

2. TEORI PENUNJANG

2.1. Teori Musik

Dalam sebuah lagu terdapat beberapa komponen penting yang sangat berpengaruh. Diantaranya adalah *Tempo*, *Birama*, Hitungan/ketukan, serta Dinamika. *Tempo* adalah cepat atau lambatnya nyanyian dalam sebuah lagu. *Birama* merupakan jumlah hitungan atau ketukan dalam satu bar. Misalnya saja *birama* 2/4 menunjukkan bahwa dalam satu bar terdapat dua hitungan not seperempat. *Birama* 3/4 menunjukkan bahwa dalam satu bar terdapat tiga hitungan not seperempat. Demikian juga dengan *birama* 4/4 menunjukkan bahwa dalam satu *birama* terdapat empat hitungan not seperempat. Namun lain halnya dengan *birama* 6/4 dan *birama* 6/8. Dalam *birama* 6/4 terdapat tiga ketuk not setengah, sedangkan dalam *birama* 6/8 terdapat 6 ketuk not seperdelapan. *Birama* 6/4 menunjukkan bahwa dalam satu *birama* terdapat 3 ketuk

Birama merupakan patokan hitungan yang dipakai sebagai penentu ketukan dalam sebuah lagu. *Birama* juga dapat digunakan untuk menentukan *style* atau irama dalam sebuah *keyboard*. Misalnya, *birama* 2/4 berarti menggunakan irama *march*, sedangkan *birama* 3/4 berarti menggunakan irama *waltz*, dan *birama* 4/4 berarti menggunakan *style beat*.

Bar adalah satuan yang digunakan untuk menghitung banyaknya *ketuk* atau *hitungan* dalam suatu lagu. Sebagai contoh: dalam lagu Indonesia Raya *birama* 4/4 terdapat 128 ketuk/hitungan dari awal sampai akhir lagu. Hal ini menunjukkan bahwa lagu Indonesia Raya jika dinyanyikan dari permulaan sampai akhir berjumlah 32 *bar*.

'*tempo*' adalah cepat atau lambatnya seseorang dalam menyanyikan sebuah lagu. Misalnya di-*set tempo* 60 pada sebuah *metronome*, hal ini berarti dalam waktu 1 menit *metronome* akan menghasilkan 60 *ketukan* (60 kali bunyi tik-tok), dimana setiap ketuknya membutuhkan waktu selama 1 detik.

'*a tempo*' merupakan istilah yang sering dipakai dalam bermusik. Definisi *a tempo* adalah kembali ke *tempo* awal (*tempo* semula) atau ke *tempo* yang sebelumnya. Sebagai contoh, jika dalam sebuah lagu terdapat beberapa macam

variasi *tempo* yang digunakan, maka biasanya pada akhir lagu dijumpai istilah *a tempo*. Hal ini berarti, jika hendak kembali ke awal bait atau permulaan lagu, maka diwajibkan untuk kembali ke *tempo* semula.

Istilah *fermata* juga sering digunakan dalam dunia musik. Definisi *fermata* adalah berhenti selama beberapa saat (*pause* selama beberapa waktu). Atau dengan kata lain, *fermata* bisa disamakan dengan *delay*. Jika terdapat tanda *fermata* dalam sebuah not, maka not tersebut akan mengalami *delay* selama 1 *bar*. Misalnya dalam lagu tersebut menggunakan *birama* 4/4, maka not tersebut mengalami penundaan sesaat (*delay*) selama 4 hitungan. Demikian pula jika lagu tersebut menggunakan *birama* 3/4, maka not tersebut mengalami *delay* selama 3 hitungan.

2.2. Persamaan Lambang Notasi Musik

Untuk menulis bunyi dan tanda diam dengan bermacam-macam lama waktu atau panjang pendeknya bunyi dan tanda diam digunakan notasi irama dengan bentuk dan nilai tertentu:

- Not penuh nilainya 1 atau 2/2 atau 4/4
- Not tengahan nilainya 1/2 atau 2/4
- Not perempat nilainya 1/4 atau 2/8
- Not perdelapan nilainya 1/8 atau 2/16
- Not perenambelas nilainya 1/16
- Tanda diam penuh nilainya 1 atau 2/2 atau 4/4
- Tanda diam tengahan nilainya 1/2 atau 2/4
- Tanda diam perempat nilainya 1/4 atau 2/8
- Tanda diam perdelapan nilainya 1/8 atau 2/16
- Tanda diam perenambelas nilainya 1/16

Titik di belakang not atau tanda diam menambahkan nilai not atau tanda diam itu dengan setengah dari nilainya:

- $4/4 + 2/4 = 6/4$
- $2/4 + 1/4 = 3/4$
- $1/4 + 1/8 = 3/8$
- $1/8 + 1/16 = 3/16$

- $1/6 + 2/4 = 6/4$
- $2/4 + 1/4 = 3/4$
- $1/4 + 1/8 = 3/4$
- $1/8 + 1/16 = 3/16$

Tiap not dapat bernilai dengan perbandingan 3 : 1, jika diberi tanda trial



Gambar 2.1. Not Legatura

Tanda legatura menghubungkan dua buah not atau lebih () memperpanjang nilai not yang pertama menjadi jumlah nilai not-not yang dihubungkannya.

2.3. Teknologi Mikrokontroler AT89S8252

Mikrokontroler Atmel AT89S8252 merupakan pengembangan dari mikrokontroler standar MCS-51. Hal-hal yang terdapat pada mikrokontroler MCS-51 juga berlaku untuk mikrokontroler Atmel AT89S8252, mikrokontroler ini memiliki kelebihan yang tidak terdapat pada mikrokontroler pendahulunya yaitu MCS-51, tetapi tidak demikian sebaliknya.

2.3.1. Fitur Mikrokontroler Atmel AT89S8252

Mikrokontroler ini memiliki keunggulan sebagai berikut:

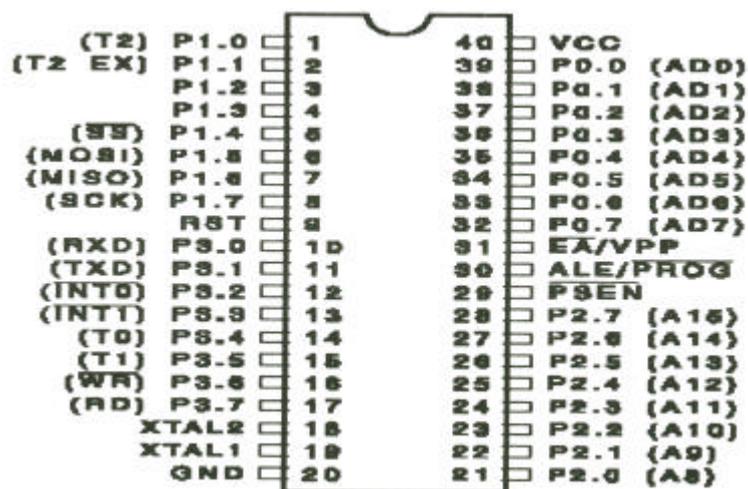
- Kompatibel dengan mikrokontroler MCS-51
- Memiliki 8Kbyte *Downloadable Flash Memory*
- 2Kbyte EEPROM
- 3 Level *program memory lock*
- 256 byte RAM internal
- 32 *Pin Input/Output* yang dapat dipakai semua
- 3 buah *Timer/Counter* 16 bit
- *Programmable* UART (*serial port*)
- *SPI Serial Interface*
- *Programmable Watchdog Timer*
- *Dual Data Pointer*

- Frekuensi kerja 0 sampai 24 MHz
- Tegangan operasi 2,7 Volt sampai 6 Volt

Di sini terlihat bahwa mikrokontroler Atmel AT89S8252 memiliki banyak fitur yang menguntungkan. Dengan adanya *Downloader flash memory* memungkinkan mikrokontroler ini bekerja sendiri tanpa diperlukan tambahan *chip* lainnya. Sementara *flash memory*-nya mampu diprogram hingga seribu kali. Hal lain yang menguntungkan dari mikrokontroler ini adalah *system* pemrograman menjadi lebih sederhana dan tidak memerlukan rangkaian yang rumit seperti rangkaian untuk memprogram MCS-51.

Timer atau *Counter* juga bertambah satu dari standar 2 buah pada MCS-51. Selain itu frekuensi kerja yang lebar dan rancangan static sangat membantu untuk proses *debugging*. Dengan adanya beberapa fitur tambahan itu, maka akan mengakibatkan bertambahnya SFR (*Special Function Register*).

Gambar berikut adalah gambar mikrokontroler Atmel AT89S8252. Tata letak *Pin-Pin* yang ada masih mengacu pada mikrokontroler MCS-51 sehingga AT89S8252 dapat menggantikan mikrokontroler MCS-51. Beberapa istilah yang asing adalah MOSI, MISO dan SCK.



Gambar 2.2. Mikrokontroler Atmel AT89S8252

Sumber: *AT89 Series Hardware Description*, Texas: Atmel Inc, 1997.

2.3.2. Fungsi-fungsi kaki (*PIN*)

- a. **Vcc** → sebagai *supply* tegangan
- b. **GND** → sebagai *Grounding*
- c. **ALE/PROG** → Keluaran **ALE** atau *Address Latch Enable* menghasilkan pulsa-pulsa untuk mengunci *bit* rendah (*low byte*) alamat selama pengaksesan *memory* eksternal. Kaki ini juga berfungsi sebagai masukan pulsa program (*the program pulse input*) atau selama pengisian *Flash* PEROM. Pada operasi normal, **ALE** akan berpulsa dengan laju 1/6 dari frekuensi kristal dan dapat digunakan sebagai pewaktuan (*timing*) atau pendetakan (*clocking*) rangkaian eksternal.

Jika dikehendaki, operasi **ALE** bisa dimatikan dengan cara mengatur *bit* 0 dari SFR lokasi 08E_H. Jika isinya '1', **ALE** hanya akan aktif selama dijumpai instruksi MOVX atau MOVC. Selain itu, kaki ini akan secara perlahan akan di-*pulled high*. Mematikan *bit* **ALE** tidak akan ada efeknya jika mikrokontroler mengeksekusi program secara eksternal.

Program Store Enable merupakan sinyal baca untuk *memory* program eksternal. Saat mikrokontroler menjalankan program dari *memory* eksternal akan diaktifkan dua kali per siklus mesin, kecuali dua aktivasi dilompati (diabaikan) saat mengakses *memory* data eksternal.

- d. **Vpp** → atau *External Access Enable* harus selalu dihubungkan ke *ground*, jika mikrokontroler akan mengeksekusi program dari *memory* eksternal lokasi 0000_H hingga FFFF_H. Selain dari itu, harus dihubungkan ke Vcc agar mikrokontroler mengakses program secara internal. Kaki ini juga berfungsi menerima tegangan 12 Volt (Vpp) selama pengisian *flash*, khususnya untuk tipe mikrokontroler 12 Volt Vpp.

2.3.3. Struktur *Port* dan Cara Kerja

Pada dasarnya mikrokontroler Atmel keluarga 51 mempunyai dua kelompok instruksi untuk mengeluarkan data ke *port parallel* (*keterangan* : tanda 'x' artinya sama seperti kondisi sebelumnya):

- Kelompok instruksi pertama bekerja pada *Port* seutuhnya, artinya 8 jalur dari *Port* bersangkutan, misalnya **MOV P3, #FFH** membuat ke-delapan jalur *Port* semuanya dalam kondisi logika '1' (atau isinya (1111 1111 dalam biner);
- Kelompok instruksi kedua hanya berpengaruh pada salah satu jalur atau *bit* dari *Port*, misalnya instruksi **SETB P3.4** artinya men-*set bit-4* dari *Port* 3 (*bit-4* dari *Port* 3 = 1 → **xxx1 xxxx**) atau instruksi **CLR P3.3** digunakan untuk me-nolkan *bit-3* dari *Port* 3 (*bit-3* dari *Port* 3 = 0 → **xxxx 0xxx**).

Selain itu *Port parallel* bisa pula dipakai untuk menerima masukan sinyal digital dari luar mikrokontroler:

- Instruksi **MOV A, P3** digunakan untuk membaca data (digital) pada seluruh *bit* (*bit* 0 hingga 7 = 8-*bit*) *Port* 1 kemudian menyimpannya dalam *accumulator*;
- Pembacaan data bisa juga dilakukan hanya pada satu *bit Port* saja, misalnya instruksi **JNB P3.7, \$** digunakan untuk memantau *bit P3.7*, jika **P3.7=0**, mikrokontroler akan kembali melaksanakan instruksi tersebut (lompat ke label '\$' artinya ke lokasi tersebut lagi), mikrokontroler akan meneruskan kembali instruksi berikutnya jika **P3.7=1** (*catatan* : instruksi yang berlawanan dengan **JNB** adalah **JB**).

Satu hal penting yang perlu diperhatikan adalah agar data dari luar, (bisa dibaca dengan benar) sebelumnya, jalur *Port* yang bersangkutan harus di inialisasi terlebih dahulu (dengan cara mengisi dengan logika '1'). Terutama jika satu *port* berfungsi ganda. (sebagai masukan dan keluaran sekaligus), kecuali jika sejak awal dan seterusnya *Port* yang bersangkutan hanya digunakan sebagai masukan saja.

Keempat *Port* pada Atmel keluarga 51 tersebut bersifat dua arah dan masing-masing memiliki sebuah pengancing (*latch*) yang terdapat dalam program sebagai *Register Fungsi Khusus* (**RFK** atau **SFR**) sebagai **P0, P1, P2** dan **P3**. Selain itu juga memiliki sebuah penggerak keluaran (*output driver*) dan sebuah penyangga masukan (*input buffer*) pada masing-masing kaki *Port*.

Penggerak-penggerak keluaran *Port* 0 dan *Port* 2 serta penyangga masukan dari *Port* 0 digunakan dalam pengaksesan *memory* eksternal. Pada aplikasi semacam ini, *Port* 0 mengeluarkan *byte* rendah alamat *memory* eksternal,

di-*multiplex* secara waktu dengan *byte* yang akan dituliskan atau dibaca (dari dan ke *memory* eksternal). *Port 2* mengeluarkan *byte* tinggi dari alamat *memory* eksternal jika (lebar alamatnya 16-bit). Selain itu, kaki-kaki *Port 2* tetap meneruskan menghasilkan isi **SFR** dan **P2**.

Semua kaki-kaki *Port 3* dan dua kaki *Port 1* (pada seri Atmel keluarga 51 yang bertipe 52 memiliki 3 *Timer*) memiliki beragam fungsi. Kaki-kaki *Port* tersebut tidak hanya sekedar kaki-kaki *Port parallel*, namun juga menyediakan beberapa fungsi khusus seperti yang ditunjukkan dalam tabel berikut ini :

Tabel 2.1. Fungsi-fungsi Khusus Kaki-kaki *Port 3*

Kaki Port	Fungsi Alternatif	Keterangan
P3.0 (1)	T2	Masukan eksternal <i>Timer 2</i>
P3.1 (1)	T2EX	Pemicu <i>Capture/Reload</i> waktu <i>Timer 2</i>
P3.0	RXD	Saluran masukan serial
P3.1	TXD	Saluran keluaran serial
P3.2	INT0	Interupsi eksternal 0
P3.3	INT1	Interupsi eksternal 1
P3.4	T0	Masukan eksternal <i>Timer 0</i>
P3.5	T1	Masukan eksternal <i>Timer 1</i>
P3.6	WR	Sinyal tanda baca <i>memory</i> data eksternal
P3.7	RD	Sinyal tanda tulis <i>memory</i> data eksternal

Semua *Pin* pada mikrokontroler Atmel AT89S8252 adalah sama dengan MCS-51, namun ada beberapa perbedaan khusus, yaitu AT89S8252 memiliki fungsi-fungsi khusus pada *port 1*, antara lain:

Port Pin	Fungsi Khusus
P1.0	T2 (external input untuk <i>Timer/ Counter 2</i>)
P1.1	T2 EX (<i>Timer/Counter 2 capture/reload trigger</i> dan control arah)
P1.2	-
P1.3	-

- P1.4 SS (*Slave port select input*)
 P1.5 MOSI (*Master Data output, Slave data input* untuk port SPI)
 P1.6 MISO (*Master Data input, Slave data output* untuk port SPI)
 P1.7 SCK (*Master clock output, Slave clock input* untuk port SPI)

2.3.4. SFR tambahan pada Atmel AT89S8252

Selain memiliki SFR (*Special Function Register*) seperti halnya pada MCS-51, mikrokontroler AT89S8252 memiliki tambahan SFR. Hal ini tidak lain karena terdapatnya fitur tambahan pada mikrokontroler ini. Jadi SFR tambahan ini ada SFR untuk mengontrol alat tambahan pada mikrokontroler AT89S8252.

SFR tambahan ini meliputi:

- T2CON (*Timer 2 Register* dengan alamat 0C8_H)
- T2MOD (*Timer 2 Mode* dengan alamat 0C9_H)
- WMCON (*Watchdog and Memory Control Register* dengan alamat 096_H)
- SPCR (*SPI Control Register* dengan alamat 0D5_H)
- SPSR (*SPI Status Register* dengan alamat 0AA_H)
- SPDR (*SPI Data Register* dengan alamat 086_H)

2.3.5. SFR untuk *Timer 2*

Pada mikrokontroler AT89S8252 terdapat tambahan sebuah *Timer/Counter* yang diberi nama *Timer 2*, sehingga AT89S8252 memiliki 3 *Timer/Counter* yaitu *Timer/Counter 0*, *Timer/Counter 1*, *Timer/Counter 2*.

Pada *Timer/Counter 2* ini dikendalikan oleh *Special Function Register* yang bernama T2CON (*Timer 2 Control*), T2MOD (*Timer 2 Mode*) dan sepasang *register* RCAP2H, RCAP2L merupakan *register capture/reload* untuk *Timer 2* dalam 16 *bit capture mode* atau *reload mode*.

Register T2CON yang beralamat di 0C8_H memiliki *bit-bit* sebagai berikut:

MSB							LSB
TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2	CP/RL2

Bit Keterangan

TF2 *Overflow flag Timer 2*, di-set oleh *Timer 2* dan harus *diclear* lewat *Software*. TF2 tidak akan di-set jika RCLK = 1 atau TCLK = 1.

EXF2 *External Flag Timer 2* di-set saat suatu *capture* atau reload disebabkan oleh transisi *negative* pada T2EX dan EXEN2 = 1. Jika sela *Timer 2* diaktifkan, EXF2 = 1 akan menyebabkan CPU mencabang ke rutin sela *Timer 2*. EXF2 harus di-clear dengan *Software*. EXF2 tidak menyebabkan sela pada mode up/down *Counter* (DCEN = 1).

RCLK *Receive clock enable*. Jika di-set menyebabkan *serial port* menggunakan pulsa *overflow Timer 2* sebagai detak penerimaan pada *serial port* untuk mode 1 dan 3. Jika RCLK = 0 menyebabkan pulsa *overflow Timer 1* yang digunakan sebagai detak.

TCLK *Transmit clock enable*. Jika di-set menyebabkan *serial port* menggunakan pulsa *overflow Timer 2* sebagai detak pengiriman. Jika TCLK = 0 menyebabkan pulsa *overflow Timer 1* yang digunakan sebagai detak pengiriman.

EXEN2 *Timer 2 eksternal enable*. Jika di-set memungkinkan *capture* atau reload terjadi sebagai hasil dari transisi *negative* pada *Pin T2EX* jika *Timer 2* tidak sedang digunakan sebagai *baud rate generator* untuk *serial port*. Jika EXEN2 = 0 menyebabkan *Timer 2* akan melakukan suatu hal pada *Pin T2EX*.

TR2 *Bit* untuk mengatur *start/stop* untuk *Timer 2*. Jika TR2 = 1, maka *Timer* akan aktif

C/T2 *Bit* pemilihan *Timer* atau *Counter* untuk *Timer 2*. Jika C/T2 = 0, maka terpilih fungsi *Timer*. C/T2 = 1 berfungsi sebagai *counter*.

CP/RL2 *Pemilihan Capture/Reload*. Jika di-set maka proses *capture* akan terjadi pada transisi *negative* pada *Pin T2EX* (jika *bit* EXEN2 = 1). Jika *bit* ini di-clear maka proses *reload* otomatis akan terjadi saat *Timer 2 overflow* atau transisi *negative* terjadi pada *Pin T2EX* (saat *bit* EXEN2 = 1). Jika *bit* RCLK atau TCLK di-set maka *bit* ini menjadi tidak diperhitungkan atau diabaikan. Hal ini karena *Timer 2* dipakai sebagai *baud rate generator* pada *serial port*. SFR ini memiliki nilai pada saat *Reset*: 0000 0000_B

Timer 2 juga memiliki SFR yang bernama T2MOD (*Timer 2 Mode Control Register*) yang beralamat di 0C9_H dan memiliki nilai pada saat *Reset* XXXX XX00_B. Bit-bit pada T2MOD adalah sebagai berikut.

MSB							LSB
-	-	-	-	-	-	T2OE	DCEN

Bit	Keterangan
T2OE	<i>Timer 2 Output Enable bit</i>
DCEN	Jika di- <i>set</i> memungkinkan <i>Timer/Counter</i> sebagai <i>up/down counter</i>

2.3.6. SFR untuk Watchdog dan Memory

Untuk menggunakan *Watchdog Timer* atau *memory*, maka dapat dilakukan dengan mengatur SFR yang bernama WMCON dengan alamat 096_H. Bit-bit pada SFR ini dapat dijelaskan sebagai berikut.

Nilai SFR ini pada saat *Reset* adalah 0000 0000_B.

MSB							LSB
PS2	PS1	PS0	EEMWE	EEMEN	DPS	WDTRST	WDTEN

Bit	Keterangan
PS2	-
PS1	-
PS0	Ketiga <i>bit</i> ini merupakan <i>bit prescaler</i> untuk <i>Watchdog Timer</i> . Jika Ketiga <i>bit</i> di- <i>clear</i> , maka periode <i>Watchdog Timer</i> adalah 16ms. Jika Ketiga <i>bit</i> di atas di- <i>set</i> , maka nominal perioda waktu <i>Watchdog Timer</i> adalah 2048 ms (<i>millisecond</i>).
EEMWE	<i>Bit</i> pengaktif penulisan EEPROM Data Memory. <i>Bit</i> ini harus di- <i>set</i> sebelum penulisan ke EEPROM dengan instruksi <i>movx</i> . Setelah selesai penulisan, maka <i>bit</i> ini harus di- <i>clear</i> .
EEMEN	<i>Bit</i> pengaktif pengaksesan internal EEPROM. Saat EEMEN = 1 instruksi <i>movx</i> dengan DPTR akan mengakses internal EEPROM bukan pada data <i>memory</i> luar. Jika EEMEN = 0, instruksi <i>movx</i> dengan DPTR akan mengakses data <i>memory</i> luar.

- DPS *Data Pointer Register Select*. Jika *bit* ini di-clear, akan memilih *bank* pertama dari *Data Pointer Register* (DPO). Jika *bit* ini di-set, akan terpilih *bank* kedua (DP1).
- WDTRST *Watchdog Timer Reset* dan *bit* bendera EEPROM *ready/busy*. Tiap saat *bit* ini di-set ke 1 oleh *Software* pengguna, suatu pulsa akan dihasilkan untuk me-reset *Watchdog Timer*. *Bit* ini kemudian secara otomatis akan *diclear*. *Bit* ini bersifat hanya dapat ditulisi. *Bit* ini juga sebagai *bit* bendera *ready/busy* pada *mode Read Only* selama penulisan EEPROM. RDY/BSY = 1 berarti bahwa EEPROM siap untuk diprogram. Selama operasi pemrograman berlangsung, *bit* ini akan berlogika '0' dan secara otomatis akan di-Reset ke '1' saat pemrograman selesai.
- WDTEN *Bit* pengaktif *Watchdog Timer*. Jika WDTEN = 1 akan mengaktifkan *Watchdog Timer*. Jika WDTEN = 0 maka akan menjadi tidak aktif.

2.3.7. SFR pengontrol SPI

Berbeda dengan mikrokontroler MCS-51, mikrokontroler AT89S8252 memiliki fasilitas SPI (*Serial Peripheral Interface*). Fasilitas ini memungkinkan transfer data kecepatan tinggi secara sinkron antara mikrokontroler AT89S8252 dengan peripheral atau antar mikrokontroler AT89S8252

Fitur SPI pada Atmel AT89S8252 meliputi :

- a. *Full Duplex*, 3 kawat dengan transfer data secara sinkron.
- b. Operasi *Master* atau *Slave*.
- c. Frekuensi maksimum 6MHz.
- d. Sistem Data Transfer MSB dahulu atau LSB dahulu.
- e. 4 *bit rate* terprogram.
- f. *Pause Flag* pada akhir transmisi.
- g. *Write Collision Flag Protection*.

2.3.8. Data Memory (EEPROM) dan RAM

Mikrokontroler AT89S8252 juga dilengkapi dengan data *memory* yang berupa EEPROM (*Electrically Erasable Programmable Read Only Memory*).

EEPROM yang ditanamkan ini besarnya 2 kilobyte (2K) dan dipakai untuk penyimpanan data.

EEPROM on-chip ini diakses dengan men-*set bit* EEMEN pada *register* WMCON pada alamat 096_H. Alamat *EEPROM* ini adalah 000_H sampai 7FF_H. Instruksi *movx* digunakan untuk mengakses *EEPROM* internal ini. Namun jika ingin mengakses data *memory* luar (kecuali mikrokontroler AT89S8252) dengan menggunakan instruksi *movx* ini, maka *bit* EEMEN harus dibuat '0'.

Bit EEMWE pada *register* WMCON harus di-*set* ke 1 sebelum semua lokasi pada *EEPROM* dapat ditulisi. Program pengguna harus me-*reset bit* EEMWE ke '0', jika proses penulisan ke *EEPROM* tidak diperlukan lagi. Proses penulisan ke *EEPROM* dapat dilihat dengan membaca *bit* RDY/BSY pada SFR WMCON. Jika *bit* ini berlogika rendah, maka berarti penulisan *EEPROM* sedang berlangsung. Jika *bit* ini berlogika tinggi (*high*) maka berarti penulisan sudah selesai dan penulisan lain dapat dimulai lagi.

Mikrokontroler AT89S8252 juga memiliki RAM yang berkapasitas 256 byte.

2.3.9. Programmable *Watchdog Timer*

Pada mikrokontroler AT89S8252 ini juga dilengkapi dengan *Watchdog Timer*. *Watchdog Timer* ini menggunakan detak tersendiri. Untuk mengatur rentang waktu (periode) pada WDT ini, maka terdapat *bit prescaler* yang dapat mengatur rentang waktu yang dibutuhkan.

Bit prescaler ini adalah *bit* PS0, PS1 dan PS2 pada *register* WMCON. Periode waktu pada WDT ini berkisar dari 16 mili detik sampai 2048 mili detik. Karena *bit prescaler*-nya ada tiga, maka akan ada 8 buah kemungkinan seperti yang tertera pada tabel di bawah ini.

Tabel 2.2 Pemilihan periode waktu *Watchdog Timer*

PS1	PS0	Periode	
0	0	0	16 mili detik
0	0	1	32 mili detik
0	1	0	64 mili detik
0	1	1	128 mili detik
1	0	0	256 mili detik
1	0	1	512 mili detik
1	1	0	1024 mili detik
1	1	1	2048 mili detik

Watchdog Timer (WDT) dapat dinon-aktifkan melalui *Power On Reset(POR)* dan selama *Power Down*. WDT diaktifkan dengan men-setting bit WDTEN pada SFR WMCON (alamat 096_H). Jika perhitungan waktu WDT telah selesai (*time out*) tanpa ada *Reset* atau dinon-aktifkan, maka suatu pulsa *Reset* internal akan dihasilkan untuk me-reset CPU.

2.3.10. *Timer 2*

Pada mikrokontroler AT89S8252 terdapat tambahan *Timer 2*. *Timer* yang lain adalah *Timer 0* dan *Timer 1* seperti yang terdapat pada MCS-51. *Timer/Counter* dapat digunakan sebagai *generator baud rate* untuk *serial port*. Pada standard MCS-51 biasanya digunakan *Timer 1* sebagai penghasil *baud rate* (untuk menjaga kompatibilitas dengan MCS-51), selain itu juga dapat menggunakan *Timer 2* sebagai penghasil *baud rate* untuk *serial port*.

Timer 2 ini merupakan *Timer/Counter* berukuran 16 *bit* yang dapat beroperasi sebagai *Timer* (dengan detak dari system detak mikrokontroler) atau dapat beroperasi sebagai penghitung kejadian (*event counter*) dengan detak dari luar. Untuk mengatur fungsi ini dilakukan dengan mengatur *bit C/T2* pada SFR T2CON. Terlihat bahwa jika *bit* ini berlogika 'high', maka akan terpilih fungsi *Counter* (C). Tetapi jika *bit* ini berlogika 'low', maka akan terpilih fungsi *Timer 2* (T2).

Timer 2 ini memiliki 3 mode operasi yaitu *capture*, *auto reload* (*up* dan *down counting*) dan *baud rate* generator. Untuk memilih mode ini dilakukan dengan mengatur *bit* pada SFR T2CON (*Timer 2 Control Register*).

Timer 2 ini terdiri dari 2 buah *Timer 8 bit register* yaitu TH2 dan TL2. Pada fungsi *Timer*, register TL2 dinaikkan (*increment*) tiap siklus mikro. Karena siklus mikro terdiri dari 12 periode osilasi, maka *count rate* menjadi 1/12 dari frekuensi *oscillator*.

Pada fungsi *counter*, register dinaikkan berdasarkan tanggapan adanya transisi tinggi ke rendah pada *pin* yang bersesuaian (dalam hal ini *Pin T2* atau P1.0). Pada fungsi ini, masukan luar akan di-sampling selama S5P2 dari setiap siklus mikro. Jika hasil sampling menunjukkan logika 'high' selama satu siklus dan 'low' pada siklus selanjutnya, maka akan terdeteksi transisi 'high' ke 'low' dan akibatnya penghitungan akan dinaikkan. Nilai penghitungan yang baru akan

muncul pada *register* selama S3P1 dari siklus Setelah transisi ‘high’ ke ‘low’ terdeteksi.

Tabel berikut menunjukkan mode operasi yang dapat dijalankan pada *Timer 2*.

Tabel 2.3. Mode Operasi *Timer 2*

RCLK+TCLK	CP/RL2	TR2	Mode
0	0	1	16 bit auto reload
0	1	1	16 bit capture
1	X	1	baud rate generator
X	X	0	off

2.3.11. Mode Capture

Pada mode ini ada dua pilihan dipilih oleh *bit* EXEN2 pada SFR T2CON. Jika EXEN2 = 0, *Timer 2* merupakan 16 bit *Timer* atau *Counter* yang jika telah *overflow* akan men-*set bit* TF2 pada T2CON.

Bit ini kemudian dapat digunakan untuk menghasilkan *delay* atau waktu tunggu. Jika EXEN2 =1, *Timer 2* akan berlaku sama, tetapi suatu transisi ‘high’ ke ‘low’ (‘1’ ke ‘0’) pada *Pin* T2EX (P1.1) akan menyebabkan nilai sekarang pada TH2 dan TL2 untuk disimpan ke RCAP2H dan RCAP2L. Sebagai tambahan transisi ‘high’ ke ‘low’ pada T2EX menyebabkan *bit* EXF2 pada T2CON diset. *Bit* EXF2 sama halnya dengan *bit* TF2 yang juga dapat menghasilkan sela.

2.3.12. Auto Reload (Up atau Down Counter)

Timer 2 dapat diprogram untuk menghitung naik atau menghitung mundur jika dikonfigurasi sebagai mode 16 bit auto reload. Fitur ini dapat dimatikan dengan mengatur *bit* DCEN (Down Counter Enable) pada SFR T2MOD. Pada saat *Reset* DCEN akan berlogika ‘low’, maka *Timer 2* akan memiliki default untuk menghitung maju (Up Counter). Jika *bit* DCEN di-*set*, *Timer 2* dapat menghitung maju atau mundur tergantung pada nilai logika pada *Pin* T2EX.

Pada saat DCEN di-*set* ke ‘1’ dan di-*clear* ke ‘0’, maka *Timer 2* secara otomatis menghitung naik/maju. Pada mode ini, 2 pilihan dipilih oleh *bit* EXEN2 pada SFR T2CON. Jika EXEN2 = 0, *Timer 2* akan naik menjadi 0FFFFH dan kemudian akan men-*set bit* TF2, jika mengalami *overflow*. *Overflow* juga

menyebabkan terjadinya *register Timer* diisi kembali dengan nilai 16 *bit* dari RCAP2H dan RCAP2L.

Jika EXEN2 = 1 maka 16 *bit* reload dapat diaktifkan (di-*triger*) baik oleh *overflow* atau oleh transisi '*high*' ke '*low*' pada *Pin T2EX*. Transisi ini juga akan men-*set bit* EXF2. Kedua *bit* TF2 dan EXF2 dapat menimbulkan sela waktu jika diaktifkan.

Men-*set bit* DCEN akan mengaktifkan *Timer 2* untuk menghitung naik atau mundur. Pada mode ini *Pin T2EX* akan mengontrol arahnya yaitu menghitung maju atau mundur. Suatu logika '1' pada *Pin T2EX* membuat *Timer 2* menghitung maju (*count up*). *Timer* akan *overflow* pada hitungan 0FFFF_H dan men-*set bit* TF2. *Overflow* ini juga menyebabkan nilai 16 *bit* pada RCAP2H dan RCAP2L diisikan kembali (reloaded) ke *register Timer* yaitu TH2 dan TL2.

Suatu logika '0' pada *Pin T2EX* membuat *Timer 2* menghitung mundur (*count down*). *Timer* akan mengalami *underflow* saat TH2 dan TL2 sama dengan nilai yang tersimpan pada RCAP2H dan RCAP2L. *Underflow* ini akan men-*set bit* TF2 dan menyebabkan 0FFFF_H akan diisikan kembali (*reloaded*) ke *register Timer*.

2.4. Sistem Pemrograman Mikrokontroler AT89S8252

Sistem Pemrograman yang digunakan untuk memprogram mikrokontroler AT89S8252 tidak terlalu rumit. Mikrokontroler AT89S8252 menggunakan *Downloadable Flash Memory* sehingga dapat membuat alat pemrograman menjadi relatif sederhana. Bahkan tidak diperlukan komponen apapun untuk memprogram mikrokontroler AT89S8252. Beberapa komponen dasar yang dipakai seperti resistor dan kapasitor sebenarnya hanya untuk menyediakan *Reset* pada saat catu daya mulai diberikan. *Flash Memory* yang digunakan dapat diisi sampai seribu kali.

Pada mikrokontroler AT89S8252 terdapat 8K *byte in-system reprogrammable flash PEROM Code memory* dan 2K *byte EEPROM Data Memory*. *Flash PEROM Code Memory* digunakan untuk menyimpan program yang telah dibuat. Sedangkan *EEPROM Data Memory* digunakan untuk menyimpan data. Kondisi awal saat dikirimkan dari pabrik, *memory flash PEROM* dan *EEPROM* dalam kondisi dihapus (berisi 0FF_H) dan siap untuk diprogram. Jika

diisikan sebuah program maka akan merubah kondisi $0FF_H$ ($1111\ 1111_B$) menjadi nilai-nilai tertentu sesuai dengan program yang telah diisikan.

Mikrokontroler AT89S8252 yang dibuat pada Tugas Akhir ini menggunakan *mode serial Programming* atau yang dikenal dengan *mode Low Voltage serial Programming*.

2.4.1. Sistem Pemrograman Serial (*Serial Downloading*)

Code memory dan *Data memory* dapat diprogram secara serial menggunakan bus SPI saat RST diberi logika 'high'. Hubungan *interface* SPI ini terdiri dari SCK, MOSI (*input*), dan MISO (*output*). Setelah RST di-*set* ke logika 'high', instruksi pengaktif pemrograman harus dijalankan terlebih dahulu sebelum operasi program atau hapus program dijalankan.

Alamat *Code memory* adalah 0000_H sampai $1FFF_H$ dan alamat untuk data *memory* adalah 000_H sampai $7FF_H$. Untuk mengisi program ke dalam mikrokontroler AT89S8252 diperlukan detak untuk SCK. Maksimum frekuensi Serial Clock (SCK) harus kurang dari $1/40$ dari frekuensi kristal. Jika digunakan kristal dengan frekuensi 12MHz, maka maksimum frekuensi untuk SCK adalah 300KHz.

Algoritma Pemrograman Serial ini adalah sebagai berikut :

a. Urutan *Power-Up*

Berikan catu daya antara *pin Vcc* dan *Ground* sementara *pin-pin* yang lain dibiarkan terbuka. *Set pin* RST ke logika 'high'. Jika kristal tidak dipasang pada *pin* XTAL1 dan XTAL2, maka diberikan detak 4MHz sampai 24MHz dan tunggu selama 10 milidetik sampai detak yang diberikan siap.

b. Aktifkan *Serial Programming* dengan mengirimkan instruksi pengaktif pemrograman serial pada *pin* MOSI/P1.5. Frekuensi untuk pergeseran (shift) diberikan kepada *pin* SCK/P1.7.

c. *Code memory* dan *Data memory* diprogram satu *byte* pada suatu waktu dengan memberikan alamat dan data bersamaan dengan instruksi penulisan.

d. Suatu lokasi *memory* dapat dicek (*verify*) dengan menggunakan instruksi pembacaan yang mengirimkan data pada alamat terpilih melalui *pin* MISO/P1.6. secara serial.

- e. Setelah sesi pemrograman selesai, maka RST dapat di-*set* ke logika ‘*low*’ untuk membuat operasi mikrokontroler menjadi normal.
- f. Urutan *Power-Off*
Set XTAL1 ke rendah (jika tidak ada kristal terpasang). Berikan logika ‘*low*’ pada RST, dan yang terakhir adalah matikan catu daya.

Tabel 2.4. Instruksi untuk *Serial Programming*

Instruksi	Format (MSB...LSB)	Operasi
<i>Programming Enable</i>	1010 1100 0101 0011 xxxx xxxx	Mengaktifkan serial <i>Programming RST=1</i>
Chip Erase	1010 1100 xxxx x100 xxxx xxxx	Menghapus <i>memory</i> 8K dan 2K
<i>Read Code Memory</i>	aaaa a001 <i>low address</i> xxxx xxxx	Membaca data dari <i>Code memory</i>
<i>Write Code Memory</i>	aaaa a010 <i>low address</i> data in	Menulis data ke <i>Code memory</i>
<i>Read Data Memory</i>	00aa a101 <i>Low address</i> xxxx xxxx	Membaca dari data <i>memory</i>
<i>Write Data Memory</i>	00aa a110 <i>Low address</i> Data in	Menulis data ke data <i>memory</i>
Write Lock bit	1010 1100 LLLx x111 BBB 123 xxxx xxxx	Menulis Lock bit <i>Set LB1, LB2</i> atau <i>LB3=0</i> untuk memprogram Lock bit

Sumber: *Flash Microcontroller Architectural Overview*, Texas: Atmel Inc, 1997.

Keterangan :

X = don't care (1 atau 0)

‘aaaa’ = alamat atas

2.4.2. Cara Kerja Pemrograman dari Komputer PC ke dalam AT89S8252

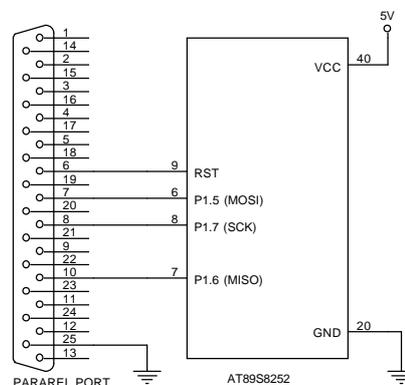
Seperti yang telah dijelaskan sebelumnya, memprogram mikrokontroler Atmel AT89S8252 dapat dilakukan dengan dua cara. Cara pertama adalah dengan menggunakan system pemrograman *parallel*. Untuk melakukan hal ini diperlukan alat pemrograman universal seperti merk Hi-Lo atau Data I/O, namun alat ini berharga sangat mahal. Jika ingin menggunakannya, maka mikrokontroler harus diletakkan ke dalam ZIF soketnya.

Apabila mikrokontroler yang telah diprogram dipasangkan ke dalam system, namun ternyata sistem tidak bekerja sesuai dengan yang diharapkan, maka program harus diperbaiki. Setelah program diperbaiki, maka mikrokontroler diambil lagi dari sistem dan dipasangkan pada ZIF soket *programmer*. Setelah selesai diprogram, mikrokontroler dipasangkan kembali ke soket sistem. Hal ini menjadi tidak praktis dan memungkinkan merusak atau mematahkan *pin-pin* yang terdapat pada IC mikrokontroler (jika tidak dilakukan dengan hati-hati).

Untuk menghindari hal di atas, maka yang terbaik adalah dengan menggunakan pemrograman secara serial (*Serial Downloading*). Dengan menggunakan cara ini, maka tidak ada lagi proses mengambil-ambil mikrokontroler dari soketnya. Mikrokontroler akan terpasang pada soket tanpa perlu dicabut-cabut lagi.

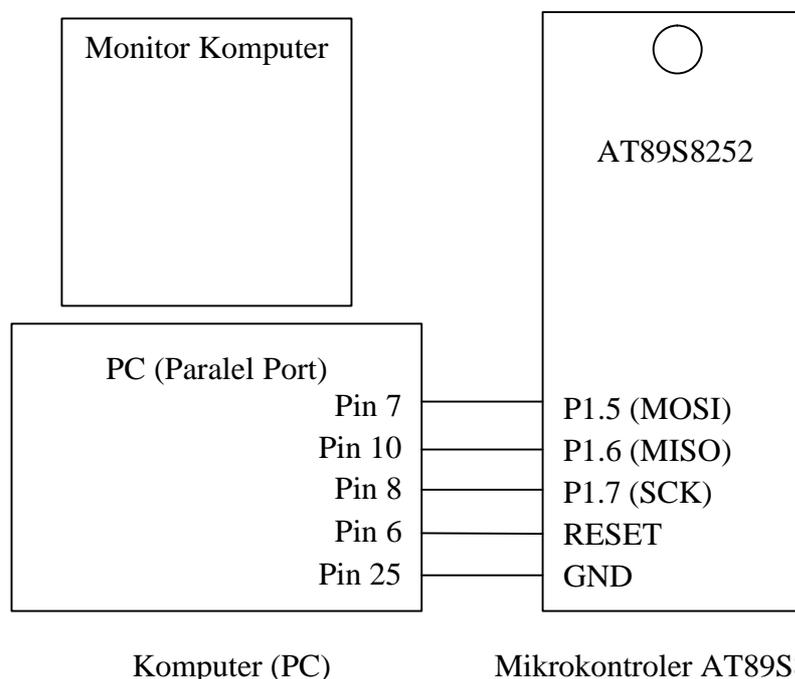
Dengan cara pemrograman seperti ini, maka akan didapat beberapa keuntungan yaitu:

- Tidak diperlukan alat pemrogram yang khusus.
- Mikrokontroler tidak perlu dicabut-cabut.
- Proses perancangan alat menjadi lebih cepat.



Gambar 2.3. Skematik IC Programmer

Skema dari pemrograman secara serial ini dapat dilihat pada gambar berikut ini.



Gambar 2.4. Hubungan antara PC dengan mikrokontroler AT89S8252

Cara menggunakan papan *board* AT89S8252 adalah sebagai berikut:

- a. Hubungkan papan *board* dengan *computer* PC pada *parallel port*-nya.
- b. Aktifkan papan *board*.
- c. *Download* program (*file** “.hex”).
- d. Atur *Reset* pada *programmer* menjadi ‘low’.
- e. Cek hasil program pada papan *board* tersebut.
- f. Jika masih terjadi kesalahan kembali ke *point* c.

Rangkaian pada papan *board* AT89S8252 adalah sangat sederhana. Hanya terdiri dari beberapa resistor, kapasitor, dan DIP *switch*.

2.4.3. Menggunakan *Software* pada PC

Software yang digunakan untuk mengisikan program ke dalam IC mikrokontroler adalah program ISP yang dapat diperoleh dari <http://www.aec-electronics.co.nz/>. Program ini gunanya untuk *men-download* program yang sudah dibuat sebelumnya (berekstensi “.hex”). Dengan menggunakan program ini, maka selain dapat memprogram *Code memory* juga dapat memprogram *Data memory*

maupun *lock bit*. Selain itu dengan adanya program ini juga mempermudah proses pengecekan (verifikasi) atau proses penghapusan (*erase*).

Cara kerja menggunakan program ISP untuk *men-download* program berekstensi '.hex' secara sistematis adalah sebagai berikut:

- a. Nyalakan Komputer (PC).
- b. Nyalakan atau buka program ISP.
- c. Buka program *downloader*-nya.
- d. *Setting* program ISP ke mode PROGRAM (PROG).
- e. Sorot Menu "*Setup*" pada program *downloader*.
- f. Isikan beberapa pilihan (contoh: mikrokontroler = AT89S8252).
- g. Sorot ke menu "*Load Hex file to Flash Buffer*"
- h. Isikan program yang akan dimasukkan ke program ISP yang berekstensi (".hex").
- i. Sorot dan tekan enter menu "*Reset*" sehingga menjadi "*high*".
- j. Sorot dan tekan enter pada menu "Program". Tunggu sampai selesai.
- k. Sorot dan tekan enter menu "*Reset*" sehingga menjadi "*low*".

Dengan cara di atas, maka program akan di-*download* dari Komputer (PC) dan diisikan ke dalam mikrokontroler AT89S8252.



Gambar 2.5. Tampilan Program ISP

Sumber : *The ISP Program CD ROM*, Auckland: AEC Electronic, 1997.