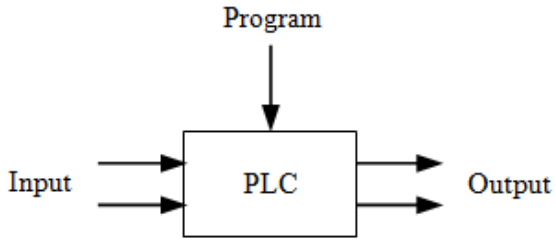
# BAB II
# TINJAUAN PUSTAKA

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

*Programmable Logic Controller* atau PLC pada awalnya dikenal sebagai *Programmable Controller* (PC) yang lahir sebaga i 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 dan intuitif.




Gambar 2.1 Logika PLC *(Programmable Logic Controller)*

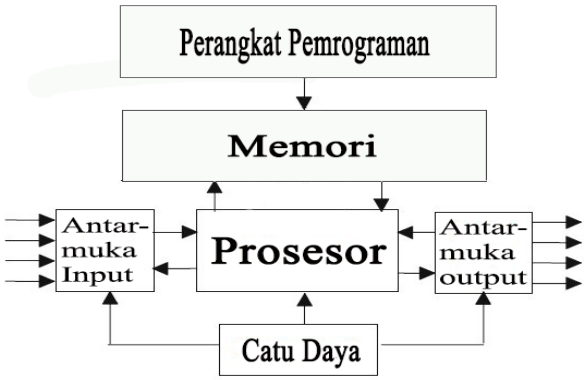
[1] William Bolton, *Programmable Logic Controller (PLC) Sebuah Pengantar Edisi Ketiga* (Jakarta: Erlangga, 2003), hlm. 03

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, katup, dll yang disambungkan ke PLC. Sang operator kemudian memasukan 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 telah diprogramkan.

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 dan penyambungan.

Perangkat PLC pertama kali dikembangkan pada tahun 1969, 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 2.2 Sistem PLC (*Programmable Logic Controller)*

### 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.
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 spacenya terdiri dari kategori *User Programable* dan Data Table. User Program 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

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 2.3 PLC Omron CP1E-E40 SDR-A

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 10 A. PLC Omron seri CP1E bekerja dengan tegangan yang bisa diubah 100 sampai 240 VAC, Program memory: 2Ksteps (EEPROM), Data memory DM: 2 Kwords. 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 10 A. PLC Omron seri CP1E memiliki sistem program dengan menggunakan *software* pemrogaman *CX-Programmer.*

### 2.1.3 *Software CX-Programmer*[2]

***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 kita bisa memprogram aneka PLC buatan omron dan salah satu fitur yang saya suka yaitu adanya fitur simulasi tanpa harus terhubung dengan PLC, sehingga kita bisa mensimulasikan ladder yang kita buat, dan simulasi ini juga bisa kita hubungkan dengan HMI PLC Omron yang telah kita buat dengan menggunakan *CX-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

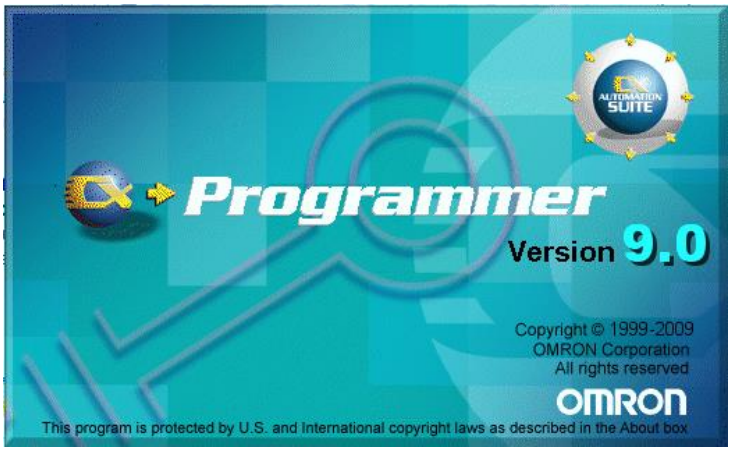
RAM 32 Mega bytes

Hard disk dengan ruang kosong kurang lebih 100 MB

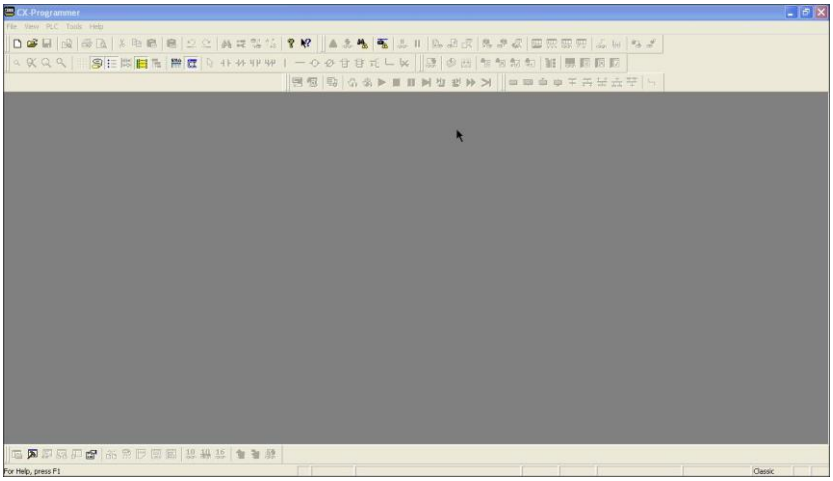
Monitor SVGA dengan resolusi 800 x 60

[2]Musbikhin._____. Pengantar CX Programmer (Seri Belajar PLC), online diakses 22 April 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 :

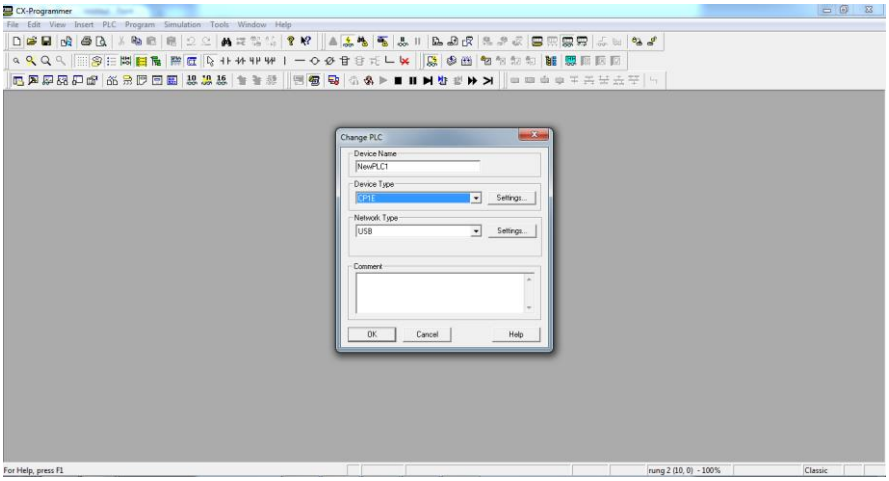



Gambar 2.4 *CX-Programmer Version 9.0 Omron*



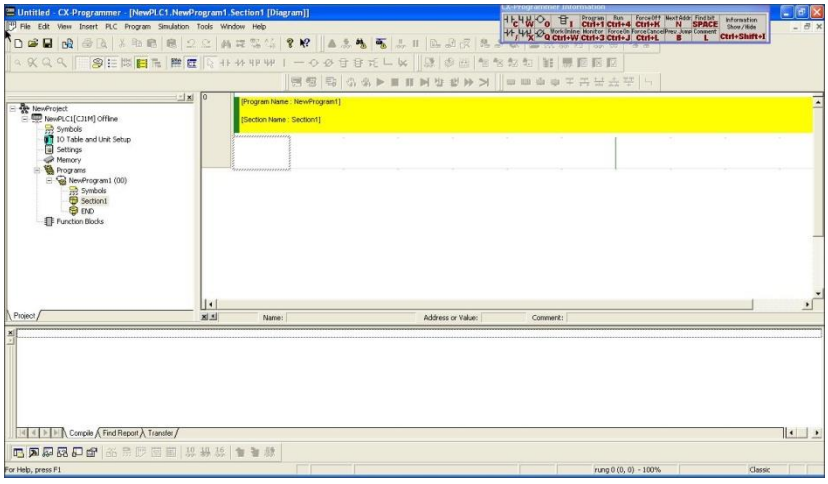
Gambar 2.5 Tampilan Pertama *Program CX-Programmer Version 9.0 Omron*

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, kalo 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 :

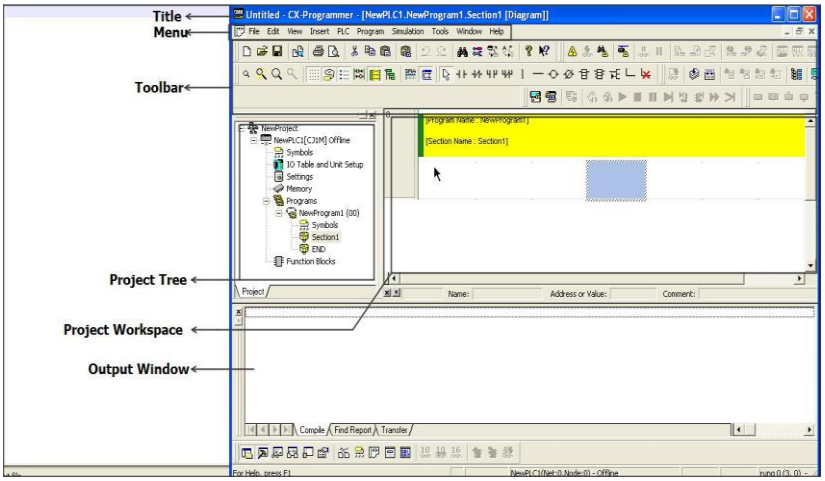



Gambar 2.6 Tampilan Pemilihan *Device* PLC Pada *Program CX-Programmer Version 9.0 Omron*

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 akan tampil tampilan berikut :




Gambar 2.7 Tampilan *Project Program CX-Programmer* Version 9.0 Omron




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

Keterangan detail untuk tampilan tersebut yaitu :

***Title Bar* :**

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

***Menus* :**

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 *software* 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. Berikut Instruksi program *CX-Programmer Version 9.0 Omron*

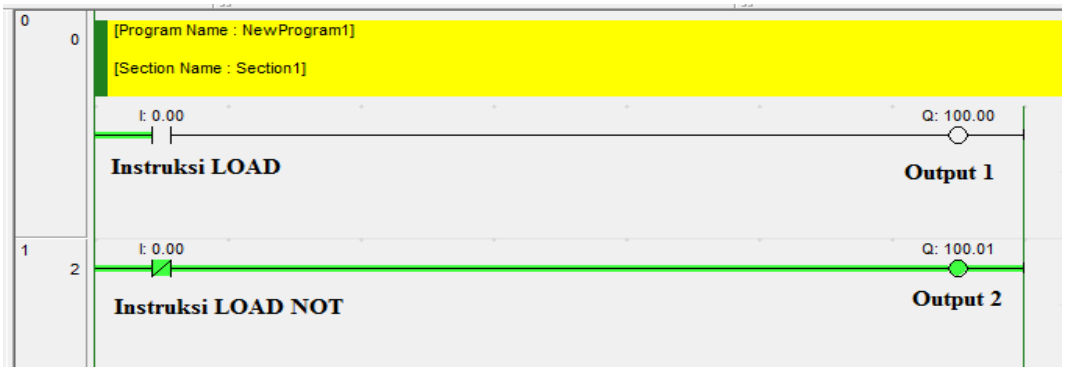
Tabel 2.1 Instruksi program *CX-Programmer Version 9.0 Omron*

| Instruksi | Kode Mnemonik |
|---|---|
| Input | I: 00.0 – I: 10.7 |
| Output | Q: 100.00 – Q: 101.07 |
| Memori | IR 000 |
| Timer | TIM 000 |
| Counter | CNT 000 |

### 2.1.5 Instruksi Dasar Pada PLC

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

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 gambar 2.9.



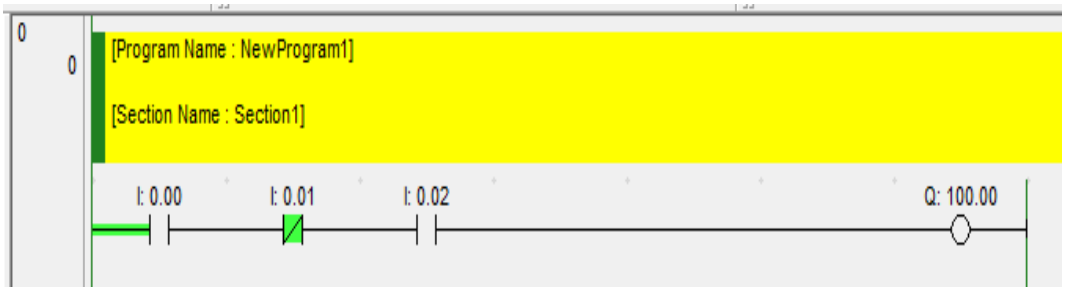

Gambar 2.9 Contoh Penggunaan Instruksi LD dan LD NOT

Tabel 2.2 Kode Mnemonik Instruksi LD dan LD NOT

| Alamat | Instruksi | Kode I/O |
|---|---|---|
| 00000 | LD | 0.00 |
| 00001 | OUT | 100.00 |
| 00002 | LD NOT | 0.00 |
| 00003 | OUT | 100.01 |

2. AND dan AND NOT

Jika terdapat dua atau lebih kondisi yang dihubungkan seri pada garis instruksi yang 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 kode mnemonic.



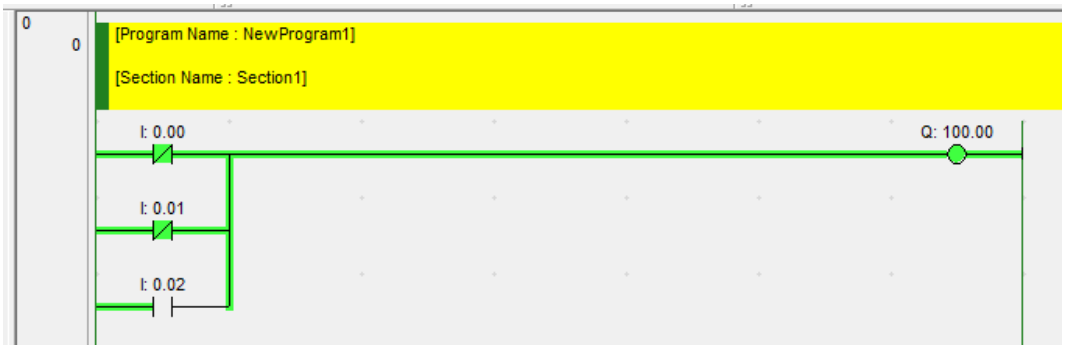

Gambar 2.10 Contoh Penggunaan Instruksi AND dan AND NOT

Tabel 2.3 Kode Mnemonik Instruksi AND dan AND NOT

| Alamat | Instruksi | Kode I/O |
|---|---|---|
| 00000 | LD | 0.00 |
| 00001 | AND NOT | 0.01 |
| 00002 | AND | 0.02 |
| 00003 | OUT | 100.00 |

3. OR dan OR NOT

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 LD NOT dan sisanya berkaitan dengan instruksi OR dan OR NOT. Gambar menunjukkan tiga buah instruksi yang berkaitan dengan instruksi LD NOT, OR NOT, dan OR. Masing-masing instruksi tersebut membutuhkan satu baris kode mnemonik.



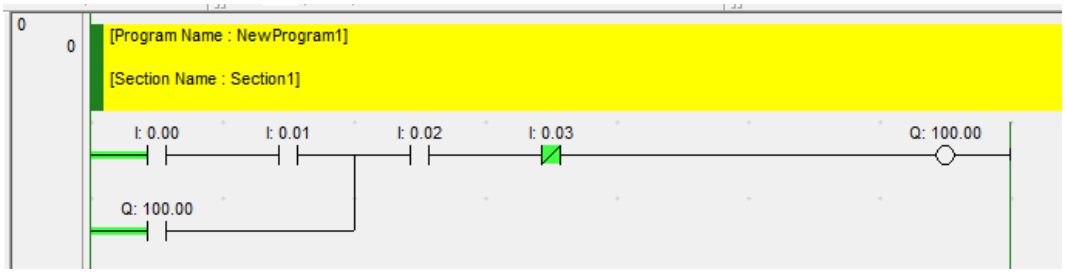

Gambar 2.11 Contoh Penggunaan Instruksi OR dan OR NOT

Tabel 2.4 Kode Mnemonik Instruksi OR dan OR NOT

| Alamat | Instruksi | Kode I/O |
|---|---|---|
| 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 2.12 Contoh Penggabungan Instruksi AND dan OR

Tabel 2.5 Kode Mnemonik Instruksi AND dan OR

| Alamat | Instruksi | Kode I/O |
|---|---|---|
| 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 Blok Logika

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 (AND LD)

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) dan blok logika kanan juga dalam keadaan ON ( 0.02 dalam kondisi ON atau 0.03 dalam kondisi OFF).


Gambar 2.13 Contoh Penggunaan Instruksi Blok Logika AND LD

Tabel 2.6 Kode Mnemonik Instruksi Blok Logika AND LD

| Alamat | Instruksi | Kode I/O |
|---|---|---|
| 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 (OR LD)

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 atau blok logika bawah dalam kondisi ON. Artinya, 0.00 dan kondisi ON dan 0.01 dalam kondisi OFF atau 0.02 dan 0.03 dalam kondisi ON).

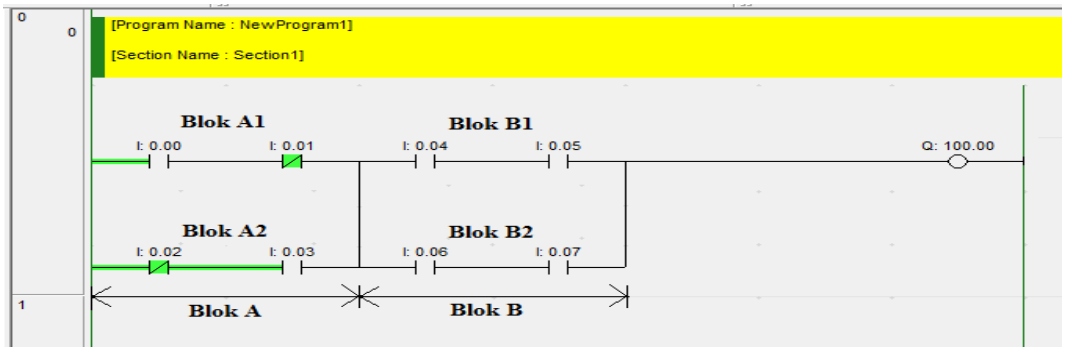

Gambar 2.14 Contoh Penggunaan Instruksi Blok Logika OR LD

Tabel 2.7 Kode Mnemonik Instruksi Blok Logika OR LD

| Alamat | Instruksi | Kode I/O |
|---|---|---|
| 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 2.15 memperlihatkan 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 blok A1 (alamat 00000 dan 00001) dan blok A2 (alamat 00002 dan 00003), kemudian digabung menggunakan instruksi blok logik OR LD (alamat 00004). Selanjutnya blok B1 dituliskan (alamat 00005 dan 00006) 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 2.15 Contoh Penggunaan Instruksi Blok Logika Kompleks

Tabel 2.8 Kode Mnemonik Instruksi Blok Logika Kompleks

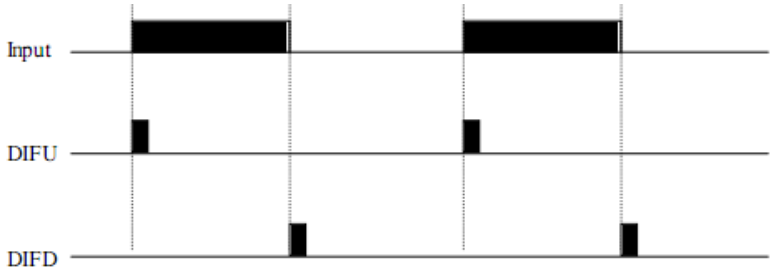
| Alamat | Instruksi | Kode I/O |
|---|---|---|
| 00000 | LD | 0.00 |
| 00001 | AND NOT | 0.01 |
| 00002 | LD NOT | 0.02 |
| 00003 | AND | 0.03 |
| 00004 | OR LD | - |
| 00005 | LD | 0.04 |
| 00006 | AND | 0.05 |
| 00007 | LD | 0.06 |
| 00008 | AND | 0.07 |
| 00009 | OR LD | - |
| 00010 | AND LD | -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

6. Intruksi Kendali Bit

Terdapat instruksi dasar yang dapat digunakan untuk mengontrol status bit secara individual, yaitu *DIFFERENTIATE UP* (DIFU), *DIFFERENTIATE DOWN* (DIFD) instruksi ini dituliskan di sisi paling kanan 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 eksternal), ini juga digunakan untuk mengontrol bit-bit lainnya.

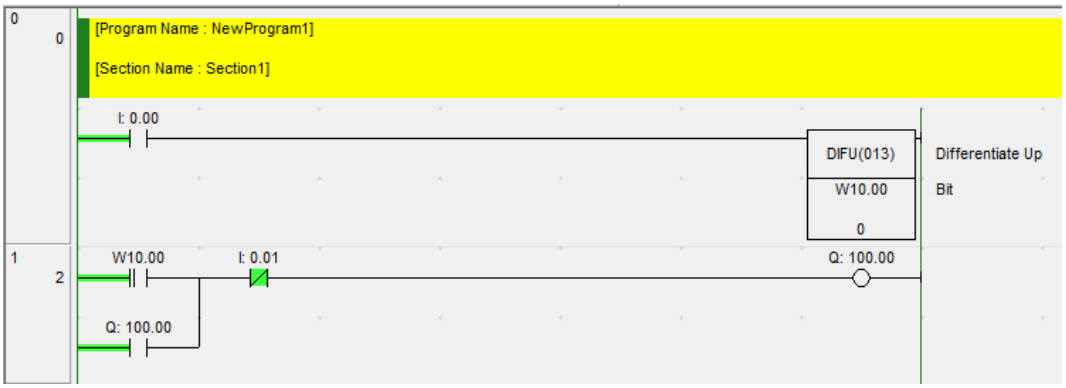
Kedua instruksi ini sangat sering sekali digunakan dalam pemrograman PLC. Kedua instruksi ini masuk ke dalam jenis *ladder instructions,* pada sub kategori *bit control instructions.*

Untuk penjelasan mengenai instruksi DIFU dan DIFD lihat gambar berikut:




Gambar 2.16 Prinsip Kerja Instruksi Kendali Bit DIFU dan DIFD

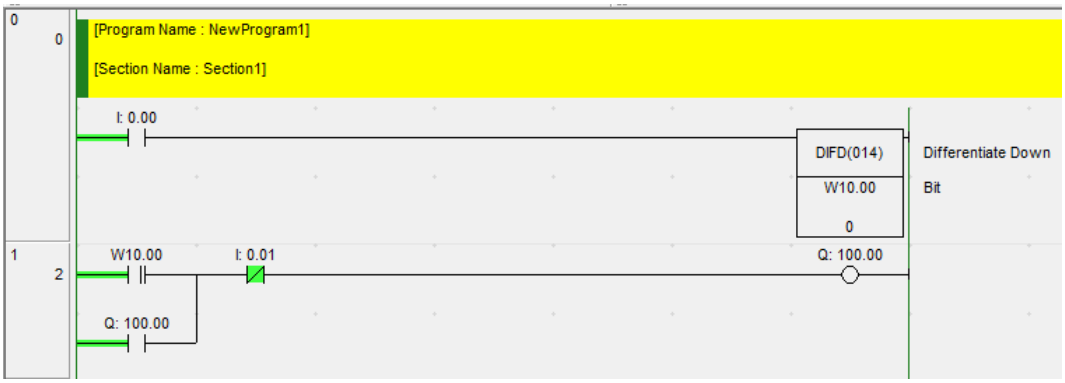
Jadi seperti terlihat pada gambar di atas, baik instruksi DIFU maupun instruksi DIFD output ON nya (warna hitam pada gambar) hanya sekali dan dalam waktu yang singkat saja, atau biasa disebut *one scan only*. sedangkan perbedaan dari instruksi DIFU dan DIFD, bahwa instruksi DIFU ini akan ON (tentunya 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 2.17 Contoh Penggunaan Instruksi Kendali Bit *DIFFERENTIATE UP* (DIFU)

Tabel 2.9 Kode Mnemonik Instruksi Kendali Bit *DIFFERENTIATE UP* (DIFU)

| Alamat | Instruksi | Kode I/O |
|---|---|---|
| 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 |



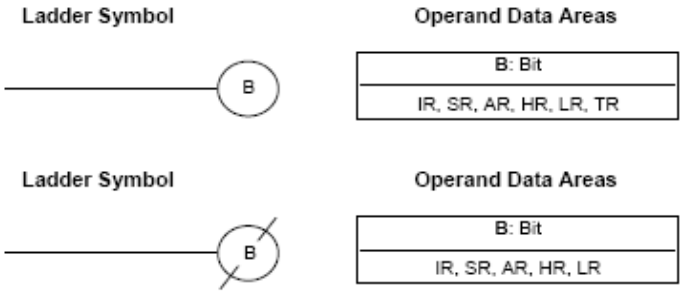

Gambar 2.18 Contoh Penggunaan Instruksi Kendali Bit *DIFFERENTIATE DOWN* (DIFD)

Tabel 2.10 Kode Mnemonik Instruksi Kendali Bit *DIFFERENTIATE DOWN* (DIFD)

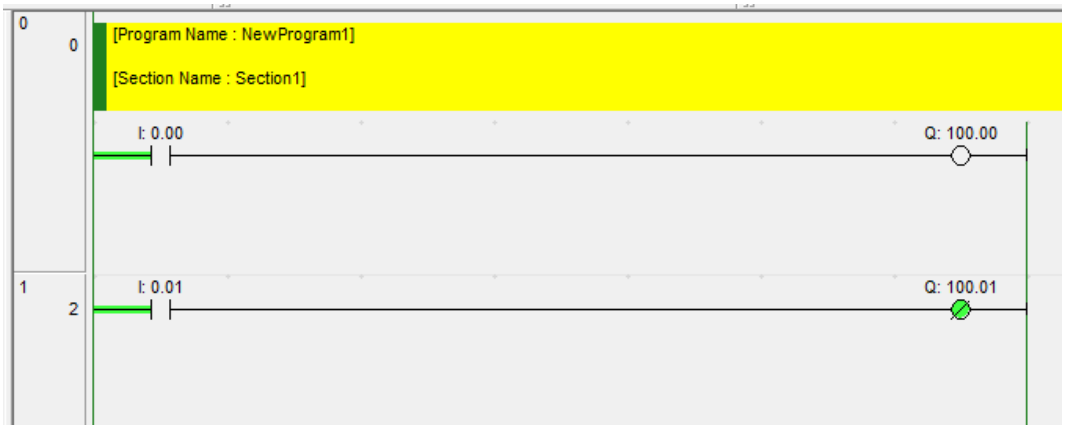
| Alamat | Instruksi | Kode I/O |
|---|---|---|
| 00000 | LD | 0.00 |
| 00001 | DIFD | W10.00 |
| 00002 | LD | W10.00 |
| 00003 | OR | 100.00 |
| 00004 | AND NOT | 0.01 |
| 00005 | OUT | 100.00 |

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

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 2.19 memperlihatkan simbol tangga dan area data operan dari instruksi OUT dan OUT NOT,sedangkan Gambar 2.20 memperlihatkan contoh implementasi kedua instruksi tersebut.




Gambar 2.19 Simbol Tangga Dan Area Data Operan Instruksi OUT dan OUT NOT




Gambar 2.20 Contoh Penggunaan Instruksi OUT dan OUT NOT

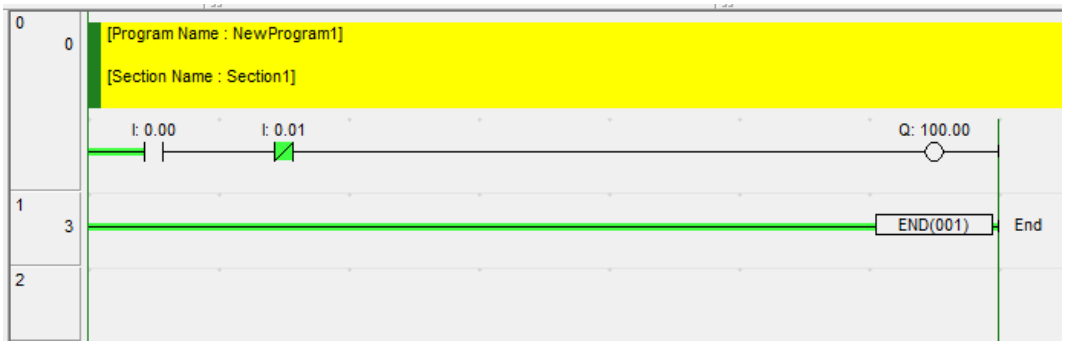
Tabel 2.11 Kode Mnemonik Instruksi OUT dan OUT NOT

| Alamat | Instruksi | Kode I/O |
|---|---|---|
| 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) sampai ditemui 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 instruksi lainnya.

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.



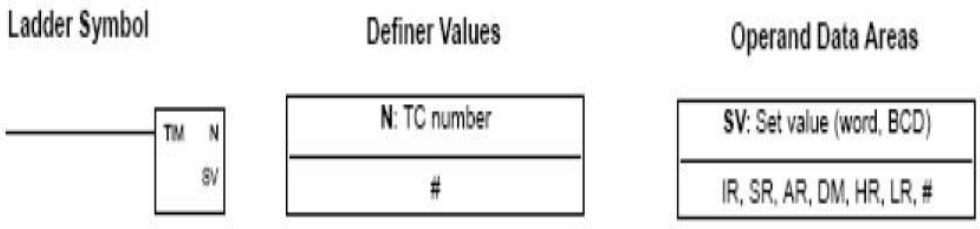

Gambar 2.21 Contoh Penggunaan Instruksi END

Tabel 2.12 Kode Mnemonik Instruksi END

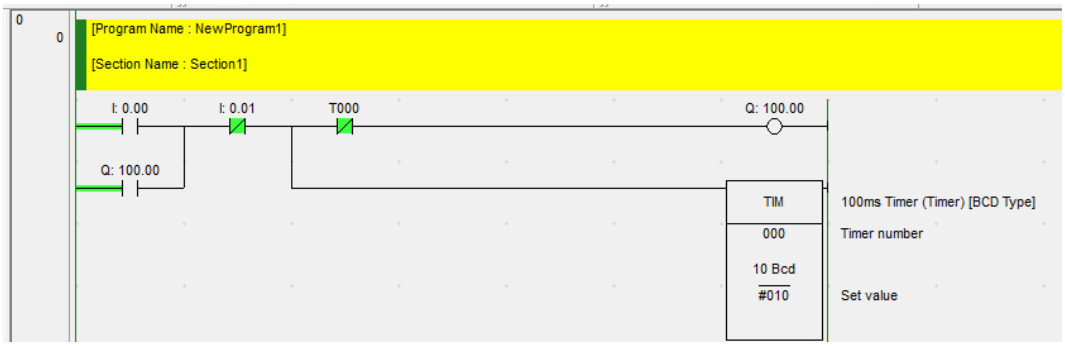
| Alamat | Instruksi | Kode I/O |
|---|---|---|
| 00000 | LD | 0.00 |
| 00001 | AND NOT | 0.01 |
| 00002 | OUT | 100.00 |
| 00003 | END(001) | - |

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 999,9 detik).




Gambar 2.22 Simbol Tangga Dan Area Data Operan Dari Instruksi TIMER (TIM)



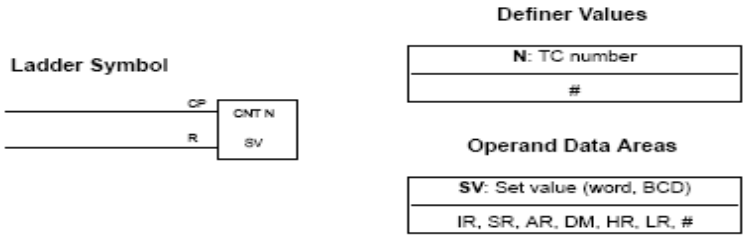

Gambar 2.23 Contoh Penggunaan Instruksi TIMER (TIM)

Tabel 2.13 Kode Mnemonik Instruksi TIMER (TIM)

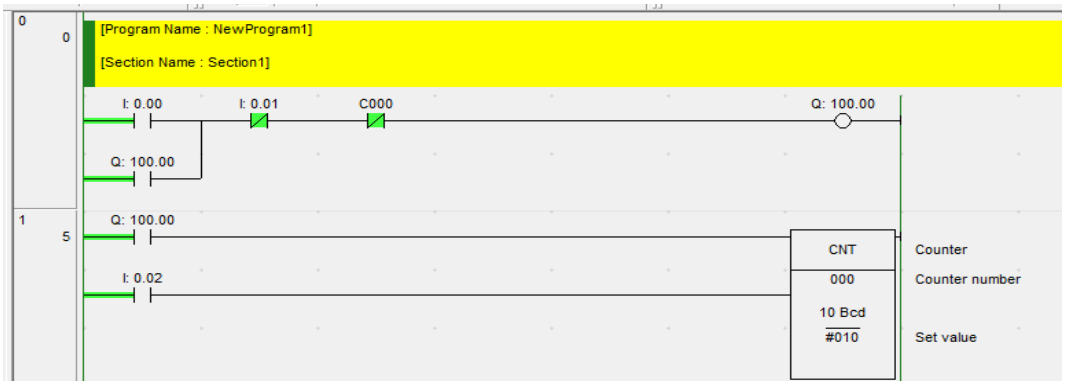
| Alamat | Instruksi | Kode I/O |
|---|---|---|
| 00000 | LD | 0.00 |
| 00001 | OR | 100.00 |
| 00002 | AND NOT | 0.01 |
| 00003 | TIM | 000 #010 |
| 00004 | AND NOT | T000 |
| 00005 | OUT | 100.00 |

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 instruksi CNT. Dan Gambar 2.25 memperlihatkan contoh penggunaan instruksi END.




Gambar 2.24 Simbol Tangga Dan Area Data Operan Dari Instruksi COUNTER (CNT)




Gambar 2.25 Contoh Penggunaan Instruksi COUNTER (CNT)

Tabel 2.14 Kode Mnemonik Instruksi COUNTER (CNT)

| Alamat | Instruksi | Kode I/O |
|---|---|---|
| 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[3]

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[4]

*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

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

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

**2.2. 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 berwarna hijau.
2. Kontak NC (*Normally Close*) Biasanya berwarna Merah.

Gambar 2.26 Tombol Tekan Kontak NO dan Kontak NC

**2.3 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 2 posisi, (ON-OFF/Start-Stop/0-1, dll) dan 3 posisi (ON-OFF-ON/Auto-Off-Manual,dll)dengan selector switch, kondisi peralatan dapat langsung di ketahui dari penunjukan tangkai *selector switch*, dengan *selector switch*, rangkaian *ON-*

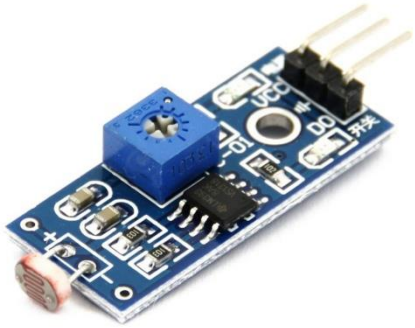
*OFF* lebih sederhana, karena *selector switch* tidak seperti tombol tekan yang hanya kontak sementara.

Gambar 2.27 *Selector Switch*

## 2.4 Sensor LDR

*Light Dependent Resistor* atau disingkat dengan LDR adalah jenis Resistor yang nilai hambatan atau nilai resistansinya tergantung pada intensitas cahaya yang diterimanya. Nilai Hambatan LDR akan menurun pada saat cahaya terang dan nilai Hambatannya akan menjadi tinggi jika dalam kondisi gelap. Dengan kata lain, fungsi LDR (*Light Dependent Resistor*) adalah untuk menghantarkan arus listrik jika menerima sejumlah intensitas cahaya (Kondisi Terang) dan menghambat arus listrik dalam kondisi gelap.

Naik turunnya nilai Hambatan akan sebanding dengan jumlah cahaya yang diterimanya. Pada umumnya, Nilai Hambatan LDR akan mencapai 200 Kilo Ohm (kΩ) pada kondisi gelap dan menurun menjadi 500 Ohm (Ω) pada Kondisi Cahaya Terang.



Gambar 2.28 Light Dependent Resistor (LDR)

## 2.5 Sensor Gas MQ-2

Sensor Gas MQ_2 adalah sensor yang mampu mendeteksi gas mudah terbakar seperti gas LP, i-butana, propana, alkohol dan gas metana. Sensor gas ini berbeda dengan sensor panas, sensor panas parameter yang di ukurnya adalah temperaturnya, sedangkan sensor gas bekerja apabila mendeteksi adanya gas yang bisa memicu terjadinya api. Sensor gas ini digunakan sebagai simulasi apabila terjadi kebocoran gas yang akan mengakibatkan adanya sumber api


Gambar 2.29 Sensor Pendeteksi Gas / Gas Sensor MQ-2

## 2.6 Lampu Tanda

Lampu tanda biasanya 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 kontak

NO, 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.

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

Gambar 2.30 Lampu Tanda Sebagai Lampu Penerangan Rumah

## 2.7 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 2.31 Relay

## 2.8 *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 2.32 *Buzzer*

## 2.9 Motor Arus Searah (DC)[5]

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*, 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 di industri.

Gambar 2.33 Motor Arus Searah (DC)

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

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

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.

### 2.9.1 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.

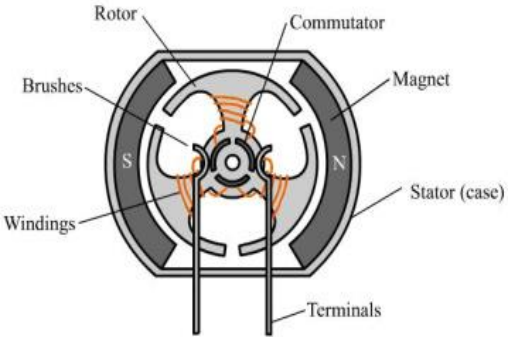
2. Kumparan Motor DC

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 2.34 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.