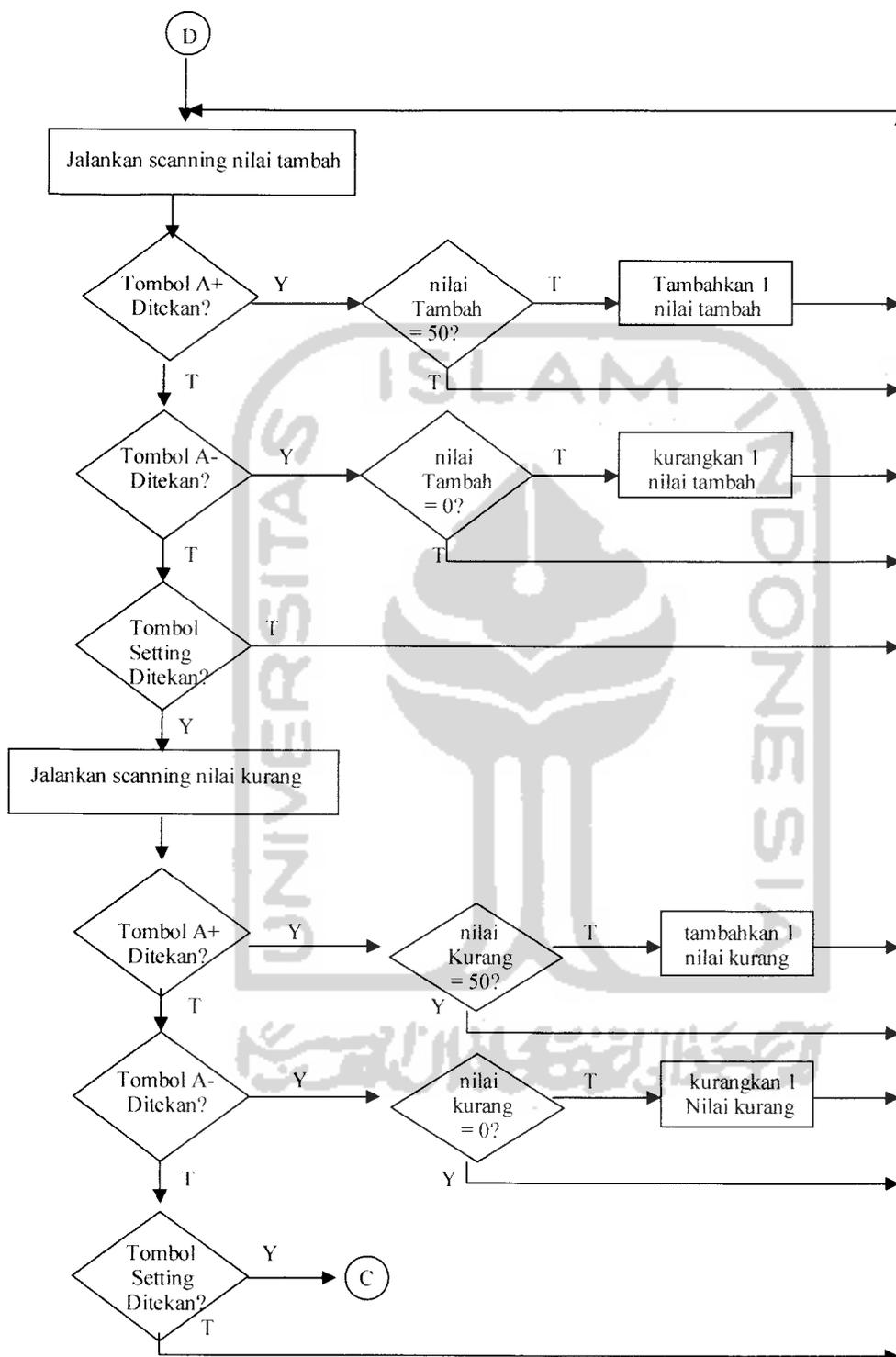
program dibuat dengan menggunakan ekstensi .ASM kemudian di *compile* dengan menggunakan program ASM51.EXE sehingga didapat sebuah program dengan ekstensi .OBJ, file .OBJ ini selanjutnya diubah menjadi format .HEX dengan menggunakan program OH.EXE.

Untuk memudahkan rancangan penyusunan program, terlebih dahulu dibuat *flow chart* atau diagram alir program, dari diagram alir inilah kemudian disusun program aslinya. Langkah lain untuk memudahkan pemeriksaan apakah program yang dibuat telah benar adalah dengan menggunakan program simulasi untuk menjalankan program yang telah selesai dibuat. Ada beberapa program untuk simulasi ini yang salah satu contohnya adalah TS-Emulator 8051. Dengan program simulasi ini, dapat diperiksa apakah potongan-potongan program yang dibuat telah benar atau belum. Gambar berikut ini adalah diagram alir dari program yang dibuat.









Gambar 3.5 Diagram alir program

Program dibagi dalam dua bagian yang pertama yaitu program pada saat acara sudah berlangsung, dimana program ini langsung diperuntukkan untuk menghitung nilai dari ketiga peserta, dimana dengan menekan tombol + atau - dari masing-masing peserta, maka nilai peserta yang bersangkutan akan berubah sesuai dengan penambahan atau pengurangan dengan nilai yang telah terekam dalam memori mikrokontroler, default awal nilai tambah yang ada dalam memori mikrokontroler adalah 10 sedangkan nilai kurang adalah 5. jadi dengan menekan nilai tambah pada salah satu peserta maka nilai peserta tersebut akan bertambah 10, dan dengan menekan nilai kurang, nilai peserta tersebut akan berkurang 5.

Program yang kedua adalah program untuk menyeting nilai penambah atau nilai pengurang yang telah ada didalam memori mikrokontroler tadi. Penyetingan dimulai dengan menekan tombol seting. Pertama kali tombol seting ditekan, program akan mengarahkan penyetingan nilai penambah, penekanan berikutnya, program akan mengarahkan penyetingan nilai pengurang, dan penekanan sekali lagi akan mengembalikan fungsi program ke penghitung nilai peserta lomba.

Berikut ini adalah potongan-potongan program, yang dibuat:

Program scanning nilai peserta

```
scanning:      mov    P0, koderata
               clr    onrata
               call   tunggu
               setb   onrata
               ;-----
               mov    P0, kodepula
               clr    onpula
               call   tunggu
               setb   onpula
               ;-----
```

```

mov    P0, kodesata
clr    onsata
call   tunggu
setb   onsata
;-----
mov    P0, koderatB
clr    onratB
call   tunggu
setb   onratB
;-----
mov    P0, kodepulB
clr    onpulB
call   tunggu
setb   onpulB
;-----
mov    P0, kodesatB
clr    onsatB
call   tunggu
setb   onsatB
;-----
mov    P0, koderatC
clr    onratC
call   tunggu
setb   onratC
;-----
mov    P0, kodepulC
clr    onpulC
call   tunggu
setb   onpulC
;-----
mov    P0, kodesatC
clr    onsatC
call   tunggu
setb   onsatC
;-----
jnb    tamA, tambahA
jnb    kurA, kurangA
jnb    tamB, tambahB
jnb    kurB, kurangB
jnb    tamC, tambahC
jnb    kurC, kurangC
jnb    resetscore, reset
jnb    setting, set
sjmp   scanning

```

Kode desimal yang akan diberikan ke *seven segment* diberikan oleh mikrokontroler pada port 0, sementara untuk penyalan bergantian (*scanning*) untuk mengontrol pemberian suplai ke masing-masing *common* dari seven segment

diberikan pada port 2 dan sebagai port 3. Pada program diatas nampak bersamaan dengan dijalankannya *scanning* juga dijalankan pemeriksaan tombol-tombol yang ada pada alat.

Program penambah dan pengurang masing-masing peserta

```
tambahA:      call  keybounce
              mov  tampungrat, rata
              mov  tampungpul, pula
              mov  tampungsat, satA
              mov  A, nilaitambah
              call  tambahkan
              call  okA
              jmp  scanning
```

```
kurangA:     call  keybounce
              mov  tampungrat, rata
              mov  tampungpul, pula
              mov  tampungsat, satA
              mov  A, nilaikurang
              call  kurangkan
              call  okA
              jmp  scanning
```

Potongan program diatas adalah merupakan potongan program untuk menambah dan mengurangi nilai peserta, pada contoh diatas adalah untuk peserta A, rutin penting yang berkaitan dengan program tersebut adalah rutin untuk menambahkan dan mengurangi nilai, pada program diatas dipanggil dengan perintah `call tambahkan` dan `call kurangkan`, berikut ini adalah rutin program tersebut.

```
tambahkan:   inc  tampungsat
             cjne tampungsat, #10, cekA1
             mov  tampungsat, #00
             inc  tampungpul
             cjne tampungpul, #10, cekA1
             mov  tampungpul, #00
             inc  tampungrat
             cjne tampungrat, #10, cekA1
             mov  tampungsat, #9
             mov  tampungpul, #9
             mov  tampungrat, #9
```

```

                                sjmp  selesai
cekA1:                            dec   A
                                jnz   A, tambahkan
selesai:                          ret
;=====
kurangkan:                        dec   tampungsat
                                cjne  tampungsat, #0ffh, cekA2
                                mov   tampungsat, #9
                                dec   tampungpul
                                cjne  tampungpul, #0ffh, cekA2
                                mov   tampungpul, #9
                                dec   tampungrat
                                cjne  tampungrat, #0ffh, cekA2
                                mov   tampungsat, #00
                                mov   tampungpul, #00
                                mov   tampungrat, #00
                                sjmp  selesai

cekA2:                            dec   A
                                jnz   A, kurangkan
                                sjmp  selesai

```

Program rutin pengubah angka desimal ke kode penampil *seven segment*

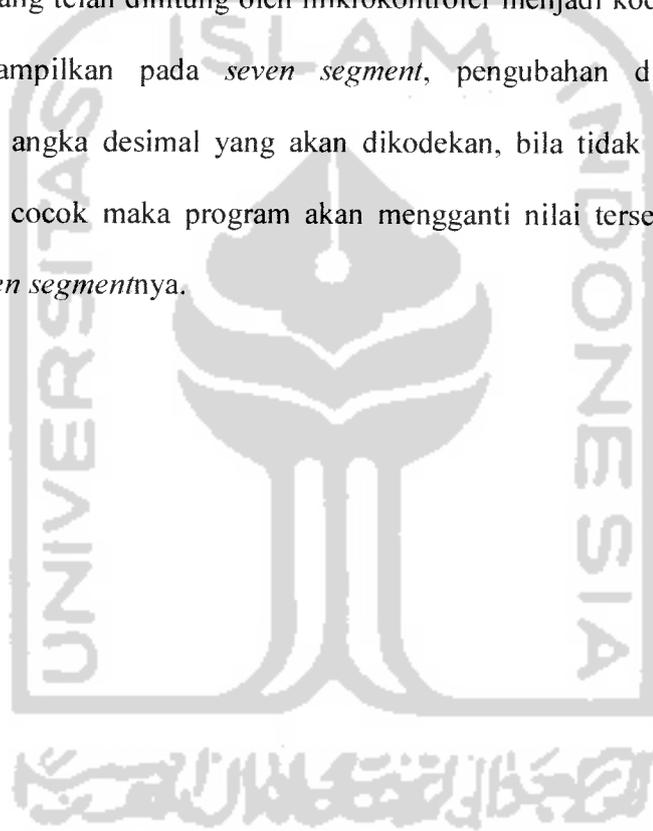
```

Banding:                          CJNE  A, #00h, angka1
                                MOV   A, #11000000b
                                SJMP  SelesaiCopy
angka1:                            CJNE  A, #01, angka2
                                MOV   A, #11111001b
                                SJMP  SelesaiCopy
angka2:                            CJNE  A, #02, angka3
                                MOV   A, #10100100b
                                SJMP  SelesaiCopy
angka3:                            CJNE  A, #03, angka4
                                MOV   A, #10110000b
                                SJMP  SelesaiCopy
angka4:                            CJNE  A, #04, angka5
                                MOV   A, #10011001b
                                SJMP  SelesaiCopy
angka5:                            CJNE  A, #05, angka6
                                MOV   A, #10010010b
                                SJMP  SelesaiCopy
angka6:                            CJNE  A, #06, angka7
                                MOV   A, #10000010b
                                SJMP  SelesaiCopy
angka7:                            CJNE  A, #07, angka8
                                MOV   A, #11111000b
                                SJMP  SelesaiCopy

```

```
angka8:      CJNE  A, #08, angka9
              MOV   A, #10000000b
              SJMP  SelesaiCopy
angka9:      MOV   A, #10010000b
selesaicopy: ret
```

Potongan program diatas adalah potongan program untuk mengubah nilai angka desimal yang telah dihitung oleh mikrokontroler menjadi kode lambang angka yang akan ditampilkan pada *seven segment*, perubahan dilakukan dengan memeriksa nilai angka desimal yang akan dikodekan, bila tidak cocok lompat ke berikutnya. Bila cocok maka program akan mengganti nilai tersebut dengan kode angka untuk *seven segment*nya.



BAB IV

PENGUJIAN SISTEM

Sebelum alat papan nilai ini dirangkai utuh dan diuji kinerjanya, dilakukan pengujian beberapa rangkaian *hardware* apakah bekerja dengan baik atau tidak. Rangkaian *hardware* yang diuji antara lain adalah, rangkaian sistem minimum mikrokontroler AT 89C51, rangkaian *display* dan rangkaian catu daya.

Untuk pengujian *software*, dapat dilakukan setelah semua blok rangkaian *hardware* telah bekerja sesuai dengan yang direncanakan. Dengan kata lain pengujian *software* juga adalah merupakan pengujian alat setelah jadi.

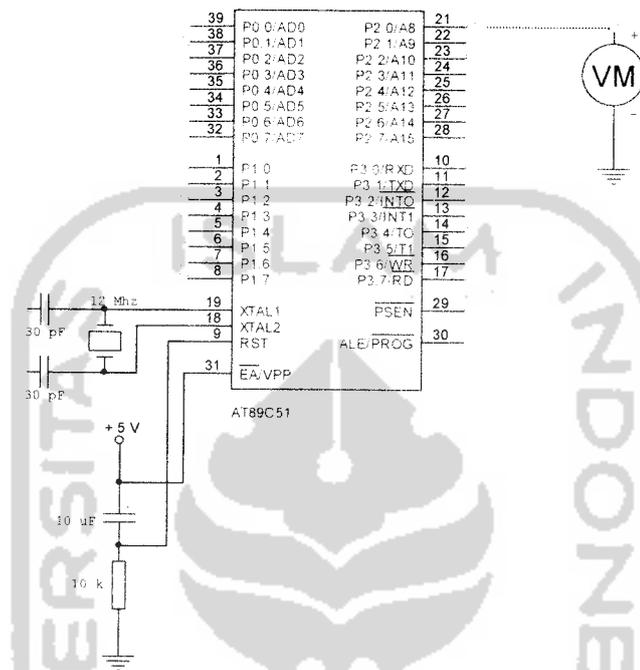
4.1. Hardware

4.1.1. Pengujian Rangkaian Sistem Minimum Mikrokontroler AT 89C51

Informasi yang diperoleh dari *datasheet* IC mikrokontroler AT 89C51 menyebutkan bahwa pada saat pertama kali mikrokontroler diberi catu daya (dihidupkan) maka secara otomatis seluruh kaki port *input/output* mikrokontroler memberikan logika 1 (*high*) atau memiliki besar tegangan yang sama atau hampir menyamai tegangan catu daya yaitu sekitar sebesar 5 Volt, dengan kondisi mikrokontroler belum diisikan program kedalam *flash memory* nya.

Mengacu pada keterangan tersebut, maka untuk pengujian rangkaian sistem minimum mikrokontroler AT 89C51 ini dilakukan dengan mengukur tegangan pada semua kaki port *input/output* pada saat mikrokontroler telah diberi catu daya (dengan

catatan mikrokontroler belum diisi program). Gambar 4.1. adalah skema rangkaian sistem minimum AT 89C51, berikut titik tempat dilakukannya pengukuran.



Gambar 4.1. Pengujian rangkaian sistem minimum mikrokontroler AT 89C51

Tabel 4.1. Data pengujian rangkaian sistem minimum AT 89C51

	Port 0	Port 1	Port 2	Port 3
0	high	high	high	high
1	high	high	high	high
2	high	high	high	high
3	high	high	high	high
4	high	high	high	high
5	high	high	high	high
6	high	high	high	high
7	high	high	high	high

Rangkaian sistem minimum mikrokontroler dapat dikatakan (relatif) baik bila dari hasil pengujian didapat data seluruh kaki port *input / output* adalah berlogika 1

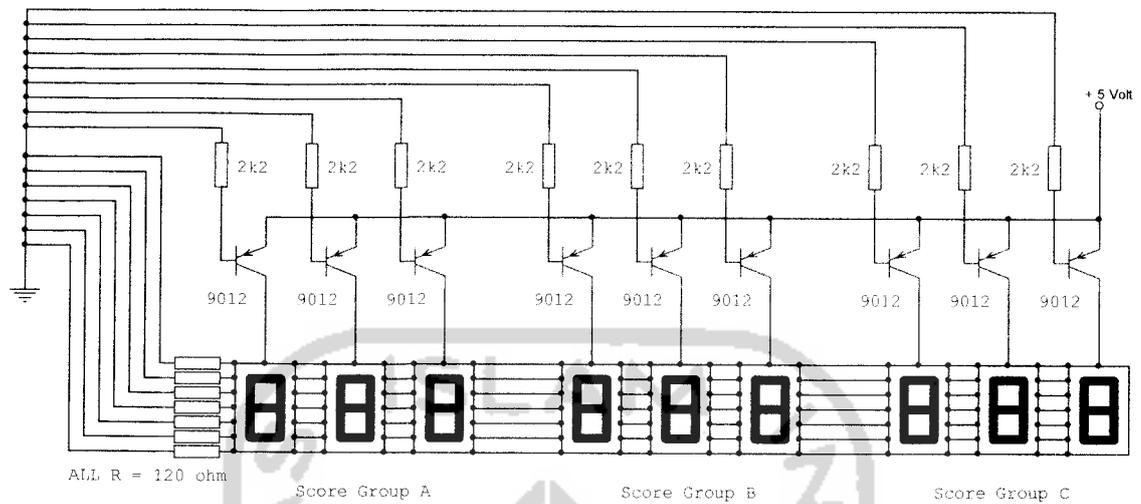
(*high*). Tabel 4.1. adalah data hasil pengamatan rangkaian sistem minimum mikrokontroler AT 89C51.

Dari tabel data pengujian diatas, diperoleh hasil bahwa semua kaki port *input / output* mikrokontroler pada saat diberi catu daya berlogika *high* hal ini dapat disimpulkan bahwa rangkaian sistem minimum telah benar.

4.1.2. Pengujian Rangkaian *Display Seven Segment*

Pengujian rangkaian *display seven segment* dilakukan dengan memberikan tegangan pada rangkaian tersebut dan masukan kontrol pada jalur masukannya, untuk masukan kontrolnya semua diberi logika 0. Pengujian rangkaian display ini adalah merupakan sekaligus pengujian transistor sebagai sakelar yang berfungsi menghubungkan catu positif pada *common* dari masing-masing seven segment. Karena transistor yang digunakan adalah merupakan transistor jenis PNP, maka dengan demikian masukan kontrol yang dibutuhkan untuk menghidupkan rangkaian *display seven segment* adalah berlogika 0 (*low*).

Rangkaian *display seven segment* dikatakan baik, bila dari hubungan yang digambarkan pada skema pengujian pada gambar 4.2. didapatkan pada *seven segment* semuanya menampilkan angka 8. Angka 8 ini mengindikasikan bahwa semua *segment* pada *seven segment* dapat menyala dengan baik. Dan dari hasil pengujian yang telah dilakukan pada rangkaian *seven segment* diatas didapat hasil semua *seven segment* menunjukkan angka 8, dengan demikian dapat dinyatakan bahwa rangkaian *display* bekerja dengan baik.



Gambar 4.2 Pengujian rangkaian display *seven segment*

4.1.3. Pengujian Rangkaian Catu Daya

Pengujian rangkaian catu daya dimaksudkan untuk memeriksa apakah catu daya yang dibuat telah mengeluarkan tegangan yang dibutuhkan oleh rangkaian. Tegangan yang dibutuhkan oleh rangkaian alat adalah sebesar 5 Volt, dengan demikian pengujian rangkaian catu yang dilakukan dengan mengukur tegangan keluarannya dengan menggunakan multimeter yang diset pada posisi pengukuran DC Volt dengan range pengukuran 10 Volt.

Dari hasil pengukuran yang dilakukan terhadap rangkaian catu daya, didapat hasil pengukuran yang menunjukkan angka pengukuran pada posisi 5 Volt dan hal ini menyatakan bahwa rangkaian catu daya bekerja dengan baik dan dapat memberikan besar tegangan catu yang direncanakan.

4.2. Software

Seperti yang telah disebutkan sebelumnya bahwa pengujian *software*, memiliki fungsi yang sama dengan pengujian alat secara keseluruhan, apabila unjuk kerja alat memberikan sesuai dengan yang yang direncanakan dapat dikatakan bahwa program yang diisikan kedalam mikrokontroler telah benar, akan tetapi apabila masih terdapat kesalahan, maka perlu diadakan perbaikan terhadap program yang telah diisikan tersebut.

Poin-poin yang akan diuji pada pengujian *software* ini adalah, pengujian penambahan dan pengurangan nilai peserta, pengujian penyetingan nilai penambah, pengujian penyetingan nilai pengurang dan pengujian penambahan dan pengurangan nilai peserta setelah nilai penambah dan pengurangnya diseting atau diubah. Berikut ini adalah data pengamatan dari pengujian beserta uraiannya.

4.2.1. Pengujian Penambah dan Pengurang Nilai Peserta

Pengujian ini dilakukan dengan menekan tombol penambah dan tombol pengurang pada masing-masing peserta dan kemudian mengamati hasilnya pada *display*. Seperti yang telah dijelaskan sebelumnya bahwa *default* awal nilai penambahan dan pengurangan yang ada didalam mikrokontroler adalah 10 dan 5. dengan demikian apabila tombol penambah pada salah satu peserta ditekan, maka nilainya akan bertambah 10 poin. Bila tombol pengurang yang ditekan, nilainya akan berkurang 5 poin. Berikut ini adalah tabel pengamatan dari pengujian tersebut.

Tabel 4.2. Pengujian Penambah dan Pengurang

Tombol yang ditekan	Even yang terjadi
A +	Nilai A bertambah 10 poin
A -	Nilai A berkurang 5 poin
B +	Nilai B bertambah 10 poin
B -	Nilai B berkurang 5 poin
C +	Nilai C bertambah 10 poin
C -	Nilai C berkurang 5 poin

Dari hasil pengujian penambah dan pengurang nilai pada masing-masing peserta didapat hasil bahwa nilai pada masing-masing peserta berubah sesuai dengan fungsi penekanan pada tombol peserta yang bersangkutan, dan ini dapat menyatakan bahwa fungsi penambah dan pengurang telah bekerja dengan benar. Pengujian penambah dan pengurang ini akan dilakukan lagi setelah nilai penambah dan nilai pengurang diset dengan nilai yang baru.

4.2.2. Pengujian Penyetingan Nilai Penambah

Pengujian penyetingan nilai penambah dilakukan dengan menekan tombol seting sekali, dengan penekanan tombol ini maka led yang berwarna hijau akan menyala dan ini mengindikasikan posisi alat sekarang pada posisi penyetingan nilai penambah. *Display* yang menyala pada saat ini hanyalah *display* digit dari peserta A. langkah pengujian selanjutnya adalah menekan tombol tambah dan kurang dari peserta dan hasilnya diamati pada *display*. Berikut ini adalah data pengamatan dari pengujian penyetingan nilai penambah.

Tabel 4.3. Data pengujian penyetingan nilai penambah

Tombol yang ditekan	Even yang terjadi
A +	Nilai penambah bertambah 1 poin
A -	Nilai penambah berkurang 1 poin

Dari pengujian didapat bahwa nilai penambah dapat diseting. Dari pengaturan program yang dilakukan, nilai untuk penambah akan memiliki nilai maksimal sebesar 50 poin. Setelah angka penambah mencapai nilai 50, penekanan tombol A + tidak akan berpengaruh apa-apa. Pada setingan nilai penambah ini, dilakukan penyetingan pada nilai 20 dan untuk membuktikan apakah setingan ini akan berpengaruh pada penekanan tombol penambah dari masing-masing peserta, akan dilakukan pengujian selanjutnya.

4.2.3. Pengujian Penyetingan Nilai Pengurang

Untuk melakukan penyetingan nilai pengurang dapat dilakukan dengan menekan tombol seting sekali lagi, dan dengan penekanan ini akan mengakibatkan led indikasi hijau mati berganti dengan led indikasi merah. Hal ini menandakan bahwa sekarang penyetingan berada pada penyetingan nilai pengurang. Langkah pengujiannya sama seperti pada penyetingan nilai pengurang. Data pengamatan dari pengujian penyetingan nilai pengurang adalah sebagai berikut:

Tabel 4.4. Data pengujian penyetingan nilai pengurang

Tombol yang ditekan	Even yang terjadi
A +	Nilai pengurang bertambah 1 poin
A -	Nilai pengurang berkurang 1 poin

Hasil pengujian mendapatkan bahwa nilai pengurang juga dapat diseting, dan pada penyetingan ini dilakukan perubahan nilai pengurang menjadi 10, dan seperti juga pada pengaturan nilai penambah, untuk mengetahui apakah perubahan nilai pengurang ini efektif, haruslah dibuktikan dengan pengujian selanjutnya.

4.2.4. Pengujian Penambah dan Pengurang Nilai Peserta Setelah Nilai Penambah dan Pengurang Diubah Menjadi 20 dan 10.

Pengujian ini sama seperti pada point 4.2.1, hanya saja kali ini adalah untuk membuktikan apakah penyetingan nilai penambah dan nilai pengurang yang telah dilakukan yaitu menjadi 20 dan 10 benar-benar telah tersimpan dan menggantikan nilai sebelumnya yaitu 10 dan 5. Berikut ini adalah data pengamatan dari pengujian tersebut

Tabel 4.5. Pengujian dan Pengurang Nilai Peserta Setelah Nilai Penambah dan Pengurang Diubah Menjadi 20 dan 10.

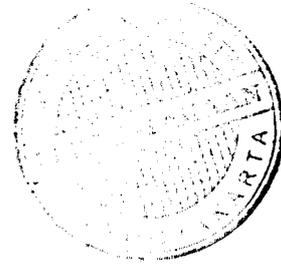
Tombol yang ditekan	Even yang terjadi
A +	Nilai A bertambah 20 poin
A -	Nilai A berkurang 10 poin
B +	Nilai B bertambah 20 poin
B -	Nilai B berkurang 10 poin
C +	Nilai C bertambah 20 poin
C -	Nilai C berkurang 10 poin

Dari hasil pengujian, didapat bahwa nilai penambah dan nilai pengurang yang telah diseting memang telah benar-benar menggantikan nilai penambah dan nilai pengurang sebelumnya. Dan dari kesemua pengujian diatas dapat disimpulkan bahwa program yang dibuat telah berfungsi dengan baik



BAB V

PENUTUP



5.1. Kesimpulan

Dari kerja yang telah dilakukan mulai dari perencanaan, perakitan dan pengujian diambil beberapa kesimpulan sebagai berikut.

1. Alat yang dibuat telah dapat berfungsi dengan baik seperti yang telah direncanakan pada awal penelitian, dimana alat tersebut dapat menampilkan nilai peserta dengan nilai penambah dan pengurang yang dapat diseting.
2. Penggunaan mikrokontroler pada alat ini sangat membantu dalam hal penyederhanaan rangkaian, fleksibilitas penggunaan alat, dan pengeluaran biaya pembuatan yang relatif murah.
3. Sistem *scanning* untuk menangani jumlah penampil *seven segment* yang banyak merupakan solusi yang sangat baik sekali untuk memaksimalkan pemakaian pin port dari mikrokontroler.

5.2. Saran

Untuk mengembangkan alat pada penelitian yang akan dilakukan selanjutnya, dibawah ini diberikan saran sebagai berikut.

1. Mendesain untuk rangkaian yang dapat menggunakan *seven segment* dalam ukuran yang besar sehingga dapat diterapkan pada pemakaian yang sesungguhnya.
2. Melengkapi fasilitas alat dengan pengaturan tombol-tombol, lampu-lampu dan bel yang juga diatur oleh mikrokontroler, sehingga didapat suatu alat pengaturan untuk acara kuis yang komplit berbasis mikrokontroler.
3. Melengkapi alat dengan SEEPROM(*Serial Electrical Erasable and Programable ROM*), sehingga apabila pada saat alat dipakai tiba-tiba aliran listrik mati, nilai masing-masing peserta masih tersimpan dengan baik.

DAFTAR PUSTAKA

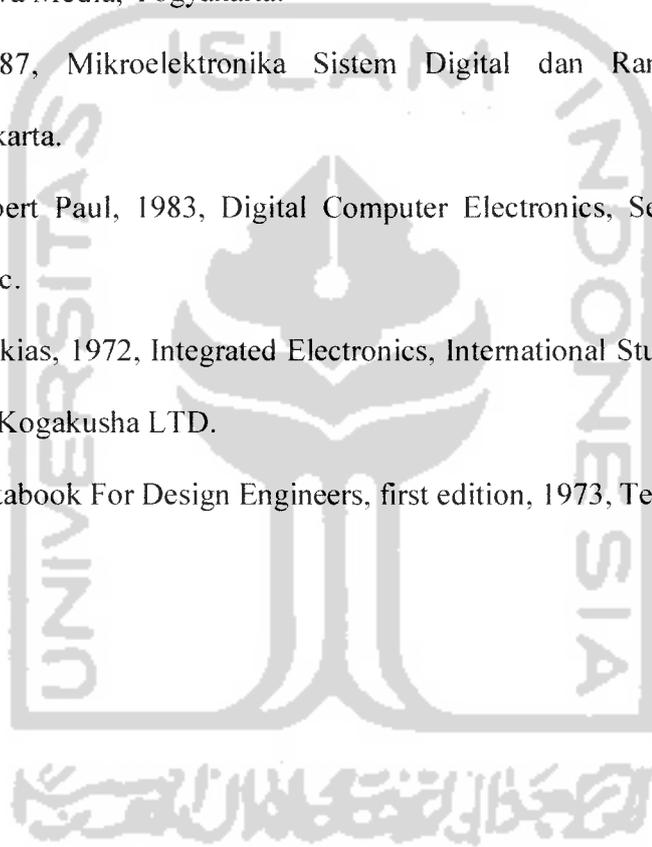
Agfianto Eko Putra, 2002, Belajar Mikrokontroler AT89C51/52/55, Teori dan Aplikasi, Gava Media, Yogyakarta.

Sutanto, 1987, Mikroelektronika Sistem Digital dan Rangkaian Analog, Erlangga, Jakarta.

Malvino Albert Paul, 1983, Digital Computer Electronics, Secon edition, Mc Graw-Hill Inc.

Millman-Halkias, 1972, Integrated Electronics, International Student Edition, Mc Graw-Hill – Kogakusha LTD.

The TTL Databook For Design Engineers, first edition, 1973, Texas Instruments.



Lampiran I

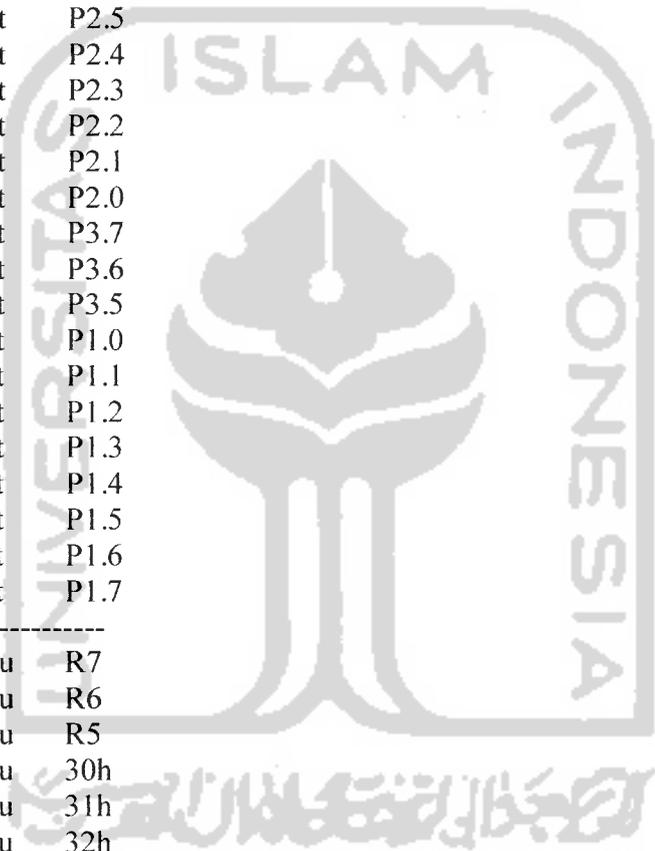
```
=====
;program papan score yang dapat diprogram,
;berbasis mikrokontroler AT89C51
;by Yudi Mulyana, FTI-Elektro-UII
=====
```

```
;Inisialisasi bit input/output
;angka untuk segmen pada port 0, -gfedcba
```

```
onratA      bit    P2.7
onpula      bit    P2.6
onsatA      bit    P2.5
onratB      bit    P2.4
onpulB      bit    P2.3
onsatB      bit    P2.2
onratC      bit    P2.1
onpulC      bit    P2.0
onsatC      bit    P3.7
ledmer      bit    P3.6
ledhij      bit    P3.5
tamA        bit    P1.0
kurA        bit    P1.1
tamB        bit    P1.2
kurB        bit    P1.3
tamC        bit    P1.4
kurC        bit    P1.5
resetscore  bit    P1.6
setting     bit    P1.7
```

```
-----
tampungrat  equ    R7
tampungpul  equ    R6
tampungsat  equ    R5
ratA        equ    30h
pula        equ    31h
satA        equ    32h
ratB        equ    33h
pulB        equ    34h
satB        equ    35h
ratC        equ    36h
pulC        equ    37h
satC        equ    38h
```

```
-----
koderatA    equ    40h
kodepulaA   equ    41h
```



Lampiran 2

```

kodesatA    equ    42h
koderatB    equ    43h
kodepulB    equ    44h
kodesatB    equ    45h
koderatC    equ    46h
kodepulC    equ    47h
kodesatC    equ    48h

```

```

;-----
nilaitambah equ    50h
nilaikurang equ    51h
koderatniltam equ    52h
kodepulniltam equ    53h
kodesatniltam equ    54h
koderatnilkur equ    55h
kodepulnilkur equ    56h
kodesatnilkur equ    57h
niltamrat   equ    60h
niltampul   equ    61h
niltamsat   equ    62h
nilkurrat   equ    63h
nilkurpul   equ    64h
nilkursat   equ    65h

```

```

;-----
ORG 0h
lag1: mov R0, 40h           ;isikan alamat awal kode segmen 40h
      mov @R0, #11000000b ;buat semua tampilan menjadi 000
      inc R0              ;naikkan nilai alamat
      cjne R0, #49h, lag1 ;kalau belum 49h ulangi
      mov tmod, #01h     ;setting timer 0, 16 bit
      mov nilaitambah, #10 ;default nilai tambah
      mov nilaikurang, #5 ;default nilai kurang
      clr A               ;untuk semua nilai awal adalah kosong
lag2: mov R0, #30h        ;isikan alamat awal nilai 30h
      mov @R0, A          ;buat semua isi alamat = 0
      inc R0              ;naikkan nilai alamat
      cjne R0, #39h, lag2 ;kalau belum 39h ulangi

```

```

;-----
;default kode angka nilai tambah dan kurang
;-----

```

```

      mov koderatniltam, #11000000b
      mov kodepulniltam, #11111100b
      mov kodesatniltam, #11000000b

```

Lampiran 3

```
mov koderatnilkur, #11000000b
mov kodepulnilkur, #11000000b
mov kodesatnilkur, #10010010b
```

```
=====
;scanning angka pada display
;=====
```

```
scanning:      mov P0, koderatA
                clr  onratA
                call tunggu
                setb onratA
                ;-----
                mov P0, kodepulA
                clr  onpulA
                call tunggu
                setb onpulA
                ;-----
                mov P0, kodesatA
                clr  onsataA
                call tunggu
                setb onsataA
                ;-----
                mov P0, koderatB
                clr  onratB
                call tunggu
                setb onratB
                ;-----
                mov P0, kodepulB
                clr  onpulB
                call tunggu
                setb onpulB
                ;-----
                mov P0, kodesatB
                clr  onsataB
                call tunggu
                setb onsataB
                ;-----
                mov P0, koderatC
                clr  onratC
                call tunggu
                setb onratC
```

Lampiran 4

```
-----  
mov    P0, kodepulC  
clr    onpulC  
call   tunggu  
setb   onpulC  
-----  
mov    P0, kodesatC  
clr    onsatC  
call   tunggu  
setb   onsatC  
-----  
jnb    tamA, tambahA  
jnb    kurA, kurangA  
jnb    tamB, tambahB  
jnb    kurB, kurangB  
jnb    tamC, tambahC  
jnb    kurC, kurangC  
jnb    resetscore, reset  
jnb    setting, set  
sjmp   scanning  
set:   ljmp   setI  
-----  
reset: mov    R0, 40h  
lag11: mov    @R0, #11000000b  
       inc    R0  
       cjne   R0, #49h, lag11  
       clr    A  
lag22: mov    R0, #30h  
       mov    @R0, A  
       inc    R0  
       cjne   R0, #39h, lag22  
       sjmp   scanning  
-----  
tambahA: call   keybounce  
         mov   tampunrat, ratA  
         mov   tampungpul, pulA  
         mov   tampungsat, satA  
         mov   A, nilaitambah  
         call  tambahkan  
         call  okaA  
         jmp   scanning
```

Lampiran 5

```
kurangA:      call    keybounce
              mov     tampungrat, ratA
              mov     tampungpul, pulA
              mov     tampungsat, satA
              mov     A, nilaikurang
              call    kurangkan
              call    okA
              jmp     scanning
```

```
;-----
tambahB:     call    keybounce
              mov     tampungrat, ratB
              mov     tampungpul, pulB
              mov     tampungsat, satB
              mov     A, nilaitambah
              call    tambahkan
              call    okB
              jmp     scanning
```

```
kurangB:     call    keybounce
              mov     tampungrat, ratB
              mov     tampungpul, pulB
              mov     tampungsat, satB
              mov     A, nilaikurang
              call    kurangkan
              call    okB
              jmp     scanning
```

```
;-----
tambahC:     call    keybounce
              mov     tampungrat, ratC
              mov     tampungpul, pulC
              mov     tampungsat, satC
              mov     A, nilaitambah
              call    tambahkan
              call    okC
              jmp     scanning
```

```
kurangC:     call    keybounce
              mov     tampungrat, ratC
              mov     tampungpul, pulC
              mov     tampungsat, satC
```

Lampiran 6

```
mov    A, nilaikurang
call   kurangkan
call   okC
jmp    scanning
;=====
tambahkan:  inc    tampungsat
           cjne   tampungsat, #10, cekA1
           mov    tampungsat, #00
           inc    tampungpul
           cjne   tampungpul, #10, cekA1
           mov    tampungpul, #00
           inc    tampungrat
           cjne   tampungrat, #10, cekA1
           mov    tampungsat, #9
           mov    tampungpul, #9
           mov    tampungrat, #9
           sjmp   selesai
cekA1:     dec    A
           jnz    A, tambahkan
selesai:   ret
;=====
kurangkan:  dec    tampungsat
           cjne   tampungsat, #0ffh, cekA2
           mov    tampungsat, #9
           dec    tampungpul
           cjne   tampungpul, #0ffh, cekA2
           mov    tampungpul, #9
           dec    tampungrat
           cjne   tampungrat, #0ffh, cekA2
           mov    tampungsat, #00
           mov    tampungpul, #00
           mov    tampungrat, #00
           sjmp   selesai
cekA2:     dec    A
           jnz    A, kurangkan
           sjmp   selesai
;=====
;rutin perubahan angka ke kode seven segmen
;=====
```

Lampiran 7

```
okA:      mov    ratA, tampungrat
          mov    pulA, tampungpul
          mov    satA, tampungsat
          mov    A, ratA
          call   banding
          mov    koderatA, A
          mov    A, pulA
          call   banding
          mov    kodepulA, A
          mov    A, satA
          call   banding
          mov    kodesatA, A
          ret
```

```
;
okB:      mov    ratB, tampungrat
          mov    pulB, tampungpul
          mov    satB, tampungsat
          mov    A, ratB
          call   banding
          mov    koderatB, A
          mov    A, pulB
          call   banding
          mov    kodepulB, A
          mov    A, satB
          call   banding
          mov    kodesatB, A
          ret
```

```
;
okC:      mov    ratC, tampungrat
          mov    pulC, tampungpul
          mov    satC, tampungsat
          mov    A, ratC
          call   banding
          mov    koderatC, A
          mov    A, pulC
          call   banding
          mov    kodepulC, A
          mov    A, satC
          call   banding
          mov    kodesatC, A
          ret
```

```
;=====
```

Lampiran 8

Banding:

```
CJNE A, #00h, angka1
MOV A, #11000000b
SJMP SelesaiCopy
angka1: CJNE A, #01, angka2
MOV A, #11111001b
SJMP SelesaiCopy
angka2: CJNE A, #02, angka3
MOV A, #10100100b
SJMP SelesaiCopy
angka3: CJNE A, #03, angka4
MOV A, #10110000b
SJMP SelesaiCopy
angka4: CJNE A, #04, angka5
MOV A, #10011001b
SJMP SelesaiCopy
angka5: CJNE A, #05, angka6
MOV A, #10010010b
SJMP SelesaiCopy
angka6: CJNE A, #06, angka7
MOV A, #10000010b
SJMP SelesaiCopy
angka7: CJNE A, #07, angka8
MOV A, #11111000b
SJMP SelesaiCopy
angka8: CJNE A, #08, angka9
MOV A, #10000000b
SJMP SelesaiCopy
angka9: MOV A, #10010000b
selesaicopy: ret
```

```
=====
;
;rutin rutin delay
;
=====
```

```
keybounce: clr TR0
mov TH0, #high (-50000)
mov TL0, #low (-50000)
setb TR0
jnb TF0, $
clr TR0
ret
```

tunggu:

Lampiran 9

```
MOV R3, #0ffh
djnz R3, $
djnz R3, $
MOV P0, #1111111B ;Padamkan tampilan
djnz R3, $
RET
```

```
=====
;setting nilai tambah dan kurang
=====
setl:      call  keybounce
           clr   ledhij ;indikasi setting nilai tambah

scanningl: mov   P0, koderatniltam
           clr   onratA
           call  tunggu
           setb  onratA
           ;-----
           mov   P0, kodepulniltam
           clr   onpula
           call  tunggu
           setb  onpula
           ;-----
           mov   P0, kodesatniltam
           clr   onsata
           call  tunggu
           setb  onsata

           jnb  tamA, plusl
           jnb  kurA, minl
           jnb  setting, kenilaikur
           sjmp scanningl
```

```
-----
plusl:     call  keybounce
           mov   A, nilaitambah
           inc   A
           call  ceklagi
           sjmp scanningl
-----
```

Lampiran 10

```
min1:      call  keybounce
           mov  A, nilaitambah
           dec  A
           call ceklagi
           sjmp scanning1
```

```
;-----
;rutin pembaharuan nilai tambah
;-----
```

```
ceklagi:   mov  nilaitambah, A
           mov  B, #100
           div  AB
           mov  niltamrat, A
           mov  A, B
           mov  B, #10
           div  AB
           mov  niltampul, A
           mov  niltamsat, B
           mov  A, niltamrat
           call banding
           mov  koderatniltam, A
           mov  A, niltampul
           call banding
           mov  kodepulniltam, A
           mov  A, niltamsat
           call banding
           mov  kodesatniltam, A
           ret
```

```
;-----
kenilaikur: setb ledhij
           clr  ledmer ;indikasi setting nilai kurang
           call keybounce
```

```
scanning2: mov  P0, koderatnilkur
           clr  onratA
           call tunggu
           setb onratA
           ;-----
           mov  P0, kodepulnilkur
           clr  onpula
           call tunggu
           setb onpula
```

Lampiran 11

```
;------  
mov    P0, kodesatnilkur  
clr    onsatA  
call   tunggu  
setb   onsatA  
  
jnb    tamA, plus1b  
jnb    kurA, min1b  
jnb    setting, score  
sjmp   scanning2  
  
;------  
score:  
call   keybounce  
setb   ledmer  
ljmp   scanning  
  
;------  
plus1a:  
call   keybounce  
mov    A, nilaikurang  
inc    A  
call   ceklagi2  
sjmp   scanning2  
  
;------  
min1a:  
call   keybounce  
mov    A, nilaikurang  
dec    A  
call   ceklagi2  
sjmp   scanning2  
  
;------  
;rutin pembaharuan nilai kurang  
;------  
ceklagi2:    mov    nilaikurang, A  
              mov    B, #100  
              div    AB  
              mov    nilkurrat, A  
              mov    A, B  
              mov    B, #10  
              div    AB  
              mov    nilkurpul, A  
              mov    nilkursat, B
```

Lampiran 12

```
mov A, nilkurrat  
call banding  
mov koderatnilkur, A  
mov A, nilkurpul  
call banding  
mov kodepulnilkur, A  
mov A, nilkursat  
call banding  
mov kodesatnilkur, A  
ret
```

end



