

## II. TEORI PENUNJANG

### 1. PENGENALAN SISTEM TELEPON

Dengan adanya kemajuan teknologi sekarang ini, maka fungsi telepon sebagai alat telekomunikasi dua arah (*full duplex*) telah mengalami banyak perubahan baik rangkaian dasar, model serta perlengkapan telepon tersebut.

Pada mulanya hubungan antar pelanggan dilakukan secara manual oleh seorang operator melalui sentral telepon. Tetapi sejalan dengan perkembangan jaman, maka cara lama ini sudah ditinggalkan dan selanjutnya hubungan antar pelanggan ini dilakukan secara otomatis (biasa dikenal dengan nama sentral telepon otomatis).

#### 1.1 Klasifikasi Pesawat Telepon



Gambar 2.1

Klasifikasi Pesawat Telepon

Secara umum pesawat telepon dapat diklasifikasikan seperti gambar di atas. Pada dasarnya unsur-unsur telepon tersebut terdiri dari sistem bicara dan sistem pengkabelan.

a. Sistem bicara

Sebuah telepon akan menggunakan gelombang suara untuk mengontrol transmisi sinyal listrik dalam bentuk arus bolak-balik pada frekuensi yang sama pada penerimanya. Besarnya frekuensi dari percakapan biasanya berkisar antara 50 Hz sampai 10.000 Hz.

Untuk melakukan transmisi dengan range frekuensi sebesar itu akan membutuhkan biaya yang besar, karena itu sistem telepon dirancang untuk range frekuensi antara 300 Hz – 3400 Hz. Range ini cukup memadai untuk mengenali suara pembicara dan mengerti dengan jenis kata-katanya.

b. Sistem Pengkabelan

Dalam pengiriman dan penerimaan sebuah sinyal dalam sistem telepon elektronik digunakan jalur kabel, yang mana jalur ini dapat digunakan secara dua arah.

## 1.2 Mengirim Sebuah Nomor

Beberapa jenis pesawat telepon mengirimkan nomor telepon yang dituju menggunakan pulsa-pulsa listrik (*dial pulses*) sedangkan yang lainnya menggunakan nada (*audio tones*).

### 1.2.1 Dial Pulsing

Hampir semua tipe telepon yang menggunakan cara ini memiliki alat pemutar berbentuk lingkaran yang akan membuat rangkaian lokal *loop* berkondisi terbuka dan tertutup dengan interval waktu tertentu. Nomor yang dihasilkan dari pulsa-pulsa listrik ini tergantung pada seberapa jauh lingkaran nomor diputar sebelum dilepaskan kembali.

### 1.2.2 Dual Tone Multi frequency (DTMF)

Beberapa tipe telepon menggunakan metode yang lebih baru dengan menggunakan nada suara untuk mengirimkan nomor telepon. Sistem ini dapat digunakan hanya jika sentral telepon mempunyai piranti pemroses nada. Pesawat telepon tipe ini dilengkapi dengan tombol-tombol tekan dari nomor 0 – 9 dan simbol \* (*asterisk*) serta # (*octothorpe*). Menekan salah satu tombol akan mengakibatkan rangkaian elektronik yang ada menghasilkan dua nada *output* yang mewakili sebuah nomor.

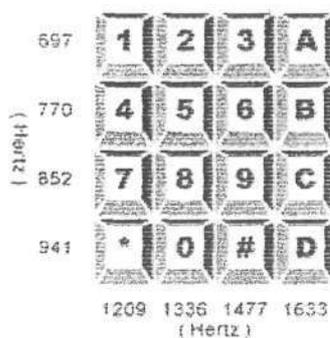
## 2. DTMF

DTMF adalah kependekan dari *dual tone multi frequency*, yang mana DTMF ini menggunakan dua nada. Setiap nada ditandai dengan angka-angka dan huruf yang dapat di lihat pada *keypad* telepon kabel maupun telepon selular, dan dibagi menjadi dua bagian nada dengan frekuensi 697 Hz sampai

dengan 941 Hz sebagai nada baris dan frekuensi 1209 Hz sampai dengan 1633 Hz sebagai nada kolom.

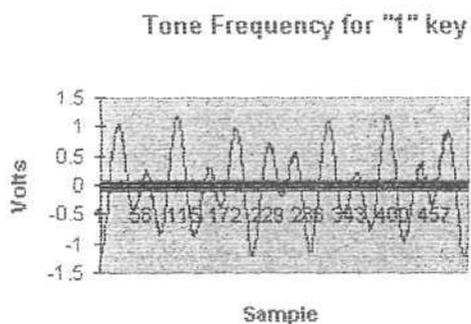
Frekuensi-frekuensi yang dipilih adalah tidak sama di sini untuk menghindari terjadinya harmonisa, hasil penjumlahan kedua frekuensi juga tidak boleh sama dengan frekuensi yang telah ada. Frekuensi yang dihasilkan harus mempunyai toleransi  $\pm 1.5\%$ , frekuensi yang lebih tinggi harus lebih keras dari frekuensi yang lebih rendah perbedaannya berkisar 4 db lebih keras.

Huruf A,B,C dan D pada *keypad* biasa digunakan untuk sistem telepon *Autovon* pada militer Amerika. Mereka menggunakannya untuk *Flash Override* – tombol A, *Flash* – tombol B, *Immediate* – tombol C dan *Priority* – tombol D. Tetapi biasanya pada pesawat telepon di Indonesia tombol A,B,C,D tidak dipakai. Kombinasi nada – nada DTMF dapat dilihat pada gambar di bawah.



Gambar 2.2

Kombinasi nada DTMF

Gambar 2.3<sup>1</sup>

Gelombang DTMF

## 2.1 Call Progress

*Call Progress* terdiri dari dua macam yaitu:

◆ Nada Sibuk

Berupa sinyal frekwensi 425 Hz yang muncul selama +/- 1/2 detik dengan interval +/- 1/2 detik pula

Gambar 2.4<sup>2</sup>

Sinyal Nada Sibuk

<sup>1</sup> DTMF Decoder/Encoder.html

<sup>2</sup> <http://www.cetrin.net.id/delta.com>, hal. 2

◆ Nada Sambung

Berupa sinyal frekwensi 425 Hz yang muncul selama +/- 2 detik dengan interval +/- 3 detik

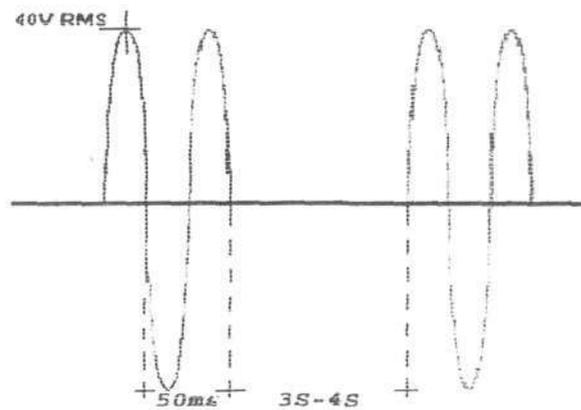


Gambar 2.5<sup>3</sup>

Sinyal Nada Sambung

## 2.2 Sinyal Dering

Sinyal ini berupa sinus dengan tegangan 40V RMS dengan frekwensi 20 Hz yang muncul selama 2 detik dan interval selama 3 - 4 detik.



Gambar 2.6<sup>4</sup>

Sinyal Dering

<sup>3</sup> Ibid, hal. 3

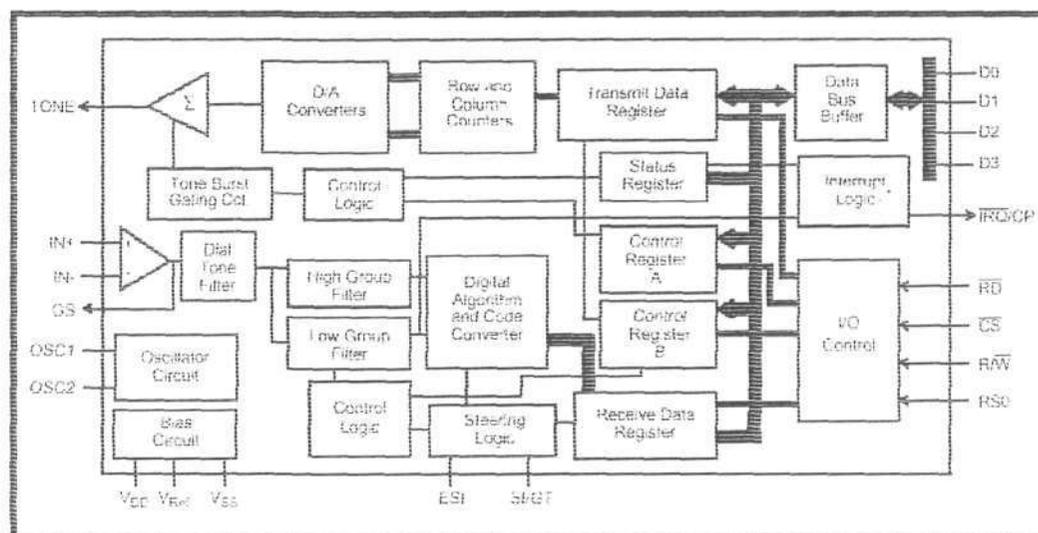
<sup>4</sup> Ibid, hal. 3

### 3. IC DTMF MT8888

#### 3.1 Deskripsi

Selain berfungsi sebagai DTMF transceiver, MT8888 ini juga dapat berfungsi sebagai *Call Progress Detector*. Sinyal *Call Progress* dideteksi dengan adanya frekwensi 425 Hz yang keluar pada kaki IRQ MT8888 dalam bentuk gelombang persegi.

Gelombang persegi frekwensi 425 Hz akan keluar pada kaki IRQ MT8888 setiap kali sinyal *call progress* yang berbentuk sinus masuk ke input ke MT8888 sehingga mikrokontroler dapat mengetahui adanya *call progress* dengan mendeteksi adanya gelombang persegi sebesar 425 Hz yang muncul pada *pin* IRQ.

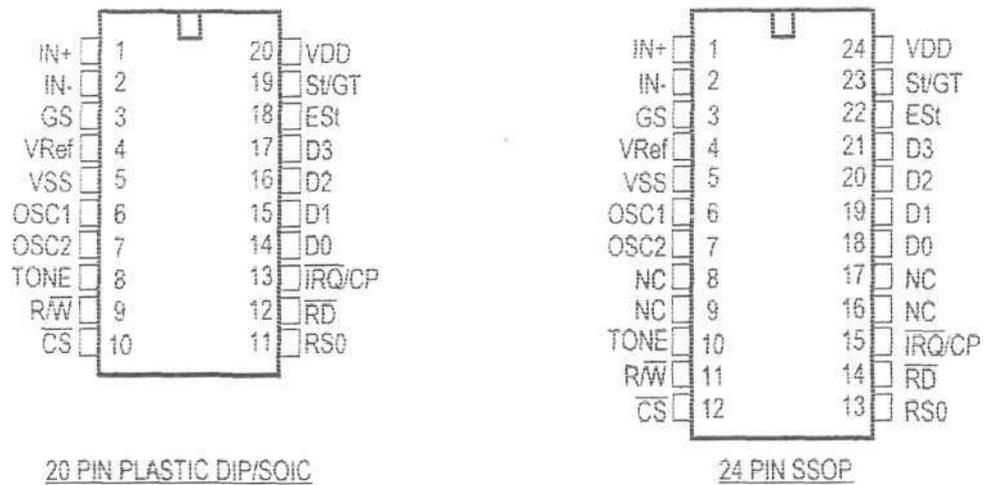


Gambar 2.7<sup>5</sup>

Blok Diagram MT8888

<sup>5</sup> Data book MT8888, hal. 4-89

### 3.2 Konfigurasi Dan Deskripsi Pin



Gambar 2.8<sup>6</sup>

#### Konfigurasi Pin MT8888

##### 3.2.1 IN+

*Non-inverting op-amp input.*

##### 3.2.2 IN-

*Inverting op-amp input.*

##### 3.2.3 GS

*Gain Select*, memberikan akses pada *differential amplifier* untuk hubungan dari *feedback resistor*.

##### 3.2.4 V<sub>Ref</sub>

*Reference Voltage output* ( $V_{DD}/2$ ).

<sup>6</sup> Ibid, hal. 4-90

### 3.2.5 $V_{DD}$ dan $V_{SS}$

*Pin* ini berfungsi sebagai sumber tegangan untuk MT8888 sebesar 5 volt dc.

### 3.2.6 OSC1

*Oscillator input*, *pin* ini juga dapat untuk mengontrol *external clock*.

### 3.2.7 OSC2

*Oscillator output*, crystal A3.579545 MHz terhubung antara OSC1 dan OSC2 lengkap dengan *oscillator circuit*. Dan bila OSC1 digunakan untuk keperluan *external* maka *pin* ini dibiarkan dalam keadaan *open circuit*.

### 3.2.8 TONE

Keluaran dari pemancar *internal* DTMF.

### 3.2.9 $\overline{WR}$

*Write*, Logika *low* pada kaki ini akan mengisi data dari D0 hingga D3 ke Data Register bila kaki RS = 0 dan mengisi data dari D0 hingga D3 ke *Control Register* bila kaki RS = 1.

### 3.2.10 $\overline{CS}$

*Chip Select*, logika *low* akan mengaktifkan IC MT8888 dan logika *high* akan menon-aktifkan IC MT8888 sehingga jalur *data bus* pada kondisi *high impedance*.

### 3.2.11 RS0

*Register Select*, logika *low* berarti akses ke *Data Register* dan logika *high* berarti akses ke *Control Register* ataupun *Status Register*.

### 3.2.12 $\overline{RD}$

*Read*, Logika *low* pada kaki ini akan mengalirkan isi *Data Register* (tempat data yang diterima atau yang dikirim oleh MT8888) bila kaki RS = 0 dan mengalirkan isi *Status Register* bila kaki RS = 1

### 3.2.13 $\overline{IRQ/CP}$

Logika *Low* terjadi pada saat sinyal DTMF masuk ke *Input* MT8888 yang bekerja pada *Mode* DTMF. Gelombang persegi terjadi pada saat sinyal *Call Progress* masuk ke *Input* MT8888 yang bekerja pada *Mode* *Call Progress*.

### 3.2.14 D0-D3

*Data Bus*, merupakan jalur aliran data pada saat *transmit*, *receive* DTMF, inisialisasi *control register* maupun pembacaan *status register*.

### 3.2.15 EST

*Early Steering Output*, pemberian sebuah *logic high* dari *algoritma digital* dapat terdeteksi sepasang nada yang *valid* (*signal condition*). Dan pada keadaan tidak ada sinyal maka EST akan kembali ke keadaan *logic low*.

### 3.2.16 St/GT

*Steering Input Guard Time Output (bidirectional)*. Sebuah tegangan lebih besar dari  $V_{Tst}$  terlihat pada St karena perlengkapan untuk *register* menangkap sepasang nada dan memperbaiki *ouput latch*. Sebuah tegangan kurang dari  $V_{Tst}$  perangkat bebas memilih sepasang nada. GT *output* bertindak untuk *me-reset external steering time-constant*.

### 3.3 Call Progress Detector

Selain berfungsi sebagai *DTMF Transceiver*, MT8888 juga dapat berfungsi sebagai *Call Progress Detector*. Sinyal *Call Progress* dideteksi dengan adanya frekuensi 425 Hz yang keluar pada kaki IRQ MT8888 dalam bentuk gelombang persegi.

Gelombang persegi frekuensi 425 Hz akan keluar pada kaki IRQ MT8888 setiap kali sinyal *call progress* yang berbentuk sinus masuk ke *input* MT8888 sehingga mikrokontroler dapat mengetahui adanya *call progress* dengan mendeteksi adanya gelombang persegi sebesar 425 Hz yang muncul pada *pin* IRQ.

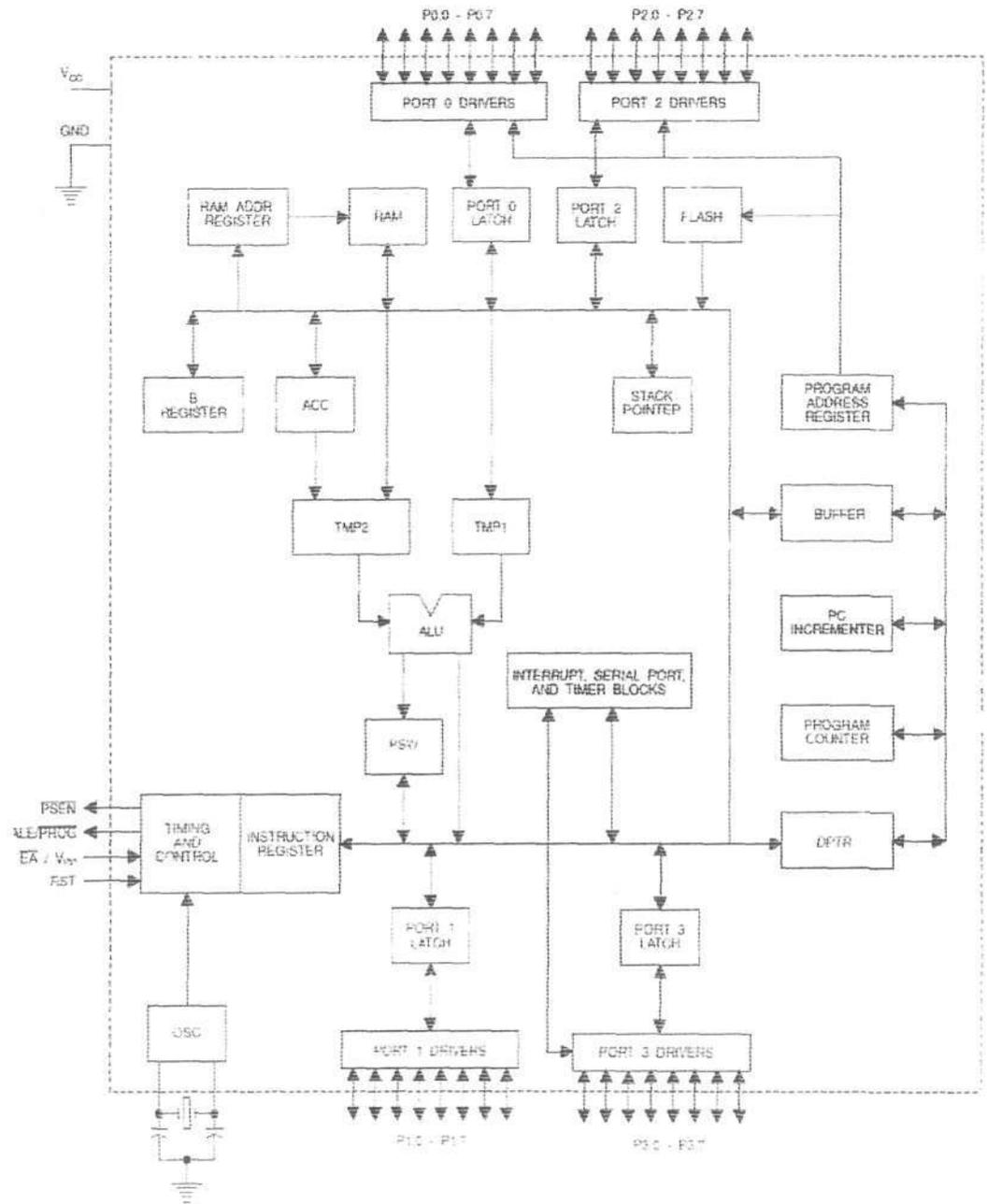
## 4. MICROCONTROLLER AT89C51

### 4.1 Deskripsi

Keluarga mikrokontroler MCS-51 merupakan keluarga mikrokontroler yang banyak digunakan untuk aplikasi pengontrolan saat ini. Selain tangguh dan efisien, keluarga ini banyak didukung oleh *vendor hardware* yang menyediakan banyak performa tambahan pada sistem mikrokontroler ini. Sebagai salah satu vendor besar di dunia, ATMEL memproduksi mikrokontroler AT89C51 yang merupakan salah satu jenis mikrokontroler yang banyak digunakan pada saat ini. Mikrokontroler AT89C51 memiliki kompatibilitas penuh dengan mikrokontroler keluarga MCS-51 terutama pada bahasa pemrogramannya.

Mikrokontroler AT89C51 memiliki spesifikasi antara lain memiliki 4 Kilo Bytes *Flash Programmable and Erasable Read Only (PEROM)* yang dapat diprogram ulang sekitar 1000 kali *write* atau *erase cycle*, 128 byte RAM, 32 jalur I/O yang dapat diprogram, dua buah 16-bit *timer / counter*, arsitektur dengan lima *vektor interrupt* dengan dua *level prioritas*, serial port yang *full duplex*, dan *on-chip oscillator*. I/O pinnya dapat menerima arus *sink* yang cukup kuat untuk menyalakan sebuah LED, dan mempunyai dua mode hemat energi yaitu *idle mode* dan *power down mode*.

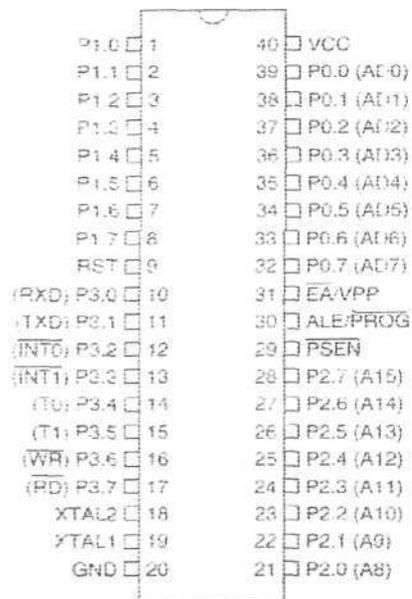
Blok diagram dari microcontroller AT89C51 dapat dilihat pada gambar berikut:

Gambar 2.9<sup>7</sup>

## Blok Diagram Microcontroller AT89C51

<sup>7</sup> Data book MCS51, hal 2

## 4.2 Konfigurasi Dan Deskripsi Pin



Gambar 2.10<sup>8</sup>

### Konfigurasi PIN Microcontroller AT89C51

#### 4.2.1. VCC (pin40 ) dan GND ( pin 20 )

Berfungsi sebagai sumber tegangan untuk mikrokontroler.

#### 4.2.2. Port 0 ( pin 32 s/d pin 39 )

Port 0 terdiri dari 8-bit *open-drain bi-directional I/O port* yang dapat berfungsi sebagai *input* maupun *output*. Sebagai *output port*, tiap *pin* dapat menerima delapan buah *input* TTL. Bila *port* ini ber-*logic '1' (high)*, maka *port* ini dapat difungsikan sebagai *input* dengan impedansi tinggi. *Port 0* juga dapat dikonfigurasi

<sup>8</sup> Ibid, hal 1

menjadi *multiplexed low-order address data bus* saat mengakses program dan memori *data eksternal*.

#### 4.2.3. Port 1 ( pin 1 s/d pin 8)

*Port 1* terdiri dari 8-bit *bi-directional I O port* dengan *internal pullups* (penguat arus secara internal) dan dapat difungsikan sebagai *input* maupun *output*. *Buffer output* dari *port 1* dapat menerima empat buah input TTL. Ketika *Port 1* mendapat *logic '1' (high)*, maka *port 1* akan di-*puled high internal pull-up* sehingga dapat difungsikan sebagai *input*.

#### 4.2.4. Port 2 ( pin 21 s/d pin 17 )

*Port 2* ini sama dengan *port 1*, terdiri dari 8-bit *bi-directional I O port* dengan *internal pullups* dan dapat difungsikan sebagai *input* maupun *output*. Sama dengan *port 1*, *port 2* juga memiliki *buffer* pada *output*-nya yang dapat menerima empat buah *input* TTL.

#### 4.2.5. Port 3 ( pin 10 s/d pin 17 )

*Port 3* juga terdiri dari 8-bit *bi-directional I O port* dengan *internal pull ups* dan dapat berfungsi sebagai *input* maupun *output*. *Buffer output* dari *port 3* dapat menerima empat buah *input* TTL.

Sebagai *input*, *pin port 3* yang secara *external di-pull low* akan mengeluarkan arus (  $I_{IL}$  ) karena adanya *pull-up*. *Port 3* pada AT89C51 juga menyediakan berbagai fungsi khusus, sebagai mana diberikan dalam tabel berikut:

Tabel 2.1

Fungsi Lain Port 3 AT89C51

PIN PORT	FUNGSI LAIN
Port 3.0	RXD (serial input port)
Port 3.1	TXD (serial output port)
Port 3.2	INT0 (external interrupt 0)
Port 3.3	INT1 (external interrupt 1)
Port 3.4	T0 (timer 0 external input)
Port 3.5	T1 (timer 1 external input)
Port 3.6	WR (external data memory write strobe)
Port 3.7	RD (external data memory read strobe)

#### 4.2.6. Pin Reset ( pin 9 )

Saat *input pin* ini mendapat *logic '1' (high)*, maka *reset* akan terjadi setelah dua *machine cycle*, dan *me-reset* nilai PC menjadi nol serta *me-reset* semua nilai *register*.

#### 4.2.7. Pin ALE/ $\overline{\text{PROG}}$ ( pin 30 )

*Pulsa output* dari *Address Latch Enable (ALE)* digunakan untuk mengunci *low byte* dari *address* selama mengakses ke *memory eksternal*. *Pin* ini juga merupakan *pulsa program input (PROG)* selama *flash programming*. Jika diinginkan, ALE operation dapat di non-aktifkan dengan *men-set bit 0* dari SFR pada 8EH.

Dengan *set bit* ini, ALE akan aktif hanya dengan menggunakan perintah MOVX atau MOVC.

#### 4.2.8. Pin PSEN ( pin 29 )

*Programmable Strobe Enable* adalah pembacaan *strobe* pada program memori *eksternal*. Saat AT89C51 sedang mengeksekusi data dari program memori *eksternal*, PSEN yang sedang aktif diloncati selama tiap akses ke data memori *eksternal*.

#### 4.2.9. Pin EA/VPP ( pin 31 )

*External Access Enable (EA)* harus dihubungkan dengan GND agar dapat mengaktifkan *device* untuk mengambil data dari program memori *eksternal* yang berlokasi dari  $0000_{16}$  sampai  $FFFF_{16}$ . Sebaliknya EA harus dihubungkan dengan VCC untuk dapat mengeksekusi program secara *internal*.

#### 4.2.10. XTAL1 ( pin 19 )

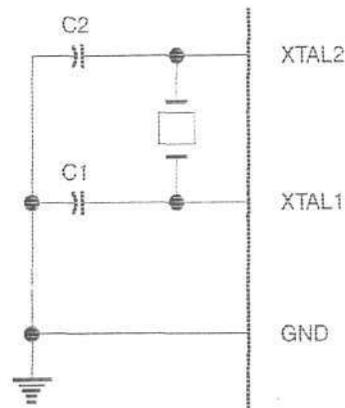
Sebagai *input* untuk *inverting oscillator amplifier* dan sebagai *input* pada *clock operation circuit internal*.

#### 4.2.11. XTAL2 ( pin 18 )

Merupakan *output* dari *inverting oscillator amplifier*.

### 4.3 Karakteristik Osilator

XTAL1 dan XTAL2 adalah *input output inverting amplifier* yang dapat berfungsi sebagai *on-chip oscillator* dengan menambahkan kristal *quartz* ataupun resonator keramik. Untuk konfigurasi dapat dilihat pada gambar di bawah ini.

Gambar 2.11<sup>9</sup>

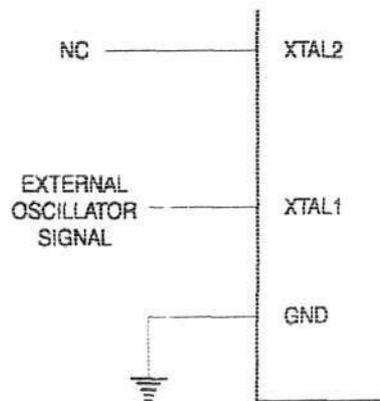
## Konfigurasi Osilator Menggunakan Kristal

Nilai C1 dan C2 adalah  $30 \text{ pF} \pm 10 \text{ pF}$  bila menggunakan kristal, sedangkan untuk resonator keramik nilai C1 dan C2 yang digunakan adalah  $40 \text{ pF} \pm 10 \text{ pF}$ .

Untuk menggunakan *external clock*, XTAL2 harus dibiarkan dalam kondisi tidak terhubung. Untuk konfigurasinya dapat dilihat pada gambar 2.12 berikut ini.

---

<sup>9</sup> Ibid, hal 4

Gambar 2.12<sup>10</sup>

Konfigurasi Osilator Menggunakan  
External Oscillator Signal

## 5. RANGKAIAN DRIVER LCD

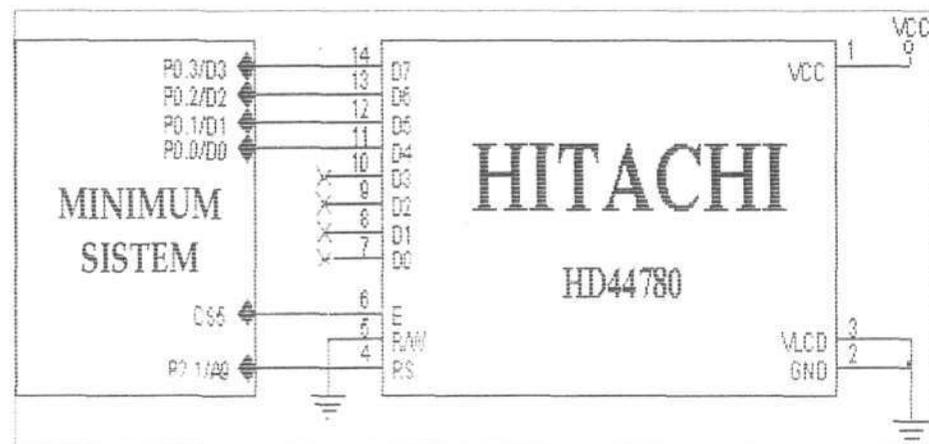
LCD yang digunakan adalah LCD 2 baris dengan 16 karakter per baris. Sistem *data bus* yang digunakan adalah *data 4 bit*. Untuk menghemat I/O, maka operasi LCD dibatasi hanya untuk menulis karakter dan *flag* LCD tidak dilakukan, tetapi diganti dengan prosedur *delay* yang dilakukan dimaksudkan agar LCD tidak sedang *busy* ketika ingin dilakukan penulisan berikutnya, sehingga dapat dihindari terjadinya kehilangan data. LCD memiliki 4 macam *mode* operasi yaitu *mode* penulisan data, serta pembacaan *busy flag* dan data. Untuk memilih *mode* operasi yang dilakukan, ada dua *pin control* yang harus diatur. Untuk lebih jelasnya dapat dilihat di tabel 2.2 berikut :

<sup>10</sup> Ibid. hal 5

Tabel 2.2

Mode – mode Operasi LCD

RS	R/W	Mode Operasi
0	0	Tulis Instruksi Operasi internal : Display center
0	1	Baca Busy Flag dan address counter ( DB0-DB6 )
1	0	Tulis Data ( DR ke DD RAM atau CG RAM )
1	1	Baca Data ( DD RAM atau CD RAM ke DR )



Gambar 2.13

Interfacing LCD ke Minimum Sistem

## 6. KEYBOARD IBM PC

Untuk saat ini *keyboard* PC merupakan hal mutlak yang harus ada pada PC, selain untuk PC juga banyak digunakan untuk keperluan – keperluan

yang lain. Karena *keyboard* ini dapat digunakan sebagai sarana input yang murah, dengan sistem pengkabelan yang sedikit dan berbasis mikrokontroler dengan mudah bisa dilengkapi dengan banyak tombol untuk mengisikan karakter berupa *text* maupun angka.

### 6.1 Cara kerja keyboard PC

Pada waktu tombol *keyboard* ditekan atau dilepas, maka *keyboard* akan mengirimkan kode ke PC atau mikrokontroler. Kode yang dikirimkan oleh *keyboard* tersebut dinamakan sebagai *scan code*.

*Scan code* tombol 'A' adalah  $1C_{11}$  dalam heksadesimal angka ini sama dengan angka biner yaitu  $00011011_B$ . ketika tombol 'A' ditekan maka *keyboard* akan mengirimkan kode  $1C_H$  secara terus menerus sampai tombol 'A' dilepaskan. Pada saat tombol dilepas *keyboard* juga mengirimkan kode  $F0_{11}$ , jadi pada saat pelepasan tombol 'A' *keyboard* mengirimkan kode  $F0_H$  dan  $1C_H$ .

Pengiriman kode – kode tersebut oleh *keyboard* dikirim secara seri, yang mana artinya dikirimkan satu demi satu / per bit. Misalkan kode yang dikirim adalah  $1C_H$  ( $00011100_B$ ), pertama dikirim '0' dan sesaat kemudian diteruskan '0' selanjutnya '1', sampai selesai 8 *bit* terkirim semua '00011100'. (pengiriman dimulai dari *bit* LSB kemudian bergeser satu per satu sampai *bit* MSB).

Masing-masing tombol pada *keyboard* mempunyai *scan code* sendiri, pengenalan *scan code* tersebut beserta implementasinya

tergantung dari PC atau mikrokontroler itu sendiri. Gambar di bawah menunjukkan *scan code* masing masing tombol *keyboard* PC. Pada gambar dapat dilihat scan code tidak berupa kode ASCII yang biasa dipakai untuk mewakili huruf, dan ditentukan secara acak. Sehingga *kode* yang telah diterima (oleh PC atau mikrokontroler) harus dirubah terlebih dahulu menjadi kode ASCII dengan cara pencarian tabel.

Esc 76	F1 05	F2 06	F3 04	F4 0C	F5 03	F6 0B	F7 83	F8 0A	F9 01	F10 09	F11 7B	F12 07	PrnSc *1	ScrL 7E	Brk *2	*1 is E012E07C *2 is E11477E1F014F077						
~ 0E	1 16	2 1E	3 26	4 25	5 2E	6 36	7 3D	8 3E	9 46	0 45	- 4E	= 55	← 66	INS E070	Home E06C	PgUp E07D	NumL 77	/ E04A	* 7C	- 7B		
TAB 0D	Q 15	W 1D	E 24	R 2D	T 2C	Y 35	U 3C	I 43	O 44	P 4D	 54	 5B	\ 5D	DEL E071	END E069	PgDn E07A	7 6C	8 75	9 7D	+ 79		
Caps 58	A 1C	S 1B	D 23	F 2B	G 34	H 33	J 3B	K 42	L 4B	; 4C	' 52	← 5A					4 6B	5 73	6 74			
Shift 12	Z 1A	X 22	C 21	V 2A	B 32	N 31	M 34	. 41	, 49	/ 4A		Shift 59	↑ E075				1 69	2 72	3 7A	↔ 7A		
Ctrl 14	Alt 11	Space 29						Alt E011	Ctrl E014	← E06B	↓ E072	→ E074	0 70	.71	E05A							

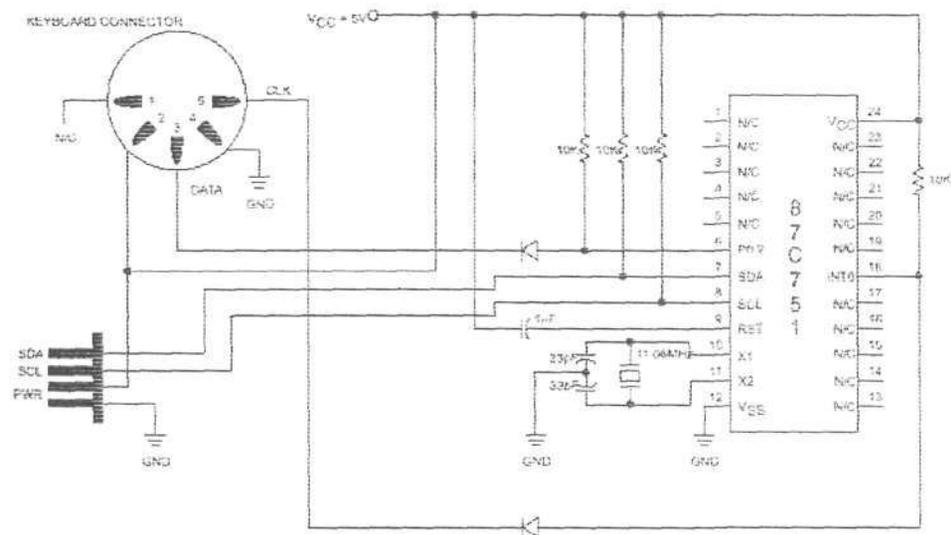
Gambar 2.14

## Keyboard PC dan Scan Code

## 6.2 Rangkaian penghubung

*Keyboard* PC ini dibangun dengan mikrokontroler MCS48, untuk keperluan membentuk rangkaian penghubung tidak perlu diketahui

bagaimana cara kerja mikrokontroler dalam *keyboard*, tetapi cukup meninjau rangkaian elektronik bagian penghubung.



Gambar 2.15

Bagian penghubung di dalam Keyboard PC

Pada rangkaian ini bisa dipakai untuk komunikasi dua arah, yakni *keyboard* dapat mengirimkan kode-kode *scan* ke PC, begitu pula PC dapat mengirimkan perintah-perintah ke *keyboard*, misalnya menyalakan lampu-lampu yang berada di *keyboard*.