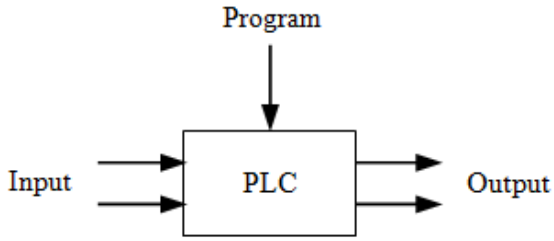
BAB II
TINJAUANPUSTAKA

2.1 PLC *(Programmable Logic Controller)*[1]

Programmable Logic Controller atau PLC pada awalnya dikenal sebagai *Programmable Controller* (PC) yang lahir sebagai produk yang kompak, dapat diprogram dan direprogram seperti komputer, tidak memakan tempat dan energi yang besar, berbasiskan teknologi digital, yang dapat menggantikan rangkaian relay dan *hardwire*.*Programmable logic controller* singkatnya PLC merupakan suatu bentuk khusus pengontrolan berbasis mikroprossesor yang memanfaatkan memori yang dapat diprogram untuk menyimpan intruksi-intruksi dan untuk mengimplementasikan fungsi-fungsi semisal logika, *sequencing*, pewaktuan (*timing*), pencacahan (*counting*) dan aritmetika guna mengontrol mesin-mesin dan proses-proses seperti (gambar 2.1) dan dirancang untuk dioperasikan oleh para insinyur yang memiliki pengetahuan mengenai bahasa pemrograman. Piranti ini dirancang sedemikian rupa agar tidak hanya programer komputer saja yang dapat membuat dan mengubah program- programnya. Oleh karena itu, para perancang PLC telah menempatkan program awal di dalam piranti ini (*pre-program*) yang memungkinkan program-program kendali dimasukkan dangan menggunakan suatu bentuk bahasa pemrograman yang sederhana danintuitif.



Gambar 2.1 Diagram Blok Logika PLC *(Programmable LogicController)*[2]

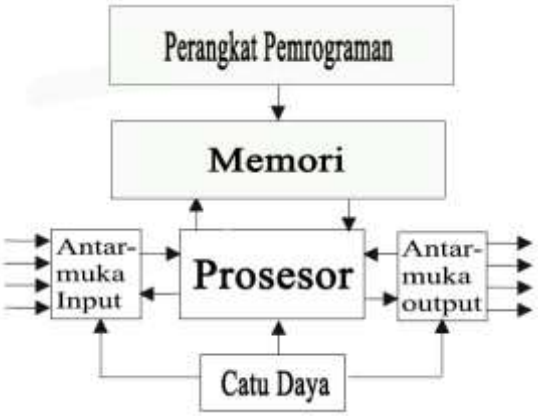
[1]William Bolton, *Programmable Logic Controller (PLC) Sebuah Pengantar Edisi Ketiga* (Jakarta: Erlangga, 2003), hlm. 03
[2]William Bolton, *Programmable Logic Controller (PLC) Sebuah Pengantar Edisi Ketiga* (Jakarta: Erlangga, 2003), hlm. 03

Istilah logika (*logic*) dipergunakan karena pemrograman yang harus dilakukan sebagian besar berkaitan dengan pengimplementasian operasi- operasi logika dan penyambungan saklar. Perangkat-perangkat *input*, yaitu sensor-sensor semisal saklar dan perangkat-perangkat *Output* di dalam sistem yang di kendali, misalnya motor DC, motor servo, dan lain-lainya yang disambungkan ke PLC. Sang operator kemudianmemasukan serangkai intruksi, yaitu sebuah program ke dalam memory PLC. Perangkat pengontrol tersebut kemudian memantau *input-input* dan *output-output* sesuai dengan intruksi-intruksi di dalam program dan melaksanakan aturan-aturan kendali yang telahdiprogramkan.

PLC serupa dengan komputer namun, bedanya komputer dioptimalkan untuk tugas-tugas perhitungan dan penyajian data, sedangkan PLC dioptimalkan untuk tugas-tugas pengontrolan dan pengoperasian di dalam lingkungan industri. Dengan demikian PLC memiliki karakteristik:

1. Kokoh dan dirancang untuk tahan terhadap getaran, suhu, kelembaban, dan kebisingan.
2. Antaramuka input dan output telah tersedia secara *built-in* didalamnya.
3. Mudah diprogram dan menggunakan sebuah bahasa pemrograman yang mudah dipahami, yang sebagian besar berkaitan dengan operasi- operasi logika danpenyambungan.

Perangkat PLC pertama kali dikembangkan pada tahun 1969. Dewasa ini PLC secara luas digunakan dan telah dikembangkan dari unit-unit kecil yang berdiri sendiri *(self-contained)* yang hanya mampu menangani sekitar 20 *input/output* menjadi sistem-sistem modular yang dapat menangani *input/output* dalam jumlah besar, menangani *input/output* analog maupun digital, dan melaksanakan mode-mode kontrol proporsional integral derivatif.



Gambar 2.2Diagram Blok Sistem PLC (*Programmable Logic Controller)*[3]

2.1.1 *Hardware*

Umumnya, sebuah sistem PLC memiliki lima komponen dasar. Komponen-komponen ini adalah unit processor, memori, unit catu daya, bagian antarmuka *input/output*, dan perangkat pemrogaman.

1. *Unit processor* atau *central processing unit* (unit pengolahan pusat) *(CPU)* adalah unit yang berisi mikroprosessor yang mnginterpretasikan sinyal-sinyal *input* dan melaksanakan tindakan-tindakan pengontrolan, sesuai dengan program yang tersimpan di dalam memori, lalu mengkomunikasikan keputusan-keputusan yang diambilnya sebagai sinyal kontrol ke antarmuka *output.*
2. *Unit catu daya* diperlukan untuk mengkonversikan tegangan AC sumber menjadi tegangan rendah DC (5V) yang dibutuhkan oleh*processor* dan rangkaian-rangkaian di dalam modul-modul antarmuka *input* dan *output*.
3. *Perangkat pemrogaman* dipergunakan untuk memasukkan program yang dibutuhkan ke dalam memori. Program tersebut dibuat dengan menggunakan perangkat ini dan kemudian dipindahkan ke dalam unit memori PLC.
4. *Unit memori* adalah tempat dimana program yang digunakan untuk melaksanakan tindakan-tindakan pengontrolan oleh mikroprosessor disimpan.

[3]William Bolton, *Programmable Logic Controller (PLC) Sebuah Pengantar Edisi Ketiga* (Jakarta: Erlangga, 2003), hlm. 04

5. *Bagian input dan output* adalah antarmuka di mana prosesor menerima infomasi dari dan mengkomunikasikan informasi kontrol ke perangkat-perangkat eksternal. Sinyal-sinyal input, oleh karenanya, dapat berasal dari saklar-saklar.

Tahap dasar untuk penyiapan awal untuk memudahkan dan memasukkan program dalam PLC dengan mempersiapkan daftar seluruh peralatan *input* dan *output* beserta lokasi I/O bit, penempatan lokasi *word* dalam penulisan data. Untuk pemrograman sebuah PLC dahulu kita harus mengenal atau mengetahui tentang organisasi dan memorinya. Ilustrasi dari organisasi memori adalah sebagai peta memori (memori map), yang spesifikasinya terdiri dari kategori *User Programable* dan Data Tabel.*User Programable* adalah dimana program *Logic Ladder* dimasukkan dan disimpan yang berupa instruksi – instruksi dalam format *Logic Ladder*. Setiap instruksi memerlukan satu *word* didalam memori.

2.1.2 PLC Omron CP1E-E40 SDR-A[4]

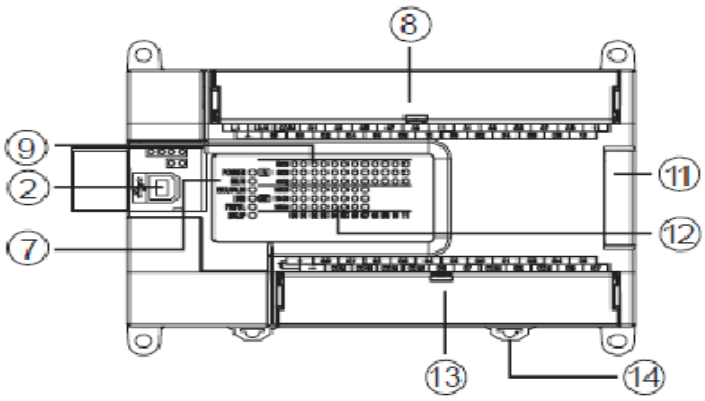
Merupakan Jenis dari PLC Omron seri CP1E, sedangkan arti dari E40 merupakan jumlah dari output dan input yang terdapat pada PLC. PLC jenis ini dapat di implementasikan pada penggerak mekanisme alat industri, alat rumah tangga, dan tugas teknik lainnya, yang mana bersifat logika elektronika.

Gambar 2.3PLC Omron CP1E-E40SDR-A

[4]Omron, "SYSMAC CP-Series CP1E CPU Units Introduction Manual". Introduction Manual, omron, hlm. 03.

PLC Omron seri CP1E memiliki I/0 sebanyak 40 yang dimana 24 *input* bisa diubah menjadi analog, yaitu bekerja dengan tegangan 5 sampai 24 volt dan memiliki *output* sebanyak 16 yang dimana masing-masing *output* tersebut juga memiliki *internal relay* yang bekerja dengan arus hingga 10A. PLC Omron seri CP1E bekerja dengan tegangan yang bisa diubah 100 sampai 240 VAC, Program memory: 2Ksteps (EEPROM), Data memory DM: 2K*words*. Dan memiliki minimal tegangan kerja 5 VDC dan maksimum tegangan kerja 24 VDC pada *input* PLC. Kemudian pada masing-masing *output* PLC memiliki internal relay yang memiliki maksimum arus kerja sebesar 10A. PLC Omron seri CP1E memiliki sistem program dengan menggunakan *software* pemrogaman CX-*Programmer*.

Bagian-bagian dari PLC Omron CP1E-40 SDR-A[5]



Gambar 2.4 Bagian-BagianPLC Omron CP1E-E40 SDR-A[6]

Keterangan :

2 ⟶ Port USB periferal

Digunakan untuk koneksi ke komputer. Komputer bisa digunakan untuk pemrograman dan monitoring

7 ⟶ Indikator operasi

Menunjukkan status operasi CP1L. Status yang terindikasi meliputi status daya, mode operasi, kesalahan, dan status komunikasi USB perifer

[5]Omron ,"Programmable Controller SYSMAC CP1L/CP1E", SYSMAC CP1L/CP1E introduction manual, hlm. 18-20

[6] Omron ,"Programmable Controller SYSMAC CP1L/CP1E", SYSMAC CP1L/CP1E introduction manual, hlm. 18

8 ⟶ Blok catu daya, *ground,* dan *input*

Digunakan untuk menghubungkan jalur catu daya, *ground* , dan jalur *input*

9 ⟶ Indikator *input*

Indikator bila kontak terminal *input* yang sesuai adalah *ON* kecuali untuk terminal *input* analog

11 ⟶ Konektor unit I / O

Digunakan untuk menghubungkan unit I / O CP-*series* dan unit ekspansi

12 ⟶ Indikator *output*

Indikator bila kontak terminal keluaran yang sesuai adalah *ON* kecuali untuk terminal keluaran analog.

13 ⟶ Pasokan daya eksternal dan blok terminal keluaran

• Terminal catu daya eksternal:

Unit yang menggunakan catu daya AC memiliki terminal *power supply* 24VDC eksternal dengan kapasitas maksimal 300mA. Ini bisa digunakan sebagai *power supply* servis untuk perangkat *input*.

Unit CPU CP1E E10 / 14/20 (S) atau N14 / 20 tidak memiliki terminal catu daya eksternal.

• Terminal *output*: Digunakan untuk menghubungkan jalur *output*

14 ⟶ Pasokan daya eksternal dan blok terminal keluaran

• Terminal catu daya eksternal:

Unit yang menggunakan catu daya AC memiliki terminal *power supply* 24VDC eksternal dengan kapasitas maksimal 300mA. Ini bisa digunakan sebagai *power supply* servis untuk perangkat *input*.

Unit CPU CP1E E10 / 14/20 (S) atau N14 / 20 tidak memiliki terminal catu daya eksternal.

• Terminal *output*: Digunakan untuk menghubungkan jalur *output*

2.1.3 *Software* CX-*Programmer*[7]

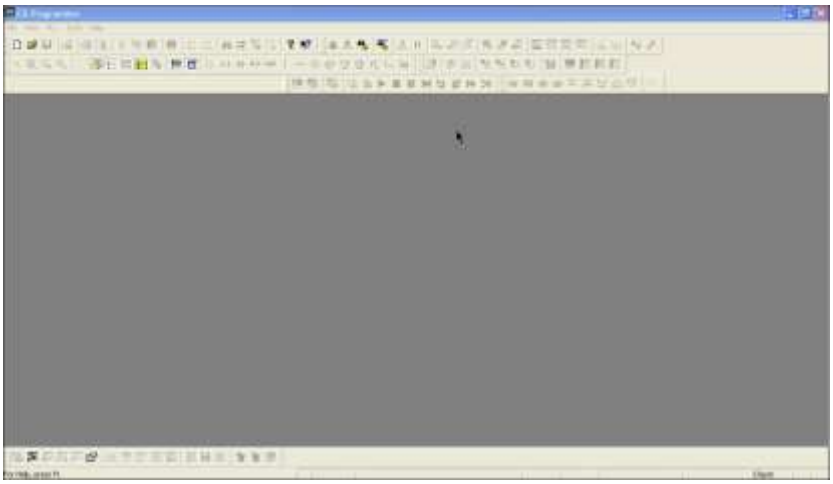
CX-*Programmer* merupakan software khusus untuk memprogram PLC buatan OMRON.CX *Programmer* ini sendiri merupakan salah satu software bagian dari CX-*One*. Dengan CX-*Programmer* ini bisa memprogram aneka PLC buatan omron dan salah satu fitur yaitu adanya fitur simulasi tanpa harus terhubung dengan PLC, sehinggamensimulasikan *ladder* yang buat, dan simulasi ini juga bias dihubungkan dengan HMI PLC Omron yang telah kita buat dengan menggunakanCX-*Designer* (bagian dari CX-*One*).

Software ini beroperasi di bawah sistem operasi *Windows*, oleh sebab itu pemakai *software* ini diharapkan sudah familier dengan sistem operasi *Windows* antara lain untuk menjalankan *software* program aplikasi, membuat *file*, menyimpan *file*, mencetak *file*,menutup *file*, membuka *file*, dan keluar dari (menutup) *software* program.Ada beberapa persyaratan minimum yang harus dipenuhi untuk bisa mengoperasikan CX-*Programmer* secara optimal yaitu:
Komputer IBM PC/AT kompatibel
CPU Pentium I minimal 133 MHz
RAM 32 Mega bytes
Hard disk dengan ruang kosong kurang lebih 100 MB
Monitor SVGA dengan resolusi 800 x 600

Gambar 2.5 CX-*Programmer Version* 9.0 Omron

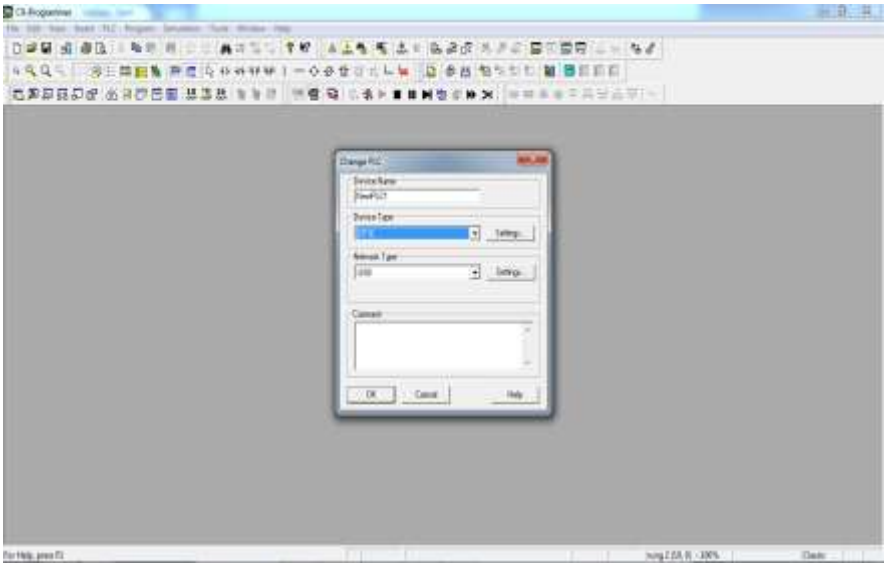
[7]Musbikhi.Pengantar CX Programmer (Seri Belajar PLC), online diakses 25 Mei 2017 pada pukul 15.32 WIB dari (http://www.musbikhin.com/pengantar-cx-programmer-seri-belajar-plc).

Berikut tampilan dari CX-*Programmer* saat pertama kali dibuka :



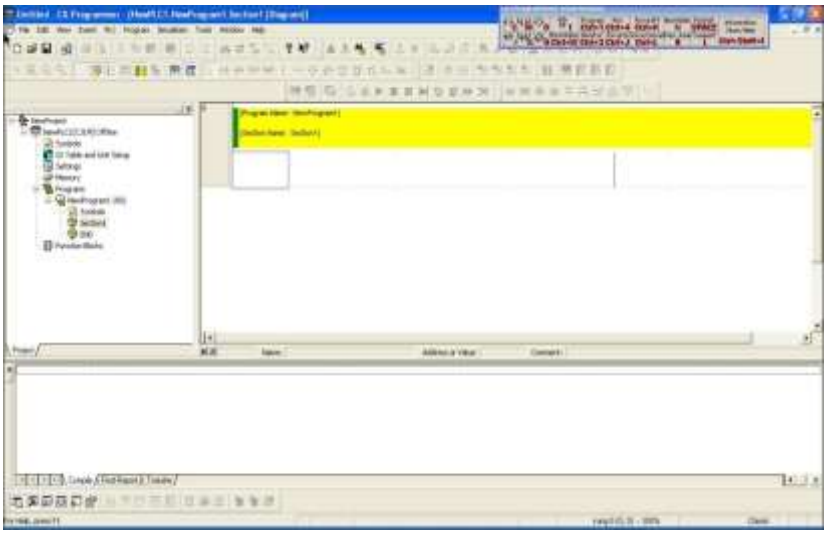
Gambar 2.6Tampilan Pertama Program CX-*Programmer Version* 9.0 Omron

Untuk memulai menggunakan CX-*Programmer* ini yaitu pada menu pilih *file ->new* atau bisa langsung pada *toolbar* klik gambar kertas putih untuk memulai membuat *project* baru, untuk membuka *file project* yang sudah dibuat sebelumnya yaitu pilih *file ->open* atau pada *toolbar* pilih gambar disamping kertas putih maka akan muncul tampilan berikut :



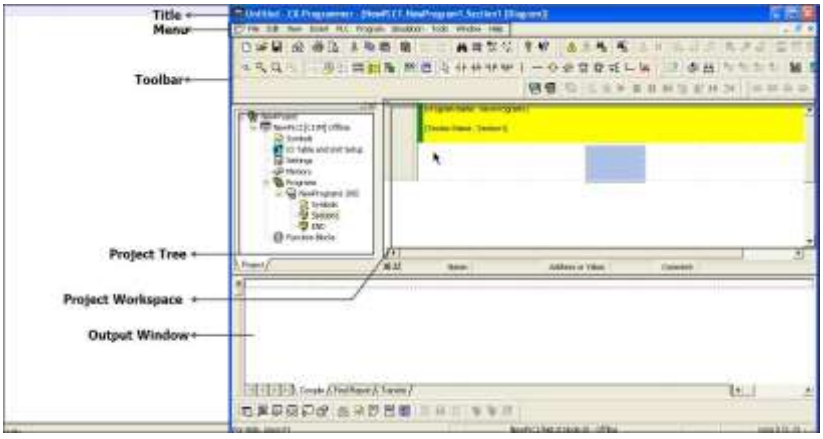
Gambar 2.7 Tampilan Pemilihan *Device*PLC Pada ProgramCX-*ProgrammerVersion* 9.0 Omron

Setelah memilih tipe PLC yang akan digunakan, misalnya PLC CP1E dan *Network type* yang akan digunakan yaitu USB untuk *setting* lebih dalam bisa diklik *setting*, kemudian klik OK maka akantampilan sebagi berikut :



Gambar 2.8 Tampilan Project Program CX-*Programmer Version* 9.0 Omron

Keterangan detail untuk tampilan tersebut yaitu :



Gambar 2.9 Tampilan Keterangan Project Program CX-*Programmer Version* 9.0 Omron

***Title Bar* :**

Menunjukan nama file atau data tersimpan dan dibuat pada CX- *Programmer*

***Menu* :**

Pilihan Untuk memilih Menu

***Toolbar* :**

Pilihan untuk memilih fungsi dengan menekan tombol.

Select[view] Toolbar, Kemudian dapat memilih *toolbar* yang ingin ditampilkan.

***Section* :**

Dapat membagi program kedalam beberapa blok. Masing-masing blok dapat dibuat atau ditampilkan.

***Project WorkSpace Project Tree* :**

Mengatur program dan data. Dapat membuat duplikat dari setiap elemen dengan melakukan Drag dan Drop diantara proyek yang berbeda atau melalui suatu proyek.

***Ladder Window* :**

Layar sebagai tampilan atau membuat diagram tangga.

***Output Window* :**

Menunjukan informasi *error* saat melakukan *compile (error check).* Menunjukan hasil dari pencarian kontak/koil didalam list form. Menunjukan detail dari *error* yang ada pada saat *loading* suatu proyek.

Status Bar :

Menunjukan suatu informasi seperti nama PLC, status *on line/offline*, lokasi dari *cell* yang sedang aktif.

***Information Window* :**

Menampilkan *window* yang menunjukan *shortcut key* yang digunakan pada CX – *Programmer.*

***Symbol Bar* :**

Menampilkan nama, alamat atau nilai dan *comment* dari simbol yang sedang dipilih *cursor.*

2.1.4 Program PLC

Suatu *sofware* yang berfungsi sebagai pengontrol otomatis yang berupa *softcontact* yang diimplementasikan kedalam suatu bentuk bilangan logika.Sehingga dapat mengatur sistem suatu alat industri elektronika dan mekanik.

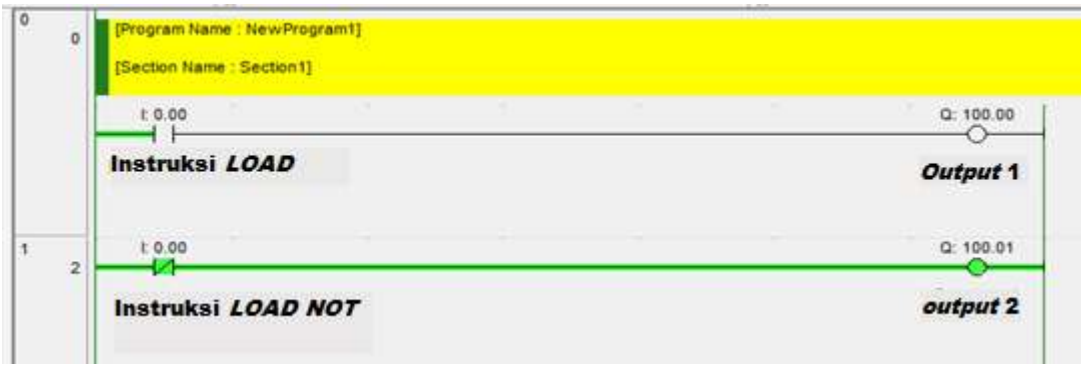
Ada 2 sistem pemprograman pada PLC Omron CP1E-E40SDR-A :

1. *Function Block Diagram*: Jenis Teknik Pemograman Logic yang tersususn dari *block-block* diagram dalam1 fungsi blok diagram khusus.
2. *Ladder Diagram*: Jenis Teknik Pemrograman *Logic* yang disusundalam satuan-satuan kontak untuk menghasilkan fungsi tertentu dalam menghasilkan logika yang terdiri dari kontak NC, NO, Timer, Counter, dan lain-lain.

2.1.5 Insruksi Dasar Pada PLC

1. *LOAD (LD)* dan *LOAD NOT (LDNOT)*

Kondisi pertama yang mengawali sembarang blok logika di dalam diagram tangga berkaitan dengan instruksi *LOAD (LD*) atau *LOAD NOT (LD NOT).* Contoh instruksi ini ditunjukkan pada gambar2.9.



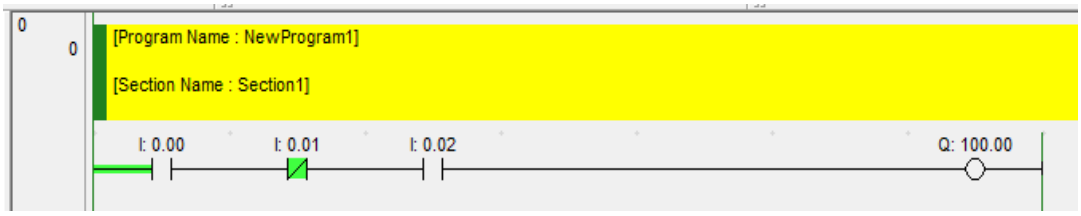
Gambar 2.10Contoh Penggunaan Instruksi *LD* dan *LDNOT*

Tabel 2.1Kode Mnemonik Instruksi LD dan LDNOT

Alamat	Instruksi	Operan
00000	*LD*	0.00
00001	*OUT*	100.00
00002	*LD NOT*	0.00
00003	*OUT*	100.01

2. *AND* dan *ANDNOT*

Jika terdapat dua atau lebih kondisi yang dihubungkan seri pada garis instruksiyang sama maka kondisi pertama menggunakan instruksi *LD*atau *LD NOT*, dan sisanya menggunakan instruksi *AND*atau *AND NOT*.Gambar menunjukkan suatu penggalan diagram tangga yang mengandung tiga kondisi yang dihubungkan secara seri pada garis instruksi yang sama dan berkaitandengan instruksi *LD, AND NOT*, dan *AND*. Masing-masing instruksi tersebut membutuhkan satu baris kodemnemonic.



Gambar 2.11 Contoh Penggunaan Instruksi *AND* dan *ANDNOT*

Tabel 2.2Kode MnemonikInstruksi *AND* dan *ANDNOT*

Alamat	Instruksi	Operan
00000	*LD*	0.00
00001	*AND NOT*	0.01
00002	*AND*	0.02
00003	*OUT*	100.00

3. *OR* dan *ORNOT*

Jika dua atau lebih kondisi yang dihubungkan paralel, artinya dalam garisinstruksi yang berbeda kemudian bergabung lagi dalam satu garis instruksi yang sama maka kondisi pertama terkait dengan instruksi *LD*dan *LDNOT* dan sisanya berkaitan dengan instruksi *OR* dan *OR NOT*.Gambar menunjukkan tiga buah instruksi yangberkaitan dengan instruksi *LD NOT, OR NOT*, dan *OR*. Masing-masing instruksi tersebut membutuhkan satu baris kode mnemonik.

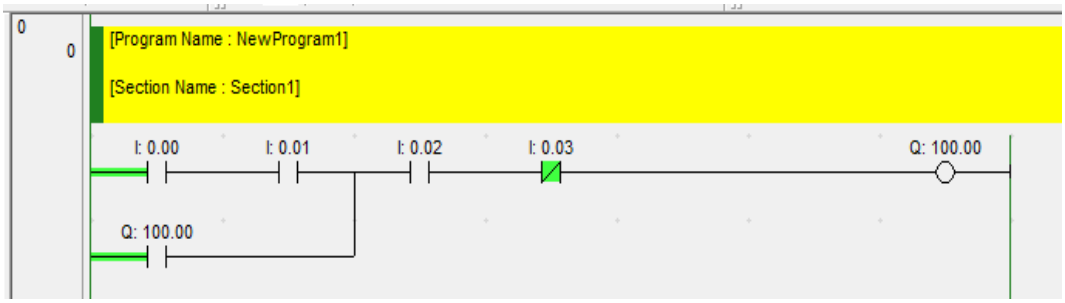
Gambar 2.12 Contoh Penggunaan Instruksi *OR* dan *ORNOT*

Tabel 2.3Kode Mnemonik Instruksi *OR* dan *ORNOT*

Alamat	Instruksi	Operan
00000	*LD NOT*	0.00
00001	*OR NOT*	0.01
00002	*OR*	0.02
00003	*OUT*	100.00

4. Kombinasi instruksi *AND* dan*OR*

Jika instruksi *AND*dan *OR* digabung atau dikombinasikan dalam suatu rangkaian tangga yang kompleks maka bisa dipandang satu persatu, artinya bisa dilihat masing-masing hasil gabungan dua kondisi menggunakan instruksi *AND* atau *OR* secara sendiri-sendiri kemudian menggabungkannya menjadi satu kondisi menggunakan instruksi *AND*atau *OR* yang terakhir. Gambar 2.12 menunjukkan contoh diagram tangga yang mengimplentasikan cara seperti tersebut di atas.



Gambar 2.13 Contoh Penggabungan Instruksi *AND* dan*OR*

Tabel 2.4 Kode MnemonikInstruksi AND danOR

Alamat	Instruksi	Operan
00000	*LD*	0.00
00001	*AND*	0.01
00002	*OR*	100.00
00003	*AND*	0.02
00004	*AND NOT*	0.03
00005	*OUT*	100.00

5. Instruksi-instruksi BlokLogika

Instruksi-instuksi blok logika tidak berhubungan dengan suatu kondisi tertentu pada diagram tangga, melainkan untuk menyatakan hubungan antar blok-blok logika, misalnya instruksi *AND LD*akan meng-*AND*-logik-kan kondisi eksekusi yang dihasilkan oleh dua blok logika, demikian juga dengan *OR LD*untuk meng-*OR* logikkan kondisi eksekusi yang dihasilkan dua blok logika.

a. *AND LOAD (ANDLD)*

Gambar 2.13 menunjukkan contoh penggunaan blok logika *AND LD* yang terdiri atas dua blok logika, yang akan menghasilkan kondisi *ON* jika blok logika kiri dalam kondisi *ON* (salah satu dari 0.00 atau 0.01 yang *ON*) danblok logika kanan juga dalam keadaan *ON*(0.02dalam kondisi *ON* atau 0.03 dalam kondisi*OFF*).

Gambar 2.14 Contoh Penggunaan Instruksi Blok Logika *ANDLD*

Tabel 2.5Kode Mnemonik Instruksi Blok Logika ANDLD

Alamat	Instruksi	Operan
00000	*LD*	0.00
00001	*OR*	0.01
00002	*LD*	0.02
00003	*OR NOT*	0.03
00004	*AND LD*	-
00005	*OUT*	100.00

b. *OR LOAD (ORLD)*

Instruksi ini digunakan untuk meng-*OR*-logik-kan dua blok logika. Gambar 2.14 menunjukkan contoh penggunaan blok logika*OR LD*yang terdiri atas dua blok logika. Kondisi eksekusi *ON*akan dihasilkan jika blok logik atas ataublok logika bawah dalam kondisi *ON*. Artinya, 0.00 dan kondisi *ON* dan 0.01 dalam kondisi *OFF* atau0.02 dan 0.03 dalam kondisi *ON*).

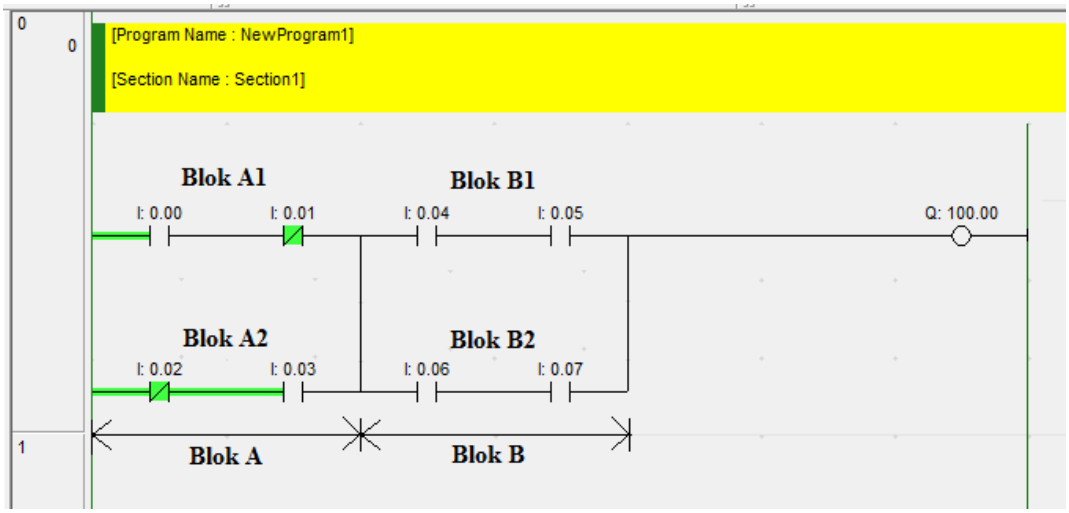
Gambar 2.15 Contoh Penggunaan Instruksi Blok Logika *OR LD*

Tabel 2.6Kode MnemonikInstruksi Blok Logika *OR LD*

Alamat	Instruksi	Operan
00000	*LD*	0.00
00001	*AND NOT*	0.01
00002	*LD*	0.02
00003	*AND*	0.03
00004	*OR LD*	-
00005	*OUT*	100.00

c. Logika Kompleks

Untuk membuat kode mnemonik diagram tangga yang kompleks, caranya dengan cara membagi membagi diagram tersebut ke dalam blok-blok logika yang besar, kemudian membagi lagi blok yang besar tersebut menjadi blok- blok logika yang lebih kecil, demikian seterusnya hingga tidak perlu lagi dibuat blok yang lebih kecil lagi. Blok-blok ini kemudian masing-masing dikodekan, mulai dari yang kecil, dan digabungkan satu per satu hingga membentuk diagram tangga yang asli. Instruksi blok logika*AND LD*dan *OR LD*hanya digunakan untuk menggabungkan dua blok logika saja(blok logika yang digabungkan berupa hasil penggabungan sebelumnya, atau hanya sebuah kondisi tunggal). Gambar 2.15memperlihatkan suatu contoh diagram tangga yang kompleks, yang dapat dibagi dua blok besar (blok A dan B). Blok A dapat dibagi lagi menjadi dua blok yang lebih kecil (blok A1 dan A2), dan blok B dibagi menjadi dua blok yang lebih kecil, yaitu blok B1 dan B2. Kemudian blok-blok logika yang kecil ini ditulis terlebih dahulu, diawali dengan menuliskan blokA1 (alamat 00000 dan 00001) dan blok A2 (alamat 00002 dan 00003), kemudian digabung menggunakan instruksi blok logika*OR LD*(alamat 00004). Selanjutnya blokB1 dituliskan (alamat 00005 dan00006) dilanjutkan dengan blok B2 (alamat 00007 dan 00008) dan digabung dengan instruksi blok logik *OR LD*(alamat 00009). Hasilnya berupa blok A dan blok B yang kemudian juga digabung menggunakan blok logika *AND LD* (alamat 00010).



Gambar 2.16 Contoh Penggunaan Instruksi Blok Logika Kompleks

Tabel 2.7 Kode MnemonikInstruksi Blok Logika Kompleks

Alamat	Instruksi	Operan
00000	*LD*	0.00
00001	*AND NOT*	0.01
00002	*LD NOT*	0.02
00003	*AND*	0.03
00004	*OR LD*	-
00005	*LD*	0.04
00006	*AND*	0.05
00007	*LD*	0.06
00008	*AND*	0.07
00009	*OR LD*	-
00010	*AND LD*	-
00011	*OUT*	100.00

6. Intruksi KendaliBit

Terdapat instruksi dasar yang dapat digunakan untuk mengontrol status bit secara individual, yaitu *DIFFERENTIATE UP* (DIFU),*DIFFERENTIATE DOWN* (DIFD) instruksi ini dituliskan disisi palingkanan diagram tangga dan

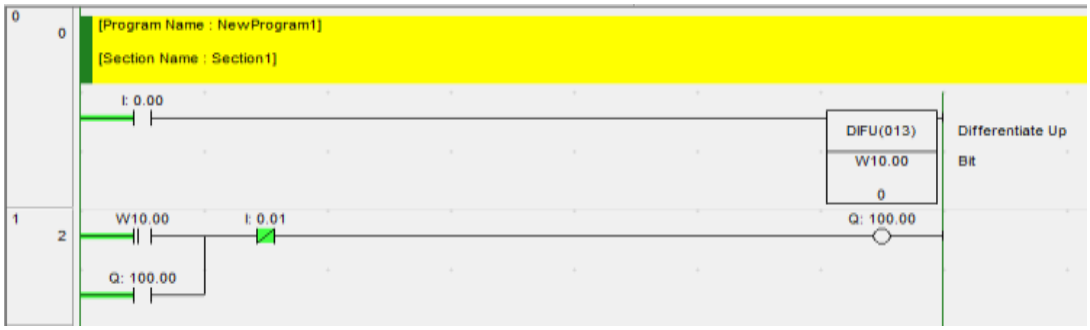
membutuhkan sebuah alamat bit sebagai operan. Selain instruksi-instruksi ini digunakan untuk membuat bit-bit keluaran *ON* atau *OFF* dalam area IR (ke piranti *eksterna*l), ini juga digunakan untuk mengontrol bit-bitlainnya.

Kedua instruksi ini sangat sering sekali digunakan dalam pemrograman PLC. Kedua instruksi ini masuk ke dalam jenis *ladder instructions,* pada sub kategori *bit control instructions.*

Untuk penjelasan mengenai instruksi DIFU dan DIFD lihat gambar berikut:

Gambar 2.17Prinsip Kerja Instruksi Kendali Bit DIFU dan DIFD

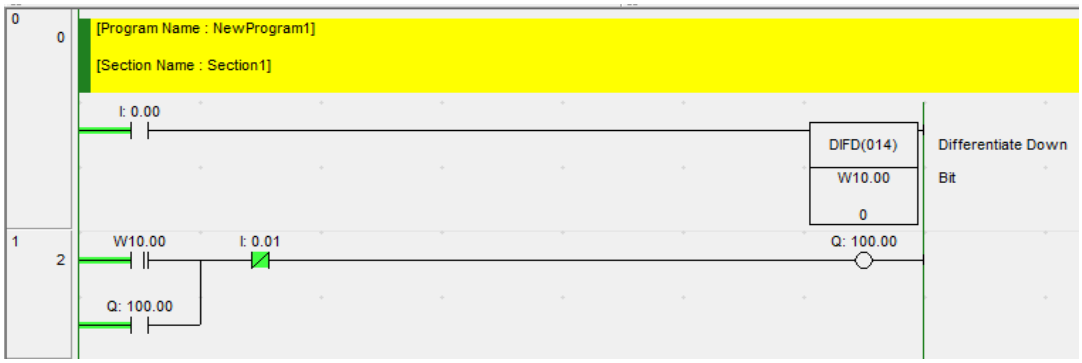
Jadi seperti terlihat pada gambar di atas, baik instruksi DIFU maupun instruksi DIFD *output ON* nya (warna hitam pada gambar) hanya sekali dan dalam waktu yang singkat saja, atau biasa disebut *one scan only*. sedangkan perbedaan dari instruksi DIFU dan DIFD, bahwa instruksi DIFU ini akan *ON* (tentunya dalam waktu singkat saja) saat input baru saja mengalami perubahan dari *OFF* ke *ON*. Sedangkan pada instruksi DIFD, akan*ON* (dalam waktu singkat saja) saat input baru saja mengalami perubahan dari *ON* ke *OFF*.



Gambar 2.18 Contoh Penggunaan Instruksi Kendali Bit *DIFFERENTIATE UP* (DIFU*)*

Tabel 2.8Kode Mnemonik InstruksiKendali Bit *DIFFERENTIATE UP* (DIFU)

Alamat	Instruksi	Operan
00000	*LD*	0.00
00001	*DIFU*	W10.00
00002	*LD*	W10.00
00003	*OR*	100.00
00004	*AND NOT*	0.01
00005	*OUT*	100.00



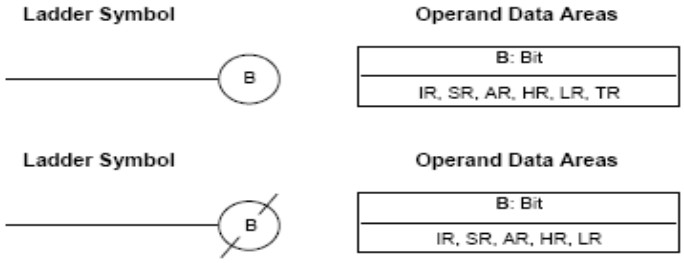
Gambar 2.19 Contoh Penggunaan Instruksi Kendali Bit *DIFFERENTIATE DOWN* (DIFD)

Tabel 2.9Kode Mnemonik Instruksi Kendali Bit *DIFFERENTIATE DOWN* (DIFD)

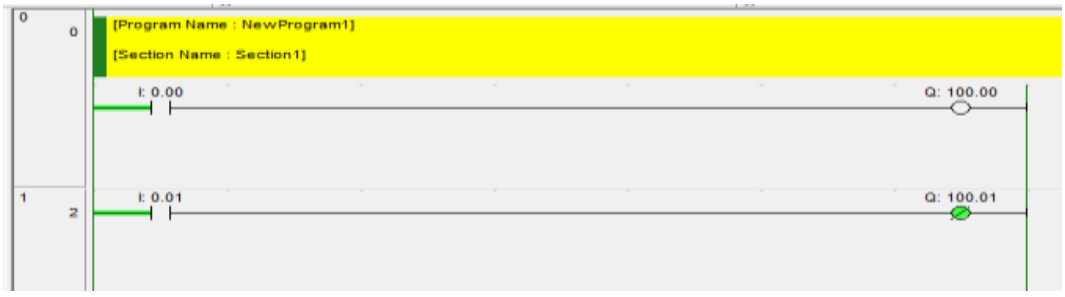
Alamat	Instruksi	Operan
00000	*LD*	0.00
00001	*DIFD*	W10.00
00002	*LD*	W10.00
00003	*OR*	100.00
00004	*AND NOT*	0.01
00005	*OUT*	100.00

7. Instruksi *OUTPUT (OUT)* dan *OUTPUT NOT (OUTNOT)*

Instruksi ini digunakan untuk mengontrol operan yang berkaitan dengan kondisieksekusi (apakah *ON* atau *OFF*). Dengan menggunakan instruksi *OUT*, maka bit operanakan menjadi *ON* jika kondisi eksekusinya juga *ON*, sedangkan *OUT NOT* akanmenyebabkan bit operan menjadi *ON* jika kondisi eksekusinya *OFF*. Gambar 2.19memperlihatkan simbol tangga dan area data operan dari instruksi *OUT* dan *OUT NOT*,sedangkan Gambar 2.20 memperlihatkan contoh implementasi kedua instruksitersebut.



Gambar 2.20 Simbol Tangga Dan Area Data Operan Instruksi *OUT* dan*OUT NOT*



Gambar 2.21 Contoh Penggunaan Instruksi *OUT* dan*OUTNOT*

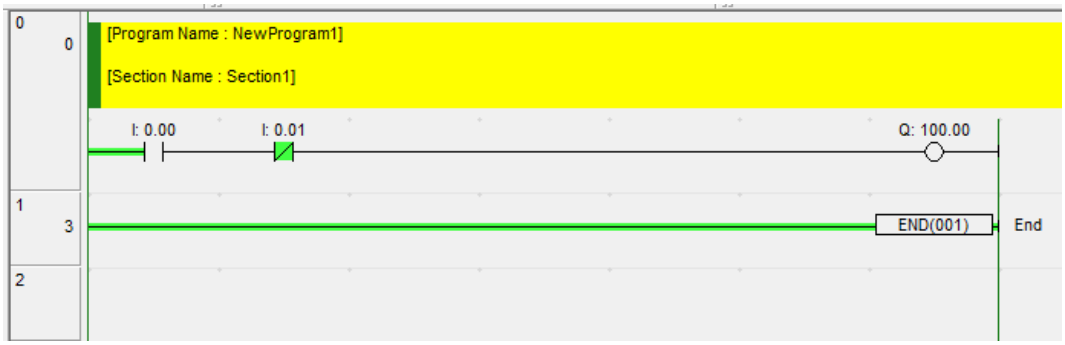
Tabel 2.10 Kode Mnemonik Instruksi *OUT* dan*OUTNOT*

Alamat	Instruksi	Operan
00000	*LD*	0.00
00001	*OUT*	100.00
00002	*LD*	0.01
00003	*OUT NOT*	100.01

8. Instruksi*END*

Instruksi *END* merupakan instruksi terakhir yang harus dituliskan atau digambarkan dalam diagram tangga. CPU pada PLC akan mengerjakan semua instruksi dalam program dari awal (baris pertama) sampaiditemui instruksi *END* yang pertama, sebelum kembali lagi mengerjakan instruksi dalam program dari awal (artinya instruksiinstruksi yang ada di bawah instruksi *END* akan diabaikan). Instruksi *END* tidak memerlukan operan dan tidak boleh diawali dengan suatu kondisi seperti pada instruksilainnya.

Suatu diagram tangga atau program PLC harus diakhiri dengan instruksi *END*, jika tidak maka program tidak dijalankan sama sekali. Angka yang dituliskan pada instruksi *END* pada kode mnemonik merupakan kode fungsinya.Gambar 2.21 memperlihatkan contoh penggunaan instruksi*END*.



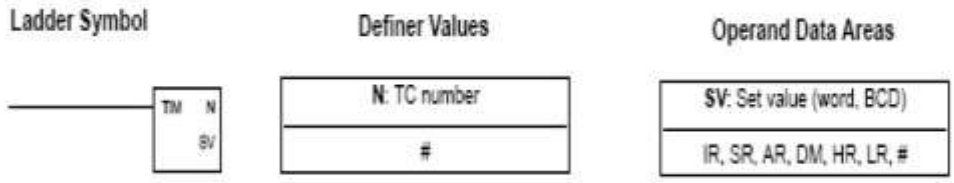
Gambar 2.22 Contoh Penggunaan Instruksi *END*

Tabel 2.11 Kode Mnemonik Instruksi *END*

Alamat	Instruksi	Operan
00000	*LD*	0.00
00001	*AND NOT*	0.01
00002	*OUT*	100.00
00003	*END(001)*	-

9. Instruksi *TIMER*(TIM)

Instruksi TIM dapat digunakan sebagai timer (pewaktu) ON-delay pada rangkaian relai. Gambar 2.22 memperlihatkan simbol tangga dan area data operan dari instruksi TIM. Instruksi TIM membutuhkan angka *timer* (N), dan nilai set (SV) antara 0000 sampai 9999 (artinya 000,0 sampai 999,9detik).



Gambar 2.23 Simbol Tangga Dan Area Data Operan Dari Instruksi*TIME*R (TIM)

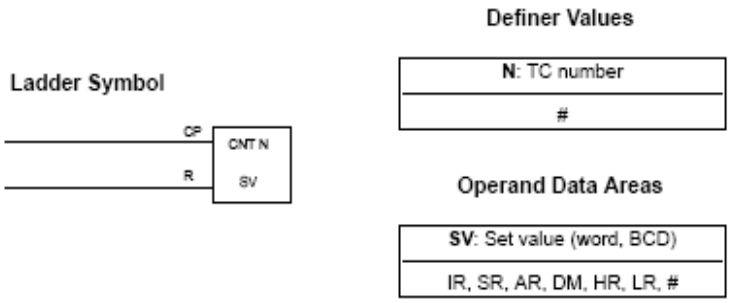
Gambar 2.24 Contoh Penggunaan Instruksi*TIMER* (TIM)

Tabel 2.12 Kode Mnemonik Instruksi*TIMER* (TIM)

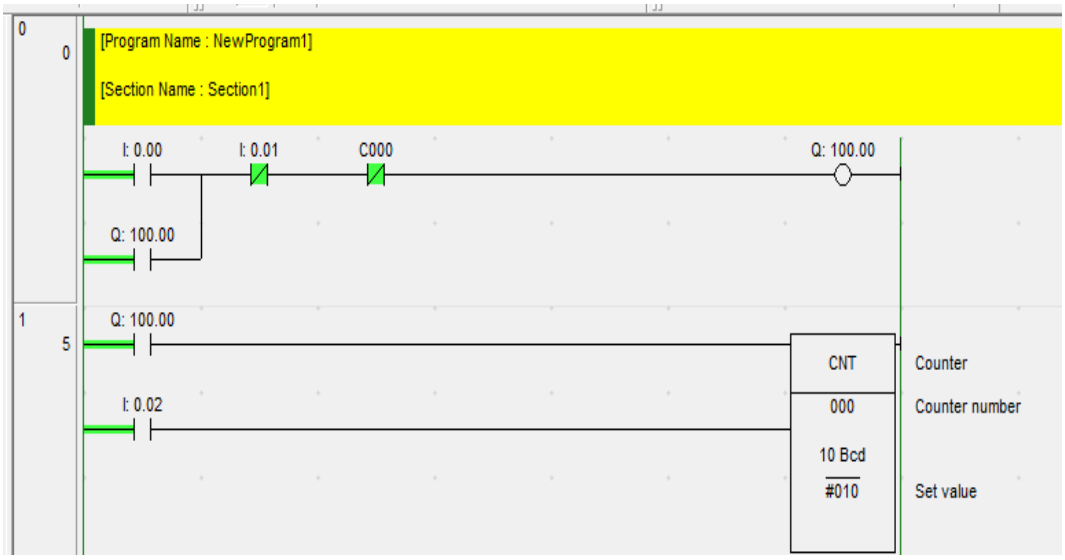
Alamat	Instruksi	Operan
00000	*LD*	0.00
00001	*OR*	100.00
00002	*AND NOT*	0.01
00003	*TIM*	000 #010
00004	*AND NOT*	T000
00005	*OUT*	100.00

10. Instruksi *COUNTER*(CNT)

CNT yang digunakan di sini adalah counter penurunan yang diset awal. Penurunan satu hitungan setiap kali saat sebuah sinyal berubah dari *OFF* ke *ON*. Counter harus diprogram dengan input hitung, input reset, angka counter, dan nilai set (SV) Nilai set ini adalah 0000 smpai 9999. Gambar 2.24 memperlihatkan simbol tangga dan area data operan dari instruksiCNT.Dan Gambar 2.25 memperlihatkan contoh penggunaan instruksi*END*.



Gambar 2.25 Simbol Tangga Dan Area Data Operan Dari Instruksi*COUNTER* (CNT)



Gambar 2.26 Contoh Penggunaan Instruksi *COUNTER* (CNT)

Tabel 2.13Kode Mnemonik Instruksi *COUNTER* (CNT)

Alamat	Instruksi	Operan
00000	*LD*	0.00
00001	*OR*	100.00
00002	*AND NOT*	0.01
00003	*AND NOT*	C000
00004	*OUT*	100.00
00005	*LD*	100.00
00006	*LD*	0.02
00006	*CNT*	000 #010

2.1.6 Perangkat – Perangkat *Input*[8]

Pada dasarnya PLC memerlukan sebuah input untuk melakukan perintah yang sudah diprogram dan di *download* ke PLC. Umumnya PLC hanya memerlukan *input* oleh sebuah saklar mekanis dan sensor.Sebuah saklar mekanis menghasilkan sinyal (atau sinyal-sinyal) hidup / mati sebagai akibat dari tertutup atau terbukanya saklar oleh suatu input mekanis. Dipasaran tersedia saklar-saklar dengan kontak *normally open* (normal terbuka) (NO) atau *normally closed* (normal tertutup) (NC) atau kontak yang dapat diatur sesuai kebutuhan dengan memilih konntak-kontak yang tepat.

2.1.7 Perangkat-perangkat *Output*[9]

Port-port output sebuah PLC dapat berupa tipe relay atau tipe isolator-optik dengan transistor atau tipe triac, bergantung pada perangkat yang tersambung padanya, yang akan dikontrol. Secara umum, sinyal digital dari salah satu kanal *output* sebuah PLC digunakan untuk mengontrol sebuah *actuator* yang pada giliranya mengontrol sebuah proses. Istilah *actuator* digunakan untuk perangkat yang mengubah sinyal listrik menjadi gerakan-gerakan mekanik yang kemudian

[8]William Bolton, *Programmable Logic Controller (PLC) Sebuah Pengantar Edisi Ketiga* (Jakarta: Erlangga, 2003), hlm. 15.
[9]William Bolton, *Programmable Logic Controller (PLC) Sebuah Pengantar Edisi Ketiga* (Jakarta: Erlangga, 2003), hlm. 24.

digunakan untuk mengontrol proses. Berikut ini adaalah beberapa contoh dari *output* PLC.

2.2 Arduino

Arduino adalah pengendali mikro *single-board* yang bersifat *open-source*, diturunkan dari *Wiring platform*, dirancang untuk memudahkan penggunaan elektronik dalam berbagai bidang. Hardwarenya memiliki prosesor ***Atmel AVR*** dan softwarenya memiliki bahasa pemrograman sendiri.

Arduino juga merupakan *platform hardwar*e terbuka yang ditujukan kepada siapa saja yang ingin membuat purwarupa peralatan elektronik interaktif berdasarkan hardware dan *software* yang fleksibel dan mudah digunakan.Mikrokontroler diprogram menggunakan bahasa pemrograman arduino yang memiliki kemiripan *syntax* dengan bahasa pemrograman C. Karena sifatnya yang terbuka maka siapa saja dapat mengunduh skema *hardware* arduino dan membangunnya.

Gambar 2.27 Arduino Atmega 2560

2.3 Tombol Tekan

Prinsip kerja tombol tekan hampir sama dengan saklar tekan yang digunakan pada instalasi penerangan, bedanya jika saklar tekan jenis yang mempunyai togel akan langsung mengikat/mengunci, sedangkan pada tombol tekan tidak ada. Jadi tombol tekan setelah ditekan tidak akan mengunci, tetapi kembali keadaannya semula. Ada dua kontak yang dapat dilakukan oleh tombol tombol tekan, yaitu:

1. Kontak NO (*Normally Open*) Biasanya berwarnahijau.
2. Kontak NC (*Normally Close*) Biasanya berwarnaMerah.

Gambar 2.28Tombol Tekan Kontak NO dan Kontak NC

2.4 Saklar Pemilih (*Selector Switch*)

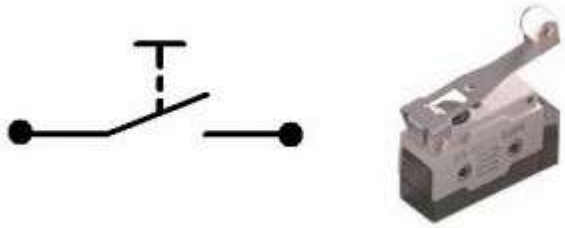
Saklar jenis ini pada umumnya tersedia dua, tiga atau empat pilihan posisi, dengan berbagai tipe knop. Saklar pemilih biasanya dipasang pada panel kontrol untuk memilih jenis operasi yang berbeda, dengan rangkaian yang berbeda pula. Saklar pemilih memiliki beberapa kontak dan setiap kontak dihubungkan oleh kabel menuju rangkaian yang berbeda, misal untuk rangkaian putaran motor cepat dan untuk rangkaian putaran motor lambat.

Selector Switch, alat ini di gunakan untuk memilih, banyak sekali type selector switch, tapi biasanya hanya dua type yang sering di gunakan, yaitu 2posisi,(*ON-OFF/Start-Stop*/0-1,dan lain-lain)dan 3 posisi (*ON-OFF-ON/Auto-Off-*Manual,dan lain-lain)dengan selector switch, kondisi peralatan dapat langsung di ketahui dari penunjukan tangkai *selector switch*, dengan *selector switch*, rangkaian *ON-OFF* lebih sederhana, karena *selector switch* tidak seperti tombol tekan yang hanya kontak sementara.

Gambar 2.29 *Selector Switch*

2.5 Limit Switch

Limit switch merupakan jenis saklar yang dilengkapi dengan katup yang berfungsi menggantikan tombol. Prinsip kerja *limit switch* sama seperti saklar *Push ON* yaitu hanya akan menghubung pada saat katupnya ditekan pada batas penekanan tertentu yang telah ditentukan dan akan memutus saat saat katup tidak ditekan. *Limit switch* termasuk dalam kategori sensor mekanis yaitu sensor yang akan memberikan perubahan elektrik saat terjadi perubahan mekanik pada sensor tersebut. Penerapan dari *limit switch* adalah sebagai sensor posisi suatu benda (objek) yang bergerak. Simbol *limit switch* ditunjukan pada gambar berikut.
imboldan bentuk *Limit Switch*

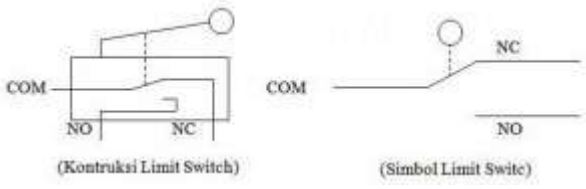


Limit switch umumnya digunakan untuk :

- Memutuskan dan menghubungkan rangkaian menggunakan objek atau benda lain.
- Menghidupkan daya yang besar, dengan sarana yang kecil.
- Sebagai sensor posisi atau kondisi suatu objek.

Prinsip kerja limit switch diaktifkan dengan penekanan pada tombolnya pada batas/daerah yang telah ditentukan sebelumnya sehingga terjadi pemutusan atau penghubungan rangkaian dari rangkaian tersebut. Limit switch memiliki 2 kontak yaitu NO (*Normally Open*) dan kontak NC (*Normally Close*) dimana salah satu kontak akan aktif jika tombolnya tertekan. Konstruksi dan simbol limit switch dapat dilihat seperti gambar di bawah.

Konstruksi Dan Simbol Limit Switch



Gambar 2.30 Limit Switch

2.6 Sensor RFID

RFID adalah singkatan dari *Radio Frequency Identification*. RFID adalah sistem identifikasi tanpa kabel yang memungkinkan pengambilan data tanpa harus bersentuhan seperti *barcode* dan *magnetic card* seperti ATM. RFID kini banyak dipakai diberbagai bidang seperti perusahaan, supermarket, rumah sakit bahkan terakhir digunakan dimobil untuk identifikasi penggunaan BBM bersubsidi bahkan dapat diaplikasikan untuk sistem parkiran otomatis.

Gambar 2.31 RFID

2.7 LampuTanda

Lampu tanda atau biasa disebut juga *pilot lamp* digunakan pada peralatan kontrol untuk menandai bekerja atau tidaknya suatu peralatan atau rangkaian, dapat juga sebagai kondisi/keadaan beban.Jika lampu tanda dipergunakan untuk menandai suatu peralatan yang sedang bekerja, maka lampu tanda dipasang seri pada kontakNO, sedangkan apabila lampu tanda digunakan untuk menandai tidak bekerjanya suatuperalatan, maka lampu tanda dipasang paralel pada kontak NC pada rangkaian yang mengontrol peralatan tersebut.Jika lampu tanda dipergunakan untuk menandai keadaan suatu peralatan/beban, maka lampu tanda mempergunakan warna-warna yang berbeda-beda bergantung pada kondisi peralatan/beban yang ditandai. Tabel 2.1 dibawah ini merupakan warna- warna yang menunjukkan fungsi dari lampu tanda.

Tabel 2.14Fungsi Warna LampuTanda

Kondisiperalatan/Beban	Warnalampu
Kondisi tidak normal, beban lebih bahaya.	Merah
Hati-hati,perhatian	Kuning
Posisi siap, mulaiberoperasi	Hijau
Beroperasinormal	Putih

Lampu tanda tidak jauh berbeda dengan lampu penerangan biasa, biasanyalampu ini mempunyai tahanan dalam yang besar sehingga dayanya rata-rata kecil.Lampu tanda juga sama seperti lampu penerangan biasa yang mempunyai bentuk bermacam-macam yang biasa dilihat pada gambar 2.29 dibawah ini.

Gambar 2.32 LampuTanda

2.8 Relay

Relay adalah suatu piranti yang bekerja berdasarkan elektromagnetik untuk menggerakan sejumlah kontaktor (saklar) yang tersusun. Relay akan tertutup (*ON*) atau terbuka (*OFF)* karena efek induksi magnet yang dihasilkan kumparan (induktor) ketika dialiri arus listrik. Berbeda dengan saklar dimana pergerakan Relay (*ON/OFF*) dilakukan manual tanpa perlu arus listrik.

Sebagai komponen elektronika, relay mempunyai peran penting dalam sebuah sistem rangkaian elektronika dan rangkaian listrik untuk menggerakan sebuah perangkat yang memerlukan arus besar tanpa terhubung langsung dengan perangkat pengendali yang mempunyai arus kecil. Dengan demikian relay dapat berfungsi sebagai pengaman.

Ada beberapa jenis relay berdasarkan cara kerjanya yaitu:

- *Normaly On* : Kondisi awal kontaktor terturup (*On*) dan akan terbuka (*Off*) jika

relay diaktifkan dengan cara memberi arus yang sesuai pada kumparan (*Coil*) relay. Istilah lain kondisi ini adalah *Normaly Close*(NC).

- *Normaly Off* : Kondisi awal kontaktor terbuka *(Off*) dan akan tertutup jika relay diaktifkan dengan cara memberi arus yang sesuai pada kumparan (*Coil*) relay. Istilah lain kondisi ini adalah *Normaly Open*(NO).
- *Change-Over* (CO) atau *Double-Throw* (DT) : Relay jenis ini memiliki dua pasang terminal dengan dua kondisi yaitu *Normaly Open* (NO) dan *Normaly Close*(NC).

Gambar 2.33 Relay

2.9 *Buzzer*

Buzzer adalah sebuah komponen elektronika yang berfungsi untuk mengubah getaran listrik menjadi getaran suara. Pada dasarnya prinsip kerja *buzzer* hampir sama dengan *loud speaker*, jadi *buzzer* juga terdiri dari kumparan yang terpasang pada diafragma dan kemudian kumparan tersebut dialiri arus sehingga menjadi elektromagnet, kumparan tadi akan tertarik ke dalam atau keluar, tergantung dari arah arus dan polaritas magnetnya, karena kumparan dipasang pada diafragma maka setiap gerakan kumparan akan menggerakkan diafragma secara bolak-balik sehingga membuat udara bergetar yang akan menghasilkan suara. *Buzzer* biasa digunakan sebagai indikator bahwa proses telah selesai atau terjadi suatu kesalahan pada sebuah alat (alarm).

Gambar 2.34 Buzzer

2.10 *Liquid Crystal Display* (LCD) 20 x 4

Liquid Crystal Display(LCD) adalah suatu jenis media tampil yang menggunakan kristal cair sebagai penampil utama. LCD sudah digunakan diberbagai bidang misalnya alal–alat elektronik seperti televisi, kalkulator, atau pun layar komputer. Pada postingan aplikasi LCD yang dugunakan ialah LCD dot matrik dengan jumlah karakter 20 x 4. LCD sangat berfungsi sebagai penampil yang nantinya akan digunakan untuk menampilkan status kerja alat.

- Fitur LCD 20 x 4

Adapun fitur yang disajikan dalam LCD ini adalah :

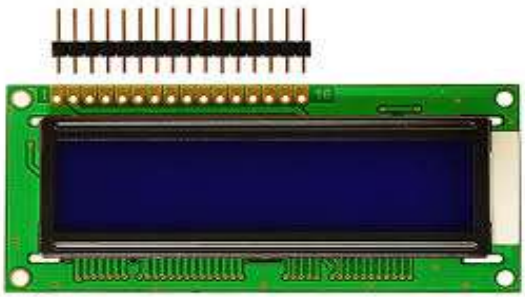
a. Terdiri dari 20 karakter dan 4 baris.

b. Mempunyai 192 karakter tersimpan.

c. Terdapat karakter generator terprogram.

d. Dapat dialamati dengan mode 4-bit dan 8-bit.

e. Dilengkapi dengan back light.



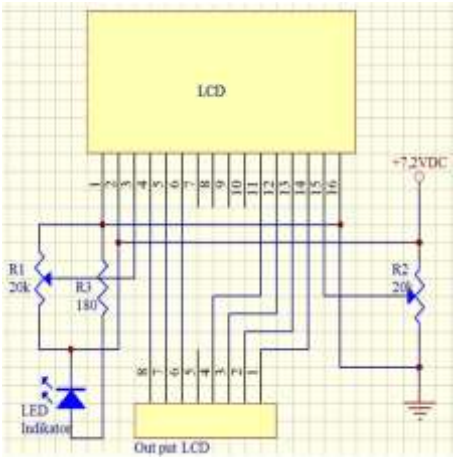
Gambar 2.35 Bentuk Fisik LCD 20 x 4

- Cara Kerja LCD Secara Umum

Pada aplikasi umumnya RW diberi logika rendah “0”. Bus data terdiri dari 4-bit atau 8-bit. Jika jalur data 4-bit maka yang digunakan ialah DB4 sampai dengan DB7. Sebagaimana terlihat pada table diskripsi, interface LCD merupakan sebuah parallel bus, dimana hal ini sangat memudahkan dan sangat cepat dalam pembacaan dan penulisan data dari atau ke LCD. Kode ASCII yang ditampilkan sepanjang 8-bit dikirim ke LCD secara 4-bit atau 8 bit pada satu waktu. Jika mode 4-bit yang digunakan, maka 2 nibble data dikirim untuk membuat sepenuhnya 8-bit (pertama dikirim 4-bit MSB lalu 4-bit LSB dengan pulsa clock EN setiap nibblenya). Jalur kontrol EN digunakan untuk memberitahu LCD bahwa mikrokontroller mengirimkan data ke LCD. Untuk mengirim data ke LCD program harus menset EN ke kondisi high “1” dan kemudian menset dua jalur kontrol lainnya (RS dan R/W) atau juga mengirimkan data ke jalur data bus.Saat jalur lainnya sudah siap, EN harus diset ke “0” dan tunggu beberapa saat (tergantung pada datasheet LCD), dan set EN kembali ke high “1”. Ketika jalur RS berada dalam kondisi low “0”, data yang dikirimkan ke LCD dianggap sebagai sebuah perintah atau instruksi khusus (seperti bersihkan layar, posisi kursor dll). Ketika RS dalam kondisi high atau “1”, data yang dikirimkan adalah data ASCII yang akan ditampilkan dilayar. Misal, untuk menampilkan huruf “A” pada layar maka RS harus diset ke “1”. Jalur kontrol R/W harus berada dalam kondisi low (0) saat informasi pada data bus akan dituliskan ke LCD. Apabila R/W berada dalam kondisi high “1”, maka program akan melakukan query (pembacaan) data dari LCD. Instruksi pembacaan hanya satu, yaitu Get LCD status (membaca status LCD), lainnya merupakan instruksi penulisan. Jadi hampir setiap aplikasi yang menggunakan LCD, R/W selalu diset ke “0”. Jalur data dapat terdiri 4 atau 8 jalur (tergantung mode yang dipilih pengguna), DB0, DB1, DB2, DB3, DB4, DB5, DB6 dan DB7. Mengirim data secara parallel baik 4-bit atau 8-bit merupakan 2 mode operasi primer. Untuk membuat sebuah aplikasi interface LCD, menentukan mode operasi merupakan hal yang paling penting.Mode 8-bit sangat baik digunakan ketika kecepatan menjadi keutamaan dalam sebuah aplikasi dan setidaknya minimal tersedia 11 pin I/O (3 pin untuk kontrol, 8 pin untuk data). Sedangkan mode 4 bit minimal hanya membutuhkan 7-

bit (3 pin untuk kontrol, 4 pin untuk data).

Bit RS digunakan untuk memilih apakah data atau instruksi yang akan ditransfer antara mikrokontroller dan LCD. Jika bit ini di set (RS = 1), maka byte pada posisi kursor LCD saat itu dapat dibaca atau ditulis. Jika bit ini di reset (RS = 0), merupakan instruksi yang dikirim ke LCD atau status eksekusi dari instruksi terakhir yang dibaca.untuk gambar skematik LCD 20 x 4 adalah sebagai berikut:



Gambar 2.36 Skematik LCD 20 x 4

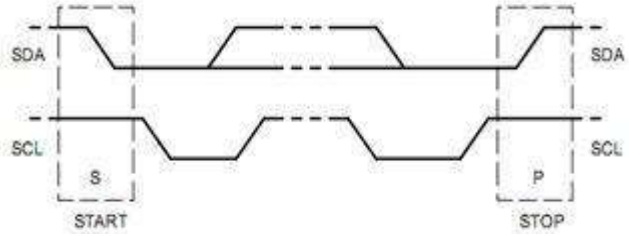
2.11 I2C[10]

Inter Integrated Circuit atau sering disebut I^2C adalah standar komunikasi serial dua arah menggunakan dua saluran yang didisain khusus untuk mengirim maupun menerima data. Sistem I^2C terdiri dari saluran SCL *(Serial Clock)* dan SDA *(Serial Data)* yang membawa informasi data antara I^2C dengan pengontrolnya. Piranti yang dihubungkan dengan sistem I2C Bus dapat dioperasikan sebagai *Master* dan *Slave*. *Master* adalah piranti yang memulai transfer data pada I^2C Bus dengan membentuk sinyal *Start*, mengakhiri transfer data dengan membentuk sinyal *Stop*, dan membangkitkan sinyal *clock*. *Slave* adalah piranti yang dialamati master.

Sinyal *Start* merupakan sinyal untuk memulai semua perintah, didefinisikan sebagai perubahan tegangan SDA dari "1" menjadi "0" pada saat SCL "1".

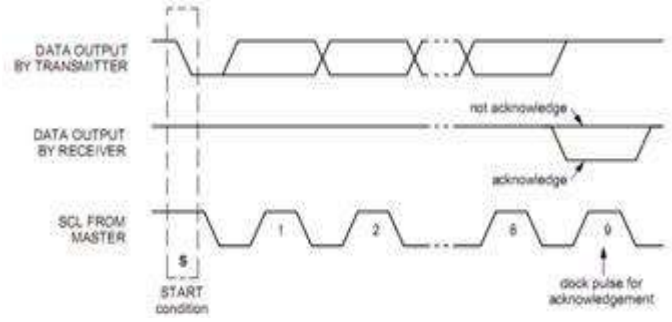
[10]Sejati,Purnomo,*Mengenal Komunikasi I2C (Inter Circuit),* (https://purnomosejati.wordpress.com/2011/08/25/mengenalkomunikasii2cinterintegrated-circuit/), diakses 3 Mei 2017).

Sinyal *Stop* merupakan sinyal untuk mengakhiri semua perintah, didefinisikan sebagai perubahan tegangan SDA dari “0” menjadi “1” pada saat SCL “1”. Kondisi sinyal *Start* dan sinyal *Stop* seperti tampak pada Gambar 1.



Gambar 2.37 Kondisi Sinyal *Start* dan

Sinyal dasar yang lain dalam I^2C Bus adalah sinyal *acknowledge* yang disimbolkan dengan ACK Setelah transfer data oleh *master* berhasil diterima *slave, slave* akan menjawabnya dengan mengirim sinyal *acknowledge*, yaitu dengan membuat SDA menjadi “0” selama siklus *clock*ke 9. Ini menunjukkan bahwa *Slave* telah menerima 8 bit data dari *Master*. Kondisi sinyal *acknowledge* seperti tampak pada Gambar 2.



Gambar 2.38 Sinyal I2C

2.12Motor Arus Searah(DC)[11]

Motor listrik merupakan perangkat elektromagnetis yang mengubah energi listrik menjadi energi mekanik. Energi mekanik ini digunakan untuk, misalnya memutar *impeller* pompa, *fan* atau *blower*, menggerakan kompresor, mengangkat bahan dan lain-lain. Motor listrik digunakan juga di rumah (*mixer*,

[11]Zuhal, *Dasar Teknik Tenaga Listrik dan Elektronika Daya* (Jakarta: Gramedia, 1988).hlm. 57

bor listrik, kipas angin) dan di industri. Motor listrik kadangkala disebut "kuda kerja" nya industri sebab diperkirakan bahwa motor menggunakan sekitar 70% beban listrik total diindustri.

Gambar 2.39 Motor Arus Searah (DC)[12]

Motor DC memerlukan suplai tegangan yang searah pada kumparan medan untuk diubah menjadi energi mekanik. Kumparan medan pada motor DC disebut stator (bagian yang tidak berputar) dan kumparan jangkar disebut rotor (bagian yang berputar). Jika terjadi putaran pada kumparan jangkar dalam pada medan magnet, maka akan timbul tegangan (GGL) yang berubah-ubah arah pada setiap setengah putaran, sehingga merupakan tegangan bolak-balik. Prinsip kerja dari arus searah adalah membalik phasa tegangan dari gelombang yang mempunyai nilai positif dengan menggunakan komutator,dengan demikian arus yang berbalik arah dengan kumparan jangkar yang berputar dalam medan magnet.Bentuk motor paling sederhana memiliki kumparan satu lilitan yang bisa berputar bebas di antara kutub-kutub magnet permanen.

Motor DC memiliki 2 bagian dasar :

- Bagian yang tetap/stasioner yang disebut stator. Stator ini menghasilkan medan magnet, baik yang dibangkitkan dari sebuah koil (elektro magnet) ataupun magnet permanen.
- Bagian yang berputar disebut rotor. Rotor ini berupa sebuah koil dimana arus listrik mengalir.

[12]Zuhal, *Dasar Teknik Tenaga Listrik dan Elektronika Daya* (Jakarta: Gramedia, 1988).hlm. 57

2.12.1 Bagian-Bagian Motor DC

Bagian Atau Komponen Utama Motor DC adalah sebagai berikut :

1. Kutub Medan Magnet

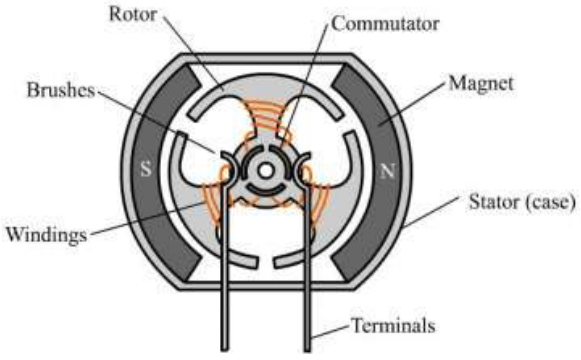
Secara sederhada digambarkan bahwa interaksi dua kutub magnet akan menyebabkan perputaran pada motor DC. Motor DC memiliki kutub medan yang stasioner dan kumparan motor DC yang menggerakan bearing pada ruang diantara kutub medan. Motor DC sederhana memiliki dua kutub medan: kutub utara dan kutub selatan. Garis magnetik energi membesar melintasi bukaan diantara kutub-kutub dari utara ke selatan. Untuk motor yang lebih besar atau lebih komplek terdapat satu atau lebih elektromagnet. Elektromagnet menerima listrik dari sumber daya dari luar sebagai penyedia struktur medan.

2. Kumparan Motor DC

Bila arus masuk menuju kumparan motor DC, maka arus ini akan menjadi elektromagnet. kumparan motor DC yang berbentuk silinder, dihubungkan ke as penggerak untuk menggerakan beban. Untuk kasus motor DC yang kecil, kumparan motor DC berputar dalam medan magnet yang dibentuk oleh kutub-kutub, sampai kutub utara dan selatan magnet berganti lokasi. Jika hal ini terjadi, arusnya berbalik untuk merubah kutub-kutub utara dan selatan kumparan motor DC.

3. Kommutator Motor DC

Komponen ini terutama ditemukan dalam motor DC. Kegunaannya adalah untuk membalikan arah arus listrik dalam kumparan motor DC. Commutator juga membantu dalam transmisi arus antara kumparan motor DC dan sumber daya.



Gambar 2.40 Komponen Motor DC

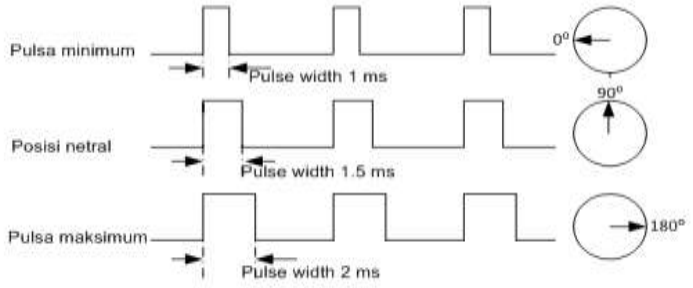
Keuntungan utama motor DC adalah sebagai pengendali kecepatan, yang tidak mempengaruhi kualitas pasokan daya. Motor ini dapat dikendalikan dengan mengatur :

- Tegangan dinamo - meningkatkan tegangan dinamo akan meningkatkan kecepatan.
- Arus medan - menurunkan arus medan akan meningkatkan kecepatan.

2.13 Motor Servo[13]

Motor servo adalah sebuah perangkat atau aktuator putar (motor) yang dirancang dengan sistem kontrol umpan balik loop tertutup (servo), sehingga dapat di set-up atau di atur untuk menentukan dan memastikan posisi sudut dari poros output motor. motor servo merupakan perangkat yang terdiri dari motor DC, serangkaian gear, rangkaian kontrol dan potensiometer. Serangkaian gear yang melekat pada poros motor DC akan memperlambat putaran poros dan meningkatkan torsi motor servo, sedangkan potensiometer dengan perubahan resistansinya saat motor berputar berfungsi sebagai penentu batas posisi putaran poros motor servo.

Terdapat 2 tipe motor servo, yaitu servo standar dan servo *continous*, dimana biasanya standar hanya dapat melakukan pergerakan sebesar 180^0, sedangkan untuk tipe *continuous* dapat melakukan rotasi atau 360^0.Sebagaimana diperlihatkan pada gambar dibawah ini.



Gambar 2.41 Perubahan sudut putar motor servo[14]

[13]Mada sanjaya, Panduan praktis pemprograman"ROBOT VISION" menggunakan Matlab dan Arduino, hlm 22.

[14]Mada sanjaya, Panduan praktis pemprograman"ROBOT VISION" menggunakan Matlab dan Arduino, hlm 23.

2.14 Pengenalan parkir[15]

Parkir adalah keadaan tidak bergerak suatu kendaraan yang tidak bersifat sementara, sedangkan fasilitas parkir adalah lokasi yang ditentukan sebagai tempat pemberhentian kendaraan yang tidak bersifat sementara untuk me-lakukan kegiatan pada suatu kurun waktu.Berdasarkan jenisnya parkir dapat dibedakan dalam beberapa tipe, yakni:

2.14.1 Parkir Menurut Tempat

1. *On Street Parking*

   Parkir jenis ini mengambil tempat di sepanjang jalan, dengan atau tanpa melebarkan jalan untuk faslitas parkir.

2. *Off Street Parking*

   Parkir jenis ini menempati pelataran parkir tertentu diluar badan jalan baik halaman terbuka atau di dalam bangunan khusus untuk parkir.

2.14.2 Parkir Menurut Posisi

1. Parkir Sejajar Sumbu jalan (180°)
2. Parkir Bersudut 30°, 45°, dan 60° dengan sumbu jalan
3. Parkir tegak Lurus sumbu Jalan (90°)

2.14.3 Parkir Menurut Status

1. Parkir Umum

   Perparkiran yang menggunakan tanah-tanah, jalan-jalan atau lapangan yang dimiliki dan dikelola oleh Pemerintah Daerah.

2. Parkir Khusus

   Perparkiran yang menggunakan tanah-tanah yang dikuasai dan pengelolaannya diselenggarakan oleh pihak ketiga.

3. Parkir Darurat

   Perparkiran di tempat-tempat umum, baik di tanah, jalan, lapangan

[15]Direktorat Jendral Perhubungan Darat.(1996).*Pedoman Teknis Peyelenggaraan Fasilitas Parkir*.Jakarta:Direktorat Jendral Perhubungan Darat.

milik Pemerintah Daerah atau swasta karena kegiatan insidentil.

4. Taman Parkir

Suatu areal bangunan perparkiran yang dilengkapi fasilitas sarana perparkiran yang pengelolaannya diselenggarakan oleh Pemerintah Daerah.

5. Gedung Parkir

Bangunan yang dimanfaatkan untuk tempat parkir yang diselenggarakan oleh Pemerintah Daerah atau pihak yang mendapat izin dari Pemerintah Daerah.