

BAB II

LANDASAN TEORI

2.1 Programmable Logic Controller (PLC).

Kemajuan teknologi yang berkembang pesat dewasa ini, mengakibatkan industri sebagai produsen/penghasil barang menggunakan cara-cara otomatisasi untuk meningkatkan jumlah hasil barang yang diproduksinya secara efektif dan efisien. Salah satu peralatan kontrol otomatis yang saat ini banyak digunakan adalah PLC.

PLC adalah kendali logika terprogram merupakan suatu piranti elektronik yang dirancang untuk beroperasi secara digital dengan menggunakan memori sebagai media penyimpanan instruksi-instruksi internal untuk menjalankan fungsi-fungsi logika, seperti fungsi pencacah, fungsi urutan proses, fungsi pewaktu, fungsi aritmatika, dan fungsi yang lainnya dengan cara memprogramnya. Program-program dibuat kemudian dimasukkan dalam PLC melalui programmer/monitor. Pembuatan program dapat dilakukan melalui komputer sehingga dapat mempercepat hasil pekerjaan. PLC dapat digunakan untuk memonitor jalannya proses pengendalian yang sedang berlangsung, sehingga dapat dengan mudah dikenali urutan kerja (*work squence*) proses pengendalian yang terjadi pada saat itu (Budyanto. M, 2003:1)

PLC pertama kali digunakan sekitar pada tahun 1960-an untuk menggantikan peralatan konvensional yang begitu banyak. Perkembangan PLC saat ini terus mengalami perkembangan sehingga bentuk dan ukurannya semakin kecil. Saat ini terdapat PLC yang dapat dimasukkan dalam saku karena bentuk

dan ukurannya yang sangatlah kecil, dan dalam perkembangannya, dimasa yang akan datang akan diperkenalkan PLC dengan bentuk dan ukuran sebesar kotak rokok.

Pada tahun 1980-an harga PLC masih terhitung mahal, namun saat ini dapat dengan mudah ditemukan dengan harga yang relatif murah. Beberapa perusahaan komputer dan elektronik menjadikan PLC menjadi produk terbesar yang terjual saat itu. Pertumbuhan pemasaran PLC mencapai jumlah 80 juta dolar di tahun 1978 dan 1 milyar dolar pertahun hingga tahun 2000 dan angka ini terus berkembang, mengingat penggunaan yang semakin luas, terutama untuk proses pengontrolan di industri, pada alat-alat kedokteran, alat-alat rumah tangga.

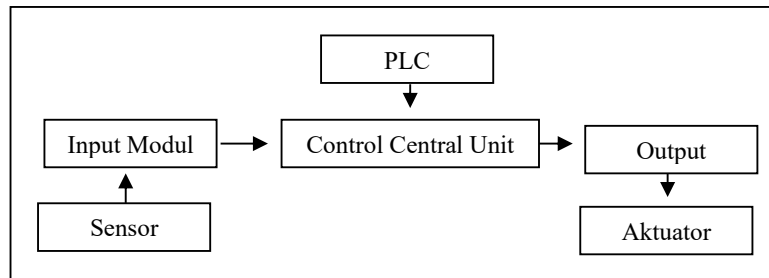
Pabrik pembuat PLC mendesain sedemikian rupa sehingga pengguna dapat dengan mudah menguasai fungsi-fungsi dan logika-logika hanya dalam beberapa jam saja. Fungsi-fungsi dasar yang banyak digunakan antara lain: kontak-kontak logika, pewaktu (timer), pencacah (counter), dan sebagainya. Bagi yang mempunyai latar belakang logika-logika digital akan dengan mudah menguasainya dalam beberapa jam saja, berlainan halnya dengan orang yang tidak memiliki latar belakang ini akan memakan waktu agak lama untuk menguasai fungsi dan logika-logika kendali PLC.

Seperti halnya komputer, PLC juga mempunyai kelengkapan yaitu CPU (*Central Processing Unit*), memori (*RAM dan ROM*), programmer/monitor, dan modul I/O (input/output).

2.1.1 PLC (Programmable Logic Controller)

PLC atau biasa disebut PC (*Programmable Controller*) adalah suatu perangkat yang dapat dengan mudah diprogram untuk mengontrol peralatan. PLC

sederhana mempunyai komponen utama berupa CCU (*Central Control Unit*), Unit I/O, *Programing Console*, *Rack* atau *Mounting assembly* dan catu daya, sistem komponen dari PLC adalah seperti gambar dibawah ini.



Gambar 2.1. Sistem Komponen Dari sebuah PLC

2.1.2 Central Control Unit (CCU)

CCU merupakan unit pusat pengolah data yang digunakan untuk melakukan proses pengolahan data dalam PLC. CCU merupakan sebuah mikroprocessor, adapun jenis processor yang dipergunakan sesuai dengan merk dan tipe dari PLC, untuk PLC FESTO DIDACTIC SERI FPC 100 menggunakan mikrokontroler 8031.

2.1.3 Unit Input Output

Fungsi dari sebuah modul input adalah untuk mengubah sinyal input dari sensor ke PLC untuk diproses dibagian Central Control Unit, sedangkan modul output adalah kebalikannya, mengubah sinyal PLC ke dalam sinyal yang sesuai untuk menggerakkan aktuator.

Fungsi terpenting dari sebuah modul input adalah sebagai berikut:

- a. mendeteksi sinyal masukan
- b. mengatur tegangan kontrol untuk batas tegangan logika masukan yang diijinkan

- c. melindungi peralatan elektronik yang sensitive terhadap tegangan luar
- d. menampilkan sinyal masukan tersebut

Fungsi terpenting dari sebuah modul output adalah sebagai berikut:

- a. Mengatur tegangan kontrol untuk batas tegangan logika keluaran yang diijinkan
- b. Melindungi peralatan elektronik yang sensitive terhadap tegangan luar.
- c. Memberikan penguatan sinyal output sebelum dikeluarkan sehingga cukup kuat untuk menggerakkan aktuator
- d. Memberikan perlindungan terhadap arus hubungan singkat (*short-circuit*) dan pembebanan lebih (*overload*)

2.1.4 Kelebihan dan Kekurangan PLC

Beberapa kelebihan yang dimiliki oleh *PLC* dibanding dengan kontrol reley konvensional, adalah:

1. *Fleksibel*

Sebelum ditemukannya *PLC*, setiap mesin mempunyai alat kontrol/pengendali tersendiri dimisalkan terdapat 15 buah mesin, maka alat pengendali yang diperlukan juga terdapat 15 buah. Lain halnya sekarang ini dengan adanya *PLC* maka untuk beberapa mesin hanya memerlukan 1 buah *PLC* saja.

2. Deteksi dan koreksi kesalahan lebih mudah.

Setelah desain program control telah selesai dibuat, kemudian dimasukkan dalam *PLC* dengan cara memprogramnya, maka program tersebut dapat dengan mudah diubah dengan menggunakan keyboard hanya dalam beberapa menit saja. Setelah itu program kembali dapat dijalankan, jika masih terdapat

kesalahan maka dapat dikoreksi dengan menggunakan diagram tangga (*ladder diagram*) sehingga koreksinya dapat dengan segera dilaksanakan.

3. Harga relatif murah.

Perkembangan teknologi memungkinkan untuk meningkatkan beberapa fungsi dengan bentuk ukuran yang semakin kecil. Tentunya hal ini juga akan menurunkan harga pembuatan yang mahal. Salah satu fungsi yang terus ditingkatkan adalah modul I/O (masukkan/keluaran). Saat ini kita mendapatkan *PLC* dengan jumlah masukkan dan keluaran yang banyak hanya dengan beberapa *dollar* saja.

4. Pengamatan visual (*visual observation*)

Operasi *PLC* saat menjalankan program yang telah dibuat dapat dilihat dengan teliti dengan menggunakan *layer CRT* (Cathode Ray Tube), sehingga ini sangat memudahkan dalam proses pencarian, pengamatan, atau dalam pembenahan program. Dengan demikian proses pembenahan hanya membutuhkan waktu yang relative singkat.

5. Kecepatan operasi (*speed of operation*)

Kecepatan operasi *PLC* sangatlah cepat. Kecepatan operasi ini adalah untuk mengaktifkan fungsi-fungsi logika hanya dalam waktu beberapa milidetik, dikarenakan menggunakan rangkaian elektronik sehingga operasinya sangatlah cepat, berlainan saat digunakan relai magnetik, yang mempunyai kecepatan operasinya lebih lambat.

6. Lebih sederhana dan mudah dalam penggunaannya, memodifikasi lebih mudah tanpa tambahan biaya.

Beberapa kekurangan yang dimiliki oleh PLC dibanding dengan kontrol relay konvensional, adalah:

1. Teknologi baru, sehingga dibutuhkan waktu untuk mengubah *system* konvensional yang telah ada.
2. Keadaan lingkungan. Untuk proses seperti pada lingkungan panas yang tinggi, *vibrasi* yang tinggi penggunaannya kurang cocok, karena dapat merusak *PLC*.

2.1.5 Konsep PLC

Konsep dari *PLC* sesuai dengan namanya, adalah sebagai berikut:

a. Programmable

Menunjukkan kemampuannya yang dapat dengan mudah diubah-ubah sesuai program yang dibuat.

b. Logic

Menunjukkan kemampuan dalam memproses input secara aritmetik (membandingkan, menjumlah, membagi dan sebagainya).

c. Controller

Kemampuan dalam mengontrol dan mengatur proses sehingga menghasilkan output yang diinginkan.

2.1.6 Fungsi PLC

Fungsi dari *PLC* dapat dibagi secara umum dan secara khusus. Secara umum fungsi *PLC* adalah sebagai berikut :

a. Control Sequence

PLC memproses input sinyal biner menjadi sinyal output yang digunakan untuk keperluan pemrosesan teknik dan yang secara berurutan (*sequence*). *PLC* menjaga agar semua step dalam proses *sequence* berlangsung dalam urutan yang tepat.

b. Monitoring Plant

PLC secara terus-menerus memonitor status suatu *system* (misalnya temperature, tekanan, tingkat ketinggian) dan mengambil tindakan yang diperlukan sehubungan dengan proses yang dikontrol (misalnya nilai telah melebihi batas) atau menampilkan pesan tersebut pada operator.

Sedangkan fungsi *PLC* secara khusus adalah memberikan input ke *CNC* (*Computerized Numerical Control*). Beberapa *PLC* dapat memberikan input ke *CNC* untuk kepentingan pemrosesan lebih lanjut. *CNC* bila dibandingkan dengan *PLC* mempunyai ketelitian yang lebih tinggi dan lebih mahal harganya. *CNC* biasanya dipakai untuk proses finishing, membentuk benda kerja, digunakan pada unit press, moulding.

2.1.7 Perbandingan antara PLC dengan kontrol konvensional

Kontrol konvensional yang menggunakan relay atau kontraktor mempunyai keuntungan dan kerugian bila digunakan sebagai rangkaian kontrol bila dibandingkan kontrol dengan menggunakan *PLC*.

Relay sendiri merupakan kontrol elektronik, karena terdapat koil/kumparan yang akan menggerakkan kontak membuka atau menutup bila kumparannya diberi arus listrik. Berikut ini adalah keuntungan dan kerugian menggunakan relay atau kontraktor:

Keuntungan :

- a. Mudah diadaptasikan untuk tegangan yang berbeda.
- b. Tidak banyak dipengaruhi oleh temperature sekitarnya. Relay terus beroperasi pada temperature 353 K (80 derajat celcius) sampai 240 K (-33 derajat celcius).
- c. Tahanan yang relative tinggi antara kontak kerja pada saat terbuka.
- d. Beberapa sirkuit terpisah dapat dihidupkan.
- e. Sirkuit yang mengontrol relay dan sirkuit yang membawa arus yang terhubung.
- f. Fisik terpisah satu sama lainnya.

Kerugian :

- a. Kontak dibatasi pada keausan dari bunga api atau dari oksidasi (material kontak yang terbaik adalah platina, emas, perak). Menghabiskan banyak tempat dibandingkan dengan transistor.
- b. Menimbulkan bunyi selama proses kontak.
- c. Kecepatan kontak yang terbatas 3 ms sampai 17 ms.
- d. Kontaminasi (debu) dapat mempengaruhi umur kontak.

Keuntungan PLC atas kontrol konvensional:

1. Aplikasi universal
2. Produksi yang besar
3. Harga semakin besar
4. Bidang aplikasi baru
5. Pemrogramman yang ampuh
6. Mudah diubah

7. Commissioning mudah dengan menggunakan fungsi-fungsi yang tersedia
8. Teks dan grafiks

2.1.8 Jenis PLC

Salah satu PLC yang dimiliki STIKOM dan digunakan untuk praktikum adalah PLC FESTO dari Jerman, seri FPC 101 B-LED dan FPC 101 AF. PLC ini mempunyai kelebihan dapat mengenal program dengan bahasa pemrograman tingkat tinggi (high level language), yaitu statement list atau STL, selain menggunakan ladder diagram yang sudah umum dan menggunakan pemrograman matrix MAT. Bahkan untuk seri tertentu dapat diprogram dengan menggunakan bahasa BASIC atau function chart FUC (Indrijono Dwi,1999:1)

PLC FPC 101B-LED memiliki spesifikasi yaitu:

- a. Indikator untuk status dan error
- b. Pemrograman yang mudah melalui PC dengan ladder Diagram dan Statement List
- c. Perlindungan output dari short-circuit
- d. Pengaman polaritas power suplay
- e. LED indikator untuk *input* dan *output*

Data teknik PLC FPC 101B-LED

- a. 21 *input*
- b. 14 *output*
- c. 32 *timer*
- d. 16 *counter*
- e. 64 *register*

- f. 256 flag
- g. 12 kBytes user memory
- h. 7,5 W untuk tiap output

Sensor – sensor yang digunakan di Laboratorium PLC adalah:

- a. Push button switch
- b. Switch toggle
- c. Sensor capacitive
- d. Sensor induktive
- e. Sensor optik
- f. Limit switch

Aktuator yang digunakan pada Laboratorium PLC adalah:

- a. Single selenoid
- b. Double selenoid
- c. Indikator Lamp
- d. Buzzer

2.1.9 Bahasa Pemrograman

Terdapat banyak pilihan bahasa untuk membuat program dalam PLC. Masing-masing bahasa mempunyai keuntungan dan kerugian sendiri-sendiri tergantung dari sudut pandang kita sebagai user. *Diagram ladder* adalah bahasa yang dimiliki oleh setiap PLC. Selain itu ada beberapa jenis PLC yang mendukung bahasa pemrograman lain seperti PLC FESTO yang juga mendukung bahasa *statement list* (STL) yang nanti akan digunakan penulis sebagai bahasa pemrograman untuk PLC Festo.

A. Ladder Diagram (LDR)

Ladder diagram menggambarkan program dalam bentuk grafik. Diagram ini dikembangkan dari kontak-kontak relay yang terstruktur dan menggambarkan aliran arus listrik. Dalam ladder diagram ini terdapat dua buah garis vertikal. Garis vertikal sebelah kiri dihubungkan dengan sumber tegangan positif catu daya aktif sedangkan garis sebelah kanan dengan sumber tegangan negatif catu daya pasif.

A.1 Elemen Program LDR

Diantara dua garis ini dipasang kontak-kontak yang menggambarkan kontrol dari switch, sensor atau output. Satu baris dari diagram disebut dengan satu rung. Input menggunakan simbol “[]” (kontak, normal open) dan “[/]” (negasi kontak, normal closed). Output mempunyai simbol “()” yang terletak paling kanan menempel garis vertikal kanan.

Selama pemrograman setiap simbol yang diberikan adalah alamat PLC sesungguhnya atau merupakan alamat simbolik (misalnya S1, S2, S3, H1).

B. Statement List (STL)

Statement list adalah bahasa pemrograman tingkat tinggi. Semua hubungan logika dan kontrol sequence dapat diprogram dengan menggunakan perintah dalam bahasa ini.

B.1 Elemen Program STL

Perintah-perintah yang digunakan adalah mirip dengan bahasa tingkat tinggi seperti pascal. Terdapat kontrol untuk perulangan, jump dan sebagainya.

Misalnya:

IF	I1.0	Jika input 1.0 aktif
THEN	SET T6	maka aktifkan timer T6.

B.1.1 Struktur Statement List (STL)

Struktur dari statement list secara umum dapat dituliskan sebagai berikut:

PROGRAM

STEP

STATEMENT

BAGIAN KONDISI

BAGIAN PELAKSANA

C. Statement

Statement merupakan pembentuk dasar dari organisasi program. Masing-masing statement terdiri dari bagian kondisi dan bagian pelaksana. Bagian kondisi mengandung satu atau beberapa buah kondisi yang akan diuji (benar atau salah) pada saat program berjalan. Bagian kondisi selalu dimulai dengan kata IF (jika). Jika kondisi bernilai benar maka instruksi yang ditulis pada bagian pelaksana akan dijalankan. Awal dari bagian pelaksana dimulai dengan kata THEN (maka).

Contoh:

IF	I6	(jika input 6 memberikan sinyal
THEN	SET O1	maka nyalakan output 1)

IF		I6	(Jika input 6 memberikan sinyal
	AND	I2	dan input 2 memberikan sinyal)
THEN	RESET	O5	jika ya, matikan output 5,
	SET	O4	nyalakan output 4)

D. Step

Program yang tidak menggunakan instruksi STEP dapat diproses dengan cara paralel. Tetapi STL menyediakan instruksi STEP yang membagi program menjadi bagian-bagian yang lebih kecil.

Dalam sebuah program dapat berisi sampai 256 STEP (0 sampai 255). Setiap STEP dapat diberi label atau tidak, dan hanya dibutuhkan jika setiap STEP tersebut merupakan target dari instruksi JUMP.

Bentuk paling sederhana dari instruksi STEP paling sedikit mengandung satu statement, misalnya:

```
STEP mulai
IF                I1
THEN SET         O2
```

Program akan menunggu pada STEP ini sampai kondisinya benar, yaitu bagian pelaksana akan dilaksanakan dulu baru setelah itu program akan berlanjut ke STEP berikutnya, dalam sebuah STEP dapat berisi beberapa statement:

```
STEP mulai
IF                I2
THEN SET         O5
```

IF		I3
THEN	RESET	O3
	SET	O2

Jika kondisi IF terakhir salah (IF I3) maka program tidak akan berlanjut ke STEP berikutnya dan akan kembali ke statement pertama dalam STEP tersebut (IF I2). Dengan kata lain program akan menunggu sampai kondisi terakhir benar, aturan pelaksanaan STEP:

- a. Jika kondisi dari sebuah statement terpenuhi maka bagian pelaksana akan dijalankan.
- b. Jika kondisi dari statement terakhir dalam suatu STEP terpenuhi maka bagian pelaksana akan dijalankan dan program berlanjut ke STEP berikutnya.
- c. Jika kondisi dari sebuah statement dalam suatu STEP tidak terpenuhi maka program akan berpindah ke statement berikutnya dalam STEP tersebut.
- d. Jika kondisi dari statement terakhir dalam sebuah STEP tidak terpenuhi maka program akan kembali ke statement pertama dari STEP yang sekarang.

E. Instruksi NOP

Instruksi NOP dapat diletakkan pada bagian kondisi atau bagian pelaksana dari sebuah statement. Bila digunakan dalam bagian kondisi, instruksi NOP selalu bernilai benar. Dengan kata lain NOP menyebabkan pelaksanaan tanpa suatu kondisi.

IF	I3	(Jika input 3 aktif)
----	----	----------------------

THEN SET O1 (Jadi output 1 akan selalu aktif pada saat Program pergi ke STEP berikutnya).

Jika digunakan dalam bagian pelaksana pengertian NOP adalah “tidak melakukan sesuatu”. Hal ini sering digunakan pada saat program harus menunggu untuk kondisi tertentu lalu pindah ke STEP berikutnya.

2.1.10 Timer

Banyak dari kontrol industri yang memerlukan pemrograman dengan waktu. Sebagai contoh, silinder 2 akan maju jika silinder 1 telah maju lebih dahulu tetapi hanya setelah lima detik. Hal seperti ini dikenal dengan *switch-on delay*. Penundaan sinyal switch-on pada switching rangkaian power sangat dibutuhkan demi alasan keamanan.

Timer dalam PLC direalisasikan dalam bentuk modul *software* yang didasarkan pada pembangkitan timing secara digital dari generator pulsa mikroprosesor. Lamanya waktu yang diperlukan ditetapkan dalam program kontrol.

A. Komponen Timer

Masing-masing timer dalam bahasa pemrograman STL terdiri dari beberapa elemen:

- a. Timer Status Bit, penulisannya “Tn” yang berfungsi menguji apakah timer sedang aktif atau tidak. Nilai bit berubah menjadi aktif (1) pada saat timer dimulai dengan (SET). Pada saat periode waktu yang

diprogram selesai atau jika timer dihentikan (RESET) status bit berubah menjadi tidak aktif (0).

- b. Timer Preselect, penulisannya “TPn” yang berfungsi sebagai operand 16 bit yang berisi nilai awal untuk sebuah timer n.
- c. Timer Word, penulisannya “TWn” yang berfungsi sebagai operand 16 bit yang secara otomatis memiliki nilai yang sama dengan TP pada saat timer dimulai (SET). Isinya akan secara otomatis dikurangi oleh sistem pada interval yang teratur.

B. Memulai Suatu Timer

Memulai timer hanya digunakan instruksi SET dan menentukan timer yang akan dimulai:

STEP inisialisasi

THEN	LOAD	V100	Nilai 100 = 1 detik
	TO	TP6	Dimasukkan ke TP6

STEP mulai

IF		I1.0	Jika input 1.0 aktif
THEN	SET	T6	maka aktifkan timer 6

Pada saat instruksi SET Tn dijalankan, yang terjadi adalah:

1. Nilai yang tersimpan dalam TPn di-copy ke TWn
2. Tn (Timer Status n) menjadi aktif
3. *Controller* secara otomatis mengurangi nilai yang tersimpan dalam TWn pada *interval* yang teratur, yaitu 10 ms
4. Pada saat nilai yang tersimpan dalam TWn mencapai 0, Tn menjadi tidak aktif.

C. Menghentikan Suatu Timer

Menghentikan suatu timer hanya memerlukan perintah RESET dan menentukan *timer* yang akan dihentikan:

IF		I1.0	Jika input 1.0 aktif
THEN	RESET	T6	Matikan timer 6

Pada saat instruksi RESET Tn dijalankan Timer Status Bit (Tn) menjadi 0 (tidak aktif). Jika timer tersebut sebelumnya sudah tidak aktif, tidak ada pengaruhnya jika kita jalankan perintah RESET Tn tersebut.

2.2. Mikrokontroler MCS-51

Fungsi dari mikrokontroler adalah untuk berkomunikasi dengan peralatan elektronik. Mikrokontroler sangat penting perannya dalam kehidupan sehari-hari seperti rumah tangga, kantor dan industri.

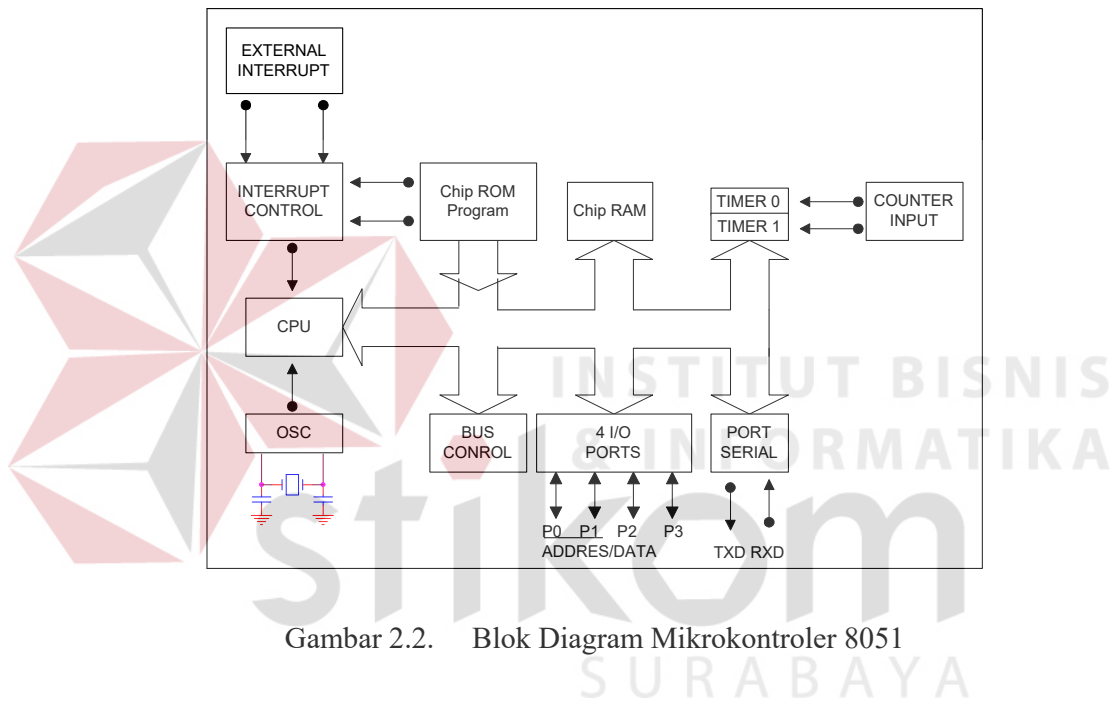
Sebuah mikrokontroler mempunyai sebuah CPU (*Central Processing Unit*) dan terdapat tambahan pemasangan sejumlah RAM (*Random Access Memory*), ROM (*Read Only Memory*) dan I/O (*Input/Output*) port dan sebuah *Timer* yang semuanya terdapat dalam satu *chip*. Dengan kata lain *processor*, RAM, ROM, I/O port dan *Timer* adalah terpasang bersama dalam satu *chip*.

Mikrokontroler 8051 adalah asli dari Intel, beberapa perusahaan juga memproduksi 8051 seperti Atmel, Phillips, AMD, Siemens, Matra dan Dallas Semiconductor.

2.2.1. Mikrokontroler 8051

Pada tahun 1981, perusahaan Intel mengenalkan sebuah mikrokontroler 8 *bit* yang disebut dengan 8051. Mikrokontroler ini mempunyai RAM sebesar 128

byte, ROM sebesar 4 Kbyte, dua *timer*, satu *serial port*, dan empat *port* (masing-masing sebesar 8 *bit*) semuanya dalam satu *chip*. 8051 adalah sebuah prosesor 8 *bit*, artinya bahwa *CPU* dapat bekerja hanya pada data sebesar 8 *bit* pada waktu yang sama. Data yang lebih besar dari 8 *bit* harus dipecah menjadi 8 *bit* setelah itu diproses oleh *CPU*. 8051 mempunyai 4 *I/O Port* masing-masing sebesar 8 *bit*, lihat Gambar 2.2.



Gambar 2.2. Blok Diagram Mikrokontroler 8051

8051 menjadi sangat terkenal setelah Intel mengizinkan perusahaan lain untuk membuat dan memasarkan beberapa jenis 8051 dan mengikuti syarat program yang kompatibel dengan 8051. Disamping itu terdapat bermacam-macam versi dari 8051 dengan kecepatan yang berbeda dan jumlah dari ROM dipasarkan oleh lebih dari 12 perusahaan.

Tabel 2.1 Perbandingan dari anggota keluarga 8051

<i>Feature</i>	8051	8052	8031
ROM	4K	8K	0K

RAM	128	256	128
Timer	2	3	2
I/O Pins	32	32	32
Serial Port	1	1	1
Interrupt Source	6	8	6

8051 adalah anggota yang asli dari keluarga 8051, Intel mengacu kepada mikrokontroler ini sebagai MCS-51, Tabel 2.1 menunjukkan ciri khusus dari mikrokontroler 8051. Ada dua anggota lain dari keluarga mikrokontroler 8051 yaitu 8052 dan 8031.

2.2.2. Mikrokontroler 8052

Mikrokontroler 8052 adalah anggota lain dari keluarga 8051. Mikrokontroler 8052 mempunyai semua standar dari 8051 serta terdapat tambahan RAM sebesar 128 *bytes* dan sebuah tambahan *timer*, sehingga 8052 mempunyai RAM sebesar 256 *bytes* dan 3 *timer*. Selain itu juga mempunyai ROM sebesar 8 *Kbytes* untuk program yang terdapat di dalam *chip*, seperti pada tabel 2.1.

2.2.3. Mikrokontroler 8031

Anggota lain dari keluarga 8051 adalah mikrokontroler 8031. *Chip* ini sering dikenal sebagai 8051 tanpa ROM karena tidak mempunyai ROM di dalam *chipnya*, seperti tabel 2.1. Untuk menggunakan chip ini harus menambah *eksternal* ROM. Eksternal ROM ini berisi program yang akan diambil dan dieksekusi oleh 8031. ROM yang berisi program untuk mikrokontroler 8031 dapat sebesar 64 *Kbyte*. Dalam proses penambahan ROM *eksternal* untuk 8031, akan kehilangan 2 port. Yang tersisa hanya 2 port (dari 4 port yang tersedia) untuk I/O. Untuk memecahkan masalah ini, perlu ditambahkan eksternal I/O untuk 8031.

Sehingga diperlukan interfacing 8031 dengan memori dan I/O port seperti menggunakan IC (Integrated Circuit) 8255.

2.2.4. Mikrokontroler 8751

Mikrokontroler 8751 mempunyai 4 Kbytes UV (Ultra Violet) EPROM di dalam *chipnya*. Menggunakan *chip* ini untuk pengembangan diperlukan PROM *burner* dan *eraser* UV-EPROM sebelum mikrokontroler 8751 diprogram lagi. Pada kenyataannya penghapusan program ROM dari IC 8751 membutuhkan waktu sekitar 20 menit.

2.2.5. Atmel AT89C51

Keluarga mikrokontroler 8051 yang terkenal ini mempunyai ROM yang di dalam *IC* dalam bentuk *flash memory*. Ini ideal untuk perkembangan yang sangat cepat sejak *flash memory* dapat menghapus dalam hitungan detik dibandingkan 8751 yang memerlukan 20 menit atau lebih untuk menghapus. Untuk alasan ini, AT89C51 melengkapi 8751 untuk menghilangkan waktu tunggu yang lama untuk menghapus *IC*. Dengan cara ini kita dapat mengembangkan kecepatan menjadi lebih tinggi. Dalam menggunakan AT89C51, untuk mengembangkan sistem dasar mikrokontroler memerlukan sebuah *ROM burner* yang *support* dengan *flash memory*, maka dengan ini *ROM eraser* tidak diperlukan. Untuk memprogram ulang *flash memory*, isi yang ada di dalamnya harus dihapus lebih dahulu. Penghapusan sebuah *flash* dilakukan oleh *ROM burner* dan hal ini menunjukkan mengapa *eraser* yang terpisah tidak diperlukan. *Atmel* versi AT89C51 juga dapat diprogram melalui serial *COM port* dari sebuah IBM PC sehingga *ROM burner* tidak diperlukan lagi. Selain itu kapasitas dari

ROM pada Atmel berbeda tergantung pada jenisnya, seperti yang terdapat pada tabel dibawah ini:

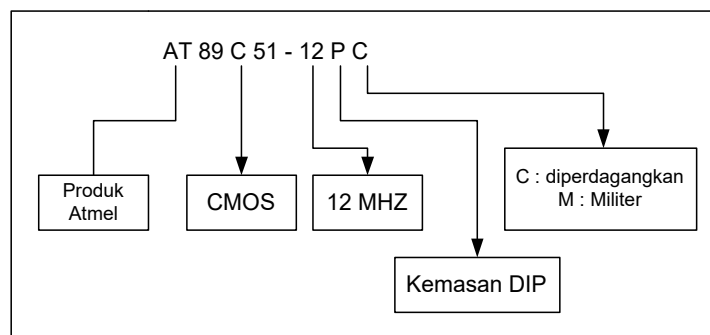
Table 2.2 Macam-macam 8051 dari ATMEL

Part Number	ROM	RAM	I/O	TIMER	INTR	VCC
AT89C51	4K	128	32	2	6	5V
AT89LV51	4K	128	32	2	6	3V
AT89C1052	1K	64	15	1	3	3V
AT89C2051	2K	128	15	2	6	3V
AT89C52	8K	128	32	3	8	5V
AT89LV52	8K	128	32	3	8	5V

Table 2.3 Macam kecepatan 8051 dari Atmel

Part Number	Speed	Pins	Kemasan	Digunakan
AT89C51-12PC	12 MHz	40	DIP plastic	Comercial
AT89C51-16PC	16 MHz	40	DIP plastic	Comercial
AT89C51-20PC	20 MHz	40	DIP plastic	Comercial

Ada bermacam-macam versi kecepatan dan kemasan dalam sebuah produk seperti pada gambar 2.3. Untuk contoh, AT89C51-12 PC dimana “C” sebelum 51 adalah untuk CMOS, yang mempunyai konsumsi daya yang kecil, “12” indikasi 12 MHz, “P” adalah untuk kemasan plastik DIP, dan “C” untuk diperdagangkan, sedangkan “M” untuk keperluan militer. Yang sering digunakan oleh mahasiswa untuk proyek adalah AT89C51-12 PC.



Gambar 2.3 Program AT89C51

2.2.6. DS5000 Dallas Semiconductor

Versi populer yang lain dari 8051 adalah IC DS5000 dari *dallas semiconductor*. ROM yang terdapat dalam chip DS5000 dalam bentuk NV-RAM. Kemampuan membaca dan menulis NV-RAM memperbolehkan mengisi program ke dalam ROM ketika NV-RAM berada dalam sebuah sistem. Ini dapat dilakukan melalui serial COM port IBM PC. Kemampuan NV-RAM untuk mengubah isi ROM beberapa bytes setiap waktu. Dibandingkan dengan UV-EPROM dan flash memory yang mana isi ROM harus dihapus dahulu sebelum diprogram ulang.

Table 2.4 Dallas Semiconductor's Soft Mikrokontroler

Part Number	ROM	RAM	I/O	Timers	Interrupt	Vcc	Packaging
DS5000-8	8K	128	32	2	6	5V	40
DS5000-32	32K	128	32	2	6	5V	40
DS5000T-8	8K	128	32	2	6	5V	40
DS5000T-8	32K	128	32	2	6	5V	40

Tabel 2.5 Macam-macam kecepatan Dallas Semiconductor

Part Number	NV-RAM	Kecepatan
DS5000-8-8	8K	8 MHz
DS5000-8-12	8K	12 MHz
DS5000-32-8	32K	8 MHz
DS5000T-32-8	32K	8 MHz (dengan RTC)
DS5000-32-12	32K	12 MHz
DS5000-8-12	8K	12 MHz (dengan RTC)

2.2.7. Phillips

Keluarga 8051 yang lain adalah perusahaan Phillip. Beberapa produk ini memiliki ciri khusus seperti adanya *A-to-D converter*, *D-to-A converter*, *I/O ports* dan *OTP* serta *flash* (Mazidi, 2000 : 28).

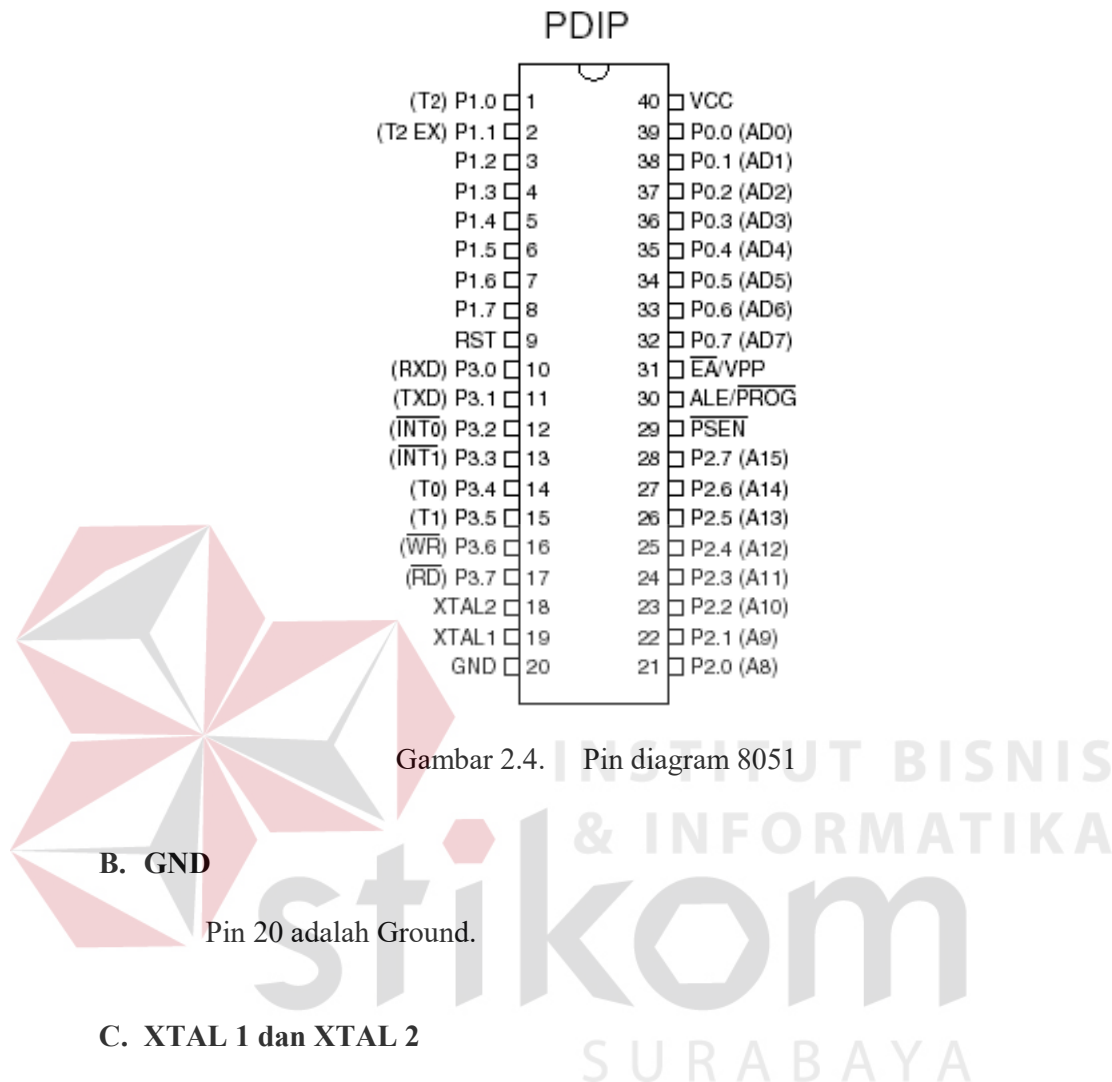
2.2.8. I/O Port dan Programming MCS51

Keluarga 8051 anggota (8751, 89C51, DS5000) mempunyai kemasan yang berbeda, seperti DIP (*dual in-line package*), QFP (*quad flat package*), dan LLC (*leadless chip carrier*). Keluarga 8051 mempunyai 40 pin, dimana masing-masing pin mempunyai bermacam-macam fungsi seperti I/O, RD, WR, alamat data, dan *interrupt*. Selain 40 pin, keluarga 8051 juga mengeluarkan 20 pin dengan fungsi yang hampir sama.

Gambar 2.4 adalah mikrokontroler yang mempunyai 40 pin, 32 pins digunakan sebagai *port* yang terdiri dari 4 *port* yaitu P0, P1, P2 dan P3 masing-masing *port* mempunyai 8 pin. Sisa pin didesain sebagai Vcc, GND, XTAL1, XTAL2, RST, \overline{EA} , \overline{PSEN} dan ALE. Dari 8 pin, enam diantaranya (Vcc, GND, XTAL1, XTAL2, RST dan \overline{EA}) digunakan oleh keluarga 8051 dan 8052. Dan dua pins yang lain adalah \overline{PSEN} dan ALE, biasanya digunakan 8031. Berikut ini pembahasan tentang fungsi masing-masing pin :

A. Vcc

Pin 40 digunakan sebagai catu daya dengan tegangan sumber sebesar 5V



B. GND

Pin 20 adalah Ground.

C. XTAL 1 dan XTAL 2

Di dalam IC Mikrokontroler 8051 terdapat osilator, tetapi memerlukan sebuah *eksternal clock* untuk menjalankannya. Sebuah kristal yang dihubungkan ke XTAL1 (pin 19) dan XTAL2 (pins 18) salah satu kakinya dihubungkan ke kapasitor sebesar 30 pF sedangkan kaki kapasitor lainnya dihubungkan ke *ground*.

Keluarga 8051 mempunyai kecepatan yang bervariasi, kristal yang digunakan harus sama atau kurang dari kecepatan yang dimiliki oleh *chip* tersebut. Misalkan mikrokontroler mempunyai kecepatan maksimum sebesar 12 MHz, maka kristal yang digunakan harus sama dengan 12 MHz atau kurang.

D. RST

Pin 9 adalah reset dengan kondisi *active high*. Kondisi ini biasanya disebut sebagai *power-on-reset*. Jika terjadi *power-on-reset* semua aplikasi berhenti dan semua nilai yang terdapat pada register semuanya hilang. Tabel 2.6 menunjukkan sebagian nilai *register* apabila terjadi *reset*.

Tabel 2.6 Kondisi *reset*

<i>Register</i>	Nilai <i>Reset</i>
PC	0000
ACC	0000
B	0000
PSW	0000
DPTR	0000

PC (*program counter*) bernilai 0 jika terjadi *reset*, lalu CPU akan mengambil program pertama dari ROM pada lokasi 0000H. Ini artinya jika meletakkan program harus terdapat di alamat ROM yang ke 0000H. Untuk *input reset* yang efektif, reset harus mempunyai durasi 2 *machine cycle*.

E. \overline{EA}

Anggota keluarga 8051 yang mempunyai ROM *internal* adalah 8751, 89C51 atau DS5000. Untuk menyimpan program ke ROM *internal*, \overline{EA} harus dihubungkan ke Vcc.

Lain dengan 8031 dan 8032, dimana di dalam chip tidak terdapat ROM *internal* sehingga untuk menyimpan program menggunakan ROM eksternal. Oleh karena itu \overline{EA} dihubungkan ke GND.

F. $\overline{\text{PSEN}}$

Untuk penyimpanan program di ROM *eksternal*, $\overline{\text{PSEN}}$ dihubungkan dengan $\overline{\text{OE}}$ pada ROM *eksternal* untuk mengambil satu instruksi. $\overline{\text{PSEN}}$ ini digunakan di mikrokontroler 8031 dan keluarga 8051 yang menggunakan ROM *eksternal*.

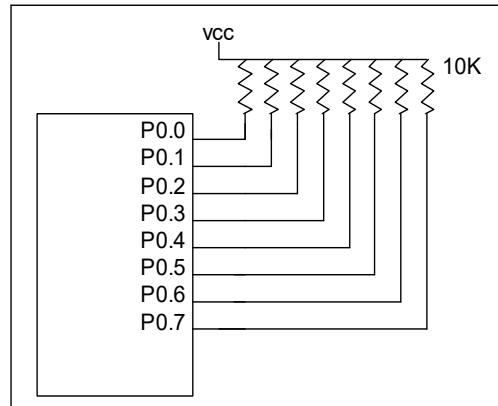
G. ALE

ALE (*alamat latch enable*) adalah suatu *output* pin yang mempunyai *active high*. Ketika 8031 menggunakan memori *eksternal*, port 0 menyediakan alamat dan data. ALE digunakan untuk *demultiplexing* alamat dan data dengan menghubungkan ALE ke G (74ls573).

H. Port 0

Port 0 sebanyak 8 pins (pin 32-39) digunakan sebagai *input* maupun *output*, dimana masing-masing pin dihubungkan ke tahanan *pull up* sebesar 10K, seperti gambar 2.5. Pada kenyataannya P0 adalah *open drain*, tidak seperti P1, P2 dan P3. *Open drain* digunakan untuk MOS *chip* sama seperti *open collector* yang digunakan oleh TTL *chip*.

Selain sebagai *output*, port 0 juga dapat digunakan untuk *input*. Sebelum data dibaca oleh port 0 maka pada semua pin data ditulis dengan nilai 1 (*high*). Setelah itu data dapat dibaca oleh port 0.



Gambar 2.5 Tahanan *pull-up*

I. Port 1

Port 1 sebanyak 8 pin (pin 1-8). digunakan sebagai *input* maupun *output* pada *port* ini tidak memerlukan tahanan *pull up* karena sudah tersedia *internal pull up*.

Selain sebagai *output port 0* juga dapat digunakan untuk *input*. Sebelum data dibaca oleh *port 1* maka pada semua pin data ditulis dengan nilai 1(*high*) setelah itu data dapat dibaca oleh *port 1*.

J. Port 2

Port 2 sebanyak 8 pin (pin 1-8) digunakan sebagai *input* maupun *output*, dan tidak memerlukan tahanan *pull up* karena sudah tersedia *internal pull up*.

Selain sebagai *output, port 0* juga dapat digunakan untuk *input*. Sebelum data dibaca oleh *port 2*, maka pada semua pin data ditulis dengan nilai 1(*high*) setelah itu data dapat dibaca oleh *port 2*.

K. Port 3

Port 3 sebanyak 8 pin (pin 10-17) digunakan sebagai *input* atau *output* dan tidak memerlukan tahanan *pull-up*, sama dengan P1 dan P2. Port 3 mempunyai fungsi tambahan seperti *interrupts*. Tabel 2.7 adalah informasi mengenai fungsi P3 yang digunakan oleh kedua *chip* 8051 dan 8031.

P3.0 dan P3.1 digunakan untuk sinyal komunikasi serial RxD dan TxD. P3.2 dan P3.3 digunakan untuk *interrupt* eksternal. P3.4 dan P3.5 digunakan untuk timer 0 dan 1. Pada akhirnya, P3.6 dan P3.7 digunakan untuk sinyal \overline{WR} dan \overline{RD} (Mazidi, 2000 : 90).

Table 2.6 Fungsi port 3

P3 Bit	Fungsi	Pin
P3.0	RxD	10
P3.1	TxD	11
P3.2	$\overline{INT0}$	12
P3.3	$\overline{INT1}$	13
P3.4	$\overline{T0}$	14
P3.5	$\overline{T1}$	15
P3.6	\overline{WR}	16
P3.7	\overline{RD}	17

2.2.9 Memori Eksternal

Arsitektur MCS-51 menyediakan kapasitas program memori eksternal dan data memori eksternal sebesar 64K. Tambahan ROM dan RAM dapat diberikan jika diperlukan begitu pula *interface* untuk menambahkan I/O.

Ketika memori *eksternal* digunakan *port 0* tidak dapat digunakan sebagai I/O. *Port 0* menjadi *multiplexed* alamat (A0-A7) dan data (D0-D7) *bus* dan *port 2* digunakan untuk *alamat bus* dengan *byte* tinggi (A8-A15).

A. Pengaksesan Memory Program eksternal.

Program memori eksternal adalah memori yang hanya dapat dibaca dan diaktifkan oleh sinyal $\overline{\text{PSEN}}$. Ketika program eksternal digunakan, kedua *port 0* dan *port 2* tidak dapat digunakan sebagai *port* pada umumnya.

Untuk mengakses memori *eksternal* memerlukan $\overline{\text{PSEN}}$ dan ALE. Dimana $\overline{\text{PSEN}}$ dihubungkan ke $\overline{\text{OE}}$ untuk mengambil program dalam ROM. Sedangkan ALE digunakan untuk *demultiplexing* alamat dan data yang menghubungkan ALE ke G (74ls573).

B. Pengaksesan Memori Data eksternal

Data memori eksternal adalah memori yang dapat dibaca atau ditulis oleh $\overline{\text{RD}}$ dan $\overline{\text{WR}}$. Instruksi yang dapat mengakses memori *eksternal* adalah *movx*, sedangkan register untuk mengakses memori eksternal adalah DPTR (16 bit data) dan R0 atau R1 untuk alamat *register*.

Untuk mengakses RAM agar dapat dibaca atau ditulis oleh mikrokontroler, menghubungkan $\overline{\text{RD}}$ pada mikrokontroler dihubungkan dengan $\overline{\text{OE}}$ yang ada di RAM untuk membaca data sedangkan $\overline{\text{WR}}$ pada mikrokontroler dihubungkan dengan $\overline{\text{W}}$ pada RAM. Untuk menghubungkan alamat dan data *bus* sama dengan ROM.

Ketika sebuah instruksi *MOVX @DPTR,A* dijalankan maka pin $\overline{\text{WR}}$ menjadi berlogika 0 (*low*) sedangkan pin $\overline{\text{RD}}$ berlogika 1 (*high*), proses diatas adalah proses menulis (I. Scott MacKensi,1999).

2.2.10 Program Counter di MCS-51

Register terpenting lain pada MCS-51 adalah PC yang nilainya berdasarkan instruksi yang di akses. Program *counter* di MCS-51 mempunyai lebar 16 bit data, sehingga mikrokontroler dapat mengakses dari alamat 0000H sampai FFFFH.

Pertama kali mikrokontroler mendapatkan daya nilai dari program *counter* adalah 0000H, sehingga instruksi yang diakses oleh mikrokontroler berada pada alamat 0000H. Ini artinya untuk penulisan didalam ROM harus diletakkan pada alamat 0000H.

A. Penempatan program dalam ROM

Untuk mendapatkan pengertian yang lebih baik dari aturan program *counter* dalam mengambil dan mengeksekusi sebuah program dijelaskan dalam bab ini. Pertama harus mengerti penempatan program pada ROM pada MCS-51 seperti tabel di bawah ini.

Tabel 2.8 Eksekusi Program

ROM alamat	Bahasa mesin	Bahasa assembly
0000	7D25	MOV R5,#10h
0002	7400	MOV A,#01H
0004	2D	ADD A,R5

Setelah program dimasukkan kedalam ROM, opcode dan operand ditempatkan di memori ROM yang lokasinya di alamat 0000H.

Tabel 2.9 Isi ROM

Alamat	Code
0000	7D
0001	25
0002	74

0003	01
0004	2D

Pada alamat 0000H berisi 7DH *MOV* R5 dan alamat 0001H berisi *operand* bernilai 25H, dimana program ini untuk memindahkan *operand* 25H kedalam R5. Alamat 0002H berisi 74 dan alamat 0003H berisi 01H dengan instruksi *MOV* A,01H. Sedangkan untuk proses penjumlahan *ADD* A,R5 berada pada alamat 0004H.

B. Eksekusi sebuah program

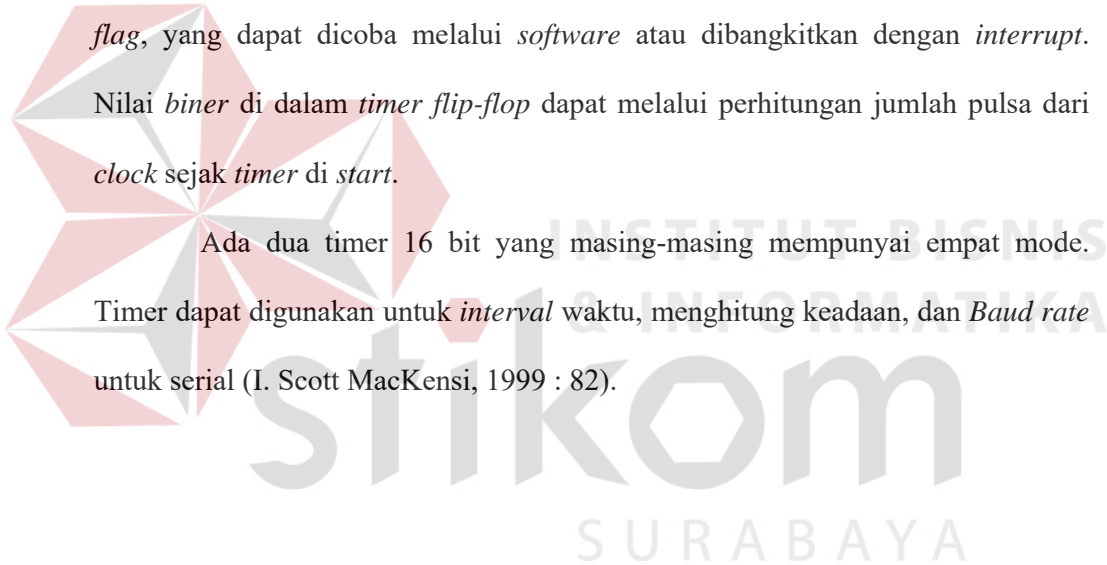
Asumsi tentang program yang di masukkan kedalam ROM MCS-51, ikuti langkah dibawah ini yang menggambarkan aksi dari MCS-51 setelah power dihidupkan :

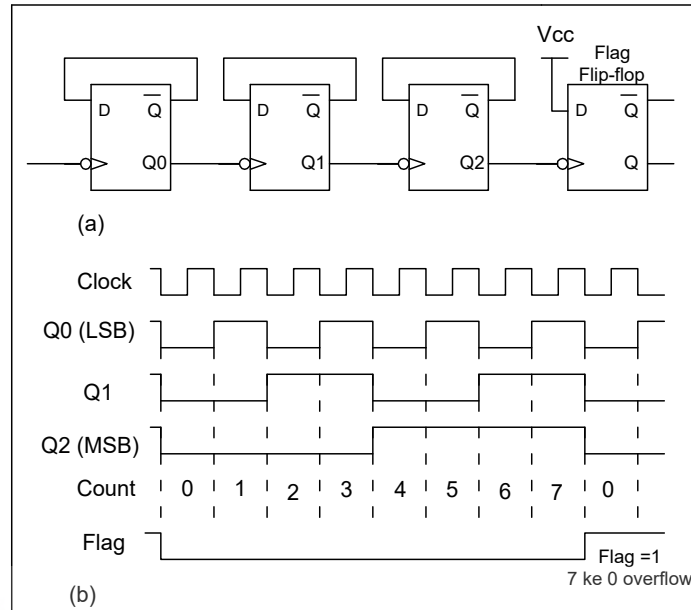
- a) Ketika daya MCS-51 dihidupkan, program *counter* mempunyai nilai 0000H dan mengambil program kedalam alamat memori 0000H. Alamat 0000H yang diambil oleh CPU adalah 7D, setelah instruksi tersebut dijalankan, CPU mengambil *operand* 25 untuk dimasukkan kedalam R5.
- b) Setelah mengksekusi *opcode* 74H, nilai *operand* 01H dipindahkan ke dalam R7 lalu program *counter* ditambah menjadi 0003H.
- c) Sekarang PC = 0004H, instruksi selanjutnya adalah “ADD A,R5”. Instruksi ini mempunyai 1 byte instruksi, sehingga PC = 00004H (Mazidi, 2000 : 46).

2.3.11 Operasi *Timer*

Timer adalah suatu rangkaian *flip-flop* pembagi dua yang menerima sinyal *input* sebagai *clock*. *Clock* dihubungkan ke *flip-flop* yang pertama, yang membagi frekuensi *clock* menjadi 2. *Output* dari *flip-flop* yang pertama dihubungkan ke *clock flip-flop* yang ke dua, dimana yang membagi frekuensi *clock* menjadi 2 dan seterusnya, seperti pada gambar 2.5. Karena masing-masing *flip-flop* dibagi dua, sebuah *timer* dengan langkah ke n sama dengan 2^n frekuensi *clock*. *Output flip-flop* yang terakhir adalah sebuah *timer overflow flip-flop* atau *flag*, yang dapat dicoba melalui *software* atau dibangkitkan dengan *interrupt*. Nilai *biner* di dalam *timer flip-flop* dapat melalui perhitungan jumlah pulsa dari *clock* sejak *timer* di *start*.

Ada dua timer 16 bit yang masing-masing mempunyai empat mode. Timer dapat digunakan untuk *interval* waktu, menghitung keadaan, dan *Baud rate* untuk serial (I. Scott MacKensi, 1999 : 82).





Gambar 2.6

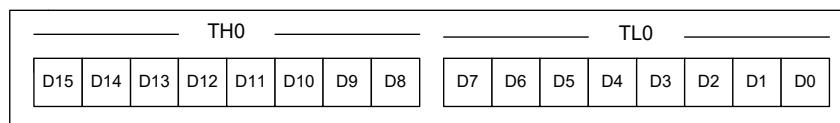
3 bit timer (a) Schematics (b) Timing Diagram

A. Register Timer

MCS-51 mempunyai dua *timer* yaitu *timer 0* dan *timer 1*, kedua *timer* tersebut mempunyai panjang 16 *bit*. MCS-51 mempunyai arsitektur 8 bit, masing-masing *timer* mengakses dua bagian dari *register byte* rendah dan *byte* tinggi.

A.1. Register Timer 0

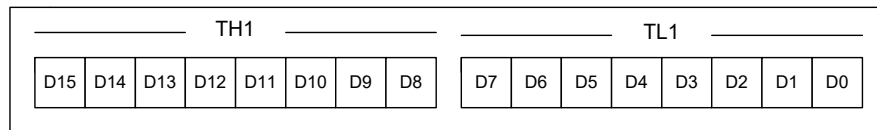
Timer 0 mengakses *byte* rendah dan *byte* tinggi dari *register 16 bit*. *Byte register* rendah disebut sebagai TL0 (*timer 0 low byte*) dan *Byte register* tinggi disebut sebagai TH0 (*timer 0 high byte*).



Gambar 2.7 Register Timer 0

A.2. Register *Timer 1*

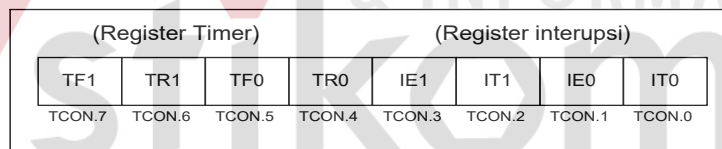
Timer 1 mengakses *byte* rendah dan *byte* tinggi dari *register* 16 bit. *Byte register* rendah disebut sebagai TL1 (*timer 1 low byte*) dan *byte register* tinggi disebut sebagai TH1 (*timer 1 high byte*)(Mazidi, 2000 : 158).



Gambar 2.8 Register *Timer 1*

A.3. *Timer* atau *Counter Control Register*

Register yang berfungsinya menghubungkan dengan *timer* ada empat yaitu TCON4, TCON5, TCON6 dan TCON7. *Register* ini bersifat *Addressable bit* sehingga *bit* TF1 disebut sebagai TCON.7, TR1 disebut sebagai TCON.6 dan seterusnya.



Gambar 2.9 Register TCON

TCON.7 atau TF1 : *Timer 1 overflow* flag diset jika *timer overflow*. *Bit* ini dapat dibersihkan oleh *software* .

TCON.6 atau TR1: 1 : *Timer 1* aktif
0 : *Timer 1 non-aktif*

TCON.5 atau TF0: *Timer 0 overflow* flag diset jika *timer overflow*. *Bit* ini dapat dibersihkan oleh *software* .

TCON.4 atau TR0: 1 : *Timer 0* aktif

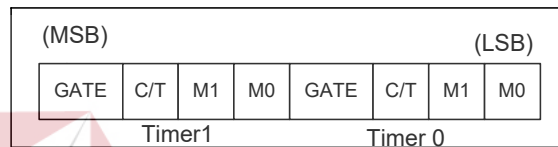
0 : *Timer 0 non-aktif*

TCON.3 sampai dengan TCON.0 dibahas pada bagian *interrupt*

(Paulus Andi Nalwan, 2003 : 33).

A.4. Register TMOD (*timer mode*)

Dua *timer 0* dan *1* yang menggunakan *register TMOD*, untuk *setting* macam mode operasi *timer*. TMOD adalah sebuah *timer 8 bit register* dimana 4 bit rendah untuk *timer 0* dan 4 bit yang tinggi untuk *timer 1*.



Gambar 2.10 TMOD Register

A.4.1. C/T (clock atau counter)

Bit ini dalam *register TMOD* digunakan untuk memilih *timer* atau *counter*. Jika bit C/T = 0 yang digunakan adalah *timer*. Sumber *clock* dari *timer* menggunakan frekuensi dari kristal.

Meskipun macam sistem MCS-51 mempunyai kristal dengan frekuensi dari 10 MHz sampai 40 MHz, yang sering digunakan untuk proyek MCS-51 adalah kristal 11,0592 MHz (Mazidi, 2000 : 160)

A.4.2. Mengaktifkan dan Non aktifkan Timer

Metode untuk mengaktifkan dan non aktifkan dari sebuah *timer* ada dua cara yaitu secara *software* dan secara *hardware*. Untuk menggunakan metode *software*

GATE=0, sedangkan controlnya terletak pada bit TRx saja, dimana TR0 untuk timer 0 dan TR1 untuk timer 1.

Selain menggunakan metode diatas untuk mengontrol *timer* dapat menggunakan metode *hardware* yaitu dengan menggunakan $\overline{\text{INTx}}$ (*interrupt*). Terlebih lebih dahulu seting GATE=1 setelah itu *timer* dapat dikontrol melalui $\overline{\text{INTx}}$. Dengan GATE = 1 dan TR0 = 1, ketika $\overline{\text{INT0}} = 1$ maka timer 0 aktif, apabila $\overline{\text{INT0}} = 1$ maka timer 0 *non-aktif* (I. Scott MacKensi, 1999 : 88).

A.4.3. M1 dan M0 (mode operasi timer)

M0 dan M1 digunakan untuk memilih mode *timer*. Ada 4 mode dalam *timer* yaitu mode 0 adalah *timer* 13 bit, mode 1 adalah *timer* 16 bit, mode 2 adalah *timer* 8 bit *auto reload* dan mode 3 adalah *timer* mode *split*. Untuk mencari nilai TLx dan THx dapat menggunakan rumus dibawah ini (Mazidi, 2000 : 159):

- Jumlah bit ≤ 8 bit :

$$T = (\text{jumlah bit} - \text{TLx} + 1) \times 1,085 \quad (2.1)$$

Jumlah bit maksimal bernilai 255.

- 9 bit \geq Jumlah bit ≤ 16 bit

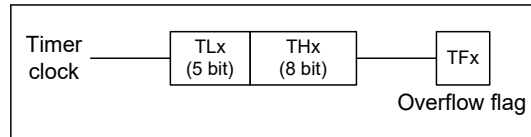
$$T = (\text{jumlah bit} - \text{THxTLx} + 1) \times 1,085 \quad (2.2)$$

Jumlah bit maksimal bernilai 65535.

a) Timer Mode 0

Timer mode 0 adalah sebuah *timer* 13 bit yang disediakan oleh MCS-51. untuk *byte* tinggi menggunakan 8 bit MSB dari THx dan untuk *byte* rendah menggunakan 5 bit yang LSB dari TLx. Nilai yang diperbolehkan *timer* 13 bit

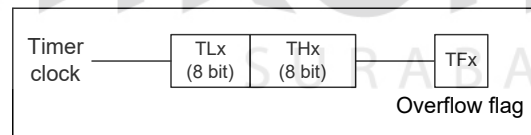
adalah 0000H sampai dengan 1FFFH. Jika *timer* telah mencapai maksimal 1FFFH dan berubah menjadi 0000H maka flag *overflow* di set ($TF_x = 1$).



Gambar 2.11 Mode 0

b) Timer Mode 1

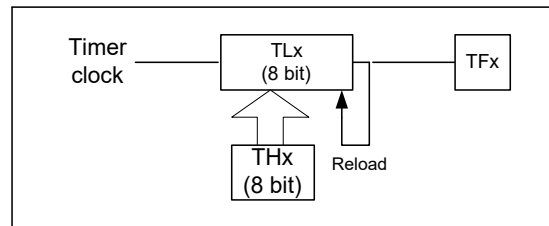
Mode 1 adalah *timer* 16 bit yang disediakan oleh MCS-51. Timer ini menggunakan *register byte* rendah dan *byte* tinggi (THx dan TLx). Jika menerima pulsa *clock*, maka *timer* menghitung naik dari : 0000H,0001H,0002H dan lain-lain. Maksimal nilai yang diperbolehkan oleh timer 16 bit adalah 0000H sampai dengan FFFFH. Overflow terjadi apabila ada perubahan dari FFFFH ke 0000H (I. Scott MacKensi, 1999 : 86)



Gambar 2.12 Mode 1

c) Timer Mode 2

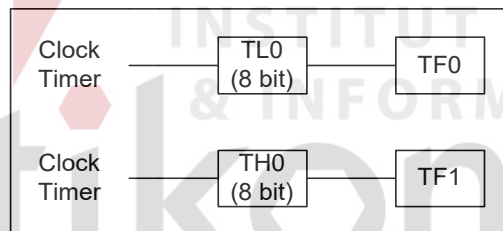
Mode 2 adalah 8 bit *auto reload*. *Register byte* tinggi THx digunakan untuk mengisi nilai kedalamnya, lalu dikirimkan secara otomatis ke dalam TLx. Setelah itu TLx menghitung sampai maksimal FFH. *Overflow* terjadi apabila ada perubahan dari FFH ke 00H (I. Scott MacKensi, 1999 : 86).



Gambar 2.13 Mode 2

d) Timer Mode 3

Mode 3 adalah *mode timer split*. Timer 0 pada mode 3 terpisah menjadi 2 timer 8 bit. TL0 *flag timer overflow* menggunakan TF0, sedangkan TH0 *flag timer overflow* menggunakan TF1. Pada mode 3 *timer 1* tidak aktif, karena TF1 sudah digunakan oleh TH0 (I. Scott MacKensi, 1999 : 86).



Gambar 2.14 Mode 3

2.3 Komunikasi Serial

Komunikasi data berarti pengiriman data antara dua komputer, antara sebuah komputer dengan terminal, atau antara terminal dengan terminal yang lain.

Komunikasi data dapat dilakukan dengan dua cara: *paralel* dan *serial*. Dalam transfer data *paralel*, sering 8 atau lebih jalur (konduktor kabel) digunakan untuk mentransfer data ke suatu device yang berjarak hanya beberapa kaki. Contoh transfer *paralel* adalah printer dan *hard disk* yang menggunakan kabel dengan banyak jalur. Meskipun dalam kasus-kasus seperti ini banyak data bisa

ditransfer dalam waktu singkat dengan menggunakan banyak kabel yang disusun paralel, tetapi jaraknya tidak bisa jauh dan biayanya relatif lebih mahal. Untuk mentransfer data ke suatu device yang terletak sejauh beberapa meter, digunakan metode serial. Dalam komunikasi serial, data dikirim satu bit dalam suatu waktu, berbeda dengan komunikasi paralel, dalam mana data dikirim satu byte atau lebih dalam suatu waktu. Kelebihan metode serial ini adalah selain ia dapat digunakan dalam jarak yang jauh, ia juga memerlukan biaya yang lebih murah jika dibandingkan dengan metode paralel.

Komunikasi data serial menggunakan dua metode, asinkronous dan sinkronous. Metode sinkronous mentransfer suatu blok data (karakter) pada suatu waktu sedangkan asinkronous mentransfer suatu byte tunggal pada suatu waktu. Ada kemungkinan untuk membuat software untuk digunakan dengan metode di atas, tetapi programnya bisa membosankan dan panjang. Karena itu, ada chip-chip IC khusus yang dibuat oleh banyak pabrik untuk komunikasi data serial. Chip-chip ini secara umum dikenal sebagai UART (*Universal Asynchronous Receiver-Transmitter*) dan USART (*universal synchronous-asynchronous receiver-transmitter*). Port COM pada IBM PC menggunakan UART 8250.

2.3.1 Transmisi Half dan Full-duplex

Jika data bisa ditransmisikan dan diterima, itu disebut transmisi duplex. Berbeda dengan transmisi simplex seperti printer, yang mana computer hanya mengirim data. Transmisi duplex bisa half atau full duplex, tergantung pada transfer datanya bisa sekaligus atau tidak. Jika data dapat ditransmisikan satu jalan pada suatu waktu, ini disebut half duplex. Dan bila data bisa melewati dua jalan pada waktu yang sama, itulah full duplex. Tentu saja, full duplex

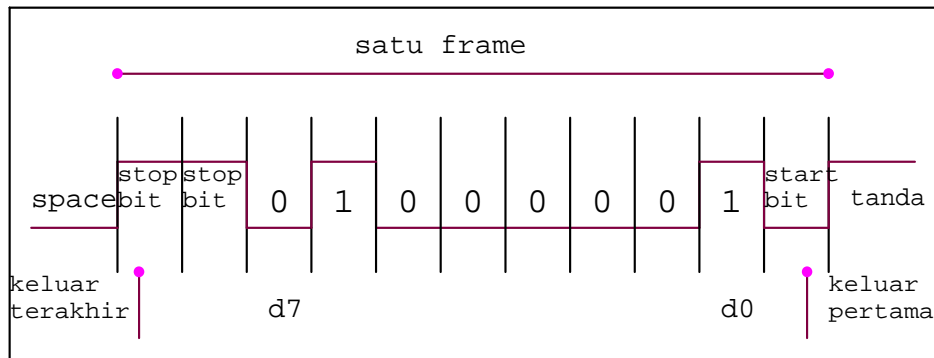
membutuhkan dua kabel konduktor (selain ground), satu untuk transmisi dan satu untuk penerimaan, agar bisa mentransfer dan menerima data secara sekaligus.

2.3.2 Komunikasi serial asinkronous dan data framing

Data yang masuk pada akhir penerimaan dari jalur data dalam transfer data serial semuanya dalam program 0 dan 1, sangat sulit untuk memahami data kecuali pengirim dan penerima menyepakati seperangkat peraturan, sebuah protokol, tentang bagaimana data dipaketkan dan berapa banyak bit *constitute* sebuah karakter, dan kapan data mulai dan berakhir.

2.3.3 Start dan stop bit

Komunikasi data serial asinkronous digunakan secara luas untuk transmisi berorientasi-karakter, dan transfer data berorientasi-blok yang menggunakan metode sinkronous. Dalam metode asinkronous, setiap karakter diletakkan antara start dan stop bit, ini disebut framing. Dalam data framing untuk komunikasi asinkronous, data, seperti karakter-karakter ASCII, dipaketkan di antara sebuah start bit dan sebuah stop bit. Start bit selalu satu bit tetapi stop bit bisa satu atau dua bit. Start bit selalu '0' (low) dan stop bit adalah '1' (high). Sebagai contoh, lihat gambar 2.14 dimana karakter ASCII "A", biner 0100 0001, diframe di antara start bit dan 2 stop bit. Perhatikan bahwa LSB-nya dikirim dulu.



Gambar 2.15 Framing ASCII “A” (41H)

Dalam gambar diatas, ketika tidak ada sinyal transfer yang bernilai 1 (high), yang disebut sebagai *mark* dan 0 (low) disebut sebagai *space*. Perhatikan bahwa transmisi dimulai dengan start bit yang diikuti oleh D0 (LSB), kemudian sisa bit-bit sampai MSB (D7), dan akhirnya, 2 stop bit yang menunjukkan akhir dari karakter “A”.

Dalam komunikasi serial asinkronous, chip-chip peripheral dan modem bisa diprogram untuk data selebar 5, 6, 7, atau 8 bit. Ini sebagai tambahan dari jumlah stop bit, 1 atau 2. Sementara dalam sistem yang lebih lama karakter-karakter ASCII adalah 7-bit tapi dengan adanya karakter ASCII extended, dibutuhkan 8-bit untuk setiap karakter. Keyboard non-ASCII kecil menggunakan karakter-karakter 5 dan 6-bit.

Dalam beberapa sistem lama, disebabkan kelambatan peralatan mekanik yang menerima. 2 stop bit digunakan untuk memberikan peralatan tersebut cukup waktu untuk mengorganisasi dirinya sendiri sebelum transmisi dari byte berikutnya. Tetapi pada PC modern penggunaan 1 stop bit adalah umum. Dengan asumsi bahwa kita mentransfer file text dari karakter ASCII menggunakan 2 stop bit sehingga total untuk tiap karakter 11 bit yaitu 8 bit untuk

program ASCII-nya, dan 1 dan 2 bit masing-masing untuk start dan stop bit. Oleh karena itu, untuk setiap karakter 8-bit ada 3 bit ekstra, atau lebih dari 25%.

Dalam beberapa sistem untuk menangani integritas data, bit parity dari byte karakter dimasukkan dalam data frame. Ini berarti bahwa untuk setiap karakter (7- atau 8-bit, tergantung pada sistem) kita punya bit parity tunggal sebagai tambahan dari start bit dan stop bit. Bit parity adalah ganjil atau genap. Dalam bit parity-ganjil jumlah total bit data, termasuk bit parity adalah ganjil dari 1-an. Serupa dengan itu, dalam sebuah bit parity-genap jumlah total bit data termasuk bit parity adalah genap. Sebagai contoh, karakter ASCII "A", biner 0100 0001, punya 0 untuk bit parity-genap. Chip-chip UART memungkinkan pemrograman bit parity-ganjil, parity-genap, dan pilihan tanpa-parity, seperti terdapat dalam pembahasan berikutnya. Jika suatu sistem membutuhkan parity, bit parity ditransmisikan setelah MSB, dan diikuti oleh stop bit.

2.4. Motor

Terdapat tiga jenis motor listrik dalam hal perbedaan kelakuan dan penggunaannya, yaitu:

1. Motor AC, dioperasikan oleh arus listrik bolak-balik.
2. Motor DC, dioperasikan oleh arus listrik searah.
3. Motor *stepper*, dioperasikan oleh pulsa-pulsa listrik.

Masing-masing jenis motor dapat memiliki bentuk yang bermacam-macam, pada prinsipnya setiap motor listrik dapat dimungkinkan pengontrolan secara elektronik terhadap kecepatan dan dayanya, sesuai dengan yang dikehendaki.

2.4.1. Motor AC

Untuk pengadaan tenaga listrik, penggunaan arus bolak balik lebih menguntungkan daripada arus searah. Dengan menggunakan transformator, tegangan bolak balik dengan mudah dapat diubah menjadi tegangan yang sangat tinggi. Dengan demikian energi listriknya dapat dipindahkan lebih ekonomis.

Keuntungan lain dari penggunaan arus bolak balik adalah konstruksi yang sederhana dari motor-motor arus bolak-balik dan motor-motor ini sering bisa dihubungkan langsung dengan jaringan.

A. Prinsip Kerja Motor Tiga Fasa.

Dalam sebuah makalah seminar mengatakan, apabila sumber tegangan tiga fasa dihubungkan pada kumparan stator maka kumparan tersebut akan timbul medan putar dengan kecepatan sinkron (Rujito dari Sekolah Tinggi Teknologi Nasional Yogyakarta). Medan putar stator memotong batang konduktor rotor akibatnya pada kumparan rotor timbul tegangan induksi sebesar

$$E_r = 4,44 \times F_r \times N_r \times \Phi_m \quad (\text{untuk satu fasa}) \quad (2.3)$$

Dimana: E_r : Tegangan induksi pada saat motor berputar (Volt).

F_r : Frekwensi arus rotor (Hz).

N_r : Jumlah lilitan rotor.

Φ_m : Fluksi maksimum (webber/telsa).

Karena kumparan rotor merupakan rangkaian tertutup maka pada kumparan tersebut mengalir arus. Arus yang mengalir pada penghantar rotor yang berada dalam medan magnet putar dari stator, maka penghantar rotor tersebut timbul gaya yang berpasangan dan berlawanan arah. Gaya tersebut

menimbulkan torsi yang cenderung memutar rotor, rotor akan berputar dengan kecepatan putar (N_r). Karena adanya perbedaan medan putar stator dan perputaran rotor timbul slip (S), dengan meliputi persamaan bentuk:

$$S = N_s - N_r / x 100\% \quad (2.4)$$

Dimana:

S = Slip

N_s = Kecepatan medan putar stator

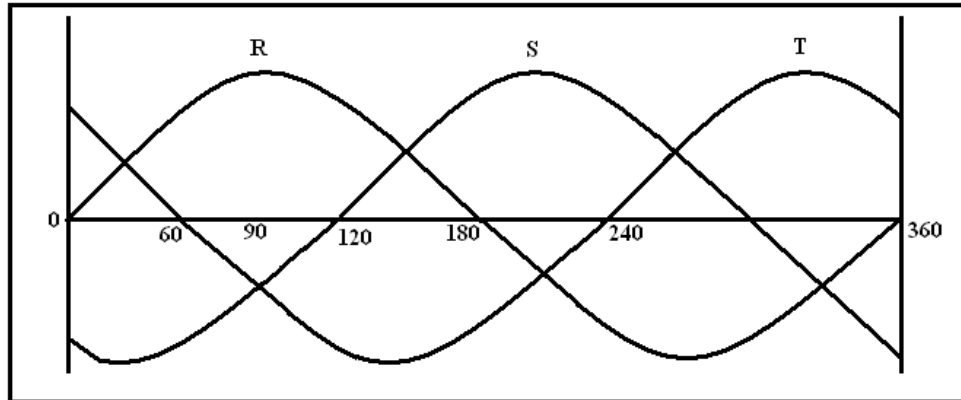
N_r = Kecepatan putar rotor

Bila $N_s = N_r$ maka tegangan tidak akan terinduksi dan arus tidak mengalir pada kumparan jangkar rotor dengan demikian tidak menghasilkan kopel. Kopel akan timbul jika $N_s < N_r$, bila kopel yang dihasilkan oleh gaya pada rotor cukup besar untuk memikul beban, maka rotor akan berputar searah dengan medan putar stator.

B. Bentuk Gelombang Motor Tiga Fasa.

Untuk menggerakkan mesin-mesin, kebanyakan digunakan motor arus bolak-balik fasa-tiga. Stator motor-motor ini membangkitkan suatu medan putar. Karena itu motor-motor ini juga dinamakan motor arus putar.

Motor-motor ini dihubungkan dengan jaringan arus bolak-balik fasa-tiga. Kalau jaringannya terdiri dari empat hantaran, hanya hantaran-hantaran fasanya saja yang dihubungkan. Hantaran netralnya tidak digunakan, dibawah ini merupakan bentuk dari gelombang motor tiga fasa.



Gambar 2.16 Bentuk Gelombang Motor Tiga Fasa

C. Arah Putar Dan Urutan Fasa Motor Fasa-Tiga

Dalam praktek arah putar sebuah motor biasanya ditentukan secara eksperimental. Akan tetapi adakalanya arah putarnya harus diketahui lebih dahulu, misalnya untuk motor-motor besar dengan kecepatan putar tinggi dan dengan *ventilator* atau sistem pelumasnya yang kerjanya tergantung pada arah poros motornya.

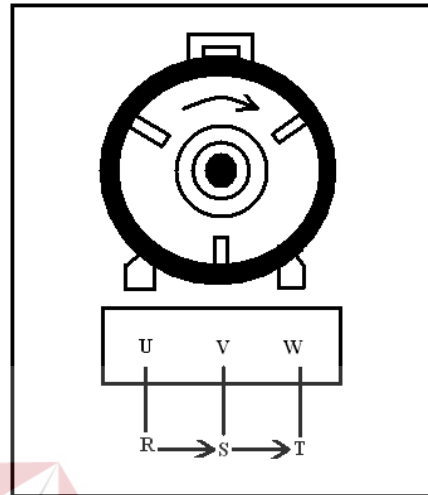
Karena itu oleh pabrik-pabrik motor listrik di negara-negara MEE (Masyarakat Ekonomi Eropa) telah dibuat ketentuan sebagai berikut:

Arah putar sebuah motor, dilihat menghadap sisi puli porosnya, akan ke kanan kalau terminal U dihubungkan dengan fasa R, terminal V dengan fasa S dan terminal W dengan fasa T, seperti yang terlihat pada gambar 2.17

Ketentuan di atas berlaku untuk motor-motor yang kotak terminalnya berada di tempat normal, yaitu disebelah kanan rumah motor, kalau dilihat menghadap sisi pulinya.

Untuk motor-motor dengan kaki yang kotak terminalnya harus berada di sebelah kiri, rumah motornya dibalik. Karena itu kalau dihubungkan dengan

urutan fasa U-R, V-S dan W-T, motornya akan berputar ke kiri, dilihat menghadap sisi pulinya.



Gambar 2.17 Arah Putar Motor Tiga Fasa

D. Hubungan Star Dan Delta Motor-Motor Fasa-Tiga

Jaringan distribusi tegangan rendah PLN umumnya memiliki tegangan 220/380 V atau 127/220 V. Di kemudian hari untuk distribusi lokal hanya akan digunakan sistem tegangan 220/380 V saja.

Sebuah motor harus digunakan dalam hubungan star atau hubungan delta, tergantung pada tegangan jaringnya. Tegangan yang harus dihubungkan dengan motor, biasanya dinyatakan di pelat merknya, misalnya 220/380 V atau 380/660 V. Tegangan yang lebih rendah ialah tegangan yang harus dihubungkan dengan kumparan-kumparan motor. Kalau sebuah motor diberi tanda tegangan 380/660 V misalnya, kumparan-kumparannya harus mendapat 380 V. Jadi kalau dihubungkan dengan jaringan 220/380 V, motor ini harus digunakan dalam

hubungan delta. Kalau digunakan dalam hubungan star, kumparan-kumparannya hanya akan mendapat 220 V saja.

Tegangan yang terlalu rendah juga dapat merusak motor. Perbedaan tegangan atau frekuensi yang tidak melebihi + 5% atau - 5% dari nilai nominalnya, biasanya tidak membahayakan motor. Untuk sebuah motor yang diberi tanda tegangan 220/380 V, hubungan yang harus digunakan adalah sebagai berikut :

a. Kalau sistem tegangan jaringnya 220/380 V, motor ini harus digunakan dalam hubungan star, karena kumparan-kumparannya harus mendapat 220V.

b. Kalau sistem tegangan jaringnya 127/220 V, motor ini harus digunakan dalam hubungan delta.

Kalau daya motor ini sama dengan 6,6 kVA, pada beban penuh arusnya akan sama seperti yang terlihat pada gambar 5.7.

a. Untuk sistem tegangan jaring 220/380 V:

$$I_n = 6600 : (380 \times \sqrt{3}) = 10 \text{ A} \quad (2.5)$$

Arus ini ialah arus yang juga mengalir dalam kumpara-kumparan motor.

b. Untuk sistem tegangan jaring 127/220 V:

$$I_n = 6600 : (220 \times \sqrt{3}) = 10 \text{ akar } 3 = 17,2 \text{ A} \quad (2.6)$$

Arus yang mengalir dalam kumparan-kumparan motor tetap 10 A, yaitu:

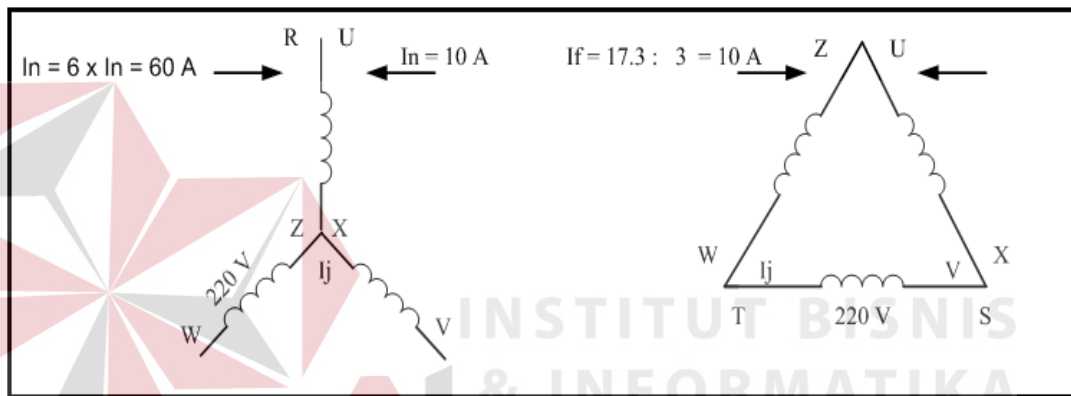
$$(17,2 : \sqrt{3}) \times \text{A} = 10 \text{ A.} \quad (2.7)$$

Tegangan kumparannya sama dengan 220V, sama seperti untuk a. Kalau motor ini dihubungkan langsung dengan jaring, arus asutnya akan 6 x arus nominalnya, jadi:

Untuk a : sama dengan $6 \times 10 \text{ A} = 60 \text{ A}$

Untuk b : sama dengan $6 \times 17,3 \text{ A} = 103,8 \text{ A}$.

Untuk sebuah motor yang diam, nilai arus asutnya ditentukan oleh impedansi kumparannya.



Gambar 2.18 Hubungan Star dan Delta

Kalau dalam kasus b, motornya lebih dahulu dihubungkan dalam hubungan star dan kemudian, setelah kecepatan putarnya sudah cukup tinggi, baru dalam hubungan delta, arus asutnya akan jauh lebih kecil. Dalam hubungan star, kumparan motornya hanya akan mendapat 127 V saja, bukan 220 V. Jadi arus asut dalam kumparan motor sekarang akan akar 3 kali lebih kecil, yaitu sama dengan:

$$6 \times (10 : \sqrt{3} \text{ A}) = 34,6 \text{ A}. \quad (2.8)$$

Karena motornya dihubungkan dalam hubungan star, arus asutnya dalam jaringan akan sama dengan arus asut dalam kumparan motor, jadi juga

sama dengan 34,6A. Jadi kalau motor ini lebih dahulu dihubungkan dalam hubungan star, arus asutnya dalam jaringan akan menjadi sepertiganya saja.

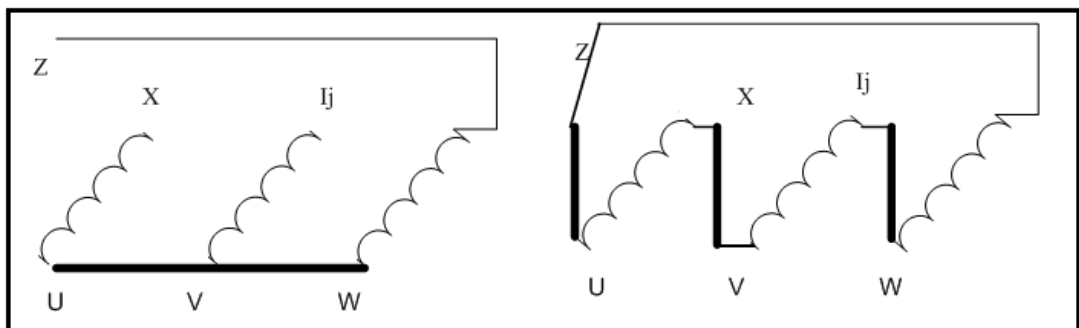
$$(34,6 : 103,8) = 1 / 3. \quad (2.9)$$

Kemungkinan untuk mengurangi arus asut ini dipraktekkan dengan menggunakan sakelar star delta untuk menjalankan motornya. Karena arus asutnya lebih kecil, kopel asutnya sekarang juga lebih kecil, sehingga kecepatan putar motornya akan meningkat lebih lamban.

Kalau digunakan sakelar star-delta, sakelarnya tidak boleh dibiarkan dalam kedudukan hubungan star. Sebab kalau dibiarkan dalam kedudukan star, arus dalam kumparan motor akan ditentukan oleh beban motor. Dalam hal contoh di atas, jika motornya diberi beban penuh, arus dalam kumparannya sekarang akan sama dengan:

$$6600 : (220 \times \sqrt{3}) = 17,3 \text{ A} \quad (2.10)$$

dan bukan 10 A. Jadi motornya akan menjadi terlalu panas, dan akhirnya akan terbakar. Gambar 2.19 memperlihatkan cara menghubungkan terminal-terminal motornya untuk hubungan star dan hubungan delta.



Gambar 2.19. Cara Menghubungkan Terminal

E. Cara-Cara Pengasutan Motor Fasa-Tiga

Cara pengasutan motor-motor yang dihubungkan dengan jaringan distribusi tegangan rendah, harus sedemikian hingga tidak menimbulkan guncangan-guncangan tegangan yang mengganggu dalam jaringan. Guncangan-guncangan demikian akan sangat mengganggu instalasi penerangan yang dihubungkan dengan jaringan itu.

Tabel di bawah ini memuat beberapa petunjuk mengenai cara pengasutan motor-motor fasa-tiga. Batas-batasnya berbeda untuk masing-masing perusahaan listrik.

Tabel 2.10. Cara pengasutan motor-motor fasa-tiga

Daya nominal motor	Cara pengasutan
Kurang dari 1,5 @ 2,25 kW	dihubungkan langsung dengan jaringan
Sampai 4 @ 6 kW	Dengan sakelar star delta
Sampai 8 @ 12 kW	Dengan sakelar star segitiga yang diperlengkapi dengan tahanan-tahanan
Lebih dari 8 @ 12 kW	Dengan transformator asut, atau motor angker gelang seret dengan tahanan asut rotor.

Kalau digunakan alat asut, arus asutnya tidak boleh melebihi 2,5 In.

Untuk motor angker gelang seret, arus asutnya tidak boleh melebihi 1,5 In.

Untuk motor-motor 20 kW atau lebih, harus benar-benar diperhatikan bahwa pengasutannya dilakukan menurut urutan yang tepat. Alat asutnya tidak boleh terlalu cepat dipindahkan ke kedudukan berikutnya. Untuk mengawasi arus

asutnya, dalam saluran ke motor dapat dipasang amperemeter. Amperemeter yang digunakan harus lebih baik. Pembagian skalanya diberi garis merah yang jelas untuk menandai arus asut maksimum yang diperbolehkan.

Kalau ada beberapa motor yang jumlah dayanya melebihi 20 kW, masing-masing motor atau semua motor sekaligus, harus diberi pengaman tegangan nol. Kadang-kadang ditentukan bahwa sakelar tegangan nol ini baru boleh dihubungkan, sesudah motor-motornya dimatikan.

Kalau instansi yang bersangkutan memiliki transformator tersendiri, daya motor terbesar yang masih boleh dihubungkan langsung dengan dapat ditentukan sebagai berikut:

$$P_m \leq P_{tr} \cdot d / u \cdot i \quad (2.11)$$

Di mana:

P_m = daya motor dalam kVA;

P_{tr} = daya transformator dalam kVA;

d = penurunan tegangan yang diperbolehkan, dalam %

u = tegangan hubung-singkat dari transformator, dalam %

i = faktor arus asut = arus asut / I_n

sebagai contoh misalkan:

$$P_{tr} = 300 \text{ kVA}; u = 3\%; i = 6; d = 6\%.$$

Untuk P_m didapat:

$$P_m \leq 300 \times 6 / 3 \times 6 \text{ kVA} \quad (2.12)$$

Atau:

$$P_m \leq 100 \text{ kVA.} \quad (2.13)$$

Kalau untuk contoh ini digunakan cara pengasutan dengan faktor arus asut 3 (jadi arus asutnya sama dengan 3 In), daya motor yang masih diperbolehkan akan meningkat jadi 200 kVA.

Untuk instalasi-instalasi dengan pembangkit listrik sendiri, daya motor yang bisa dihubungkan langsung, tergantung pada daya dan jenis generator yang digunakan dan pada pengaturan tegangan generator ini.

2.4.2 Motor DC

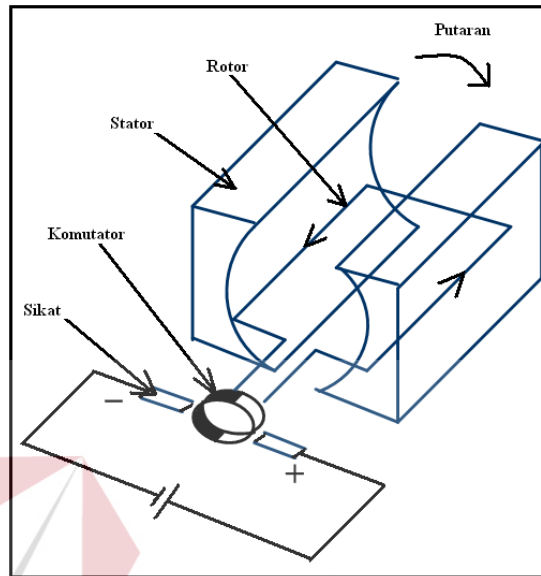
Motor arus searah ialah suatu mesin yang berfungsi mengubah tegangan listrik arus searah (DC) menjadi tenaga gerak atau tenaga mekanik, dimana tenaga gerak tersebut berupa putaran dari rotor.

Dasar dari motor DC adalah, disekitar kawat berarus listrik terdapat medan- medan magnet (teori oersted). Bila mana arus listrik yang mengalir pada kawat arahnya menjauhi kita (maju), maka medan-medan magnet yang terbentuk disekitar kawat arahnya searah dengan putaran jarum jam. Sebaliknya bila arus listrik yang mengalir pada kawat arahnya mendekati kita (mundur), maka medan magnet yang terbentuk di sekitar kawat arahnya berlawanan dengan arah putaran jarum jam (teori Maxwell).

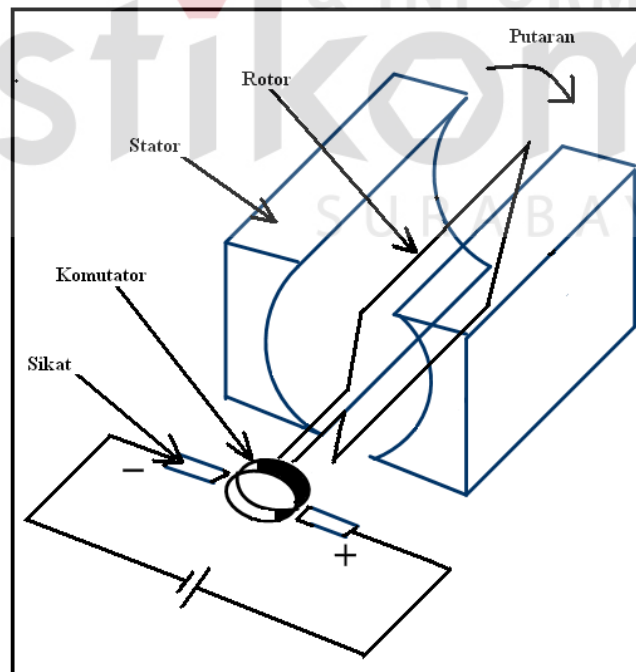
Pada penggunaan motor DC yang sesungguhnya agar dapat menghasilkan torsi dan kecepatan putar yang sesuai dengan yang dibutuhkan dapat digunakan transmisi Gear Box. Dengan transmisi Gear Box beban kerja dari motor akan menjadi lebih ringan dan motor tidak cepat rusak.

A. Prinsip Kerja Motor DC

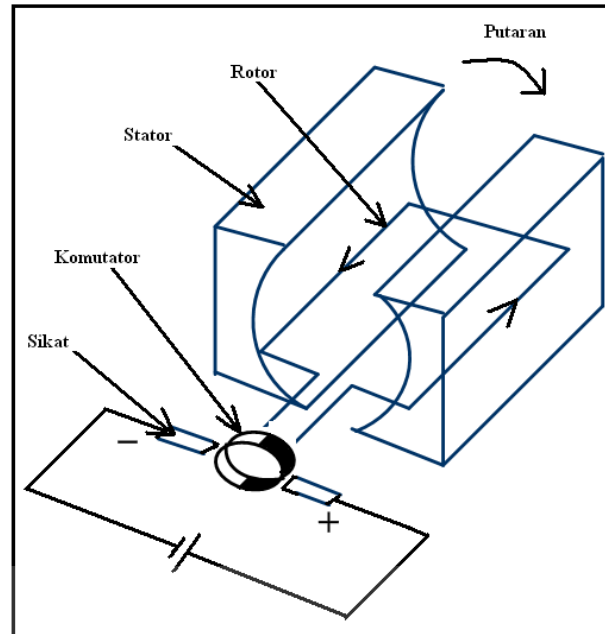
Untuk mengetahui prinsip kerja dari motor DC dapat dilihat pada gambar 2.20a, 2.20b dan 2.20c.



Gambar 2.20a Prinsip Kerja Motor Langkah 1



Gambar 2.20b Prinsip Kerja Motor DC Langkah 2



Gambar 2.20c Prinsip Kerja Motor DC Langkah 3

Arus listrik mengalir ke koil melalui sikat-sikat yang selalu berhubungan dengan komutator, yang ditekan oleh pegas. Pada posisi pada gambar 2.20a, aliran arus pada koil akan menghasilkan medan magnet yang berlawanan dengan medan magnet dari magnet stator. Sehingga menyebabkan koil berputar ke arah yang ditunjukkan oleh anak panah. Apabila aliran arus tetap mengalir dalam arah seperti pada gambar 2.20a, koil akan diam pada posisi vertikal setelah berputar sejauh 90 derajat. Karena adanya momentum dari koil, maka gerakannya akan melawati posisi vertikal. Apabila ia telah mencapai posisi seperti pada gambar 2.20b, komutator akan menyebabkan aliran arus yang mengalir melalui koil berbalik dari arah semula. Dengan demikian, aliran arus sekarang akan menghasilkan tolakan magnet yang memutar koil sejauh 90 derajat ke posisi seperti pada gambar 2.20c.

2.5 Transistor

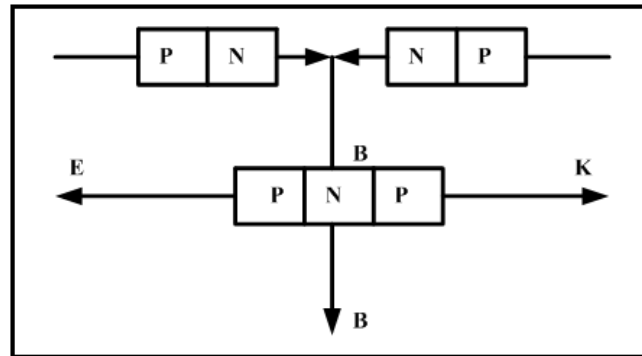
Transistor ditemukan oleh tiga orang sarjana Amerika, yang bernama *J. Barden*, *WH Brattain* dan *W Shockley* pada tahun 1948, nama transistor berasal dari kata *transfer* dan *resistor*, transfer artinya mengalihkan atau membuat perubahan sedangkan resistor adalah suatu bahan yang tidak dapat menghantarkan arus listrik, jadi arti dari transistor adalah merubah bahan yang tidak dapat menghantarkan aliran listrik menjadi bahan penghantar atau setengah penghantar atau disebut juga bahan *semikonduktor* (Rusmadi,1999:81).

Transistor pada umumnya dipergunakan sebagai *penguat* atau *amplifier*, transistor sendiri sebenarnya adalah hasil pengembangan dari dua buah jenis dioda jenis PN dan NP yang dipertemukan sehingga akan membentuk satu elektroda yang berfungsi sebagai pengontrol pertemuan antara bahan PN dan NP tersebut. Prinsip terjadinya pertemuan kedua bahan tersebut seperti yang terlihat pada gambar 2.8, bila kedua bahan yang dipertemukan bahan jenis N nya maka akan diperoleh transistor jenis PNP, ($PN + PN = PNP$).

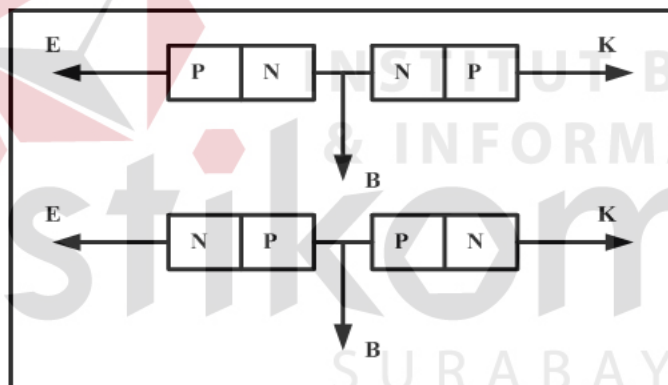
Sedangkan pada gambar 2.21 terlihat juga bila kedua bahan yang dipertemukan bahan jenis P nya, maka akan terbentuk transistor jenis NPN, $NP + PN = NPN$ dari hasil pertemuan kedua bahan P dan N tersebut akan menghasilkan sebuah transistor yang memiliki 3 buah elektroda yang membentuk 3 buah kaki yaitu:

1. Emitor disingkat E.
2. Basis disingkat B
3. Kolektor disingkat C

Berdasarkan prinsip tersebut maka dalam dalam teknik elektronika, transistor tersebut dinamakan *transistor bipolar jenis PNP* dan *NPN*.



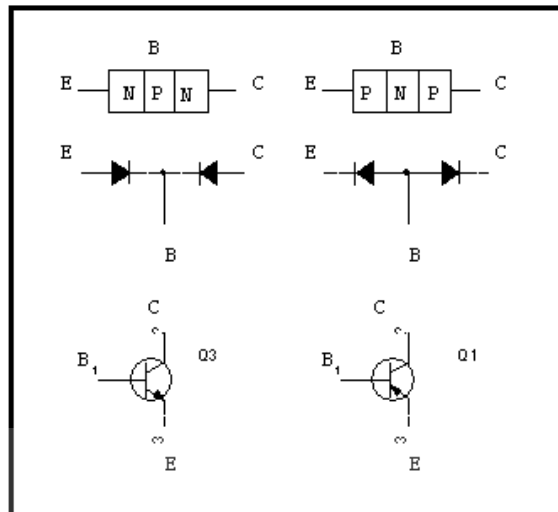
Gambar 2.21. Prinsip Transistor PNP



Gambar 2.22. Prinsip Transistor NPN

Transistor merupakan alat dengan tiga terminal basis(b), kolektor (c), emitor(e), transistor dapat dibedakan dalam dua jenis yaitu NPN dan PNP. Disebut transistor PNP karena terdiri dari bahan semi konduktor jenis P,N,dan P. dan disebut transistor NPN karena terdiri dari bahan semikonduktor jenis N,P,dan N. sehingga transistor dapat kita anggap sebagai dua jenis dioda yang saling dipertemukan. Transistor PNP memiliki simbol panah masuk, sedangkan

transistor NPN mempunyai simbol panah keluar seperti yang terlihat pada gambar 2.23.



Gambar 2.23. Simbol Transistor.

2.5.1. Transistor Pertemuan (*Junction Transistor*)

Azas kerja transistor seperti yang terlihat pada gambar 2.24 yaitu:

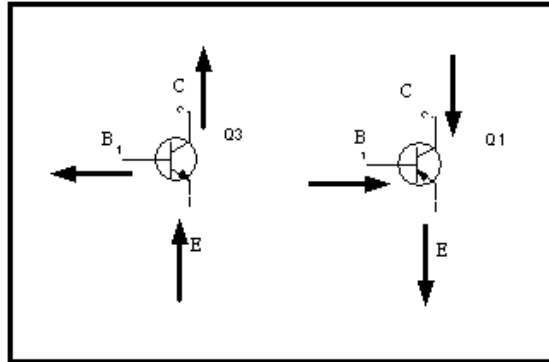
(Wasito,1995:180).

- Akan ada arus diantara terminal-terminal kolektor-emitor (arus I_C), hanya apabila, ada arus yang mengalir diantara terminal-terminal basis-emitor (arus I_B).
- Perbandingan antara kuat I_C dan kuat I_B disebut “bandingan-hantaran arus maju” (*forward current transfer ratio*), disingkat: H_{FE} ,

dengan rumus:

$$H_{FE} = I_C : I_B \quad (2.13)$$

- Untuk penguatan frekuensi tinggi, ada transistor dengan harga $H_{FE} = 1000$ atau lebih



Gambar 2.24. Arah arus pada transistor

2.5.2 Parameter-parameter pada Transistor

Ada beberapa hal yang perlu diketahui mengenai parameter-parameter pada transistor yaitu: (Wasito,1995:180).

- Parameter-parameter transistor tidaklah konstan, meskipun tipe sama, parameter dapat berbeda.
- Parameter juga akan berlainan bagi arus yang berlainan, tapi dalam prakteknya dapat kita anggap bahwa parameter-parameter adalah konstan.
- Konduktansi,

$$G_M = i_e : V_{be} \text{ ma/V (ms)} \quad (2.14)$$

i_e : arus isyarat ac di antara kolektor - emitor

V_{be} : tegangan isyarat ac antara basis - emitor

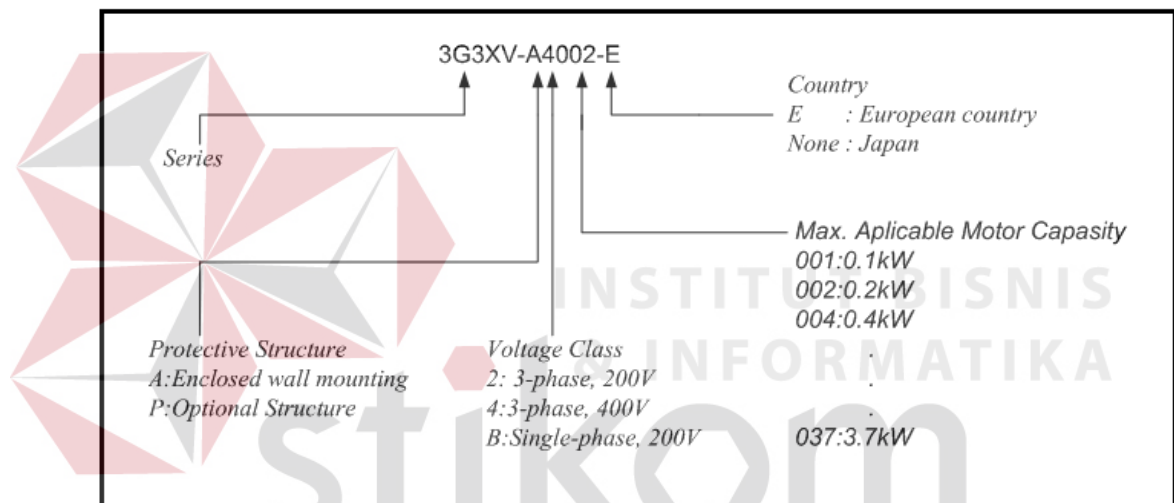
2.6. Inverter

Inverter motor tiga fasa disini berfungsi sebagai jembatan antara motor tiga fasa dengan DAC dan inverter berfungsi juga sebagai perubah tegangan DC ke tegangan AC, karena motor tiga fasa merupakan motor AC, dengan adanya

inverter maka motor tiga fasa bisa dikontrol dengan peralatan luar seperti PLC, mikrokontroler atau peralatan lainnya.

2.6.1. OMRON 3G3XV-A2004

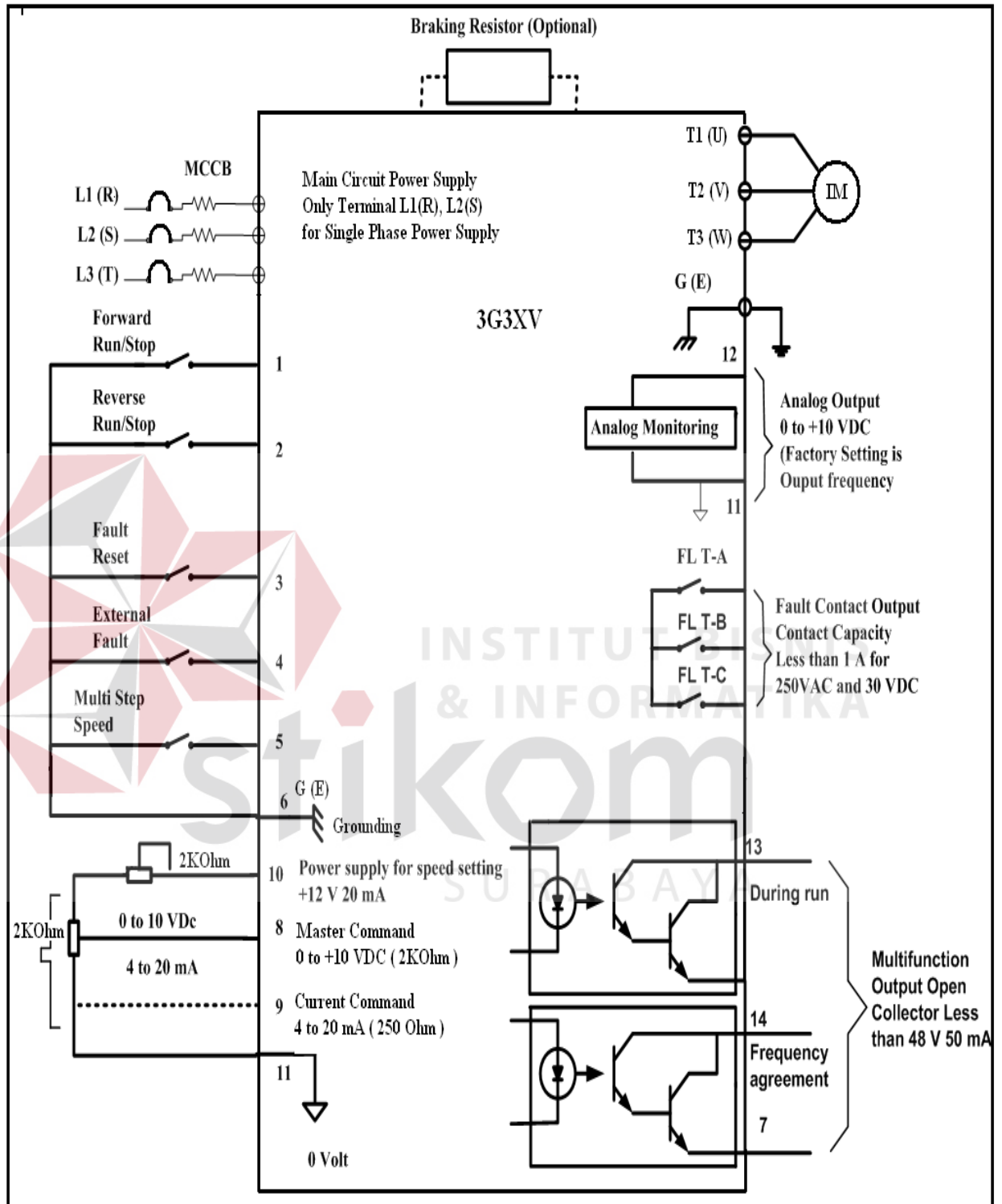
Inverter yang ada di Laboratorium Kontrol di STIKOM adalah “3G3XV-A2004”, dengan *Input Specification* adalah *AC3PH 200-230V 50Hz/60Hz* dan *Output Specification* adalah *AC3PH 0-230V 1.2KVA 3A*. Inverter Model Number 3G3XV-A4002-E (Malvino, 1994:341), memiliki arti yaitu:



Gambar 2.25. Inverter Model Number 3G3XV-A4002-E

2.6.2 Standard Wiring Diagram

Inverter Model Number 3G3XV-A4002-E menggunakan *Standard Wiring Diagram*, yang dapat digunakan secara *auto* dan *manual*, seperti pada gambar 2.26.



Gambar 2.26. Standard Wiring Diagram

2.7 Dasar – dasar Konverter DAC

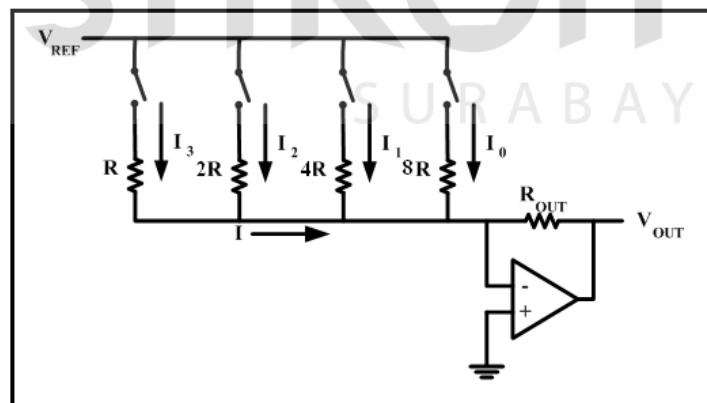
Pada rangkaian penjumlah op-amp dapat digunakan untuk menyusun suatu konverter DAC dengan memakai sejumlah hambatan masukan yang diberi bobot dalam darat biner, seperti yang terlihat pada gambar 2.14. Dari gambar tersebut terdapat V_{REF} , yang merupakan sebagai tegangan acuan (*reference*) yang tepat, semua hambatan merupakan hambatan yang presisi, dengan maksud supaya mendapatkan arus-arus masukan yang presisi pula. Pada gambar 2.27 terdapat saklar, maksudnya adalah jika saklar ditutup maka arus akan mengalir masuk, jika dibuka maka arus tidak masuk apabila semua saklar ditutup maka arus-arus masukan sama dengan:

$$I_3 = V_{REF} / R \quad (2.15)$$

$$I_2 = V_{REF} / 2R \quad (2.16)$$

$$I_1 = V_{REF} / 4R \quad (2.17)$$

$$I_0 = V_{REF} / 8R \quad (2.18)$$



Gambar 2.27. Dasar – dasar Konversi DAC





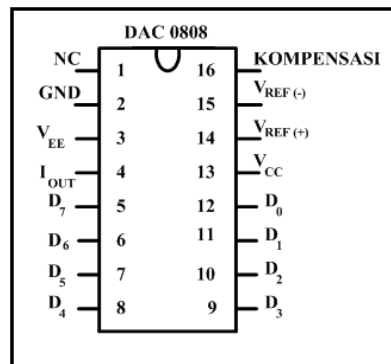




2.7.2 DAC 0808 secara Umum

Digital to Analog Converter (DAC), adalah rangkaian/IC yang digunakan untuk mengubah sinyal digital ke analog. Sinyal digital disini adalah suatu logika 0 dan logika 1, yang akan diubah ke sinyal analog. Sinyal analog disini berupa suatu tegangan tertentu, misalnya untuk inverter motor tiga fasa range tegangannya adalah 0 sampai 10 volt, jadi tegangan 0 sampai 10 volt merupakan perbandingan dengan logika 0 dan 1 atau perbandingan 0 sampai 10 volt dengan 0 sampai 255 desimal. Salah satu DAC yang umum dan mudah didapat yaitu DAC 0808 seperti yang terlihat ada gambar 2.28, yang memiliki gambaran secara umum sebagai berikut:

1. Mempunyai delapan bit input sebagai input sinyal digital.
2. Mempunyai resolusi 8 sampai 12 bit untuk harga terendah.
3. Mempunyai resolusi 16 sampai 18 bit untuk harga termahal.
4. Bersifat monotonik
5. Tingkat kesalahan kurang dari $\pm \frac{1}{2}$ LSB pada setiap tingkat keluaran.
6. Arus acuan 2mA.
7. Waktu pemantapan 150 ns.



Gambar 2.31. Konfigurasi DAC 0808