

BAB III

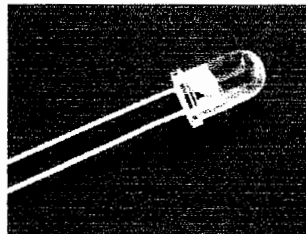
LANDASAN TEORI

3.1. Penampil Karakter

Penampil merupakan unit yang bertugas untuk menunjukkan hasil dari perhitungan aritmatik, baik sebelum, sedang, ataupun sesudah proses perhitungan terjadi. Berbagai macam teknologi penampil telah dikembangkan saat ini, di antaranya adalah *LED (Light Emitting Diode)*, dan *LCD (Liquid Crystal Display)*.

a. LED

LED (Light Emiting Dioda) adalah dioda yang mampu menghasilkan cahaya pada saat diberikan tegangan maju kepada kaki-kakinya. Bahan yang umum digunakan untuk *LED* kombinasi *Galium-Arsenida (GaAs)* dan *Galium-Fosfor (GaP)*. Sedangkan bentuk fabrikasinya dapat bermacam-macam, dari bentuk seperti tabung yang biasanya digunakan untuk indikator hingga bentuk alfanumeris untuk keperluan menampilkan huruf dan angka. Keuntungan pemakaian *LED* adalah kecepatan responnya terhadap tegangan yang diberikan, tahan guncangan, masa pemakaian yang lebih lama, efisiensinya yang tinggi, dan kemampuannya bekerja pada tegangan yang rendah. Bentuk fisik dari *LED* dapat dilihat pada Gambar 3.1.



Gambar 3.1. Bentuk fisik LED

b. *LCD*

Dalam perencanaan digunakan *display* atau penampil jenis *LCD*. *LCD* merupakan penampil karakter dengan pemakaian daya yang rendah, serta *controller CMOS LCD driver* dengan bagian *controller* sebagai berikut:

- a) Pembangkit karakter (*CG ROM/RAM*)
- b) Data penampil (*DD ROM/RAM*)

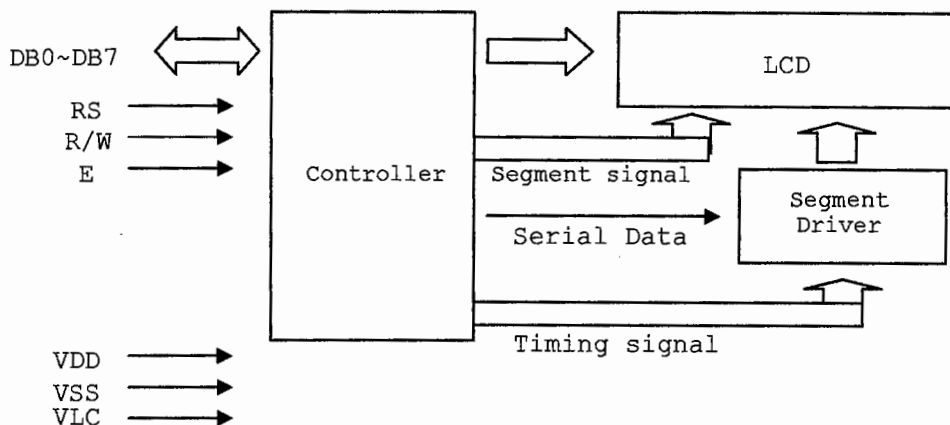
Yang berfungsi mengontrol semua perintah yang berasal dari *MPU* atau mikrokontroler dalam hal ini adalah *AT89S52*.

Bentuk fisik dari *LCD* tersebut dapat dilihat pada Gambar 3.2.



Gambar 3.2. Bentuk fisik *LCD*

Blok diagram dari *LCD* seperti pada Gambar 3.2.



Gambar 3.3. Blok diagram *LCD*

Karakter LCD:

- 1) LCD ini terdiri atas 40 karakter 16 x 2 baris dengan 1 karakter terdiri dari 5 x 7 titik (*Dot*) matrik.
- 2) Dengan 192 tipe pembangkit karakter ROM
- 3) Serta 8 buah pembangkit karakter RAM (program write).
- 4) 80 x 8 bit *display* data RAM.
- 5) Pemakaian LCD dapat disambungkan dengan MPU 4 bit dan 8 bit.
- 6) Dengan berbagai fungsi perintah seperti: *display clear*, *Cursor home*, *display on/off*, *Cursor on/off*, *display Character blink*, *Cursor shift*, dan *display shift*.
- 7) LCD dilengkapi dengan *oscillator internal*.
- 8) Dengan catu daya tunggal (5 volt).
- 9) Menggunakan teknologi CMOS.
- 10) Bekerja dengan suhu 0 °C - 50 °C.

Tabel 3.1 memperlihatkan fungsi-fungsi dari LCD.

Tabel 3.1. Fungsi LCD

Sinyal	Pin No	I/O	Tujuan	Fungsi
DB0-DB3	7 - 10	I/O	MPU	Data membaca kondisi LCD ke MPU atau tulis dari MPU ke LCD melalui bus data tetapi jika interface data 4 bit sinyal tidak dapat digunakan.
DB4-DB7	11 - 14	I/O	MPU	Data atas terdiri dari 4 data bus dua arah. Data dapat dibaca dari modul ke MPU atau ditulis, di sini DB7 digunakan sebagai busy flag.

Tabel 3.1. Fungsi LCD (lanjutan)

Sinyal	Pin No	I/O	Tujuan	Fungsi
E	6	Input	MPU	Sinyal awal pengoperasian, sinyal mengaktifkan data, dibaca dan ditulis.
R/W	5	Input	MPU	Sinyal penyeleksi baca (R) tulis (W)
RS	4	Input	MPU	Sinyal penyeleksi register, dengan kondisi : 0 = register intruksi (write) busy flag dan alamat pencacah (read).
VLC	3	-	Catu	Terminal power supply terminal untuk menjalankan LCD terang pada layer dapat divariasi dengan mengubah VLC.
VDD	2	-	Catu	Catu daya +5 volt
VSS	1	-	Catu	Terminal ground 0 volt

Sumber : LCD User Manual

Ivan (2008) menyebutkan *display* karakter pada LCD diatur oleh pin EN, RS dan RW. Jalur EN dinamakan *Enable*. Jalur ini digunakan untuk memberitahu LCD bahwa anda sedang mengirimkan sebuah data. Untuk mengirimkan data ke LCD, maka melalui program EN harus dibuat logika low "0" dan set pada dua jalur kontrol yang lain RS dan RW. Ketika dua jalur yang lain telah siap, set EN dengan logika "1" dan tunggu untuk sejumlah waktu tertentu (sesuai dengan data sheet dari LCD tersebut) dan berikutnya set EN ke logika low "0" lagi.

Jalur RS adalah jalur *Register Select*. Ketika RS berlogika low "0", data akan dianggap sebagai sebuah perintah atau instruksi khusus (seperti *clear screen*, posisi kursor, dll). Ketika RS berlogika high "1", data

yang dikirim adalah data text yang akan ditampilkan pada *display LCD*. Sebagai contoh, untuk menampilkan huruf "T" pada layar LCD maka RS harus diset logika high "1".

Jalur RW adalah jalur *layer_1 Read/Write*. Ketika RW berlogika low (0), maka informasi pada *bus data* akan dituliskan pada layar LCD. Ketika RW berlogika high "1", maka program akan melakukan pembacaan memori dari LCD. Sedangkan pada aplikasi umum pin RW selalu diberi logika low "0".

Pada akhirnya, *bus data* terdiri dari 4 atau 8 jalur (bergantung pada *mode operasi* yang dipilih oleh *user*).

Anonimaus (1987) menyebutkan beberapa perintah dasar yang harus dipahami dalam inisialisasi *LCD Character* yaitu:

a. Clear display

Perintah *clear display* digunakan untuk membersihkan semua tampilan dan mengembalikan *Cursor* ke posisi semula atau posisi awal (alamat 0) kode spasi 20h menulis semua alamat *DDRAM*, dan alamat *DDRAM* 0 diset ke AC. Jika digeser tampilan kembali seperti posisi aslinya. Setelah perintah membersihkan layar (*display clear*) dieksekusi, *mode entry increment* perintah *display clear* terdapat pada Tabel 3.2.

	RS	R/W	DB7							DB0
Code	0	0	0	0	0	0	0	0	1	

D: Ketika $D = 1$ tampilan ON (aktif)

Ketika $D = 0$ tampilan OFF

Jika $D = 0$ digunakan, maka data tampilan tinggal di *DDRAM*, mengubah 0 menjadi 1 untuk menampilkan data.

C: Ketika $C = 1$ *Cursor* tertampil aktif

Ketika $C = 0$ *Cursor* tidak tertampil

Cursor yang ditampilkan dibawah ini berbentuk karakter matrik bintang 5×7 jika *Cursor* pada posisi OFF maka data tampilan dituliskan ke *DDRAM* atau satu persatu diatur oleh I/D.

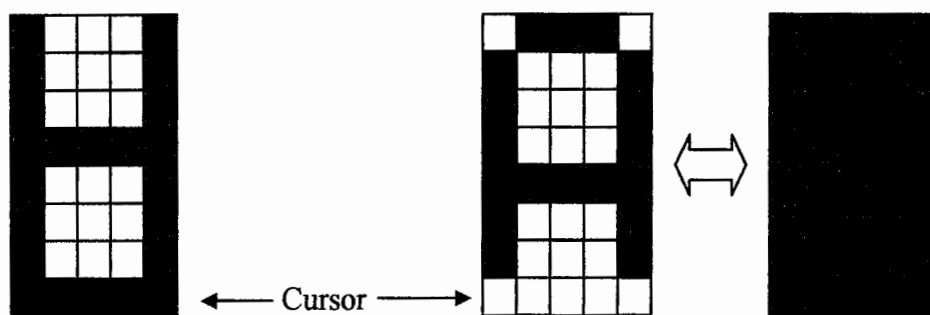
B: Ketika $B = 1$ karakter pada posisi *Cursor* tersebut mulai *blinking*.

Ketika $B = 0$ tidak *blink*

Sebuah contoh karakter matrik 5×7 dapat dilihat pada Gambar 3.4.

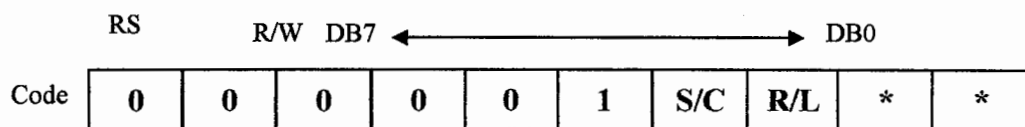
$C = 1$ (*Cursor Display*)

$B = 1$ (*blinking*)



Gambar 3.4. Sebuah karakter matrik 5×7

e. **Cursor (Display Shift)**



* = invalid bit

Adalah perintah untuk menggerakkan *Cursor* dan menggeser tampilan tanpa mengubah isi pada *DDRAM*. Posisi *cursor* setara dengan *AC*. Perintah ini dapat digunakan untuk mengoreksi dan memperbaiki tampilan, karena posisi *cursor* pada penampil dapat digeser tanpa menulis atau membaca penampil data. Selama kapasitas *DDRAM* adalah 40 karakter dan 2 baris, *Cursor* digeser dari 40 digit pada baris 1 ke digit 1 pada baris ke 2. Tampilan pada baris 1 dan 2 digeser pada saat yang sama oleh karena itu contoh tampilan pada baris 2 tidak bergeser ke baris 1. Proses pengoperasian *Cursor* (*display shift*) dapat dilihat pada Tabel 3.3 berikut:

Tabel 3.3. Proses pengoperasian *Cursor* (*display shift*)

S/C	R/L	Operasi
0	0	Posisi <i>Cursor</i> digeser ke kiri (<i>AC</i> berkurang 1)
0	1	Posisi <i>Cursor</i> digeser ke kanan (<i>AC</i> bertambah 1)
1	0	Semua tampilan digeser ke kiri oleh <i>Cursor</i>
1	1	Semua tampilan digeser ke kanan oleh <i>Cursor</i>

f. Fungsi Set

Perintah fungsi set digunakan untuk mengeset *interface* panjang data. DL: *interface* panjang data (*interface data length*) , ketika DL = 1 panjang data diset pada 8 bit (DB7 ke DB0), ketika DL = 0 panjang data diset pada 4 bit. Empat bit pertama ditransfer lebih dahulu baru kemudian diikuti 4 bit berikutnya. Perintah set fungsi harus dieksekusi lebih dahulu sampai semua perintah dijalankan, kecuali untuk *busy flag*/pembacaan alamat. Jika perintah lainnya dijalankan terlebih dahulu maka tidak ada perintah fungsi kecuali dengan mengubah eksekusi *interface* panjang data.

Di dalam LCD memiliki *Display Data Ram (DDRAM)* mempunyai kapasitas 80x8 bit dan menyimpan data penampil 80 buah kode karakter 8bit. Beberapa area penyimpanan DDram yang mana tidak dapat digunakan untuk *display* (penampil) dapat digunakan sebagai data RAM umum (*general data RAM*).

ROM pembangkit karakter (*CGROM*) mempunyai 192 tipe karakter yang tersimpan didalamnya dengan 5x7 matrik bintik dari kode karakter 8bit. Tabel 3.4 memperlihatkan kode karakter *CGROM* dan contoh-contoh karakter yang ditampilkannya.

Tabel 3.4. Data kode karakter pada CGROM

Lower 4 Bits	Upper 4 Bits																
		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
xxxx0000	CG RAM (1)			0	0	P	'	P				-	9	3	α	p	
				!	1	A	Q	a	q			•	7	4	ä	q	
xxxx0010	(2)			"	2	B	R	b	r			「	イ	ツ	×	β	θ
	(3)			#	3	C	S	c	s			」	ウ	テ	ε	ω	
xxxx0100	(4)			\$	4	D	T	d	t			、	エ	ト	μ	Ω	
	(5)			%	5	E	U	e	u			•	オ	ナ	1	6	Ü
xxxx0110	(6)			&	6	F	V	f	v			ヲ	カ	ニ	ヨ	ρ	Σ
	(7)			'	7	G	W	g	w			ア	キ	ヌ	ウ	g	π
xxxx1000	(8)			(	8	H	X	h	x			イ	ク	ネ	リ	5	×
	(1)			)	9	I	Y	i	y			ウ	ケ	ル	ル	"	4
xxxx1010	(2)			*	:	J	Z	j	z			エ	コ	ハ	レ	j	≠
	(3)			+	;	K	L	k	{			オ	サ	ヒ	ロ	*	π
xxxx1100	(4)			,	<	L	¥	l	l			ハ	シ	フ	フ	φ	π
	(5)			-	=	M	J	m	}			ユ	ズ	ハ	ン	ト	÷
xxxx1110	(6)			.	>	N	^	n	→			ヨ	セ	ホ	〃	π	
	(7)			/	?	O	_	o	+			ッ	リ	マ	〃	ö	■
xxxx1111	(8)																

Sumber: LCD User Manual

Sekarang ini, berbagai jenis LCD telah dikembangkan. Dari jenis LCD biasa, *Passive-Matrix LCD* (PMLCD), hingga *Thin-Film Transistor Active-Matrix LCD*

(TFT-AMLCD). Kemampuan *LCD* juga telah ditingkatkan, dari yang *monocrom* hingga yang mampu menampilkan ribuan warna.

Dari data penampil diatas *LCD* (*Liquid Cristal Display*) sangat efektif untuk penampil karena selain daya yang dibutuhkan kecil, *LCD* juga dapat menampilkan karakter tulisan, angka dan simbol dengan sempurna yang merupakan hasil penelitian Ivan (2008).

3.2. Keyboard

Nalwan dan Wichit (2003) menyebutkan bahwa antarmuka *keyboard* standard adalah hubungan komunikasi serial. "Serial" yang menandakan bahwa data dikirim satu bit pada suatu waktu tertentu pada satu jalur tunggal. Serial komunikasi dipilih untuk antara muka *keyboard* karena hal ini *simple* dan mudah diterapkan, dan tidak banyak data per detik yang harus dikirimkan pada antar muka *keyboard*. Ada empat jalur yang menghubungkan *keyboard* dan *PC* (*Personal Computer*). Dua dari ini adalah *power supply* dan *ground*, yang digunakan untuk memberikan daya ke rangkaian *keyboard*. Dua sinyal yang lain adalah sebagai berikut:

1. *Data Keyboard*: Disinilah jalur bit-bit data dan perintah yang akan dikirimkan ke sistem *PC* dari *keyboard*.
2. *Clock Keyboard*: Ini merupakan sinyal *clock* regular, dengan suatu nilai yang berosilasi dari logika "1" ke "0" dengan pola yang teratur. Tujuan dari sinyal *clock* ini adalah untuk mensinkronisasi *keyboard* dan sistem, sehingga mereka selalu bekerja secara bersamaan.

Keyboard PC AT yang terkoneksi, hanya menggunakan empat buah kabel, kabel ini ditunjukkan pada Gambar 3.5, untuk 5 pin DIM dan PS/2.



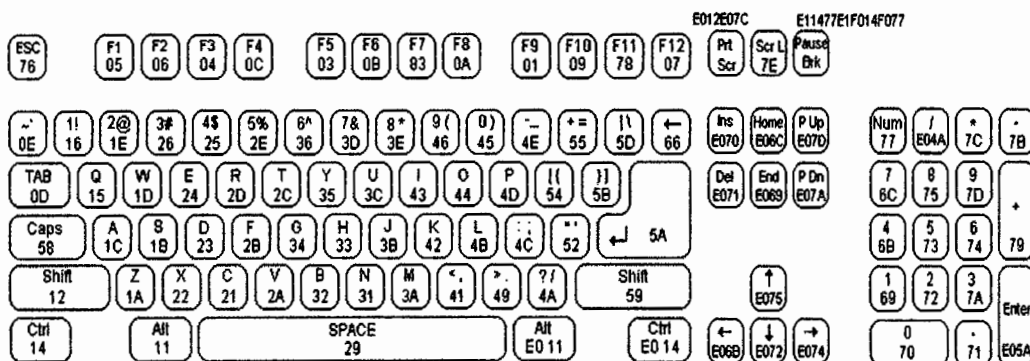
- 1.KBD Clock
- 2.KBD Data
- 3.N/C
- 4.GND
5. +5V (VCC)



1. KBD Clock
2. GND
3. KBD Data
4. N/C
5. +5V (VCC)
6. N/C

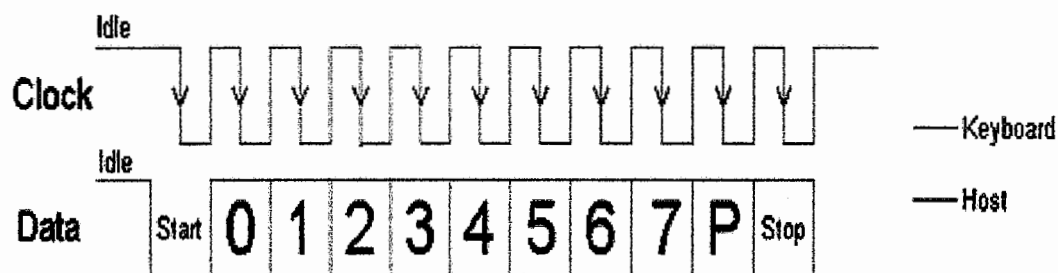
Gambar 3.5. Keyboard konektor (a) XT (b) PS/2

Ivan (2008) menyebutkan setiap kali salah satu tombol keyboard ditekan atau dilepas, maka akan mengirimkan kode ke mikrokontroller. Kode tersebut yang dinamakan scan code. Gambar 3.6 adalah gambar keyboard dengan scan code masing - masing tombol.



Gambar 3.6. Scan code Keyboard

Transmisi data, dari *keyboard* ke sistem, dilakukan dengan *frame* 11 bit. Bit pertama adalah bit *start* (logika 0), diikuti dengan 8 bit data (*LSB first*), atau bit paristas (paritas ganjil) dan bit *stop* (logika 1) yang ditunjukkan pada Gambar 3.7.

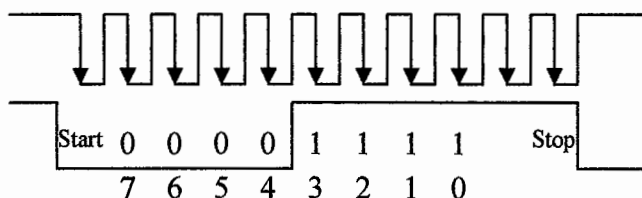


Gambar 3.7. Sinyal *clock* dan data

Setiap bit harus dibaca pada sisi turun dari *clock*. Agar dapat lebih jelas untuk pembacaan *scan code* dapat dilihat pada gambar di bawah ini adalah contoh pengambilan data *scan code* dari *keyboard*.

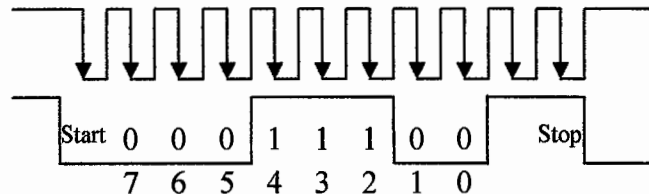
- Saat tombol tidak ditekan

Saat tombol pada *keyboard* tidak ditekan *scan code* yang dihasilkan sebesar 15 desimal atau 0f hexa.



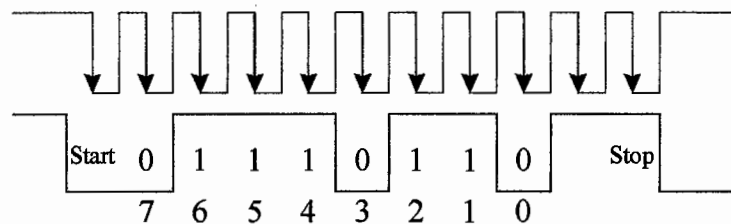
- Saat tombol A ditekan

Saat tombol A pada *keyboard* ditekan *scan code* yang dihasilkan sebesar 28 desimal atau 1C hexa.



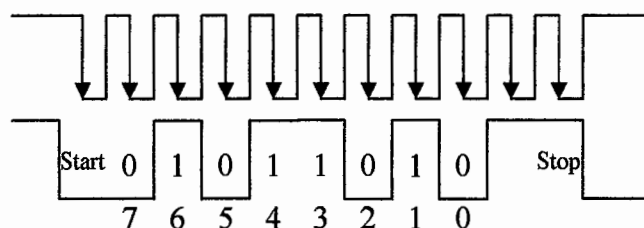
- Saat tombol *ESC* ditekan

Saat tombol *ESC* pada *keyboard* ditekan *scan code* yang dihasilkan sebesar 118 desimal atau 76 hexa.



- Saat tombol *ENTER* ditekan

Saat tombol *ENTER* pada *keyboard* ditekan *scan code* yang dihasilkan sebesar 90 desimal atau 5a hexa.



3.3. Arsitektur Mikrokontroler

Putra (2006) menyebutkan bahwa mikrokontroler AT89S52 merupakan mikrokontroler CMOS 8 bit dengan 8 Kbit *in-sistem programmable Flash memory*. Mikrokontroler ini berteknologi memori *non volatile* kerapatan tinggi dari Atmel yang kompatibel dengan mikrokontroler standar industri MCS-51 baik pin kaki IC maupun set instruksi serta harganya yang cukup murah.

Spesifikasi mikrokontroler AT89S52 antara lain:

- a. Sesuai dengan produk-produk MCS-51®.
- b. 8K Bytes *In-System Programmable (ISP) Flash Memory*. Ketahanan: 1000 *Write/Erase Cycles*.
- c. Tegangan pengoperasian 4.0 V - 5.5V.
- d. Beroperasi pada: 0 Hz - 33 MHz.
- e. Tiga tingkat penguncian memori program.
- f. 256 x 8-bit *Internal RAM*
- g. 32 jalur I/O yang dapat diprogram.
- h. Tiga buah 16-bit *Timer/Counters*.
- i. 8 sumber interupsi.
- j. Saluran serial *UART* penuh rangkap dua.
- k. Sedikit daya yang tidak terpakai dan mode penurunan daya.
- l. Penginterupsian perbaikan dari mode penurunan daya.
- m. *Watchdog Timer*.
- n. *Dual Data Pointer*.
- o. *Power-off Flag*.
- p. Waktu pemrograman singkat.
- q. Fleksibilitas Pemrograman *ISP (Byte dan Page Mode)*.

Port ini juga digunakan sebagai saluran alamat pada saat pemrograman dan verifikasi.

Penjelasan masing-masing fungsi *pin* dapat dilihat pada Tabel 3.5.

Tabel 3.5. Fungsi *pin port* 1 mikrokontroler AT89S52

Pin port	Fungsi alternative
P1.0	T2 (<i>Input</i> cacah eksternal ke <i>Timer/Counter 2, clock out</i>)
P1.1	T2EX (<i>Timer/Counter 2 capture/reload</i> dan kontrol arah)
P1.5	MOSI (digunakan untuk <i>In-sistem Programming</i>)
P1.6	MISO (digunakan untuk <i>In-sistem Programming</i>)
P1.7	SCK (digunakan untuk <i>In-sistem Programming</i>)

b. Pin 9 (RST)

Merupakan masukan reset (aktif tinggi) untuk dua siklus mesin. Bit *DISTRO* di *SFR AUXR* (alamat 8EH) dapat digunakan untuk menonaktifkan fitur ini.

c. Pin 10 sampai 17

Port 3 merupakan saluran (*bus*) dua arah *input/output* 8 bit *internal pull-up* yang memiliki fungsi pengganti. Ketika logika 1 diberikan ke port 3, maka *internal pull-up* akan menset port pada kondisi high dan port 3 dapat digunakan sebagai saluran *input*. Bila fungsi pengganti tidak dipakai, maka fungsi ini dapat digunakan sebagai port paralel 8 bit serbaguna. Selain itu, sebagian dari port 3 dapat berfungsi sebagai sinyal kontrol

pada saat proses pemrograman dan verifikasi. Port 3 juga melayani fungsi dari fitur utama pada AT89S52. Adapun fungsi penggantinya seperti pada Tabel 3.6.

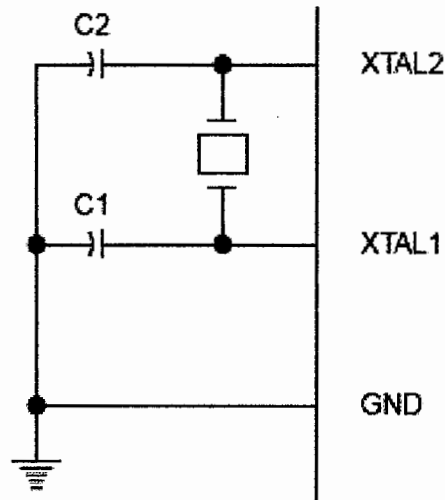
Tabel 3.6. Fungsi pengganti dari port 3

Bit	Nama	Fungsi Alternatif
P3.0	RXD	<i>port input serial</i>
P3.1	TXD	<i>port output serial</i>
P3.2	INT0'	interupsi eksternal 0
P3.3	INT1'	interupsi eksternal 1
P3.4	T0	<i>input eksternal timer/pencacah 0</i>
P3.5	T1	<i>input eksternal timer/pencacah 1</i>
P3.6	WR'	jalur menulis memori data eksternal <i>strobe</i>
P3.7	RD'	jalur membaca memori data eksternal <i>strobe</i>

d. Pin 18 dan 19

Jalur ini merupakan masukan ke penguat osilator berpenguat tinggi. Mikrokontroler ini memiliki seluruh rangkaian osilator yang diperlukan pada chip, kecuali rangkaian kristal yang mengendalikan frekuensi osilator. Oleh karena itu, pin 18 dan 19 sangat diperlukan untuk dihubungkan dengan kristal. Selain itu, XTAL 1 juga dapat dipakai sebagai *output* untuk *inverting osilator* amplifier dan *input* ke rangkaian *internal clock*. Sedangkan XTAL 2 merupakan *output* dari *inverting oscillator*

amplifier. Rangkaian osilator pada umumnya dapat dilihat pada Gambar 3.9.



Gambar 3.9. Rangkaian osilator yang umum

e. Pin 20

Merupakan *ground* sumber tegangan yang diberi simbol *GND*.

f. Pin 21 sampai 28

Pin ini merupakan *port 2* yang menjadi saluran (*bus*) *input/output* dua arah 8 bit dengan *internal pull-up*. Saat pengambilan data dari program memori eksternal dengan menggunakan alamat 16 bit (*MOVX @ DPTR*), maka *port 2* akan berfungsi sebagai saluran (*bu*) alamat tinggi (A8-A15). Sedangkan pada saat mengakses ke data memori eksternal menggunakan alamat 8 bit (*MOVX @ R1*), *port 2* mengeluarkan isi P2 ke *Special Function Register*.

g. Pin 29

Program Store Enable (PSEN') merupakan sinyal pengontrol untuk mengakses program memori eksternal yang masuk ke dalam saluran (*bus*) selama proses pemberian atau pengambilan instruksi (*fetching*) ketika AT89S52 sedang mengeksekusi kode dari memori program eksternal, *PSEN'* akan diaktifkan dua kali dalam setiap siklus mesin, kecuali aktivasi kedua *PSEN'* ditunda selama akses ke memori data eksternal.

h. Pin 30

ALE/PROG' (*Address Latch Enable*) merupakan penahanan alamat memori eksternal (pada port 1) selama menghasilkan memori eksternal. Pin ini juga sebagai pulsa (sinyal) *input* program (*PROG*) selama proses pemrograman. Pada operasi standar, sinyal *ALE* mempunyai kecepatan konstan, yaitu 1/6 dari frekuensi osilator dan sering digunakan sebagai timing eksternal atau tujuan *clocking*. *ALE* juga dapat dinonaktifkan dengan menset bit 0 dari *SFR* pada lokasi 8EH. Mengeset *ALE* pada keadaan nonkatif tidak mempunyai efek jika mikrokontroler berada pada mode eksekusi eksternal.

i. Pin 31

External Access Enable (EA) merupakan sinyal kontrol untuk pembacaan memori program, apabila diset rendah (L), maka mikrokontroler akan melaksanakan seluruh instruksi dari memori program eksternal. Sedangkan apabila diset tinggi (H) maka mikrokontroler akan melaksanakan instruksi dari memori program *internal* saat isi program *Counter*

kurang dari 4096. Port ini juga berfungsi sebagai tegangan pemrograman ($V_{pp} = +12V$) selama proses pemrograman berjalan.

j. Pin 32 sampai 39

Port 0 merupakan saluran (*bus*) *input/output* 8 bit *open collector* yang dapat digunakan sebagai *multiplex bus* alamat rendah dan *bus* data selama adanya akses ke memori program eksternal. Pada saat proses pemrograman dan verifikasi port 0 digunakan sebagai saluran (*bus*) data. eksternal *pull-up* dibutuhkan selama proses verifikasi.

k. Pin 40

Merupakan sumber tegangan positif yang diberi simbol V_{cc} .

3.3.1. Organisasi Memori

Mikrokontroler MCS51 memisahkan memori data dengan memori program. Memori data dan memori program yang bisa diakses oleh MCS51 masing-masing adalah 64kByte. Ruang memori sebesar 64kByte itu tidak seluruhnya berada di *chip (on chip memory)*, tetapi bisa ditambahkan memori eksternal.

a. Memori Program

Jika pin $\overline{EA}$ dihubungkan ke *GND*, program yang akan dikerjakan diambil semua dari memori eksternal (*ROM* eksternal). Jika $\overline{EA}$ dihubungkan ke *VCC*, program yang dikerjakan pada alamat 0000H hingga 1FFFFH diambil dari memori program internal (*on chip flash ROM*) dan program dialamat 2000H hingga FFFFFH diambil dari memori *ROM* eksternal.

b. Memori Data

AT89S52 memiliki memori data (RAM) sebesar 256 Byte. Memori ini dibedakan menjadi 128 Byte rendah dan 128 Byte atas. Memori data bagian 128 Byte atas mempunyai alamat yang sama dengan Register Fungsi Khusus/*Special Function Register (SFR)*. Hal ini berarti 128 Byte RAM atas dan SFR mempunyai alamat yang sama walaupun keduanya secara fisik terpisah. CPU membedakan memori yang akan diakses dengan cara pengalamatan yang berbeda. Dengan menggunakan pengalamatan tak langsung (*indirect addressing*) akan mengakses 128 Byte RAM atas dan dengan pengalamatan langsung (*direct Addressing*) akan mengakses ke SFR.

3.3.2. Memprogram Flash Mode Serial (In System Programming)

Kode program (instruksi) dapat diprogramkan ke dalam flash memori dengan interface ISP (*In System Programming*) secara serial dengan membuat input pin RST ke logika 1. Interface serial memakai pin SCK (*serial clock*, P1.7), MISO (*Master data In Slave Out*, P1.6) dan MOSI (*Master data Out Serial In*, P1.5). Setelah pin RST = 1, instruksi *Programming Enable* harus diberikan dahulu untuk mengaktif mode memprogram secara serial ini. Berikutnya instruksi *Chip Erase* untuk membuat isi flash terhapus dan diisi ulang. Instruksi *chip erase* akan membuat isi flash berubah ke FFH semua.

Pemrograman serial dapat dilakukan dengan memberi clock dari eksternal yang masuk ke pin XTAL1 atau memakai clock internal dengan menghubungkan kristal dan

kapasitor ke pin XTAL1 dan XTAL2. Frekuensi serial clock (SCK) haruslah kurang dari $1/6$ frekuensi oscillator. Jadi misalnya dipakai kristal 33MHz untuk frekuensi clock, maka frekuensi SCK maksimum adalah 2MHz.

3.4. Parallel Port

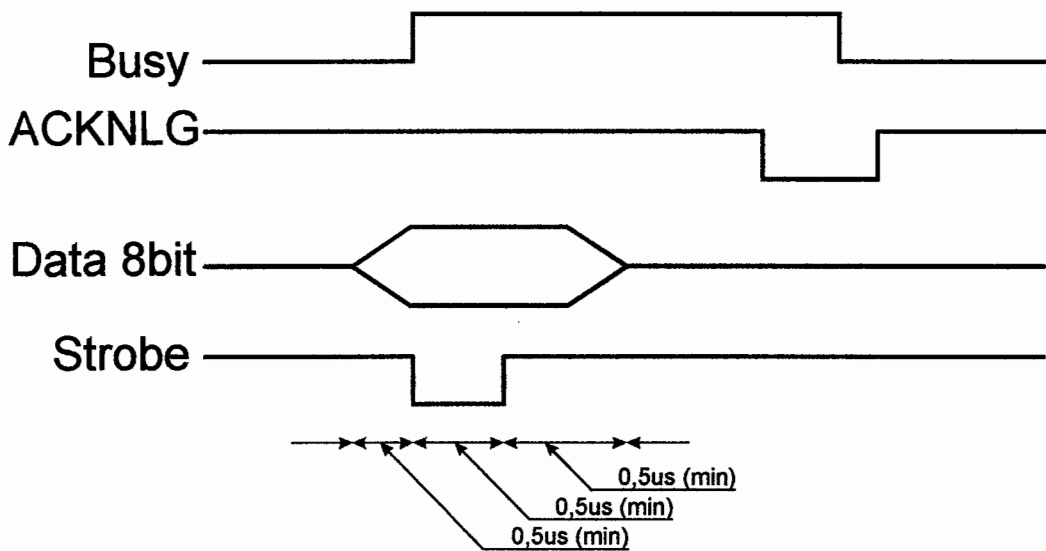
Wasito (1986) menyebutkan parallel port sering digunakan pada sistem komputer untuk menghubungkan komputer dengan device yang lain dengan pengiriman datanya secara parallel 8 bit seperti printer, control device, dll. Selain data 8 bit parallel port juga terdapat intrupsi-intrupsi yang lain untuk memberikan kode-kode perintah seperti terlihat pada Tabel 3.7.

Tabel 3.7. Konfigurasi pin parallel port

Pin	Sinyal SPP	Sinyal EPP	In/Out	Fungsi
1	Strobe	Write	Out	Saat line ini berlogika rendah mengindikasikan tulis dan saat berlogika tinggi mengindikasikan baca
2-9	Data 0-7	Data 0-7	In/Out	Data Bus dua arah
10	ACK	Interrupt	In	Mengindikasikan akhir data yang dikirim dan menandai bahwa data selanjutnya akan dikirim
11	Busy	Wait	In	Logika rendah mengindikasikan data sedang dikirim (busy)
12	Paper Out/End	Spare	In	Jika rendah, device memberikan tanda Acknowledges
13	Select	Spare	In	Extensibility flag
14	Auto Line Feed	Data Strobe	Out	Jika rendah, mengindikasikan data transfer
15	Error/Fault	Spare	In	Saat pemberian logika rendah dari device mengindikasikan reverse data dalam keadaan available
16	Initialize	Reset	Out	Reset - aktif rendah
17	Select Print	Address Strobe	Out	Jika rendah, mengindikasikan transfer alamat
18-25	GND	GND	GND	Ground

3.5. Printer Dot Matrik

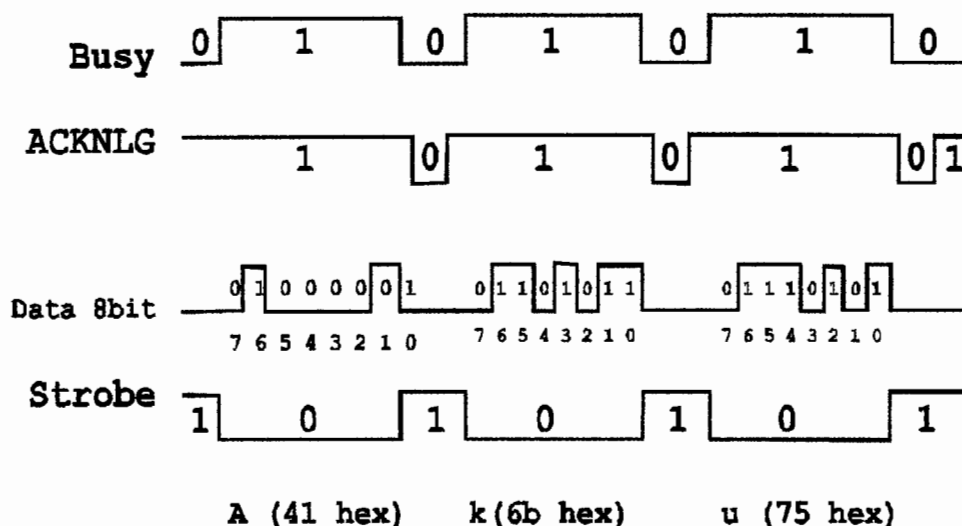
Anonimaus (1989) menyebutkan printer *dot matrik* sering digunakan di perusahaan ataupun industri, dimana keuntungan dari printer ini adalah khususnya dalam hasil pengeprintan, printer ini dapat langsung digandakan dalam satu kali pengeprintan dengan cara pemberian karbon diantara dua kertas atau lebih. Dimana printer *dot matik* ini memiliki kecepatan pengeprintan sebesar 150 karakter per detik dalam *10cpi* (*Characters per inch*) dan 180 karakter perdetik dalam *120cpi*. Transfer data dan intrupsinya dilakukan dengan menggunakan modus *parallel* seperti terlihat pada Gambar 3.10.



Gambar 3.10. Gelombang *timing* transfer data printer *dot matrik*

Terlihat pada Gambar 3.10, transfer data diawali dengan menyiapkan data 8 bit dari sistem kontrol (mikrokontroler) dan sebelum printer diberi data ASCII 8 bit terlebih dahulu sistem kontrol memeriksa apakah printer sedang sibuk atau tidak (saat sibuk pin *busy* akan memberikan logika 1). Saat dideteksi tidak sibuk maka untuk mulai mentransfer data ASCII 8 bit pin *strobe* akan diberikan pulsa 0 (*low*) selama ± 40 mikro detik yang dilanjutkan dengan memberikan logika 1 (*high*) pada pin *strobe*. Setelah data dikirimkan printer dot matrik akan memberikan sinyal *ACKNLG* dari *low* ke *high* bahwa data selanjutnya siap akan diterima oleh printer dot matrik.

Sebagai contoh untuk pengiriman data ASCII per karakter tulisan dapat dilihat pada Gambar 3.11 saat mencetak tulisan "Aku" dengan kode ASCII sebesar 31hexa (lihat Tabel 3.8).



Gambar 3.11. Gelombang *timing* transfer data printer *Dot* matrik untuk pengeprintan tulisan "Aku"

Tabel 3.8 menunjukkan kode *ASCII* pada printer dot matrik.

Tabel 3.8. Kode *ASCII* pada printer dot matrik

Fungsi	ASCII code (Hex)	Fungsi	ASCII code (Hex)
!	21	"	22
#	23	\$	24
%	25	&	26
'	27	(	28
)	29	*	2a
+	2b	,	2c
-	2d	.	2e
/	2f	0	30
1	31	2	32
3	33	4	34
5	35	6	36
7	37	8	38
9	39	:	3a
;	3b	<	3c
=	3d	>	3e
?	3f	@	40
A	41	B	42
C	43	D	44
E	45	F	46
G	47	H	48
I	49	J	4a
K	4b	L	4c
M	4d	N	4e
O	4f	P	50
Q	51	R	52
S	53	T	54
U	55	V	56
W	57	X	58
Y	59	Z	5a
[	5b	¥	5c
]	5d	^	5e
_	5f	`	60
a	61	b	62
c	63	d	64
e	65	f	66
g	67	h	68

Tabel 3.8. Kode ASCII pada printer dot
matrik (lanjutan)

Fungsi	ASCII code (Hex)	Fungsi	ASCII code (Hex)
i	69	j	6a
k	6b	l	6c
m	6d	n	6e
o	6f	p	70
q	71	r	72
s	73	t	74
u	75	v	76
w	77	x	78
y	79	z	7a
{	7b		7c
}	7d	→	7e
←	7f		

3.6. Metode-metode Perancangan

Metode perancangan merupakan suatu tindakan terhadap prosedur, teknik dan sifat alat bantu tertentu yang membantu aktivitas-aktivitas jenis tertentu yang digunakan dalam proses perancangan.

Cross (1994) menyebutkan metode perancangan dibagi dalam dua kelompok, yaitu metode kreatif dan metode rasional.

3.6.1. Metode Kreatif

Metode kreatif biasanya digunakan untuk meningkatkan produksi gagasan, menghilangkan hambatan dalam berkreatifitas, dan juga memperluas area pencarian solusi. Beberapa metode yang digunakan dalam menstimulasi pemikiran kreatif adalah sebagai berikut:

a. *Brainstorming*

Metode yang paling banyak dikenal adalah *brainstorming*. Metode ini dapat dikatakan merupakan hasil dari pendapat dan gagasan banyak orang yang terlibat pada suatu kelompok, dan sebaiknya mereka memiliki pengetahuan yang cukup mengenai permasalahan. Metode ini membangkitkan banyak gagasan (ide) dan pendapat yang akan disaring untuk memperoleh ide-ide baru yang bermanfaat untuk ditindaklanjuti. Dalam melakukan *brainstorming*, semua pihak dapat dengan bebas mengeluarkan ide-idenya tanpa harus ada kecaman dan kritikan.

b. *Sinektik*

Sinektik merupakan suatu aktivitas kelompok yang mencoba membangun, mengkombinasikan dan mengembangkan gagasan-gagasan untuk memberikan solusi kreatif terhadap masalah perancangan. Tujuan metode ini adalah mengarahkan aktivitas pemikiran yang spontan ke arah eksplorasi dan transformasi masalah-masalah perancangan. Perbedaannya dengan metode *brainstorming* adalah pada metode *brainstorming* dihasilkan gagasan sebanyak mungkin, sedangkan metode sinektik berusaha untuk menghasilkan suatu solusi tertentu.

c. Memperluas ruang pencarian

Hal yang biasanya menghalangi dalam proses berpikir kreatif adalah melihat adanya batasan dalam solusi yang akan dicapai. Pembatasan terhadap ide-ide kreatif akan membuat perancang susah dalam

mencari solusi yang tepat terhadap pekerjaannya. Pemikiran yang kreatif tersebut harus diperluas agar ide-ide perancang dapat dituangkan secara bebas sehingga diperoleh suatu hasil yang optimal.

Metode-metode yang dapat dipergunakan untuk membangkitkan ide secara kreatif antara lain:

- a. Penelitian langsung
- b. Inovasi teknologi
- c. Studi eksplorasi pemakai
- d. Integrasi teknologi dan pemasaran
- e. Teknik kreatifitas

Terdapat 5 tahap dalam teknik kreatifitas, yaitu:

- a. Identifikasi masalah
- b. Menganalisa penyebab timbulnya masalah
- c. Mengeluarkan semua alternatif pemecahan masalah
- d. Mengevaluasi keuntungan dan kekurangan dari setiap alternatif yang ada.
- e. Memilih alternatif terbaik.

3.6.2. Metode Rasional

Metode rasional lebih sering dikenal dengan sebutan metode perancangan daripada teknik kreatif. Pada dasarnya metode rasional dengan metode kreatif memiliki tujuan yang sama, misalnya memperluas ruang pencarian solusi atau memungkinkan pengadaan tim kerja dan grup pengambil keputusan.

Tidak sedikit perancang khawatir penggunaan metode rasional dapat mengekang (*straight jacket*) kreatifitas. Sesungguhnya metode rasional merupakan

metode perancangan yang sistematis tujuannya memperbaiki kualitas keputusan perancangan dan hasil akhir dari produk. Metode rasional menggabungkan aspek prosedural dari suatu perancangan dengan aspek struktural dari masalah perancangan.