II. TEORI PENUNJANG

1. SEJARAH PERKEMBANGAN PLC

Pada tahun **1968,** sekelompok insinyur dari General Motor mulai berfikir untuk menciptakan *Programmable* ***Controller*** (PLCRC), yang mula – mula mempunyai spesifikasi sebagai berikut :

- Bersifat fleksibel karena ***program*** dapat diubah–ubah menurut urutan operasinya di lapangan.
- Perawatan dan perbaikannya sederhana dan mengutamakan penggunaan bentuk – bentuk modul.
- Mempunyai harga yang lebih murah dibandingkan dengan harga komponen–komponen seperti ***panel relay*** untuk menjalankan fungsi yang sama.
- Dapat diandalkan pada proses produksi dalam perindustrian dan lebih kecil dari rangkaian ***relay*** ekivalennya.

Hal ini tentunya memberikan minat yang besar bagi ilmuwan untuk memikirkan bagaimana PLC dapat digunakan untuk kontrol dalam industri.

Perangkat industri dengan cepat berubah dari instruksi logika sederhana menjadi meliputi ***Counter*** dan ***Timer,*** kemudian lebih berkembang lagi dengan fungsi–fungsi matematika pada PLC yang lebih besar. Perkembangan pada perangkat keras juga terjadi dengan kapasitas memori yang lebih besar dan jumlah titik I/O eksternal yang lebih banyak.

Pada tahun 1976, PLC inengalami kemajuan lagi yang memungkinkan untuk mengontrol *Remote I/O* ***Rack,*** yaitu sejumlah besar I/O jarak jauh yang dapat dimonitor dan diperbarui melalui sistem komunikasi, bahkan untuk jarak beberapa ratus meter dari PLC Utama. PLC ini menggunakan sebuah mikroprosesor dimulai pada tahun 1977 oleh Allan Bradley Corporation di Amerika. PLC ini menggunakan sebuah mikroprosesor 8080 dengan tambahan prosesor khusus untuk melayani instruksi logika kecepatan tinggi.

Pesatnya perkernbangan aplikasi dari PLC dalam industri sangat didukung dengan semakin majunya teknologi dari keluarga mikroprosesor yang memiliki berbagai macam keandalan. Sekarang *Programmable Controller* sedang dikembangkan dengan menggunakan mikrokontroler dimana ditekankan pada bentuk kontroler yang kecil, jumlah **kontrol** dan jaringan komunikasi. Pemasaran kontroler yang kecil mulai tumbuh dengan cepat sejak awal tahun 1980 dimana sejumlah perusahaan Jepang memperkenalkan produk PLC dalam ukuran kecil dengan harga per-unitnya lebih murah bila dibandingkan dengan PLC lain pada waktu itu. Hal ini menjadikan PLC dijadikan salah satu kebutuhan yang utama dalam proses industri dan cenderung semakin canggih dengan harganya cenderung menurun.

Ada perbedaan istilah yang digunakan untuk menyatakan *Programmable Controller,* seluruhnya bertolak dari pengoperasian mesin yang bersangkutan.

PC : Programmable Controller (Istilah Inggris)

PLC : Programmable Logic Controller (Istilah Amerika)

PBS : Programmable Binary Sistem (Istilah Swedia)

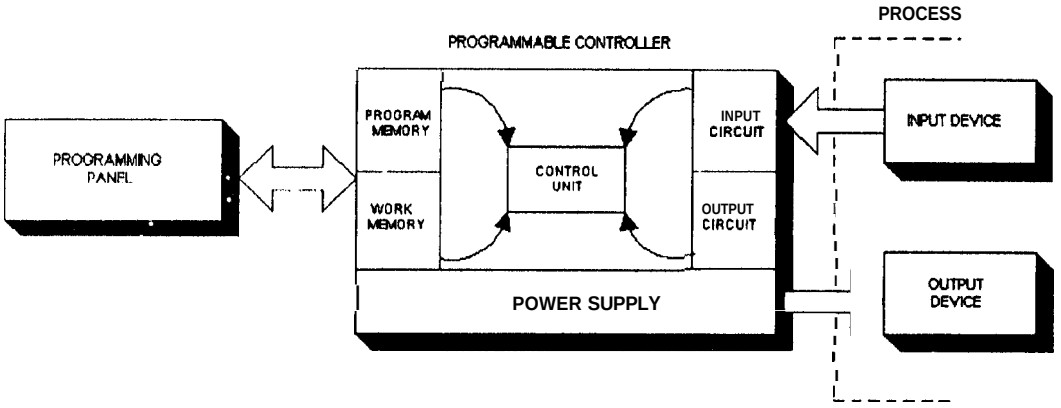
Pada mulanya istilah ini ditujukan untuk menggambarkan bahwa kontroler normalnya bekerja pada logika *ON/OFF* yaitu daerah *binary*. Namun sejak adanya *Programmable Controller* yang dilengkapi dengan proses Analog I/O maka istilah itu tidak lagi menyatakan keberadaannya. Karena alasan ini maka hampir seluruh keluarga *Programmable Controller* menggunakan istilah PROGRAMMABLE CONTROLLER yang disingkat PC, tetapi kadang-kadang untuk menghindari kekeliruan dengan *Personal Computer* (PC) maka digunakan singkatan PLC untuk PROGRAMMABLE LOGIC CONTROLLER.

2. STRUKTUR PERANGKAT KERAS

Seperti yang telah sedikit dijelaskan pada pendahuluan, PLC merupakan suatu komputer yang khusus dirancang untuk pemakaian dalam industri, yang memiliki **3** daerah fungsi yaitu :

1. **CPU**
2. MEMORI
3. I/O

Kondisi *input* yang masuk ke PLC akan diterima dan kemudian disimpan dalam memori. PLC kemudian mengolah *input* tersebut dan kemudian mengeluarkan **ke *output*** sesuai dengan *program* yang ada pada memori. Kondisi *output* yang dikeluarkan tersebut kemudian mengendalikan peralatan yang dihubungkan.



STRUKTUR PLC

Gambar 2.1[1]

Struktur PLC

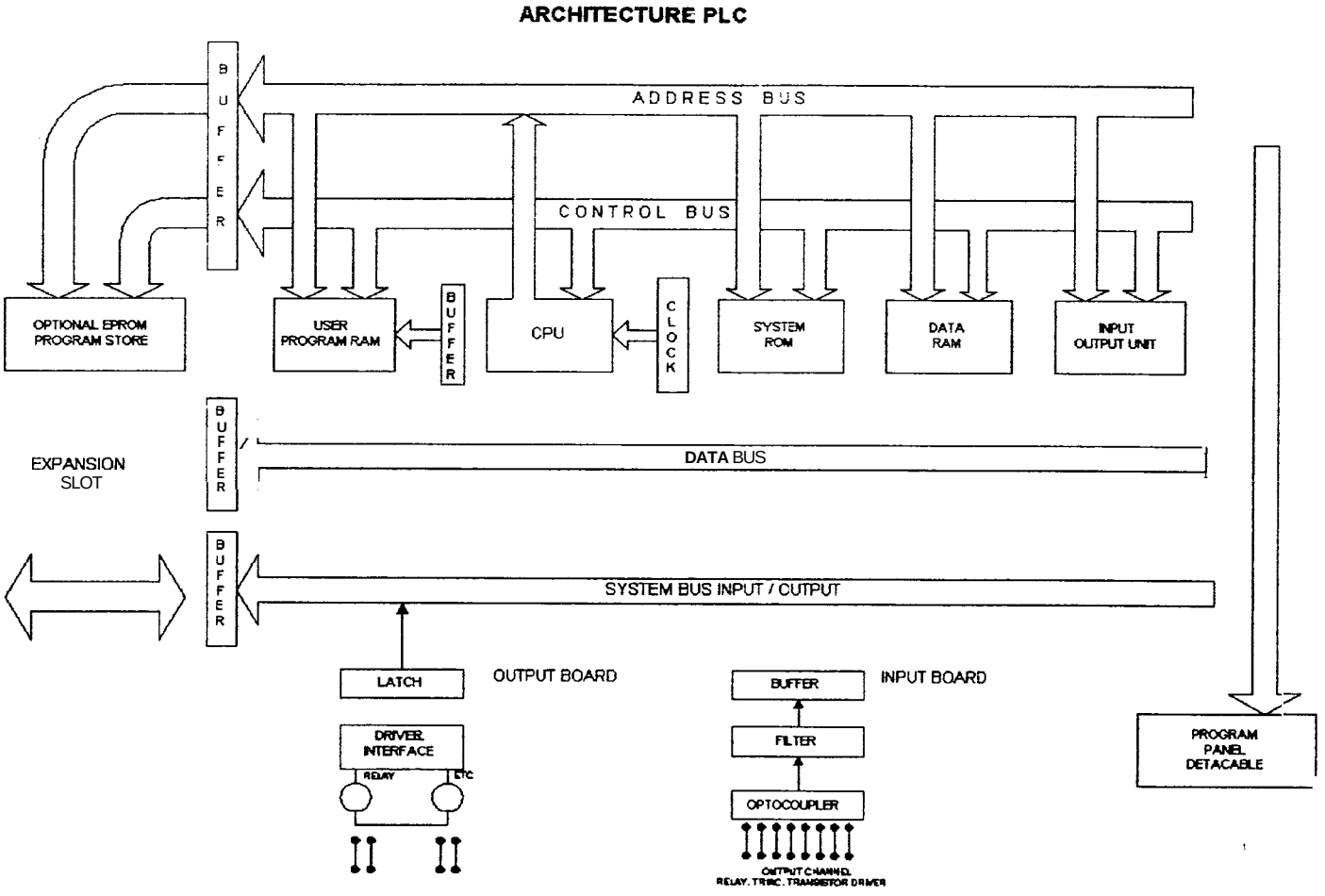
2.1. CPU

Central Processing Unit (CPU) adalah sebuah mikroprosesor yang mengatur kerja dari system PLC. Semua operasi dikontrol dan disimpan dalam CPU. CPU menjalankan instruksi – instruksi *program* yang disimpan dalam memori. Pengiriman semua informasi dibawah pengendalian CPU menggunakan sistem ***bus,*** yang menghubungkan CPU, memori dan **I/O.** CPU digerakkan dengan frekuensi ***clock*** oleh kristal luar atau oleh *RC Oscillator*, umumnya antara 1 – **8** ***MHz,*** tergantung dari jenis mikroprosesor yang digunakan serta penerapannya.

Frekuensi *clock* menentukan kecepatan operasi dari CPU dan memberikan *timing* / sinkronisasi untuk semua elemen dalam sistem.

---

[1] Warnock, Ian *G. Programmable Controller Operation And Application.* 1988. **p. 46.**



Gambar 2.2[2]

Arsitektur PLC

[2] Ibid. p. 52.

Semua PLC ***modern*** menggunakan mikroprosesor sebagai dasar dari sistem CPU-nya dan beberapa PLC yang besar juga telah dilengkapi dengan mikroprosesor tambahan untuk kontrol yang kompleks seperti ***coprocessor*** untuk proses matematika dan mikroprosessor untuk kontrol PID. Dalam gambar **2.2** akan terlihat jelas arsitektur dari PLC.

2.2. Memori

Memori adalah tempat untuk menyimpan sistem operasi, selain itu, suatu PLC mungkin juga membutuhkan memori untuk fungsi lain, yaitu

- Untuk menyimpan penyangga (*temporary buffer*) dari *I/O* ***Channel.***
- Penyimpanan sementara ***(temporary storage)*** untuk keadaan fungsi–fungsi di dalam, misalnya : ***timer, counter, relay,*** pembuatan tanda *(flag)* dan lain–lain.

2.3. *I/O*

Hampir semua PLC beroperasi diantara 5 dan 12 V_{DC}, unit I/O membentuk ***interface*** diantara mikroelektronik PLC karena itu harus diberikan beberapa pengkondisian sinyal yang diperlukan dan fungsi isolasi. Demikian sehingga memungkinkan suatu PLC dihubungkan langsung ke penggerak dan ***transducer*** tanpa memerlukan rangkaian perantara lagi. Beberapa I/O yang sering ada dalam industri antara lain :

Input : **5** V_{DC} **(** Tegangan umum TTL **)**

24 V_{DC}

110 V_{AC}

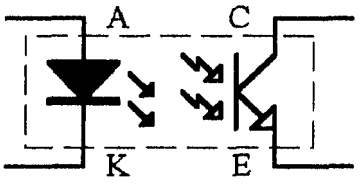
240 V_{AC}

Output : 24 V_{DC} 100 mA

110 V_{AC} 1 **A**

240 V_{AC} 2 **A** *(Relay)*

Ini adalah beberapa *standard* praktis untuk beberapa *I/O channel* yang secara listrik diisolasikan dari proses yang dikontrol dengan menggunakan rangkian *Opto Isolator* pada modul I/O seperti pada gambar **2.3.** Suatu rangkaian *Opto Isolator* terdiri dari LED *(Light Emiting Diode)* dan suatu *photo transistor* yang membentuk pasangan yang disebut *Opto Coupler* yang memperbolehkan sinyal kecil melewatinya, tetapi akan memotong lonjakan tegangan yang tinggi atau bergelombang ke tingkat yang sama tingginya. Hal ini memberikan perlindungan terhadap transien dari penyalaan dan catu daya yang bergelombang.



Gambar **2.3**

Opto Isolator

3. STRUKTUR PERANGKAT LUNAK

Dalam pemrograman PLC yang harus diperhatikan adalah kemudahan untuk memprogram PLC tersebut tanpa harus membuang waktu untuk mempelajarinya. Hal ini berarti dibutuhkan suatu bahasa pemrograman tingkat tinggi untuk dapat memberikan instruksi-instruksi yang sesuai dengan fungsi-fungsi yang ada pada PLC.

3.1. Instruksi-Instruksi Logika

Terdapat 2 metode yang digunakan dalam pemrograman PLC yaitu metode dengan menggambarkan instruksi-instruksi logika pada *personal computer* (PC) dan metode dengan menuliskan instruksi-instruksi logika pada *personal computer* (PC) atau *console*. *Diagram ladder* ataupun *text* yang dibaca dari *personal computer* akan dikonversikan ke *mnemonic* instruksi yang kemudian akan di-*compile* menjadi *machine code*. *Machine code* tersebut akan dikirimkan ke CPU dari PLC lewat komunikasi *serial*. Instruksi logika terdiri dari singkatan-singkatan yang menyatakan suatu ***mnemonic*** yang akan dieksekusikan oleh PLC. Singkatan-singkatan instruksi logika bervariasi, tergantung dari pabrik yang membuat PLC tersebut. Secara umum singkatan-singkatan instruksi logika tersebut mempunyai kesamaan dalam istilah-istilah yang menunjukkan proses logika yang harus dieksekusi oleh CPU dari PLC. Sebagai contoh perbandingan dari dua produk PLC diperlihatkan dalam tabel 2.1 berikut ini.

Tabel **2.1**[3]

Tipe Dari Instruksi

Texas Instrument		Mitsubishi F Series	
Mnemonic Action		Mnemonic Action	
STR	Store (Start a new rung of a ladder diagram)	LD	Start rung with on open contact
OUT	Output	OUT	Output
AND	Series Component	AND	Series Elements
OR	Paralel Component	OR	Paralel Elements

Karena pasangan-pasangan instruksi logika dari *programmable controller* cenderung sedikit maka dengan cepat dapat dikuasai oleh para teknisi. Setiap instruksi dibuat terdiri dari dua bagian, yaitu *mnemonic* instruksi (*opcode*) dan alamat *(operand),* sebagai contoh "OUT Y 1000", yang mempunyai arti bahwa instruksi tersebut berhubungan dengan peralatan *output* (Y) pada alamat 1000.

3.2. Cara Pengalamatan Dari Input – Output

Instruksi-instruksi logika digunakan untuk memprogram suatu rangkaian kontrol yang sudah didesain dalam bentuk *diagram ladder* dengan menetapkan *input* maupun *output* sebagai *operand* yang sesuai dengan PLC yang digunakan. Pengalamatan I/O dari suatu sistem berbeda-beda untuk setiap PLC, tetapi pada umumnya mempunyai arti yang sama. Sebagai contoh TEXAS INSTRUMENT dan MITSUBISHI menggunakan simbol X untuk menyatakan *input* dan Y untuk menyatakan

[3] Ibid. p. 64.

output. Interval alamat yang dapat dipakai untuk *input* dan *output* pada MITSUBISHI F40 adalah sebagai berikut:

24 Input → X400 – X407, X410 – X413

x500 – x507, x510 – x513

16 Output → Y430 – Y437, Y530 – Y537

Dengan memperhatikan *interval* dari alamat tersebut maka tidak akan terjadi *overlap* alamat antara fungsi-fungsi logika, bahwa 405 harus sebagai ***input*** dan 540 harus sebagai ***output.*** Sehingga simbol X dan Y sebenarnya tidak perlu dicantumkan.

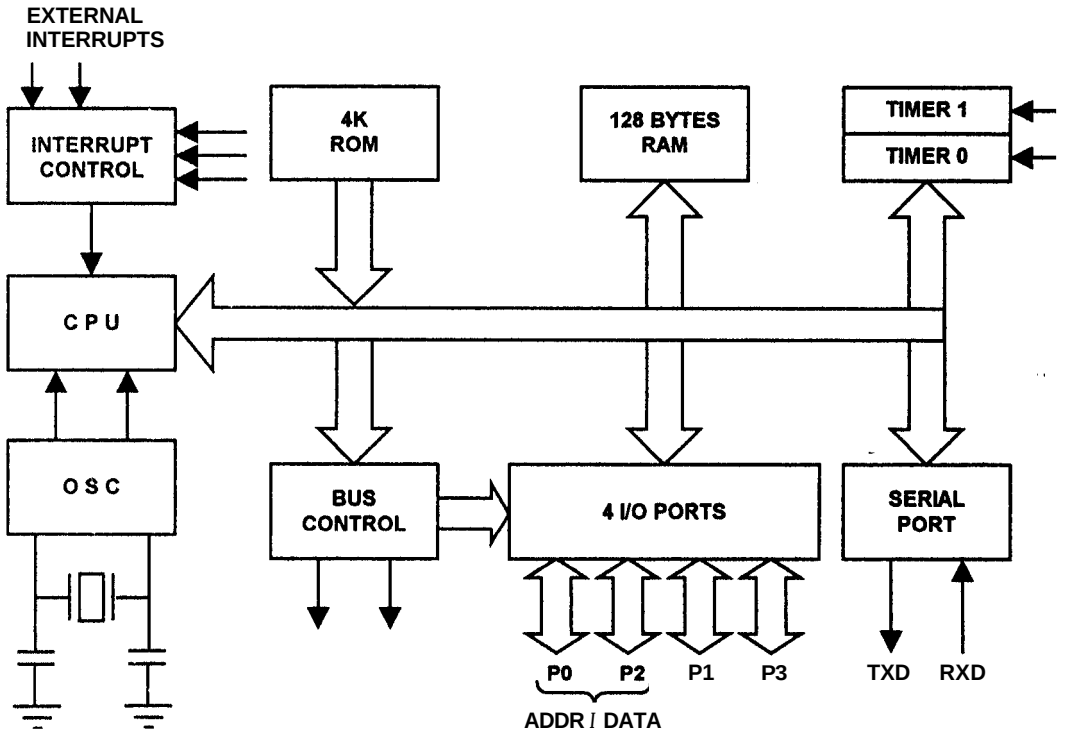
4. PROSES PELAKSANAAN INSTRUKSI DALAM PLC

Pada saat suatu *program* dimasukkan kedalam PLC, maka setiap instruksi ditempatkan dalam *Read Only Memory* (ROM) atau lebih dikenal dengan nama *Instruction Storage.* CPU memiliki *program counter register* yang bertugas menunjukkan alamat dari suatu lokasi memori. Satu instruksi yang diterima oleh CPU akan ditempatkan didalam *stack memory* untuk didekodekan dan kemudian akan dieksekusi. Berikut ini adalah contoh suatu *program* pendek yang akan memperjelas cara kerja dari CPU.

Alamat Memori	Instruksi
0000_H	LD X000
0001_H	OR X001
0002_H	OUT Y500

Kctika PLC dijalankan maka *program counter* akan menuju pada alamat memori 0000_H (lokasi dari instruksi yang pertama). Kemudian mengambil instruksi ‘‘LD X000”, didekodekan dan dieksekusi. CPU akan mengambil

logika dari ***input*** (X) pada alamat 000 kemudian CPU akan memasukkan ***data*** tersebut pada suatu ***stuck memory.*** Setelah itu maka ***program counter*** *akan* ditambah 1, dimana instruksi selanjutnya adalah "OR X001", CPU *akan* mengambil logika dari ***input*** (X) pada alamat 001 dan mengambil ***data*** pada ***stuck memory*** kemudian akan dilakukan operasi OR dan hasilnya *akan* dimasukkan pada ***stack memory.*** Selanjutnya***program counter*** akan ditambah 1, dari instruksi selanjutnya adalah "OUT Y 500"CPU akan mengambil ***data*** pada ***stack memory*** kemudian ***data*** tersebut akan dikeluarkan pada ***output*** (Y) dengan alamat 500. Proses ini *akan* terus berulang sampai CPU dimatikan.



Gambar 2.4[4]

Arsitektur Mikrokontroler AT89C52

[4] Intel Corporation. *MCS51 Microcontroller Family User's Manual.* (1994). p. **1-3.**

5. MIKROKONTROLER ATS9C52

Mikrokontroler AT89C52 termasuk salah satu *Integrated Circuit* (IC) Mikrokontroler dari keluarga MCS-51. Mikrokontroler AT89C52 ini mempunyai fitur sebagai berikut:

- ✓ CPU (*Central Processing Unit*) 8 bit.
- ✓ *Internal program memory* sebesar **8** KBytes.
- ✓ *Internal data memory* sebesar 256 Bytes.
- ✓ *External program memory* sebesar 64 KBytes.
- ✓ *External data memory* sebesar 64 KBytes.
- ✓ *Bi-directional* I/O sebanyak 32 bit (terbagi dalam **4** port).
- ✓ Tiga buah *Timer/Counter* **16** bit.
- ✓ UART (*Universal Asynchronous Receiver-Transmitter*) *full duplex.*
- ✓ Lima buah vektor *interrupt* dengan dua *level* prioritas.

Arsitektur mikrokmtroler dapat dilihat pada blok diagram (gambar **2.4).**

5.1. Deskripsi PIN untuk AT89C52

Mikrokontroler AT89C52 memiliki 40 *pin,* 32 *pin* diantaranya adalah ***hi-directional*** *I/O* yang terbagi dalam 4 ***port.*** Konfigurasi ***pin*** tersebut sebagai berikut:

- ❑ VCC *(pin* 40), merupakanpin tegangan ***+5*** Volt.
- ❑ **GND** *(pin* 20), merupakanpin tegangan 0 Volt *(ground).*

- *RESET (pin* **9),** jika diberi tegangan +5Volt maka seluruh isi dari *internal memory* dan *register-register* yang dimiliki **AT89C52** akan kembali ke kondisi *reset* (dapat dilihat pada tabel **2.1).** *Program Counter* dari *IC* tersebut berada pada *address* 0000_H. Jika diberi *ground* maka **AT89C52** akan beroperasi sesuai dengan *program* yang telah diisi didalam *internal* ROM atau *external* ROM.
- PSEN **(*pin* 29),** *Program Strobe Enable* adalah sinyal yang dikeluarkan oleh mikrokontroler **untuk** membaca memori *program (fetching).* Sinyal **PSEN** aktif berlogika '0'.
- **EA** - *External Access (pin* **3l),** jikapin **ini** diberi tegangan **+5** Volt **maka AT89C52 berada** dalam **mode access** *internal ROM,* sedangkan jika diberi tegangan 0 Volt *(ground)* maka **AT89C52** berada dalam *mode access external ROM*
- *a* ALE – *Address Latch Enable (pin* **30),** sinyal ini berupa pulsa persegi yang keluar terus menerus dengan frekuensi **1/6** dari frekuensi kristal, digunakan untuk memisah *address* ***bus*** byte rendah **(A7 –** AO) yang sebelumnya dimultipleks dengan *data bus* dalam **AD7 -ADO.**
- *a* **XTAL1** *(pin* **19)** dan **XTAL2** *(pin* **18),** jika *on-chip oscillator* akan digunakan, maka kedua *port* ini harus dihubungkan dengan sebuah kristal. Jika akan digunakan

external oscillator maka port XTAL,1 harus dihubungkan dengan oscillator sementara port XTAL 2 dibiarkan terbuka.

- Port 0 (pin 32, 33, 34, 35, 36, 37, 38, 39), berfungsi sebagai address bus byte rendah (A7 – AO) dan *data bus* (D7 – DO) yang didisain secara multipleks, sehingga *port* ini diberi nama AD7 – ADO.
- *Port* 1 (pin 8, 7, 6, 5, 4, 3, 2, I), berfungsi sebagai *port I/O* dua arah yang dapat diakses per bit. Pada AT89C52 terdapat port khusus yaitu pada *port* P1.0 dan P1.1 (dapat dilihat pada tabel 2.2).
- Port 2 (*pin* 28, 27, 26, 25, 24, 23, 22, 21), berfungsi sebagai address *bus* byte tinggi (A15 – A8).
- Port 3 (*pin* 17, 16, 15, 14, 13, 12, 11, lo), dapat digunakan dengan dua cara, yaitu dianggap sebagai port *I/O* biasa (seperti port 1) atau dianggap sebagai port khusus. Port khusus tersebut dapat dilihat pada tabel 2.2.

Tabel 2.2[5]

Fungsi Khusus Port 3 Dan P1.1, P1.0 Pada **AT89C52**

PIN	PORT	FUNGSI
1	P1.0	T2 (*External input* untuk *Timer/Counter* 2)
2	P1.1	T2 Ex (*Timer/Counter 2 capture/reload trigger & direction control*)
10	P3.0	RXD (*port input* untuk komunikasi serial)
11	P3.1	TXD (***port output*** untuk komunikasi serial)
12	P3.2	INTO (***External Interrupt 0***)
13	P3.3	INT1 (*External Interrupt* 1)
14	P3.4	T0 (*External input* untuk *Timer/Counter* 0)
15	P3.5	T1 (*External input* untuk *Timer/Counter* 1)
16	P3.6	WR (sinyal *write* untuk *external data memory*)
17	P3.7	RD (sinyal *read* untuk *external data memory*)

[5] **ATMEL** Corporation. ***Microcontroller Databook.*** (October 1995). **p.** 3-67.

Tabel 2.3[6]

Kondisi AT89C52 Saat *Power On - Reset*

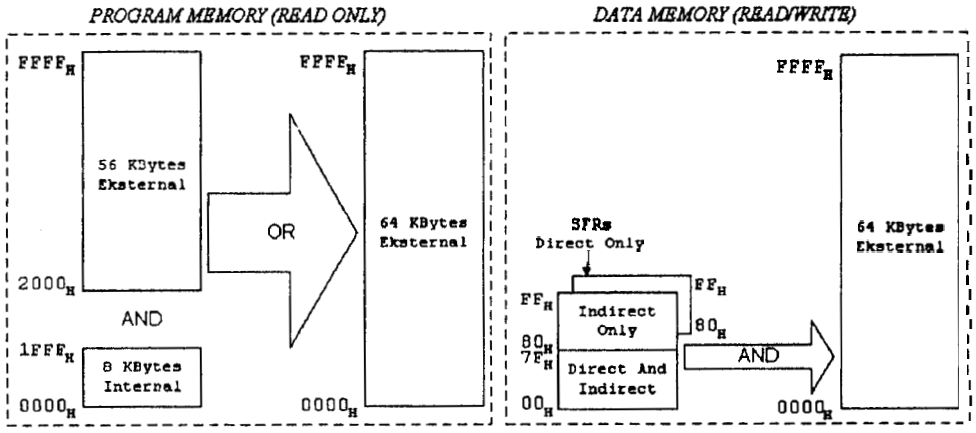
Register	Nilai dalam binari
ACC	00000000
B	00000000
PSW	00000000
SP	00000111
DPH	00000000
DPL	00000000
P0	11111111
P1	11111111
P2	11111111
P3	11111111
IP	00000000
IE	00000000
TMOD	00000000
T2MOD	00000000
TCON	00000000
T2CON	00000000
TH0	00000000
TL0	00000000
TH1	00000000
TL1	00000000
TH2	00000000
TL2	00000000
RCAP2H	00000000
RCAP2L	00000000
SCON	00000000
SBUF	*Indeterminate*
PCON	00000000

[6] Ibid. p. 2-25.

5.2. Organisasi Memori

Mikrokontroler **AT89C52** mengenal dua macam memori, yaitu *program memory* dan *data memory*. *Program memory* adalah tempat untuk menyimpan *program* yang berisikan instruksi *(instruction storage),* sedangkan *data memory* adalah tempat untuk menyimpan *data (data storage). Program memory* dapat berbentuk ROM internal atau eksternal, sedangkan *data memory* dapat berbentuk RAM internal atau eksternal. **AT89C52** adalah jenis mikrokontroler yang mempunyai internal ROM, *program memory* sebesar 64 KBytes tersebut dibagi menjadi dua bagian yaitu bagian internal dan eksternal. Besar memori dari **AT89C52** adalah sebesar **8** KBytes, *range* memorinya dimulai dari $0000_H - 1FFF_H$. Jika **EA** = '1' maka setiap pengaksesan *program memory* akan diarahkan *internal* ROM terlebih dahulu, dan selebihnya diarahkan ke *external* ROM, sedangkan jika **EA** = '0' maka setiap pengaksesan *program memory* akan diarahkan *external* ROM.

Data Memory dibagi menjadi dua bagian yaitu *internal* dan *external memory*, *external* dan *internal* RAM mempunyai struktur bagian yang terpisah berbeda dengan *program memory*. *External memory* berukuran **64** KBytes dengan *range* alamat mulai dari 0000_H -$FFFF_H$, sedangkan *internal memory* terbagi menjadi dua bagian yaitu bagian *data memory* yang berukuran ***256*** bytes dan bagian *Special Function Register* (SFR) yang diperlukan mikrokontroler untuk menyimpan *data* penting.

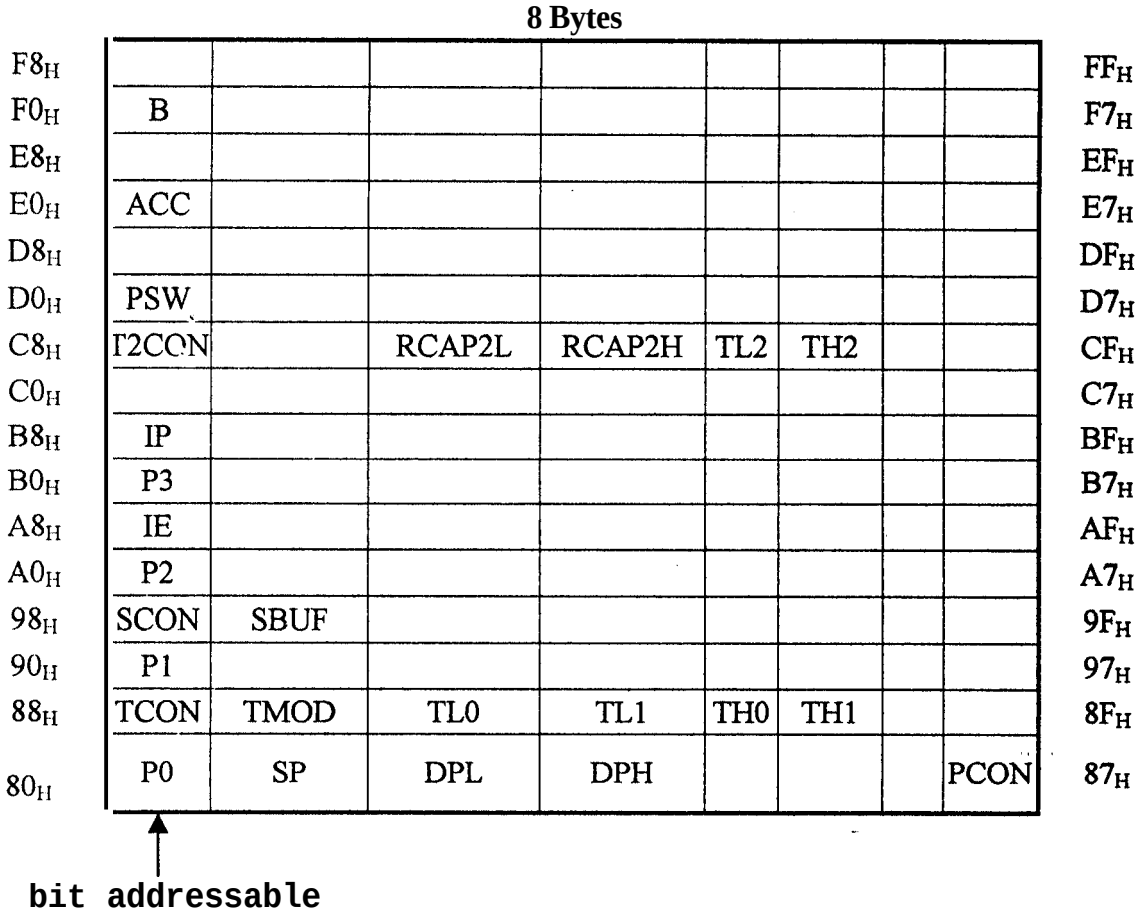


Gambar 2.5[7]

Sturktur Memori AT89C52

Cara untuk mengakses ***internal memory*** dan **SFR** pada AT89C52 mempunyai aturan tersendiri. Pada ***internal memory*** dibagi menjadi 2 bagian yaitu bagian yang dapat diakses secara ***direct*** ataupun ***indirect*** yaitu mulai dari alamat 00_H – $7F_H$, sedangkan alamat mulai dari 80_H – FF_H hanya dapat diakses secara ***indirect.*** Untuk ***Special Function Register*** (SFR) hanya dapat diakses secara ***direct.*** Tabel dari SFR dapat dilihat pada gambar **2.6.**

[7] Ibid. **p.** 2-2 1.



Gambar 2.6[8]

Special Function Register

5.3. Interrupt

Mikrokontroler AT89C52 memiliki 6 sumber *interrupt*, yaitu dua buah *interrupt* eksternal (INTO dan INT1), **tiga buah** *Timer/Counter* **(TO,** T1 **dan** T2), **serta satu buah *interrupt serial*. Keenam *interrupt* diatas memiliki *vector address* yang dapat dilihat pada tabel 2.4.**

---

[8] **Ibid. p. 2-26.**

Tabel 2.4[9]

Interrupt

Interrupt Source	Vector Address
IE0	0003h
TF0	[illegible]
IE1	0013h
TF1	001Bh
RI & TI	0023h
TF2 & EXF2	002Bh

Jika terjadi ***interrupt,*** maka ***program*** akan melompat ke alamat vektor ***interrupt*** yang bersangkutan dan baru kembali ke ***program*** utama jika menjumpai perintah RETI ***(returnfrom interrupt).***

Keenam *interrupt* ini dapat diaktifkan atau dimatikan melalui register IE ***(Interrupt Enable). Bit*** **EA** berfungsi untuk mengaktifkan sistem ***interrupt*** secara keseluruhan, sedangkan ET2, ES, ET1, EX1, ET0, EX0 berturut-turut untuk mengaktifkan *Timer/Counter 2*, *Serial*, *Timer/Counter 1*, *External Interrupt 1*, *Timer/Counter 0*, *External Interrupt 0* (dapat dilihat pada gambar 2.7).

Interrupt eksternal (INTO dan **INT1)** dapat diaktifkan dengan dua ***mode,*** yaitu ***mode*** aktif ***level (level activated)*** dan ***mode*** aktif transisi ***(transition activated).*** Jika INTO atau INT1 diberi logika '0' pada aktif *level* atau diberi perubahan transisi turun (*falling edge*) dari **logika** '1' → **logika** '0' maka akan menyebabkan terjadinya ***interrupt.***

---

[9] Ibid. **p..**2-28.

Mode untuk mengaktifkan ***level*** ataupun transisi dapat dilakukan pada ***register*** TCON (*Timer/Counter Control*), yaitu pada ***bit*** IT0 dan IT1. *Interrupt Timer/Counter* TO, T1 dan T2 terjadi jika terjadi ***overflow*** pada ***register*** *Timer/Counter* yang bersangkutan. ***Register*** *Timer/Counter* itu adalah TH0|TL0, TH1|TL1, dan TH2|TL2.

Interrupt serial terjadi jika dijumpai adanya ***stop bit*** baik pada jalur TX ***(transmit)*** maupun pada jalur RX ***(receive).***

EA	-	ET2	ES	ET1	EX1	ET0	EX0

TCON: Timer/Counter Control Register

TF1	TR1	TF0	TR0	E 1	IT1	IE0	IT0

TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/-T2	CP/-RL2

Gambar 2.7[10]

Register-Register Interrupt

5.4. Komunikasi Serial

Pada mikrokontroler AT89C52 terdapat fasilitas komunikasi *serial full duplex*. Dalam mikrokontroler ini terdapat ***register*** TX yang berfungsi untuk mengirimkan ***data*** lewat ***transmitter.***

---

[10] Ibid. pp. 2-28, 2-30, 2-33.

Register RX yang berfungsi untuk menampung ***data*** yang diterima lewat ***receiver.*** Kedua ***register*** ini terpisah secara fisik, sehingga tidak mungkin terjadi ***data*** bentrok. Dalam SFR terdapat SBUF ***(Serial Buffer),*** dimana berfungsi sebagai penghubung antar kedua ***register*** dengan *program.* Jika terjadi penulisan ***data*** ke SBUF maka ***data*** tersebut akan diteruskan ke ***register*** TX, selanjutnyajika ***data*** padajalur penerima maka *data* akan disimpan di ***register*** RX, selanjutnya ***data*** tersebut dapat dibaca oleh ***program*** dengan memberi perintah pembacaan terhadap SBUF.

Komunikasi ***serial*** mempunyai empat macam mode, yaitu mode 0, 1, **2,** dan 3. Hal ini berkaitan dengan tipe ***data*** dan ***baud rate*** yang digunakan (dapat dilihat pada tabel 2.5).

Tabel 2.5[11]

Mode Komunikasi Serial

SM0	SM1	Mode	Description	Baud Rate
0	0	0	Shift Register	F_{OSC} / 12
0	1	1	8 bit UART	Variable
1	0	2	9 bit UART	F_{OSC} / 64 or F_{OSC} / 32
1	1	3	9 bit UART	Variable

Hit SMO'dan SM1 terdapat pada ***register*** SCON ***(Serial Control),*** **yang** berfungsi untuk memilih ***mode*** komunikasi ***serial.***

[11] Ibid. **p.** 2-35.

Untuk *mode* yang menggunakan 9 *bit Universal Asynchronous Receiver Transmitter* (UART) mempunyai arti bahwa *data* akan berukuran 9 *bit,* yaitu *bit* 1 sampai *bit* 8 akan terletak pada *register* SBUF sedangkan *bit* 9 akan diletakkan pada TB8 (untuk *transmit)* dan RB8 (untuk *receive).* Kedua *bit* TB8 dan RB8 terdapat pada *register* **SCON.**

SM0	SM1	SM2	REN	TB8	RB8	TI	RI

Gambar 2.8[12]

Register Serial Control (SCON)

Pada gambar 2.8 dapat dilihat isi dari *register* SCON. Bit *SM2* berfungsi untuk mengaktifkan komunikasi multiprosesor sedangkan *bit* REN berfungsi untuk menerima *data* atau tidak menerima data, jika *bit* REN = '0', maka tidak akan dapat untuk menerima *data* tetapi jika *bit* REN = '1', maka dapat menerima *data. Bit* TI dan RI merupakan *bit* yang dapat menyebabkan terjadinya *interrupt serial,* dimana *bit* TI *(Transmit Interrupt Flag)* akan berlogika '1' (terjadi *interrupt serial)* jika pada jalur *transmit* telah mengirimkan *bit* terakhir *(Stop Bit),* sedangkan *bit* RI *(Receive Interrupt Flag)* akan berlogika '1' (terjadi *interrupt serial)* jika pada jalur *receive* telah menerima *bit* terakhir *(Stop Bit).*

Baud *rate mode 0* selalu bernilai 1/12 frekuensi *oscillator,* sedangkan *baud rate* pada ***mode*** 2 akan bernilai 1/64 frekuensi *oscillator* jika pada *register* PCON *(Power Control) bit* SMOD berlogika '0' jika

---

12 Ibid. p. 2-35.

bit SMOD berlogika '1' maka *baud rate* pada *mode 2* akan bernilai 1/32 frekuensi *oscillator.*

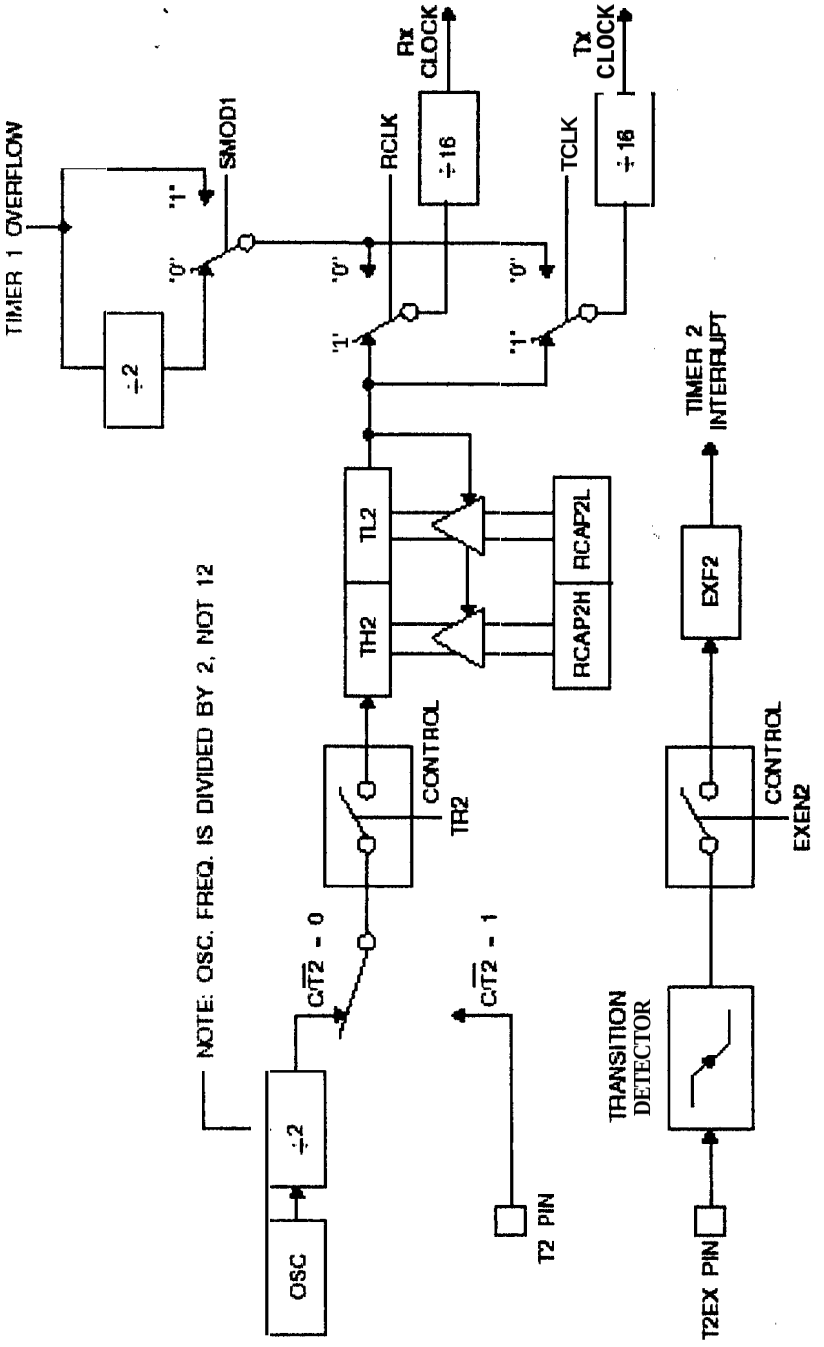
Baud rate pada mode 1 dan 3 dapat dibuat dengan memanfaatkan *overflow* dari *Timer1* atau *Timer2.*

$$BaudRate = \frac{2^{SMOD}}{32} \times \frac{OscillatorFrequency}{12x[256-(TH1)]} \qquad (2.1)$$

Nilai *baud rate* pada *mode* 1 dan 3 ini dapat dihitung dengan melihat persamaan (2.1), dimana SMOD bernilai 1 maka *baud rate* akan menjadi *2* kali lipat sedangkan jika *bit* SMOD bernilai 0 maka *baud rate* hanya dikali 1. Pada persamaan (2.1) diatas perhitungan *baud rate*-nya menggunakan *Timer1*. Sedangkan perhitungan dengan menggunakan *Timer2* dapat dilihat pada persamaan (2.2) dibawah ini.

$$BaudRate = \frac{OscillatorFrequency}{32x[65536-(RCAP2H,RCAP2L)]} \qquad (2.2)$$

Hal yang harus diperhatikan pada saat melakukan proses *serial* dengan menggunakan *Timer2* yaitu *bit* RCLK dan TCLK harus diberi logika '1', sedangkan menggunakan *Timer1* maka *bit* RCLK dan TCLK diberi logika '0'. *Bit* RCLK dan TCLK ada pada *register* T2CON yang dapat dilihat pada gambar 2.7. Proses *bit* RCLK dan TCLK pada proses perhitungan *baud rate* ini dapat dilihat pada gambar 2.9.



Gambar 2.9[13]

Baud Rate Generator Mode

[13] ATMEL Corporation. *AT89C52 Microcontroller Datasheet.* (199). p.