



**DISPLAY MATRIX
BERBASIS MIKROKONTROLER AT89S51**

TUGAS AKHIR

**Diajukan dalam rangka penyelesaian Studi Diploma III
Untuk mencapai gelar ahli Madya**

oleh :

Nama : Moh Hasanuddin
NIM : 5352302033
Prodi : Diploma III Teknik Elektro
Jurusan : Teknik Elektro
Fakultas : Teknik

**FAKULTAS TEKNIK
UNIVERSITAS NEGERI SEMARANG**

2007

ABSTRAK

Moh Hasanuddin, 2007, *Display Matrix Berbasis Mikrokontroler*, Tugas akhir, Teknik Elektro Universitas Negeri Semarang.

Kemajuan teknologi elektronik mendorong pola hidup masyarakat yang cenderung semakin praktis hal tersebut juga dirasakan oleh civitas akademika kampus baik dosen, karyawan, dan mahasiswa. Sebagai contoh dalam lingkup yang kecil adalah mahasiswa, sebagai mahasiswa tentunya sangat akrab dengan Dekan dan Pembantu Dekannya, seorang mahasiswa sudah terbiasa mencari Dekan maupun Pembantu Dekannya untuk konsultasi, bimbingan, atau keperluan lain yang menyangkut dengan perkuliahan.

Dalam proses pencarian tersebut kadang-kadang mahasiswa bingung karena tidak ada keterangan atau papan informasi yang bekerja secara otomatis di kampus mengenai ada dan tidak adanya Dekan maupun Pembantu Dekan. Untuk mencarinya sering mahasiswa bolak-balik kekampus untuk mengecek apakah orang yang di cari sudah datang apa belum dan biasanya pencarian itu di akhiri dengan rasa kecewa karena sudah di tunggu berjam-jam ataupun bola-balik dari kost ke kampus tidak datang juga.

Permasalahan yang timbul diantaranya yaitu : bagaimana display matrik ini dapat bekerja dengan baik dan secara otomatis tanpa bantuan tenaga manusia atau manual menginformasikan bahwa Dekan atau Pembantu Dekan ada dan tidak ada diruangannya, bagaimana cara membuat tulisan yang bervariasi dan menarik dibaca.

Hasil dari perancangan, pembuatan, pengujian dan percobaan alat display matrix berbasis mikrokontroler ini dapat diambil kesimpulan sebagai berikut : Informasi yang diberikan kepada mahasiswa pada alat ini sudah cukup jelas karena ditampilkan menggunakan LED sebanyak 350 buah sebagai pengganti dot matrik yaitu berupa lampu led 5X7 perbloknya, dan ada 10 blok. Waktu yang dibutuhkan oleh alat informasi ini untuk menampilkan tulisan Dekan ada dan tidak ada kepada mahasiswa relatif singkat yaitu setiap 1 menit tulisan berganti Dekan, PD I sampai PD III ada dan tidak ada.

Untuk mengatasi kesulitan membaca dari jarak jauh, maka dapat dilakukan dengan menyempurnakan pembuatan sistem mekaniknya diperbesar dan tulisannya diubah menjadi perblok menggunakan LED yang besar.

HALAMAN PENGESAHAN

Laporan Tugas Akhir ini telah dipertahankan dihadapan sidang penguji Tugas Akhir Fakultas Teknik Universitas Negeri Semarang pada:

Hari :

Tanggal :

Pembimbing

Drs. I Made Sudana, M.Pd.
NIP.131404314

Penguji II :

Penguji I:

Drs.I Made Sudana, M.Pd.
NIP.131404314

Drs. Suryono, MT.
NIP.131404314

Ketua Jurusan

Ketua Program Studi D-III

Drs. Djoko Adi Widodo, M.T.
NIP. 131570064

Drs. Agus Murnomo, M.T
NIP.131616610

Dekan,

Prof.DR. Soesanto, M. Pd.
NIP. 130875753

MOTTO DAN PERSEMBAHAN

1. Tidak ada jalan pintas menuju kesuksesan, kesuksesan akan datang pada mereka yang berusaha mendapatkannya bukan pada mereka yang hanya mengharapkannya. Jangan pernah putus asa karena yang mudah putus asa tidak pernah sukses dan orang sukses tidak pernah putus asa. Ok!!!
2. Seorang alim yang tidak mengamalkan ilmunya, maka nasehatnya akan lenyap dari hati orang yang mendengarnya, sebagaimana hilangnya setetes embun di atas batu yang halus. (Malik bin Dinar)
3. Tampilannya seperti ujian, tapi isinya adalah rahmat dan kenikmatan. Berapa banyak kenikmatan yang sungguh besar baru diperoleh setelah melalui ujian. (Miftah Darus Sa'adah, 1/299).
4. Perjalanan panjang hanya bisa ditempuh dengan keseriusan dan berjalan waktu malam. Jika seorang musafir menyimpang dari jalan, dan menghabiskan waktu malamnya untuk tidur, kapan ia akan sampai ketujuan ?. (Ibnu Qayyim)

Seiring rasa syukur atas nikmat dan rahmat ALLAH SWT, laporan ini kupersembahkan untuk :

- ♥ Ayah dan ibundaku yang senantiasa mengiringi setiap langkahku dengan doa, mendidik dan memberi dorongan untuk mampu menuju kesuksesan.
- ♥ Kakak2ku tercinta yang selalu memberikan inspirasi dan semangat.
- ♥ My De'2 yang telah memberi semangat, dan gagasan segar dalam meniti lika-liku kehidupanku
- ♥ Ikhwah fillah, trimakasih atas ukhuwah, dukungan dan nasehatnya,uhibbuka fillah.
- ♥ Dan Semuanya!

KATA PENGANTAR

Puji dan syukur kehadiran Allah SWT karena dengan rahmat dan hidayah-Nya, penulis dapat menyelesaikan tugas akhir dengan judul: “Display Matrix berbasis mikrokontroler AT 89S51 “, guna memenuhi syarat menyelesaikan program Diploma III pada program studi teknik elektro instrumentasi kendali Fakultas Teknik Universitas Negeri Semarang.

Untuk itu penulis mengucapkan terimakasih yang sebanyak-banyaknya kepada:

1. Dekan Fakultas Teknik Universitas Negeri Semarang, Bapak Prof. Dr. Soesanto, M. Pd
2. Ketua Jurusan Teknik Elektro Fakultas Teknik Universitas Negeri Semarang, Bapak Drs. Djoko Adi Widodo, M.T.
3. Dosen pembimbing Tugas Akhir, Bapak Drs. I Made Sudana, M.Pd, yang telah meluangkan waktunya untuk membimbing hingga terselesaikannya laporan ini.
4. Kepala Laboratorium Teknik Elektro, Bapak Drs. Suryono, M.T, yang telah memberikan ijin dalam penggunaan fasilitas laboratorium untuk menyelesaikan tugas akhir.
5. Ayah, Ibu, dan kakak2ku yang telah memberi dukungan, semangat dan do'anya disetiap munajatnya.
6. Saudara-saudaraku seperjuangan teknik elektro instrumentasi kendali, instalasi listrik, PTE angkatan 2002 dan 2001
7. Semua pihak yang telah banyak membantu hingga selesai kuliah.

Meskipun tugas akhir ini telah diusahakan dengan sebaik-baiknya, tetapi penulis menyadari bahwa tugas akhir ini masih banyak kekurangan,dengan segala kerendahan hati penulis mengharapkan kritik dan saran yang membangun dari pembaca.

Semarang, Januari 2007

Penulis

DAFTAR ISI

	Halaman
HALAMAN JUDUL	i
ABSTRAK	ii
HALAMAN PENGESAHAN	iii
MOTTO DAN PERSEMBAHAN	iv
KATA PENGANTAR	v
DAFTAR ISI	vii
DAFTAR GAMBAR	ix
DAFTAR TABEL	x
BAB I PENDAHULUAN	
A. Latar Belakang	1
B. Permasalahan	2
C. Pembatasan masalah.....	2
D. Tujuan	3
E. Manfaat	3
BAB II ISI	
A. Landasan Teori.....	4
1. Mikrokontroler	4
2. Transistor	18
3. Kondensator	19
4. Resistor	20
5. LED.....	21

6. IC	22
B. Pembuatan	24
1. Alat dan Bahan	24
2. Perencanaan Desain	25
3. Pembuatan Program	26
4. Proses Pembuatan Benda Kerja	40
5. Pengujian	44
6. Hasil Pengujian	46

BAB III PENUTUP

A. Kesimpulan	47
B. Saran	47

DAFTAR PUSTAKA	49
-----------------------------	----

LAMPIRAN

DAFTAR GAMBAR

Gambar 1. Diagram Blok AT89S51	5
Gambar 2. Konfigurasi Pin AT 89S51	5
Gambar 3. Peta Register Fungsi Khusus - SFR	11
Gambar 4. Simbol Transistor	18
Gambar 5. Simbol Kondensator	19
Gambar 6. Simbol Resistor	21
Gambar 7. Simbol LED.....	22
Gambar 8. Simbol IC	23
Gambar 9. Gambar Mekanik Tampak Depan dan Samping	25
Gambar 10. Flowchart.....	27
Gambar 11. Diagram Blok Display Matrix.....	45

DAFTAR TABEL

Tabel 1. Fungsi Khusus Port 3	8
Tabel 3. Daftar Bahan	25
Tabel 4. Hasil Percobaan.....	47

DAFTAR LAMPIRAN

Data Sheet IC 74HC573.....	50
Data Sheet Transistor BC327.....	57
Data Sheet IC ULN2003.....	61
Data Sheet Mikrokontroler AT89S51.....	69
Gambar Rangkaian Display Matrix.....	71

BAB I

PENDAHULUAN

A. Latar Belakang Masalah

Mikrokontroler, sebagai suatu terobosan teknologi mikroprosesor dan mikrokomputer, hadir memenuhi kebutuhan pasar (market need) dan teknologi baru. Sebagai teknologi baru, yaitu teknologi semikonduktor dengan kandungan transistor yang lebih banyak namun hanya membutuhkan ruang yang kecil serta dapat diproduksi secara massal (dalam jumlah banyak) membuat harganya menjadi lebih murah (dibandingkan mikroprosesor).

Sebagai kebutuhan pasar, mikrokontroler hadir untuk memenuhi selera industri dan para konsumen akan kebutuhan dan keinginan alat-alat bantu bahkan mainan yang lebih baik dan canggih. Kemajuan teknologi elektronika mendorong pola hidup masyarakat yang cenderung semakin praktis. Hal tersebut juga dirasakan oleh civitas akademika kampus baik dosen, karyawan, dan mahasiswa. Sebagai contoh dalam lingkup yang kecil adalah mahasiswa, sebagai mahasiswa tentunya sangat akrab dengan Dekan dan Pembantu Dekannya untuk konsultasi, bimbingan, atau keperluan lainnya yang menyangkut dengan perkuliahan. Dalam proses pencarian tersebut kadang-kadang mahasiswa bingung karena tidak adanya keterangan atau papan informasi di fakultas mengenai ada dan tidak ada Dekan maupun Pembantu Dekan tersebut.

Untuk mencarinya sering mahasiswa bolak-balik ke kampus apakah orang yang ia cari sudah datang apa belum dan biasanya pencarian itu diakhiri dengan

rasa kecewa karena sudah ditunggu berjam-jam ataupun bolak-balik dari kampus ke kos tidak datang juga. Untuk itu di buat suatu alat yang fungsinya memberi informasi kepada mahasiswa bahwa Dekan dan Pembantu Dekan ada atau tidak ada dikantor berupa display matrik dengan mikrokontroler AT89S51.

B. Permasalahan

Dari latar belakang diatas dapat diperoleh permasalahan yang timbul diantaranya yaitu :

1. Bagaimana display matrix ini dapat bekerja dengan baik dan secara otomatis tanpa bantuan tenaga manusia atau manual.
2. Bagaimana cara membuat tulisan didisplay matrik yang bervariasi dan menarik di baca.

C. Pembatasan Masalah

Untuk menghindari permasalahan baru yang tidak sesuai dengan tujuan dari tugas akhir ini dan agar tidak keluar dari pokok bahasan yang disajikan, maka perlu adanya pembatasan masalah sebagai berikut:

1. Display matrix ini menggunakan mikrokontroler AT89S51,
2. Display matrix ini menggunakan led sebanyak 5X7 perblok sebagai pengganti dot matrix untuk tampilan tulisannya .

D. Tujuan

1. Mengembangkan ilmu pengetahuan dan teknologi sebagai hasil karya yang dapat dimanfaatkan di kampus
2. Membuat program tulisan pada display matrix menggunakan led berbasis mikrokontroler.

E. Manfaat

1. Memberi kontribusi ilmu pengetahuan dan teknologi pada dunia elektronika dan sekaligus bisa diterapkan dikampus.
2. Dengan menggunakan bahasa program display matrix ini dapat diatur/diubah-ubah tulisannya.

BAB II

ISI

A. Landasan Teori

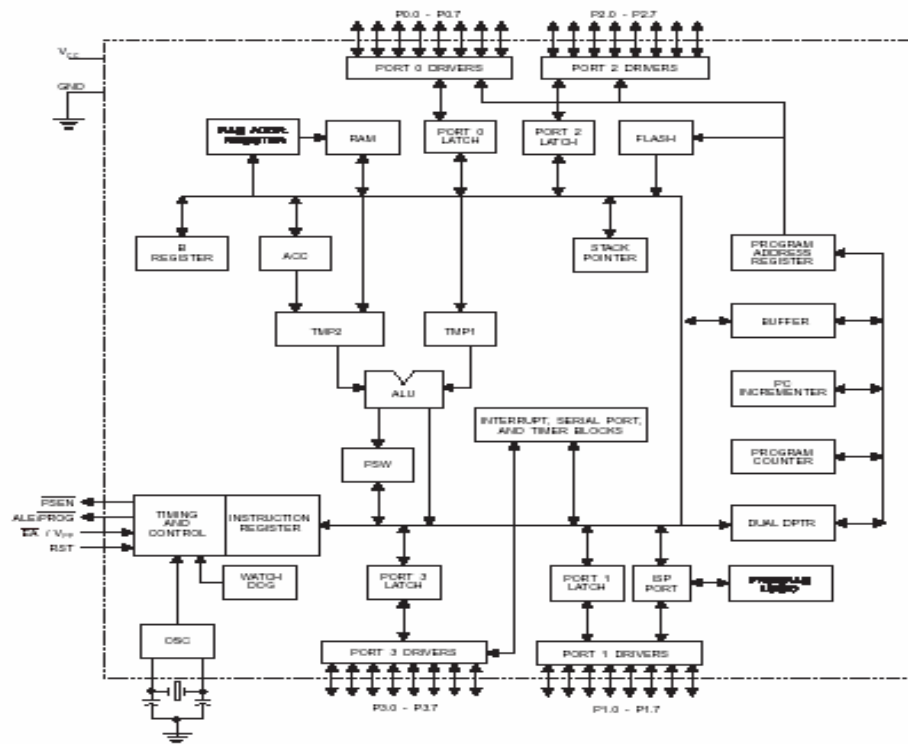
1. Mikrokontroler AT 89S51

1.1 Arsitektur

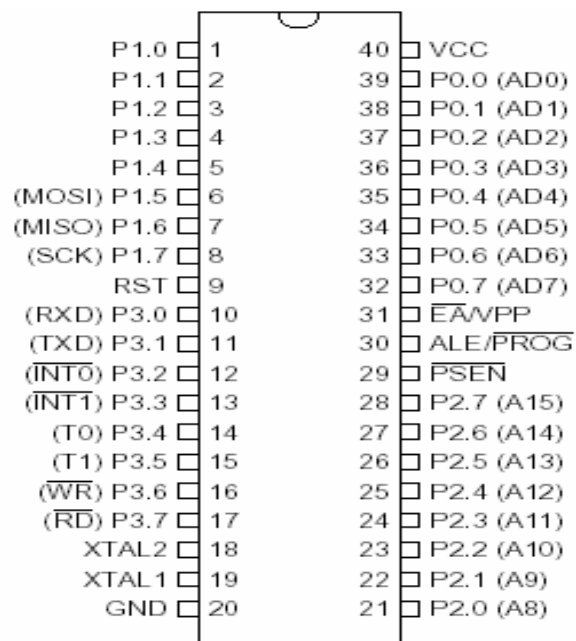
Mikrokontroler AT89S51 ialah mikrokomputer CMOS 8 bit dengan 4KB Flash Programmable dan Erasable Read Only Memory (PEROM). Mikrokontroler berteknologi memori non volatile kerapatan tinggi dari Atmel ini kompatibel dengan mikrokontroler standar industri MCS-51 (seperti mikrokontroler 8031 yang terkenal dan banyak digunakan beberapa waktu lalu) baik pin kaki IC maupun set instruksinya serta harganya yang cukup murah. Oleh karena itu, sangatlah tepat jika kita mempelajari mikrokontroler jenis ini.

AT89S51 mempunyai memori yang terdiri dari RAM internal sebesar 128 byte dengan alamat 00H-7FH dapat diakses menggunakan RAM address register. RAM Internal ini terdiri dari Register Banks dengan 8 buah register (R0-R7). Memori lain yaitu 21 buah Special Function Register dimulai dari alamat 80H-FFH. RAM ini beda lokasi dengan Flash PEROM dengan alamat 000H – 7FFH. (Afgianto Eko Putra. 2005. *Belajar Mikrokontroler AT89C51/52/55 (Teori dan Aplikasi)*)

Block Diagram



Gambar 1. Diagram Blok AT89S51



Gambar 2. Konfigurasi Pin AT 89S51
 (Afgianto Eko Putra. 2005. *Belajar Mikrokontroler AT89C51*)

1.2 Konfigurasi dan fungsi pin AT 89S51

Mikrokontroler AT 89S51 mempunyai 40 kaki, 32 kaki diantaranya digunakan sebagai port paralel. Satu port paralel terdiri dari 8 kaki, dengan demikian 32 kaki tersebut membentuk 4 buah port paralel, yang masing-masing dikenal sebagai port 0, port 1, port 2, port 3. nomor dari masing-masing jalur (kaki) dari port paralel mulai dari 0 sampai 7, jalur (kaki) pertama port disebut sebagai P0.0 dan jalur terakhir untuk port 3 adalah P3.7.

Berikut penjelasan masing-masing pin :

VCC

Berfungsi sebagai sumber tegangan +5V

GND

Berfungsi sebagai pentanahan (ground)

Port 0

Port 0 adalah masukan/keluaran 8 bit dengan nama P0.0 – P0.7 jenisnya cerat terbuka masukan dua arah (open drain bi directional I/O port). Jika port 0 berlogika 1 maka dapat digunakan sebagai masukan yang mempunyai impedansi tinggi.

Selain berfungsi sebagai masukan/keluaran, port 0 juga berfungsi sebagai :

1. Multipleks antara byte alamat rendah (A0 s/d A7) dan data (D0 s/d D7) pada saat mengakses memori program eksternal atau memori data eksternal. Pada fungsi ini, Port 0 mrmbutuhkan resisitor pullup.

2. Masukan byte kode program selama pemrograman flash memori (memori program internal atau onchip) dan keluaran saat verifikasi. Resistor pullup dibutuhkan selama verifikasi.

Port 1

Port 1 adalah masukan/keluaran 8 bit dengan nama masing-masing P1.0 s/d P1.7 yang bersifat dua arah. Port 1 sudah di pasang resistor pullup secara internal. Jika logika satu dituliskan pada port 1 maka keluaran akan berlogika satu dan dapat digunakan sebagai masukan.

Fungsi lain port 1 adalah sebagai masukan alamat rendah pada saat pemrograman memori flash internal dan verifikasi.

Port 2

Port 2 sama dengan Port 1 yaitu masukan/keluaran 8 bit dengan nama masing-masing P1.0 s/d P1.7 yang bersifat dua arah. Port 2 sudah dipasang resistor pullup secara internal. Jika logika satu dituliskan pada port 2 maka keluaran akan berlogika satu dan dapat digunakan sebagai masukan.

Fungsi lain Port 2 adalah:

1. Sebagai byte alamat tinggi (A8 s/d A15) pada saat menjalankan program pada memori program eksternal data pada memori data eksternal dengan menggunakan pengalamatan 16 bit (instruksi MOVX @ DPTR) sedangkan jika menggunakan pengalamatan 8 bit (instruksi MOVX @ RI) maka Port 2 berisi SFR P2.

2. Sebagai bit alamat atas (A8 s/d A12 untuk AT8S51 dan kendali saat pemrograman memori flash internal dan verifikasi.

Port 3

Port 3 sama dengan port 1 dan port 2 yaitu masukan/keluaran 8 bit dengan nama masing-masing P3.0 s/d P3.7 yang bersifat dua arah. Port 3 sudah dipasang resistor pullup secara internal. Jika logika satu dituliskan pada port 3 maka keluaran akan berlogika satu dan dapat di gunakan sebagai masukan.

Selain sebagai masukan/keluaran biasa, Port 3 juga mempunyai fungsi khusus seperti pada table 1

Pin Port	Fungsi Khusus	AT89S51
P3.0	RXD (port masukan serial)	Ada
P3.1	TXD (port keluaran serial)	Ada
P3.2	INT0 (interupsi eksternal 0, aktif rendah)	Ada
P3.3	INT1 (interupsi eksternal 1, aktif rendah)	Ada
P3.4	T0 (masukan eksternal timer 0)	Ada
P3.5	T1 (masukan eksternal timer 1)	Ada
P3.6	WR (signal tulis untuk memori eksternal, aktif rendah)	Ada
P3.7	RD (signal baca untuk memori eksternal, aktif rendah)	Ada

Tabel. 1 Fungsi Khusus Port 3

Fungsi lain adalah sebagai masukan signal kendali pada saat pemrograman memori flash dan verifikasi.

RST

Berfungsi sebagai masukan reset. Jika RST diberi logika tinggi dalam waktu 2 siklus mesin maka mikrokontroler akan direset.

ALE/PROG

Signal/Address Latch Enable (ALE) digunakan untuk mengaktifkan IC latch agar data alamat rendah disimpan. ALE aktif ketika mengakses program eksternal. Pin ini juga digunakan untuk memberikan pulsa pemrograman memori flash internal'

Dalam keadaan normal ALE mengeluarkan pulsa dengan frekuensi konstan yaitu $1/6$ frekuensi osilator. Sehingga dapat digunakan untuk tujuan pewaktuan eksternal.

PSEN (Program Store Enable)

PSEN adalah keluaran signal strobe untuk membaca kode program (code memory). Ketika AT89S51 mengeksekusi memori program eksternal, signal PSEN diaktifkan dua kali setiap siklus mesinnya.

EA/VPP (External Access Enable)

EA harus dihubungkan ke ground (GND) jika semua program diakses dari memori program eksternal (external code memory) yang dimulai dari alamat 0x0000 s/d 0xFFFF. Jika program yang akan dieksekusi berasal dari memori program internal dan eksternal maka EA di hubungkan ke VCC. Pin EA juga

digunakan sebagai masukan tegangan pemrograman ketika akan memprogram memori flash internal.

XTAL-1

Masukan penguat osilator membalik dan masukan rangkaian clock internal.

XTAL-2

Keluaran dari penguat osilator membalik. (*Totok Budioko. 2005. Belajar Dengan Mudah dan Cepat Pemrograman Bahasa C Dengan SDCC) Pada Mikrokontroler AT89C51/52 Teori, Simulasi, dan Aplikasi.*

1.3 SFR (SPECIAL FUNCTION REGISTER) PADA AT89S51

Tidak semua pada alamat SFR digunakan, alamat-alamat yang tidak digunakan, tidak diimplementasikan pada chip. Jika dilakukan usaha pembacaan pada alamat-alamat yang tidak terpakai tersebut akan menghasilkan data acak dan penulisannya tidak menimbulkan efek sama sekali. Pengguna perangkat lunak sebaiknya jangan menuliskan '1' pada lokasi-lokasi 'tak bertuan' tersebut, karena dapat digunakan untuk mikrokontroler generasi selanjutnya. Dengan demikian, nilai-nilai reset atau non-aktif dari bit-bit baru ini akan selalu '0' dan nilai aktifnya adalah '1' berikut akan dijelaskan secara singkat SFR-SFR beserta fungsinya :
(Afgianto Eko Putra. 2005. *Belajar Mikrokontroler AT89C51/52/55 (Teori dan Aplikasi)*)

8 Byte

F8								FF
F0	B							F7
E8								EF
E0	ACC							E7
D8								DF
D0	PSW							D7
C8	(T2CON)	(T2MOD)	(RACP2L)	(RACP2H)	(TL2)	(TH2)		CF
C0								C7
B8	IP							BF
B0	P3							B7
A8	IE							AF
A0	P2							A7
98	SCON	SBUF						9F
90	P1							97
88	TCON	TMOD	TL0	TL1	TH0	TH1		8F
80	P0	SP	DPL	DPH			PCON	87

TCON

Gambar 3. Peta Register fungsi khusus – SFR (Special Function Register) tanda (...) untuk SFR yang dijumpai dikeluarga 51 dengan 3 Timer.
(Afgianto Eko Putra. 2005. *Belajar Mikrokontroler AT89C51/52/55 (Teori dan Aplikasi)*)

1. Akumulator

Akumulator atau ACC ialah sebuah register 8 bit yang merupakan pusat dari semua operasi accumulator, termasuk dalam operasi aritmatika dan operasi logika.(lokasi **E0h**)

2. Register B

Register B digunakan selama operasi perkalian dan pembagian, untuk instruksi lain dapat diperlukan sebagai register scratch pad (“papan coret-core”) lainnya.(lokasi **F0h**)

3. Program Status Word (PSW)

Register PSW (lokasi **D0h**) berisikan informasi status program sebagaimana telah dijelaskan sebelumnya.

4. Stack Pointer

Register SP atau stack pointer (lokasi **81h**) merupakan register dengan panjang 8 bit, digunakan dalam proses simpan dan ambil dari/ke stack. Nilainya akan dinaikkan sebelum data disimpan menggunakan instruksi **PUSH** dan **CALL**. Walau stack bisa menempati lokasi dimana saja dalam **RAM**, register **SP** akan selalu diinisialisasi ke **07h** setelah adanya reset, hal ini menyebabkan stack berawal dilokasi **08h**.

5. Data Pointer

Register Data Pointer atau DPTR berupa DPTR untuk byte tinggi (DPH) dan byte rendah (DPL) yang masing-masing berada dilokasi **83h** dan **82h**, bersama-sama membentuk register yang mampu menyimpan alamat 16 bit. Dapat dimanipulasisebagai register yang terpisah.

6. Serial Data Buffer

SBUF atau Serial Data Buffer (lokasi 99h) sebenarnya terdiri dari dua register yang terpisah, yaitu register penyangga pengirim (transmit buffer) dan penyangga penerima (receive buffer). Pada saat data disalin ke SBUF, maka data sesungguhnya dikirim ke penyangga pengirim dan sekaligus mengawali transmisi data serial. Sedangkan pada saat data disalin dari SBUF, maka sebenarnya data tersebut berasal dari penyangga penerima.

7. Timer Register

- a. Pasangan register (TH0, TL0) dilokasi 8Ch dan 8Ah, (TH1, TL1) dilokasi 8D dan 8Bh serta (TH2, TL2) dilokasi CDh dan CCh merupakan register-register pencacah 16 bit untuk masing-masing Timer0, Timer1 dan Timer2.
- b. Capture Register
Pasangan register (RCAP2H, RCAP2L) yang menempati lokasi CBh dan CAh merupakan register capture untuk mode Timer 2 Capture. Pada mode ini, sebagai tanggapan terjadinya suatu transisi sinyal dikaki (pin) T2EX , TH2 dan TL2 akan disalin masing-masing ke RCAP2H dan RCAP2L. Timer 2 juga memiliki mode isi ulang otomatis 16 bit dengan RCAP2H serta RCAP2L digunakan untuk menyimpan nilai isi ulang tersebut.
- c. Control Register
Register-register IP, IE, TMOD, TCON, T2CON, T2MOD, SCON, dan PCON berisi bit-bit kontrol dan status untuk sistem interups.

1.4 Format Program Bahasa Assembly

Program bahasa assembly berisikan :

1. instruksi-intruksi mesin
2. pengarah-pengarah assembler
3. kontrol-kontrol assembler dan
4. komentar-komentar.

Instruksi-instruksi mesin merupakan mnemonik yang menyatakan suatu instruksi yang bisa dijalankan (misalnya MOV), pengarah assembler (assembler directive) merupakan instruksi ke program assembler yang mendefinisikan struktur program, symbol-simbol, data, konstanta dan lain-lain (misalnya ORG). kontrol-kontrol assembler mengatur (menentukan) mode-mode assembler dan aliran assembly langsung (misalnya \$TITLE). Komentar perlu dituliskan agar program mudah dibaca, tidak harus perinstruksi bisa juga sekumpulan instruksi yang mengerjakan suatu operasi.

Baris-baris program yang mengandung instruksi mesin atau pengarah assembler harus mengikuti aturan program assembler ASM51. Masing-masing baris atas beberapa field yang dipisahkan dengan spasi atau tabulasi. Format umumnya :

(Label :) mnemonic (operan) (,operan) (...) (; komentar)

1. Label

Sebuah label mewakili suatu alamat dari instruksi (atau data) yang mengikat label ini digunakan sebagai operan pada instruksi-instruksi percabangan (misal: SJMP TERUS). Simbol dan label adalah dua hal yang berbeda. Symbol tidak menggunakan titik dua, sedangkan label menggunakan, perhatikan contoh berikut:

```
PAR EQU 500           ;" PAR " adalah suatu symbol
                       ;dari nilai 500
START MOV A,#0FFh    ;" START " adalah label yang
```


; menunjuk pada lokasi instruksi MOV

sebuah symbol atau label harus diawali dengan sebuah huruf, tanda tanya atau garis bawah kemudian diikuti dengan huruf, angka tanda tanya atau garis bawah hingga 31 karakter. Kata tercadang (reserved words) tidak dapat digunakan sebagai symbol maupun label.

2. Mnemonik

Mnemonik instruksi atau pengarah assembler dimasukkan dalam “mnemonic field” yang mengikuti “label mnemonic”. Mnemonik instruksi misalnya ADD, MOV, INC dan lain-lain. Sedangkan pengarah assembler misalnya ORG, EQU, DB akan dibahas lebih lanjut.

3. Operan

Operan ditulis setelah mnemonik, bisa berupa alamat atau data yang digunakan instruksi yang bersangkutan. Bisa juga berupa label yang mewakili alamat suatu data atau yang berupa simbol yang mewakili suatu data konstanta. Perlu diingat bahwa ada beberapa instruksi MCS51 yang tidak memerlukan operan (RET dan lain-lain).

4. Komentar

Komentar harus diawali dengan titik koma (;), sebuah baris atau bagian dari suatu baris akan dianggap sebagai komentar jika diawali dengan titik koma. Sub-rutin dari bagian-bagian besar program yang mengerjakan suatu operasi biasanya diawali dengan blok komentar yang menjelaskan fungsi sub-rutin atau bagian besar program tersebut.

Symbol-simbol assembler khusus digunakan untuk mode-mode pengalamatan melalui register. Symbol-simbol ini ini mencakup A.R0 s/d R7.DPTR.C.PC dan AB. Juga tanda dollar (\$) yang dapat digunakan untuk menunjuk nilai pencacah program (program counter) saat itu. Perhatikan contoh berikut :

```

STEBC      C
INCDPTR    DPTR
JNB        TI,$

```

Instruksi terakhir ditulis secara efektif tanpa menggunakan label yang bisa juga dituliskan.

```

LAGI:      JNB        TI, LAGI

```

1.5 Model Pengalamatan

a. Pengalamatan Tak Langsung

Beberapa instruksi menggunakan operan berupa register yang menyimpan alamat data disimpan. Dalam hal ini digunakan tanda “at” (@) yang dapat digunakan bersama dengan R0, R1, DPTR atau PC tergantung dari instruksi yang digunakan. Misalnya :

```

ADD        A, @R0
MOVC       A, @A+PC

```

Instruksi pertama menyalin data yang tersimpan di alamat yang ditunjukkan R0 ke Akumulator. Sedangkan instruksi kedua untuk data yang disimpan dalam @A+PC.

b. Pengalamatan Langsung

Data-data langsung diawali dengan tanda pound (#) dan menyatu dengan instruksi yang bersangkutan. Seperti contoh berikut:

```

CONSTANT      EQU 100
               MOV  A, #0FEh
               ORL  40h, #CONSTANT

```

Semua operasi yang melibatkan data langsung (kecuali instruksi MOV, DPTR, # data) hanya membutuhkan data 8-bit (1 byte). Data langsung akan di evaluasi sebagai suatu konstanta 16-bit dan byte-rendah yang digunakan. Semua bit di byte-tinggi harus sama (00h atau FFh) atau nantinya akan mengakibatkan kesalahan “value will not fit in byte”. Perhatikan contoh berikut:

```

MOV  A, #0FF00h      ; benar
MOV  A, #00FFh      ; benar
MOV  A, #0FE00h      ; salah, high-byte berbeda
MOV  A, #01FFh      ; salah, high-byte berbeda
MOV  A, #-256        ; dua instruksi sama hasilnya
MOV  A, #0FF00h      ; A = 00h

```

(Afgianto Eko Putra. 2005. *Belajar Mikrokontroler AT89C51/52/55 (Teori dan Aplikasi)*)

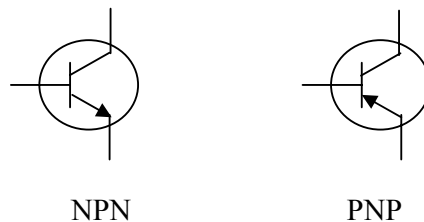
2. TRANSISTOR

Transistor adalah komponen elektronika yang mempunyai tiga buah terminal, Terminal itu disebut emitor, basis, dan kolektor. Transistor dibentuk dengan penggabungan dua buah dioda. Dioda satu dengan yang lain saling digabungkan dengan cara menyambungkan salah satu sisi dioda yang senama. Dengan cara penggabungan seperti ini dapat diperoleh dua buah dioda sehingga menghasilkan transistor NPN dan PNP.

Bahan mentah yang digunakan untuk menghasilkan bahan N dan P adalah silikon dan germanium. Oleh karena itu, dikatakan :

- Trnsistor germanium PNP
- Transistor silikon NPN
- Transistor silikon PNP
- Transistor germanium NPN.

Semua komponen didalam bagan rangkaian transistor dinyatakan dengan simbol. Anak panah yang terdapat didalam simbol menunjukkan arah yang melalui transistor.



Gambar 4. Simbol Transistor

Catatan :

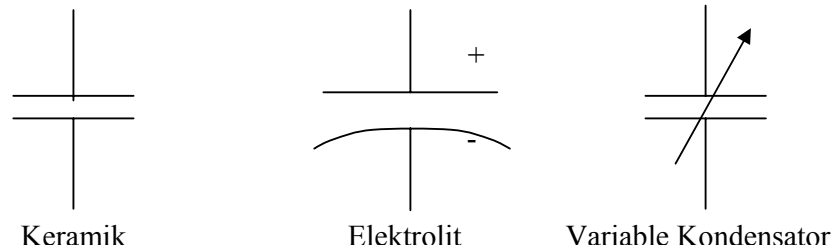
1. Transistor PNP : anak panah didalam simbol pada hubungan emitor menunjuk kedalam.
2. Transistor NPN : anak panah didalam simbol pada hubungan emitor menunjuk keluar.

[Daryanto. Drs. 2000. *Pengetahuan Teknik Elektronika*. Jakarta : Bumi Aksara]

Transistor yang digunakan dirangkaian ini adalah jenis TR BC327 sebagai penguat.

3. KONDENSATOR

Kapasitor atau kondensator memiliki struktur yang terbuat dari plat metal dan dipisahkan oleh suatu bahan dielektrik. Bahan – bahan dielektrik dikenal dengan udara vakum, keramik, gelas dan lain – lain. Apabila kedua ujung plat diberi tegangan listrik, maka muatan –muatan positif akan mengumpul pada salah satu kaki (elektroda) metalnya dan pada saat yang sama muatan – muatan negatif yang terkumpul pada ujung metal yang satu lagi. Namun demikian muatan positif tidak dapat segera mengalir menuju ujung kutub negatif, dan sebaliknya muatan negatif pun tidak bisa mengalir menuju kutub positif , karena terpisah oleh bahan dielektrik yang non konduktif . Dengan demikian muatan elektrik ini “ tersimpan” selama tidak ada konduksi pada ujung – ujung kakinya.



Gambar 5. Simbol Kondensator

Sedangkan kapasitor yang sering dipakai atau umum dipakai adalah :
kondensator keramik dan kondensator elektrolit (elco).

1. Kondensator Tetap

Yang dimaksud kondensator tetap ialah kondensator yang nilai kapasitansya tidak dapat diubah – ubah, jenis sifatnya tetap. Termasuk kondensator tetap ini diantaranya kondensator berjenis mika, film, poliester, dan keramik. Nama kondensator tersebut didasarkan atas bahan dielektrum yang dipergunakan untuk membuatnya.

2. Kondensator Tidak Tetap (Variable)

Sedangkan yang dimaksud dengan kondensator tidak tetap atau disebut juga Varco (Variable Condensator) yang dilambangkan dengan huruf “Vc” atau “Vr” saja adalah kondensator yang nilai kapasitansya dapat diubah atau diatur sesuai dengan keperluan. Biasanya hanya berkisar antara 0 sampai dengan 500 pF Komponen ini biasanya hanya dipergunakan pada rangkaian elektronika radio, dan jarang sekali dipakai dalam proyek – proyek elektronika lain.

4. RESISTOR

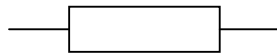
Resistor sering disebut juga dengan Werstand, tahanan atau hambatan. Resistor dinyatakan dengan notasi R dengan satuan Ohm (Ω).

Pada rangkaian elektronika, resistor berfungsi sebagai :

- a. Menghambat arus listrik
- b. Membagi arus listrik pada rangkaian paralel
- c. Membagi tegangan pada rangkaian seri

Resistor adalah komponen dasar elektronika yang dipergunakan untuk membatasi jumlah arus yang mengalir dalam satu rangkaian. Sesuai dengan namanya resistor bersifat resistif dan biasanya komponen ini terbuat dari bahan karbon. Berdasarkan hukum Ohm bahwa resistansi berbanding terbalik dengan jumlah arus yang mengalir melaluinya.

Resistor yang kita kenal umumnya berbentuk tabung dengan dua kaki dari tembaga di kiri dan kanan. Pada badannya terdapat lingkaran membentuk gelang berwarna yang merupakan kode warna untuk memudahkan pemakai untuk mengenali besar resistansi tanpa mengukur besarnya dengan Ohmmeter terlebih dahulu.



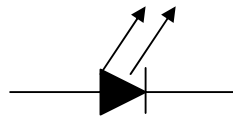
Gambar 6. Simbol Resistor

5. LED

LED adalah semikonduktor khusus yang dirancang untuk memancarkan cahaya apabila dialiri arus. Bila dioda diberi prategangan maju, elektron-elektron bebas akan jatuh kedalam lubang-lubang (hole) disekitar persambungan. Ketika seluruh dari tingkat energi lebih tinggi ke tingkat energi lebih rendah elektron-elektron bebas tersebut akan mengeluarkan energi dalam bentuk radiasi. Pada dioda penyearah, energi ini keluar dalam bentuk panas. Tetapi pada dioda pemancar cahaya (*Light Emitting Diode*) LED, energi ini memancarkan sebagai cahaya. LED ini telah dapat menggantikan lampu-lampu

pijar dalam beberapa pemakaian karena tegangannya yang rendah, umurnya yang panjang, dan dari mati ke hidup dan sebaliknya berlangsung cepat.

Dioda biasanya terbuat dari bahan *silicon*, yaitu bahan buram yang menghalangi pengeluaran cahaya. Sedangkan LED terbuat dari unsur-unsur seperti *gallium, arsen*, dan *fosfor*, warna LED diantaranya adalah merah, hijau, kuning, biru, jingga, atau bening. Penurunan tegangan LED adalah dari 1,5 V sampai 2,5 untuk arusnya diantara 10 dan 150 mA (Malvino, 1985 :).



Gambar 7. Simbol LED

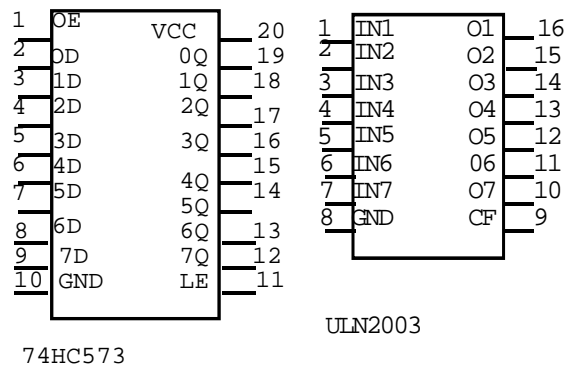
Dalam rangkaian ini menggunakan LED sebanyak 7 x 5 per bloknya x 10 blok jadi total keseluruhan ada 350 buah led sebagai pengganti dot matrik.

6. INTEGRATED CIRCUIT (IC)

Integrated circuit (IC) adalah komponen elektronika yang terdiri dari transistor, resistor, dan kapasitor atau kondensator yang dirakit atau dikemas sedemikian rupa sehingga menjadi satu rumah “DIL” atau *Dual In Line* yang terdiri dari beberapa kaki.

Digital Integrated Circuit atau IC Digital tersusun oleh beberapa gerbang-gerbang logika, gerbang logika dapat difungsikan sebagai multifibrator, oscilator atau pembangkit getaran dan dapat pula difungsikan sebagai timer. Integrated Circuit (IC) adalah komponen yang mudah rusak sehingga dalam pemasangan

perlu menggunakan soket dan diberi aluminium pendingin. IC yang digunakan pada rangkaian ini adalah 74HC573 dan ULN 2003.



Gambar 8. Simbol IC

Fungsi IC 74HC573 sebagai pengendali atau driver ke transistor sekaligus mengatur LED 7 kolom matrik perbloknya. Sedangkan ULN 2003 berfungsi sebagai input tegangan mengatur lampu LED 5 baris dalam setiap bloknya.

B.Pembuatan

1. Alat dan Bahan

Alat yang dipergunakan adalah sebagai berikut:

- a. Pensil
- b. Mistar (penggaris)
- c. Cutter (pisau potong)
- d. Penitik
- e. Palu
- f. Obeng
- g. Meisn bor
- h. Mata bor
- i. Kunci mesin bor
- j. Gergaji triplex
- k. Kikir
- l. Solder
- m. Penyedot timah (atraktor)
- n. Tang potong
- o. Tang penjepit
- p. Spray cat
- q. Eprom prgrammer

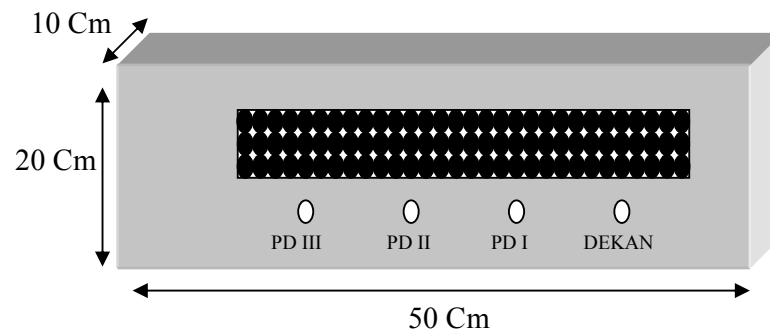
Bahan yang digunakan adalah sebagai berikut:

Tabel 3. Daftar Bahan

No	Nama Bahan	Identifikasi bahan	Jumlah
1.	Mikrokontroler	AT89S51	1 bh
2.	Transformator	2 A	1 bh
3.	Rangkaian Catu Daya	9 dan 12V	1 bh
4.	Buzzer	12 V	1 bh
5.	Saklar Reset		5 bh
6.	Led	½ W	350 bh

2. Perencanaan Desain

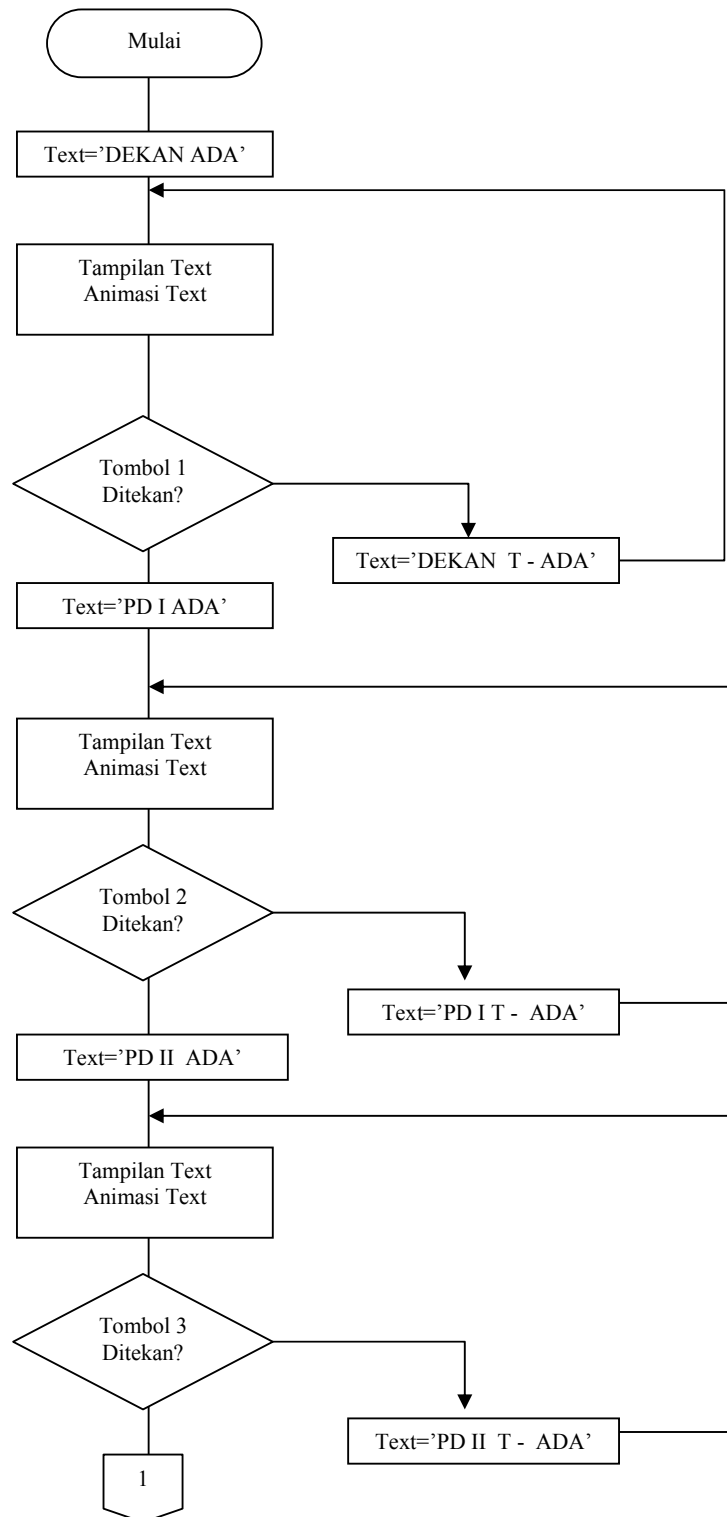
2.1 Bentuk Mekanik

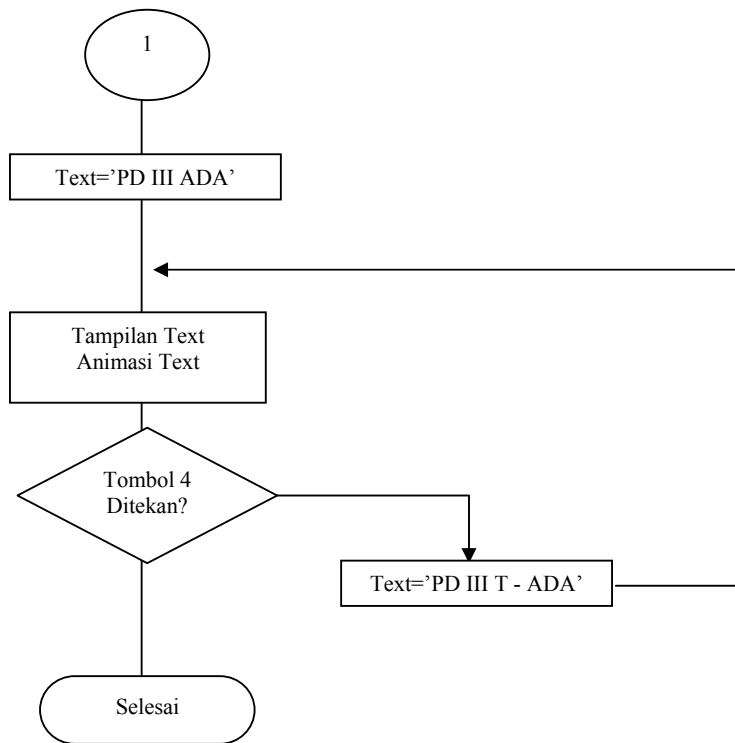


Gambar 9. Gambar Mekanik Tampak Depan dan Samping

3. PEMBUATAN PROGRAM

3.1 FLOWCHART PROGRAM





Gambar 10. Flowchart

Keterangan:

Program mulai dengan memberikan nilai pada variabel text dengan text yang sudah disiapkan misalnya mengisi nilai text dengan 'DEKAN ADA'. Kemudian program akan melanjutkan dengan menampilkan text tersebut pada display dot matrik. Setelah tampil text tersebut dianimasi dengan menggerakkan text ke kiri, ke atas dan ke bawah. Program juga mengecek tombol masukan. Bila tombol ditekan maka teks akan diubah dari 'ADA' menjadi 'TIDAK ADA'

3.2 Listing Program

```

S1   BIT   P3.7
S2   BIT   P3.6
S3   BIT   P3.5
S4   BIT   P3.4
      ORG  0AH

COUNT1  DB  1
COUNT2  DB  1

COL1 DB  1
COL2 DB  1
COL3 DB  1
COL4 DB  1
COL5 DB  1
CHR  DB  1
CHRKE DB  1
TEMP DB  1
FLAG DB  1

      ORG  0000H
      AJMP BOOT
;
      ORG  0100

BOOT:      MOV  SP,#073H
          mov  r3,#00h
          DJNZ R3,$
          MOV  P0,#0FFH
          MOV  P1,#00
          CLR  P3.0
          CLR  P3.1
          MOV  P2,#0FFH
          MOV  FLAG,#00H

BOOT1:
          LCALL  BERSIH
          JB   ACC.0,BOOT2
          MOV  DPTR,#TEXT2
          LCALL  ANIMASI
          MOV  A,FLAG
          JNB  ACC.7,BOOT3
          CLR  ACC.7
          MOV  FLAG,A

```

```
LJMP BOOT1
BOOT2:
MOV DPTR,#TEXT6
LCALL ANIMASI
MOV A,FLAG
JNB ACC.7,BOOT3
CLR ACC.7
MOV FLAG,A
LJMP BOOT1

BOOT3:
MOV A,FLAG
JB ACC.1,BOOT4
MOV DPTR,#TEXT3
LCALL ANIMASI
MOV A,FLAG
JNB ACC.7,BOOT5
CLR ACC.7
MOV FLAG,A
LJMP BOOT3

BOOT4:
MOV DPTR,#TEXT7
LCALL ANIMASI
MOV A,FLAG
JNB ACC.7,BOOT5
CLR ACC.7
MOV FLAG,A
LJMP BOOT3

BOOT5:
MOV A,FLAG
JB ACC.2,BOOT6
MOV DPTR,#TEXT4
LCALL ANIMASI
MOV A,FLAG
JNB ACC.7,BOOT7
CLR ACC.7
MOV FLAG,A
LJMP BOOT5

BOOT6:
MOV DPTR,#TEXT8
LCALL ANIMASI
MOV A,FLAG
JNB ACC.7,BOOT7
```

```

CLR ACC.7
MOV FLAG,A
LJMP BOOT5

```

BOOT7:

```

MOV A,FLAG
JB ACC.3,BOOT8
MOV DPTR,#TEXT5
LCALL ANIMASI
MOV A,FLAG
JNB ACC.7,BOOT9
CLR ACC.7
MOV FLAG,A
LJMP BOOT7

```

BOOT8:

```

MOV DPTR,#TEXT9
LCALL ANIMASI
MOV A,FLAG
JNB ACC.7,BOOT9
CLR ACC.7
MOV FLAG,A
LJMP BOOT7

```

BOOT9:

```

LJMP BOOT1

```

ANIMASI:

```

LCALL AMBIL_TEXT
LCALL GESER_KIRI
LCALL GESER_ATAS
LCALL GESER_BAWAH
LCALL BERSIH
RET

```

CEK_TOMBOL:

```

JB S1,CEK_TOMBOL1

```

CEK_TOMBOLA:

```

JNB S1,CEK_TOMBOLA
MOV A,FLAG
CPL ACC.0
SETB ACC.7
MOV FLAG,A

```

CEK_TOMBOL1:

```

JB S2,CEK_TOMBOL2

```

CEK_TOMBOLB:

```

JNB S2,CEK_TOMBOLB
MOV A,FLAG

```



```
CPL ACC.1
SETB ACC.7
MOV FLAG,A
CEK_TOMBOL2:
    JB S3,CEK_TOMBOL3
CEK_TOMBOLC:
    JNB S3,CEK_TOMBOLC
    MOV A,FLAG
    CPL ACC.2
    SETB ACC.7
    MOV FLAG,A
CEK_TOMBOL3:
    JB S4,CEK_TOMBOL_END
CEK_TOMBOLD:
    JNB S4,CEK_TOMBOLD
    MOV A,FLAG
    CPL ACC.3
    SETB ACC.7
    MOV FLAG,A
CEK_TOMBOL_END:
    RET
TAMPIL_COLOM:
    MOV P1,#00
    CLR P3.1
    CLR P3.0
    MOV P2,@R0
    SETB P3.1
    CLR P3.1

    LCALL KALI5_R0
    MOV P2,@R0
    SETB P3.0
    CLR P3.0
    MOV P1,#00000000B

    LCALL KALI5_R0
    MOV P2,@R0
    MOV P1,#10000000B
    MOV P1,#00

    LCALL KALI5_R0
    MOV P2,@R0
    MOV P1,#01000000B
    MOV P1,#00

    LCALL KALI5_R0
```

```

MOV P2,@R0
MOV P1,#00100000B
MOV P1,#00

```

```

LCALL KALI5_R0
MOV P2,@R0
MOV P1,#00010000B
MOV P1,#00

```

```

LCALL KALI5_R0
MOV P2,@R0
MOV P1,#00001000B
MOV P1,#00

```

```

LCALL KALI5_R0
MOV P2,@R0
MOV P1,#00000100B
MOV P1,#00

```

```

LCALL KALI5_R0
MOV P2,@R0
MOV P1,#00000010B
MOV P1,#00

```

```

LCALL KALI5_R0
MOV P2,@R0
MOV P1,#00000001B
MOV P1,#00

```

```

RET

```

```

KALI5_R0:

```

```

INC R0
INC R0
INC R0
INC R0
INC R0
RET

```

```

BERSIH:

```

```

MOV R2,#81
MOV R0,#01FH

```

```

BERSIH1:

```

```

MOV @R0,#0FFh
INC R0
DJNZ R2,BERSIH1
RET

```

```

AMBIL_CHAR:

```

```

PUSH DPH

```

```

    PUSH DPL
    MOV A,CHR      ;KARAKTERNYA
    CJNE A,#31H,AMBIL_CHARA1
    MOV DPTR,#ANGKA1
    LJMP AMBIL_CHAR1
AMBIL_CHARA1:
    CJNE A,#32H,AMBIL_CHARA2
    MOV DPTR,#ANGKA2
    LJMP AMBIL_CHAR1
AMBIL_CHARA2:
    CJNE A,#33H,AMBIL_CHARA3
    MOV DPTR,#ANGKA3
    LJMP AMBIL_CHAR1
AMBIL_CHARA3:
    CJNE A,#2DH,AMBIL_CHARA4
    MOV DPTR,#MIN
    LJMP AMBIL_CHAR1
AMBIL_CHARA4:
    MOV A,CHR      ;KARAKTERNYA
    MOV DPTR,#TABEL
AMBIL_CHARA:
    SUBB A,#41H
    MOV CHR,A
    LCALL      GESER_DPTR
AMBIL_CHAR1:
    MOV COUNT1,#05
    MOV R0,CHRKE
AMBIL_CHAR2:
    MOV A,#00
    MOVC      A,@A+DPTR
    CPL A
    MOV @R0,A
    INC R0
    INC DPTR
    DJNZ COUNT1,AMBIL_CHAR2
AMBIL_CHAR3:
    POP DPL
    POP DPH
    RET
GESER_DPTR:
    MOV A,CHR
    JZ GESER_DPTR_END
    INC DPTR
    INC DPTR
    INC DPTR
    INC DPTR

```

```

        INC  DPTR
        DEC  CHR
        AJMP GESER_DPTR
GESER_DPTR_END:
        RET
AMBIL_TEXT:
        MOV  CHRKE,#1FH
AMBIL_TEXTA:
        MOV  A,#00H
        MOVC A,@A+DPTR
        CJNE A,#$24,AMBIL_TEXT1
        RET
AMBIL_TEXT1:
        MOV  CHR,A
        CJNE A,#20H,AMBIL_TEXT2
        INC  CHRKE
        INC  CHRKE
        INC  CHRKE
        INC  DPTR
        AJMP AMBIL_TEXTA
AMBIL_TEXT2:
        LCALL AMBIL_CHAR
        LCALL TAMBAH_CHRKE
        INC  DPTR
        INC  CHRKE
        AJMP AMBIL_TEXTA
        RET
TAMBAH_CHRKE:
        INC  CHRKE
        INC  CHRKE
        INC  CHRKE
        INC  CHRKE
        INC  CHRKE
        RET
TAMPIL1:
        LCALL CEK_TOMBOL
        MOV  A,FLAG
        JB   ACC.7,TAMPIL_END
        MOV  R3,#10
TAMPIL2:
        MOV  P0,#01H           ; scanning
        MOV  R0,#1FH
        LCALL TAMPIL_COLOM
        LCALL DLY_01
;       LCALL DLY_01
        LCALL OFF

```

```

MOV P0,#02H
MOV R0,#20H
LCALL    TAMPIL_COLOM
LCALL    DLY_01
; LCALL    DLY_01
LCALL    OFF

MOV P0,#04H          ; scanning
MOV R0,#21H
LCALL    TAMPIL_COLOM
LCALL    DLY_01
; LCALL    DLY_01
LCALL    OFF

MOV P0,#08H          ; scanning
MOV R0,#22H
LCALL    TAMPIL_COLOM
LCALL    DLY_01
; LCALL    DLY_01
LCALL    OFF

MOV P0,#10H          ; scanning
MOV R0,#23H
LCALL    TAMPIL_COLOM
LCALL    DLY_01
; LCALL    DLY_01
LCALL    OFF
DJNZ R3,TAMPIL2
TAMPIL_END:
RET
OFF:
MOV P0,#00H
MOV P2,#0FFH
MOV P1,#0FFH
SETB P3.0
SETB P3.1
LCALL    DLY_01
; LCALL    DLY_01
RET

;*-----*
; DELAY SUB ROUTINE
;*-----*
DLY_01: PUSH    PSW ;(DELAY 1 mS)
        PUSH DPH

```

```

        PUSH DPL
;       SETB RS0
        MOV DPTR,#1000
        MOV R5,DPH
        MOV R4,DPL
ULGA:   ;LCALL    CEK_TOMBOL
        DJNZ R4,ULGA
        DJNZ R5,ULGDXB
        RET
ULGDXB: MOV R4,#0FFH
ULGDXC: DJNZ    R4,ULGDXC
        DJNZ R5,ULGDXB
        POP  DPL
        POP  DPH
        POP  PSW
        RET
DLY_50:
        PUSH PSW ;(DELAY 50 mS)
        PUSH DPH
        PUSH DPL
;       SETB RS0
        MOV DPTR,#50000
        MOV R5,DPH
        MOV R4,DPL
ULG1:  DJNZ R4,ULG1
        DJNZ R5,ULGDX3
        RET
ULGDX3: MOV R4,#0FFH
ULGDX5: DJNZ    R4,ULGDX5
        DJNZ R5,ULGDX3
        POP  DPL
        POP  DPH
        POP  PSW
        RET
DLY_05:                                     ; (DELAY 0,5 DETIK)
        PUSH ACC
        MOV  A,#10
DLY_05A:
        LCALL    DLY_50
        DEC  A
        JNZ  DLY_05A
        POP  ACC
        RET
DLY_1S:
        LCALL    DLY_05
        LCALL    DLY_05

```

```
    RET
DLY_5S:
    LCALL    DLY_1S
    LCALL    DLY_1S
    LCALL    DLY_1S
    RET
GESER_ATAS:
    MOV COUNT1,#08
GESER_ATAS1:
    LCALL    TAMPIL1
    MOV R0,#1FH
    MOV COUNT2,#80
GESER_ATAS2:
    MOV A,@R0
    RL  A
    MOV @R0,A
    INC R0
    DJNZ COUNT2,GESER_ATAS2
    LCALL    TAMPIL1
    DJNZ COUNT1,GESER_ATAS1
    LCALL    TAMPIL1
    LCALL    TAMPIL1
    LCALL    TAMPIL1
    LCALL    TAMPIL1
    LCALL    TAMPIL1
    LCALL    TAMPIL1
    LCALL    TAMPIL1
    LCALL    TAMPIL1
    LCALL    TAMPIL1
    RET
GESER_BAWAH:
    MOV COUNT1,#08
GESER_BAWAH1:
    LCALL    TAMPIL1
    MOV R0,#1FH
    MOV COUNT2,#80

GESER_BAWAH2:
    MOV A,@R0
    RR  A
    MOV @R0,A
    INC R0
    DJNZ COUNT2,GESER_BAWAH2
    LCALL    TAMPIL1
    DJNZ COUNT1,GESER_BAWAH1
    LCALL    TAMPIL1
    LCALL    TAMPIL1
```

```

LCALL    TAMPIL1
LCALL    TAMPIL1
LCALL    TAMPIL1
LCALL    TAMPIL1
LCALL    TAMPIL1
LCALL    TAMPIL1
RET

```

```

GESER_KIRI:
    MOV    COUNT2,#80
GESER_KIRI1:
    LCALL    TAMPIL1
    LCALL    PINDAH
    DJNZ    COUNT2,GESER_KIRI1
    LCALL    TAMPIL1
    LCALL    TAMPIL1
    LCALL    TAMPIL1
    LCALL    TAMPIL1
    LCALL    TAMPIL1
    LCALL    TAMPIL1
    LCALL    TAMPIL1
    LCALL    TAMPIL1
    LCALL    TAMPIL1
    RET

    RET

```

```

PINDAH:
    MOV    COUNT1,#80
    MOV    6FH,1FH
    MOV    R0,#01FH
    MOV    R1,#020H

```

```

PINDAH1:
    MOV    A,@R1
    MOV    @R0,A
    INC    R1
    INC    R0
    DJNZ    COUNT1,PINDAH1
    MOV    6EH,6FH
    RET

```

```

;          123456789ABCDEF

```

```

TEXT2:    DB    'HASAN ADA ', $24
TEXT3:    DB    'PD 1 ADA ', $24

```


TEXT4: DB 'PD 2 ADA ', \$24
 TEXT5: DB 'PD 3 ADA ', \$24

TEXT6: DB 'DKN T-ADA ', \$24
 TEXT7: DB 'PD1 T-ADA ', \$24
 TEXT8: DB 'PD2 T-ADA ', \$24
 TEXT9: DB 'PD3 T-ADA ', \$24

TABEL:

DB 03FH,044H,044H,044H,03FH ; A
 DB 07FH,049H,049H,049H,03EH ; B
 DB 03EH,041H,041H,041H,022H ; C
 DB 07FH,041H,041H,041H,03EH ; D
 DB 07FH,049H,049H,049H,049H ; E
 DB 07FH,048H,048H,048H,040H ; F
 DB 03EH,049H,049H,049H,02EH ; G 7
 DB 07FH,008H,008H,008H,07FH ; H
 DB 000H,041H,07FH,041H,000H ; I
 DB 006H,001H,041H,07EH,040H ; J
 DB 07FH,008H,014H,022H,041H ; K
 DB 07FH,001H,001H,001H,001H ; L
 DB 07FH,020H,018H,020H,07FH ; M
 DB 07FH,020H,010H,008H,07FH ; N
 DB 03EH,041H,041H,041H,03EH ; O 15
 DB 07FH,048H,048H,048H,030H ; P
 DB 03EH,049H,045H,043H,03EH ; Q
 DB 07FH,048H,04CH,04AH,031H ; R
 DB 032H,049H,049H,049H,026H ; S
 DB 040H,040H,07FH,040H,040H ; T
 DB 07EH,001H,001H,001H,07EH ; U
 DB 07CH,002H,001H,002H,07CH ; V
 DB 07FH,002H,00CH,002H,07FH ; W
 DB 063H,014H,008H,014H,063H ; X
 DB 060H,010H,00FH,010H,060H ; Y
 DB 043H,045H,049H,051H,061H ; Z
 SPASI: DB 000H,000H,000H,000H,000H ;
 MIN: DB 008H,008H,008H,008H,008H ;
 ANGKA1:
 DB 000H,041H,07FH,041H,000H ; I
 ANGKA2:
 DB 041H,07FH,041H,07FH,041H ; II
 ANGKA3:
 DB 041H,07FH,07FH,07FH,041H ; III
 End

4. Proses Pembuatan Benda Kerja

Proses pembuatan benda kerja pada alat ini meliputi tiga bagian, yaitu: bagian elektronik, pembuatan program, dan bagian mekanik.

4.1 Bagian Elektronik

Bagian elektronik meliputi:

1. Perencanaan rangkaian
2. Percobaan laboratorium
3. Proses pembuatan papan rangkain tercetak
4. Pemasangan komponen
5. Perakitan

4.1.1 Perencanaan Rangkaian

Setelah mendapat ide rancangan maka dimulailah merencanakan rangkaian dengan cara mencari beberapa alternatif rangkaian yang semacam untuk digunakan sesuai keperluan. Perencanaan rangkaian dimaksud untuk merealisasikan ide menjadi benda dalam bentuk rangkaian elektronika. Bagian ini meliputi perencanaan pemakaian komponen dan macam rangkaian yang akan digunakan.

4.1.2 Percobaan laboratorium

Percobaan laboratorium merupakan suatu cara untuk menguji rangkaian yang masih dirangkai pada proto board. Percobaan ini dimaksudkan untuk mengetahui apakah rangkaian yang nantinya akan dibuat pada papan rangkain tercetak (PRT) sudah dapat berfungsi sesuai yang telah diharapkan atau belum. Apabila rangkaian yang telah diuji dalam percobaan laboratorium sudah dapat berfungsi seperti yang diharapkan, maka langkah selanjutnya adalah membuat papan rangkaian tercetak yang akan digunakan sebagai tempat untuk memasang komponen dari rangkaian yang telah diuji pada percobaan laboratorium.

4.1.3 Proses Pembuatan Papan Tercetak (PRT)

Proses pembuatan papan rangkaian tercetak meliputi beberapa langkah yaitu:

- a. Membuat gambar rangkaian sesuai dengan gambar yang telah direncanakan pada komputer yang kemudian dicetak kedalam kertas.
- b. Menyalin gambar yang telah dicetak di dalam kertas kedalam PCB atau di sablon, yang kemudian dilarutkan menggunakan ferikolit hingga lapisan tembaga pada PCB yang tidak tertutup oleh gambar mengelupas.

- c. Kemudian bersihkan menggunakan tiner hingga bersih, dan lapisan yang menutupi tembaga pada PCB benar-benar sudah bersih.

4.1.4 Pemasangan Komponen

Untuk proses pemasangan komponen dilakukan langkah-langkah sebagai berikut:

- a. Membersihkan jalur PRT dengan cara mengamplas dan kemudian mengeringkannya.
- b. Mengolesi jalur PRT dengan lofet agar dalam penyolderan timah mudah melekat.
- c. Memasang komponen sesuai pada tempatnya kemudian menyoldernya dan memotong sisa kaki komponen dengan tang potong.
- d. Membersihkan sisa lofet dengan tiner.

Perlu diperhatikan sebelum komponen-komponen dipasang harus diperiksa terlebih dahulu apakah komponen tersebut rusak atau tidak. Gunakan solder dengan daya yang tidak terlalu besar yaitu sekitar 30 watt. Hal ini untuk menghindari terjadinya pemanasan yang berlebihan terhadap komponen-komponen. Timah solder menggunakan jenis yang berkualitas untuk memperoleh hasil penyolderan yang baik.

4.1.5 Perakitan

Setelah semua rangkaian selesai dirangkai, maka kita coba untuk merakit semua rangkaian menjadi sistem yang telah kita rencanakan. Hal ini dimaksudkan untuk mengetahui kekurangan dari rangkaian sistem, kabel penghubung, atau bahan yang kita butuhkan. Setelah semua rangkaian lengkap maka dilakukan pengetesan.

4.2 Bagian Pembuatan Program

Pemrograman dilakukan dengan menggunakan bahasa Assembler MCS51 dengan program ALDS. Adapun langkah-langkahnya adalah sebagai berikut:

- a. Membuat diagram alir (flow chart) dari program yang akan dibuat
- b. Mengubah diagram alir tersebut kedalam bahasa pemrograman Assembler MCS51.
- c. Menyimpan program tersebut.
- d. Mengkompilasi program yang telah selesai dibuat ke memori sampai program betul dan bekerja dengan baik.
- e. Hasil kompilasi program dikodekan dari kode heksa ke kode biner dan diisikan ke EPROM.

4.3 Bagian Mekanik

Proses pembuatan bagian mekanik meliputi: proses pembuatan kotak, proses pengecatan, dan proses pemberian tanda / fungsi tombol.

5. Pengujian

Pengujian kerja Display Matrik Berbasis Mikrokontroler dilakukan untuk mengetahui proses kerja, sehingga pada saat terjadi kerusakan atau gangguan yang menyebabkan tidak berfungsinya sistem secara keseluruhan ataupun sebagian dapat digunakan sebagai rujukan untuk melakukan perbaikan. Dalam pengujian untuk mendapatkan data-data spesifikasi sistem ini maka peralatan dan bahan yang digunakan adalah :

- a. Multimeter
- b. Programmer AT89S51
- c. Catu daya +12V
- d. Kabel dan jumper

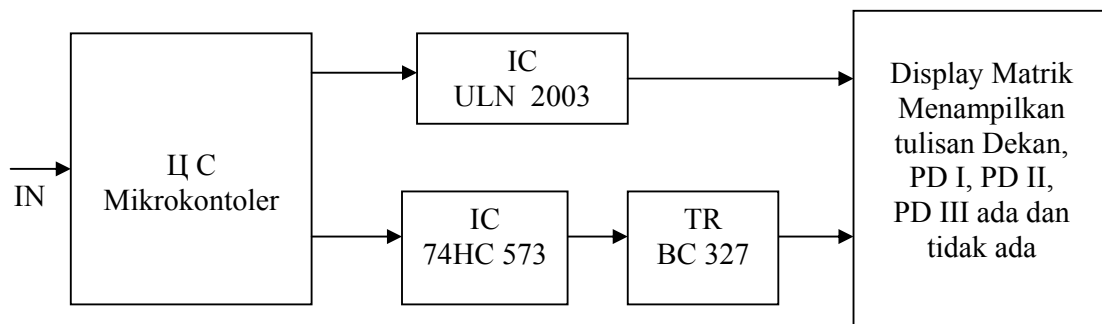
5.1 Langkah Pengujian

Langkah-langkah pengujiannya adalah :

- a. Mempersiapkan gambar rangkaian dan tata letak komponen.
- b. Mempersiapkan semua peralatan yang akan digunakan.
- c. Melakukan pengujian rangkaian.

5.2 Cara Kerja Rangkaian

Cara kerja alat “Display Matrix Berbasis Mikrokontroler” akan dijelaskan berdasarkan diagram blok dibawah ini :



Gambar 11. Diagram Blok Display Matrix Berbasis Mikrokontroler AT89S51

Alat Display Matrik ini terdiri dari 5 komponen utama yaitu : Mikrokontroler, IC ULN2003, IC 74HC573, Transistor BC 327 dan rangkaian Led / Display Matrik.

Adapun cara kerja keseluruhan rangkaian dapat dijelaskan sebagai berikut :

1. Mikrokontroler mendapat input tegangan 12 V dan mempunyai fungsi melakukan proses pengolahan data yang sudah di program dulu lewat IC downloader.
2. Kemudian keluaran dari mikrokontroler pada port P0.0 – P0.4 masuk ke IC ULN2003 yang keluarannya berfungsi sebagai penguat darlington lampu LED pada 5 baris dalam setiap bloknya.
3. Selanjutnya keluaran dari mikrokontroler pada port P2.0 – P2.6 masuk ke IC 74HC573 yang keluarannya berfungsi sebagai

penyangga atau buffer ketransistor BC327 sekaligus menjadi input tegangan lampu LED pada 7 kolom matrik perbloknya.

4. Setelah kedua keluaran dari IC ULN2003 sebagai input tegangan pada 5 baris dalam setiap bloknya dan IC 74HC573 sebagai buffer Transistor BC327 sekaligus input tegangan untuk lampu LED pada 7 kolom perbloknya maka Display matrik bisa menyala menampilkan tulisan Dekan, PD I, PD II, dan PD III ada dan tidak ada sesuai dengan program yang telah didownload oleh mikrokontroler.

6. Hasil Pengujian

Hasil pengujian yang didapatkan telah sesuai dengan perencanaan, untuk menampilkan tulisan Dekan atau Pembantu Dekan ada dan tidak ada supaya kelihatan menarik dibaca, serta waktu untuk menentukan berapa lama pergantian tulisan bisa diatur program mikrokontrolernya melalui IC downloader.

Berikut tabel hasil percobaan :

Tegangan Input (Vcc)	Tegangan Emitor (VE)	Tegangan Resistor (V Data Kolom)	Tegangan IC (V Data Baris)
5 Volt	5,1 Volt	5 Volt	5 Volt
9 Volt	9,1 Volt	9 Volt	9 Volt
12 Volt	12,1 Volt	12 Volt	12 Volt

Tabel. 4. Hasil Pengujian

BAB III

PENUTUP

A. Kesimpulan

Setelah melakukan perancangan, pembuatan, pengujian dan percobaan perlatan display matrix berbasis mikrokontroler dapat diambil kesimpulan sebagai berikut:

- 1 Informasi yang diberikan kepada mahasiswa pada alat ini sudah cukup jelas karena ditampilkan menggunakan Led sebanyak 350 buah sebagai pengganti dot metrik yaitu berupa lampu led 5X7 perbloknya, dan ada 10 blok.
- 2 Waktu yang dibutuhkan oleh alat informasi ini untuk menampilkan tulisan relatif singkat yaitu setiap 1 menit tulisan berganti Dekan, PD I sampai PDIII ada dan tidak ada.

B. Saran

Dalam pembuatan alat display matrix ini agar didapatkan hasil yang lebih baik lagi, maka perlu diperhatikan hal-hal berikut :

1. Untuk mengatasi kesulitan membaca dari jarak jauh, maka dapat dilakukan dengan menyempurnakan pembuatan sistem mekaniknya diperbesar dan tulisannya diubah menjadi perblok.

2. Meskipun alat ini cukup memadai, agar dapat dicapai hasil yang lebih optimal dan lebih akurat di masa yang akan datang perlu di adakan peninjauan kembali terhadap beberapa jenis peralatan yang sudah ada saat ini. Sehingga masih dapat ditingkatkan untuk informasi yang lebih baik lagi dan akurat.

DAFTAR PUSTAKA

Atmel, 1997, “*AT89 Series Hardware Description*”, Atmel Inc., (<http://www.atmel.com>), USA

Atmel, 1997, “*Flash Microcontroller: Architectural Overview*”, Atmel Inc., (<http://www.atmel.com>), USA

Totok Budioko. 2005. *Belajar Dengan Mudah dan Cepat Pemrograman Bahasa C Dengan SDCC (Small Device C Compiler) Pada Mikrokontroler AT 89X051/AT89C51/52 Teori, Simulasi, dan Aplikasi*. Yogyakarta : Gava Media.

Afgianto Eko Putra. 2005. *Belajar Mikrokontroler AT89C51/52/55 (Teori dan Aplikasi)*. Yogyakarta : Gava Media.

Albert Paul Malvino. PH. D., E.E. 2003. *Prinsip – Prinsip Elektronika*. Jakarta : Salemba Teknik.

Dwi Sunar Prasetyono. 2003. *Belajar Sistem Cepat Elektronika*. Yogyakarta : Absolut.

Daryanto. Drs. 2000. *Pengetahuan Teknik Elektronika*. Jakarta : Bumi Aksara