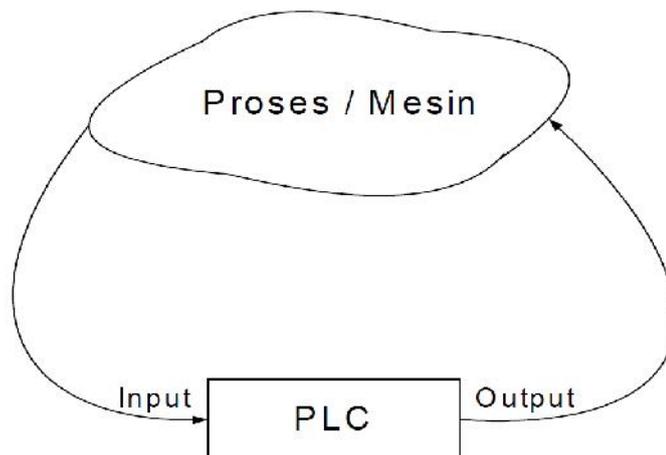


## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Programmable Logic Controller (PLC)

*Programmable logic controller* singkatnya PLC merupakan suatu bentuk khusus pengendalian berbasis mikroprosesor yang memanfaatkan memori yang dapat di program untuk menyimpan intruksi-intruksi dan untuk menerapkan (*counting*) dan aritmetika guna mengendalikan mesin-mesin dan proses-proses seperti (gambar 2.1) dan dirancang untuk dioperasikan oleh para insinyur yang memiliki sedikit 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 dengan menggunakan suatu bentuk bahasa pemrograman yang sederhana.



Gambar 2.1 Sebuah proses Programmable logic Controllers.<sup>[1]</sup>

---

<sup>[1]</sup> Setiawan, Iwan (2006).PROGRAMMABLE LOGIC CONTROLLER dan TEKNIK PERANCANGAN SISTEM KONTROL. Yogyakarta:Andi Yogyakarta.

Istilah logika (*logic*) dipergunakan karena pemrograman yang harus dilakukan sebagian besar berkaitan dengan penerapan operasi-operasi logika dan penyambungan *switch*. Perangkat-perangkat *input*, yaitu sensor-sensor semisal sakelar dan perangkat-perangkat *Output* di dalam sistem yang di kendali, misalnya motor, katup, dll yang disambungkan ke PLC. Sang operator kemudian memasukan serangkaian intruksi, yaitu sebuah program ke dalam memori 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.

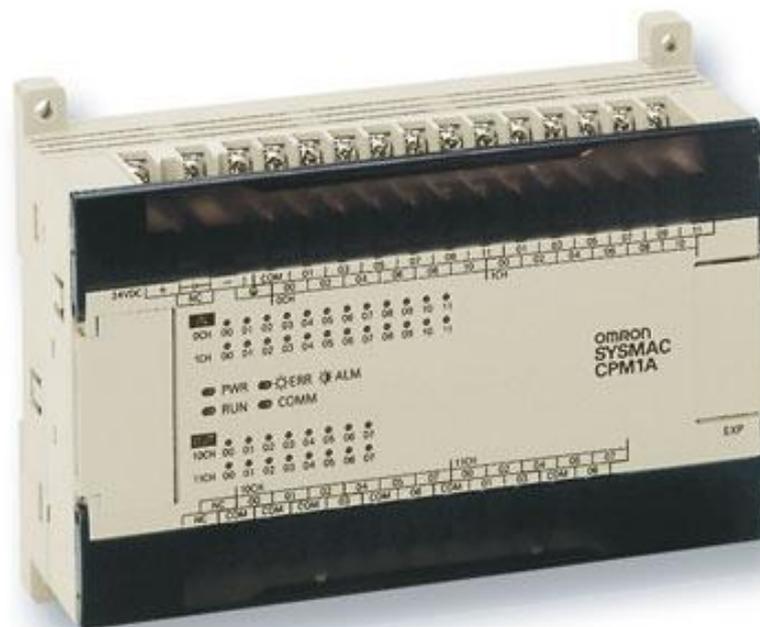
PLC serupa dengan komputer namun, bedanya komputer dioptimalkan untuk tugas-tugas perhitungan dan penyajian data, sedangkan PLC dioptimalkan untuk tugas-tugas pengendalian dan pengoperasian di dalam lingkungan industri. Dengan demikian PLC memiliki karakteristik :

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. 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 modular yang dapat menangani input/output dalam jumlah besar, menangani input/output analog maupun digital dan melaksanakan mode kendali proporsional.

### 2.1.1 Spesifikasi PLC Omron CPM 1A

PLC Omron CPM1A merupakan salah satu tipe PLC yang memiliki kecepatan yang tinggi yang dirancang untuk operasi kendali yang memerlukan jumlah I/O dari 10 sampai 100 buah I/O. Selain itu, PLC ini memiliki kemudahan dalam penginstalan, pengembangan, dan pemasangan sistem.



Gambar 2.2 PLC Omron CPM 1A<sup>[2]</sup>

Tabel 2.1 Spesifikasi PLC Omron CPM 1A

Nama	Spesifikasi
Power Supply	100 - 240 VAC ; 50/60 Hz
Operating Voltage Range	85 – 264 VAC
Inrush Current	30 A max.
Power Consumption	60 VA max.

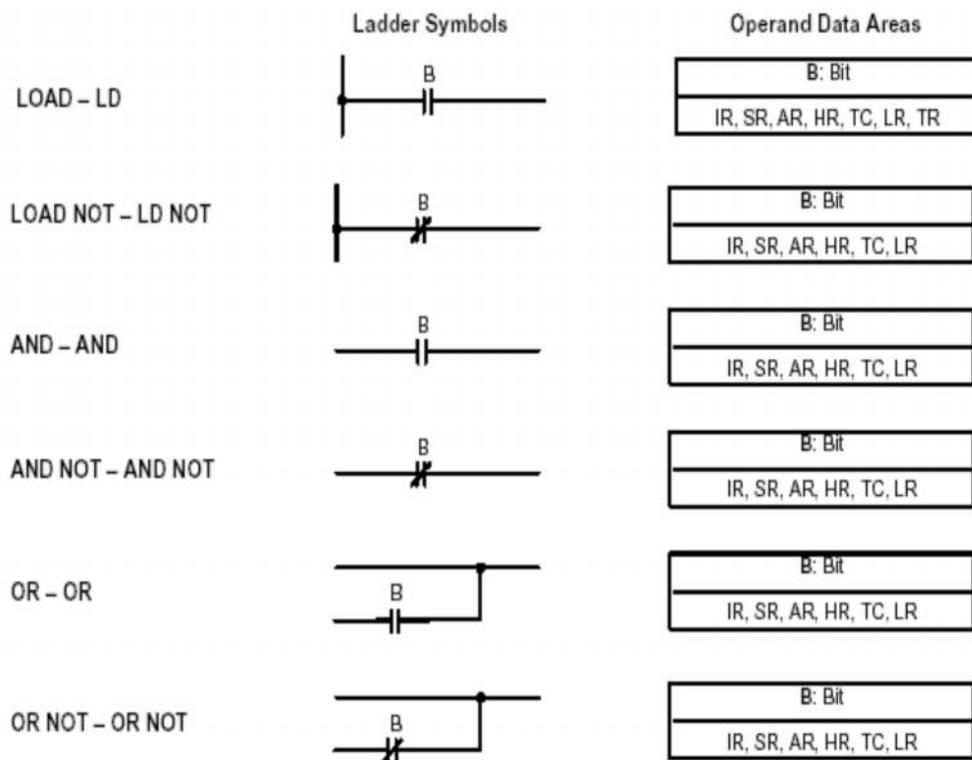
<sup>[2]</sup> Pribadi, Egi (2011). Teori Dasar PLC Omron CPM 1A. from <http://egipribadi.blogspot.com/2011/04/plc-omron-cpm-1a.html>, 28 June 2013

External power supply	24 VDC ; (300mA)
dimension	150 x 90 x 85 mm (Width x Heightx Depth)
Weight	700 gram max.
Communication connector	RS 232C

### 2.1.2 Instruksi-instruksi pada PLC

#### 1. Instruksi-instruksi Tangga (*Ladder Instructions*)

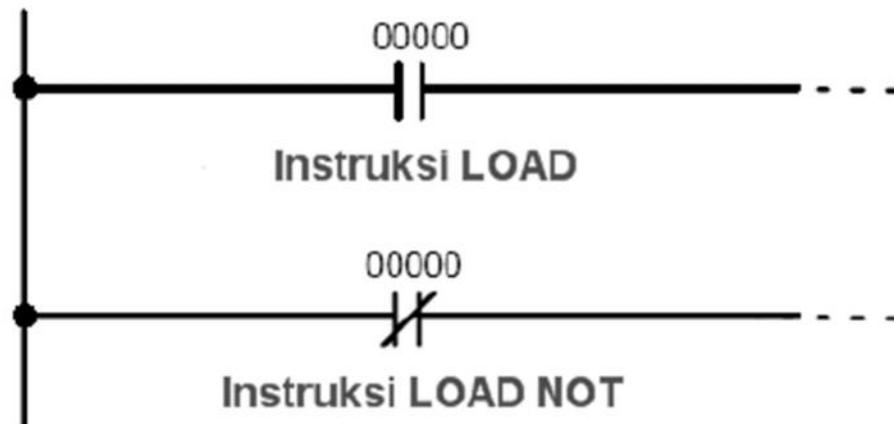
Instruksi-instruksi tangga adalah instruksi-instruksi yang terkait dengan kondisi di dalam diagram tangga. instruksi-instruksi tangga, baik yang independen maupun yang kombinasi atau gabungan dengan instruksi, akan membentuk kondisi eksekusi. gambar 2.2 memperlihatkan kode mnemonic, diagram tangga, dan area data.



Gambar 2.3 Kode mnemonic, diagram ladder dan area data operan dari instruksi-instruksi tangga.

**a. 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 2.4.



Gambar 2.4 Contoh instruksi LD dan LD NOT

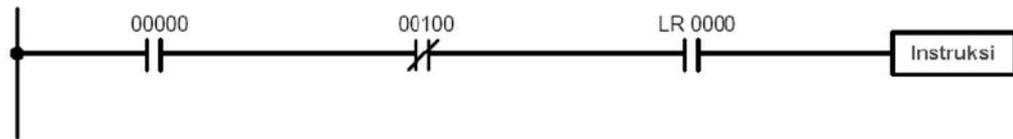
Tabel 2.2 Contoh instruksi LD dan LD NOT

Alamat	Instruksi	Operan
0000	LD	00000
0001	Instruksi	00000
0002	LD NOT	
0003	instruksi	

**b. 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 2.5 menunjukkan suatu penggalan diagram tangga yang mengandung tiga kondisi yang dihubungkan secara seri pada garis instruksi yang sama dan

berkaitan dengan instruksi LD, AND NOT, dan AND. Masing-masing instruksi tersebut membutuhkan satu baris kode mnemonic.



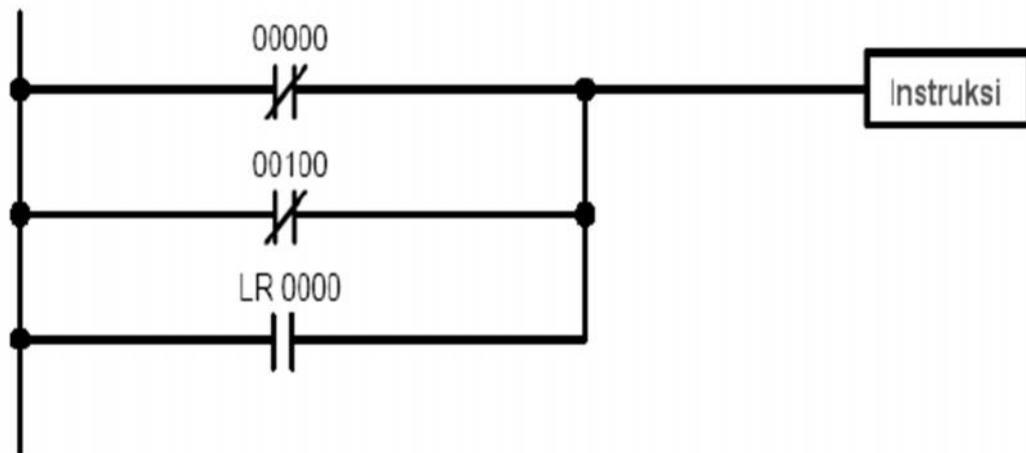
Gambar 2.5 Contoh penggunaan instruksi AND dan AND NOT

Tabel 2.3 Contoh penggunaan instruksi AND dan AND NOT

Alamat	Instruksi	Operan
0000	LD	00000
0001	AND NOT	01000
0002	AND	LR 0000
0003	Instruksi	

### c. OR dan OR NOT

Jika dua atau lebih kondisi yang dihubungkan paralel, artinya dalam garis instruksi 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 2.6 menunjukkan tiga buah instruksi yang berkaitan dengan instruksi LD NOT, OR NOT, dan OR. Masing-masing instruksi tersebut membutuhkan satu baris kode mnemonic.



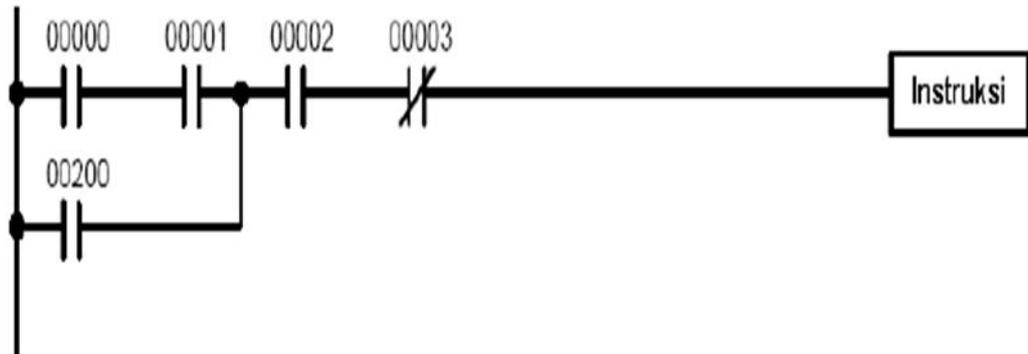
Gambar 2.6 Contoh penggunaan instruksi OR dan OR NOT

Tabel 2.4 Contoh penggunaan instruksi OR dan OR NOT

Alamat	Instruksi	Operan
0000	LD NOT	00000
0001	OR NOT	00100
0002	OR	LR 0000
0003	Instruksi	

#### d. 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 2.7 menunjukkan contoh diagram tangga yang mengimplementasikan cara seperti tersebut di atas.



Gambar 2.7 Contoh penggabungan instruksi AND dan OR

Tabel 2.5 Contoh penggabungan instruksi AND dan OR

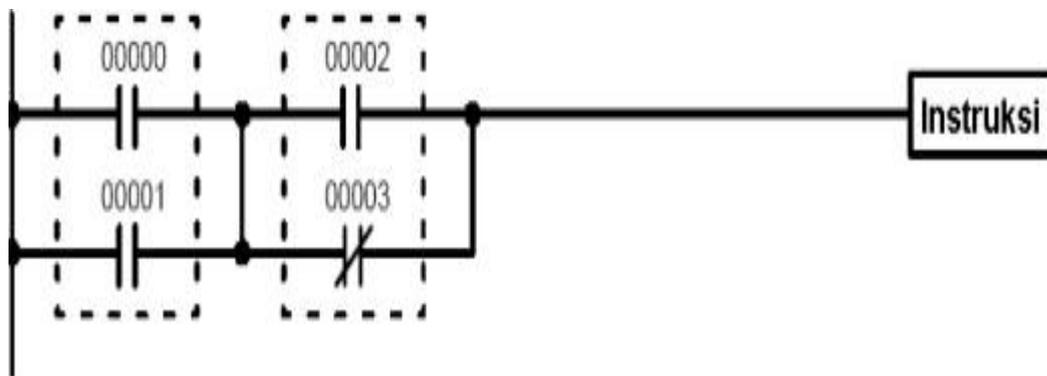
Alamat	Instruksi	Operan
0000	LD	00000
0001	AND	00001
0002	OR	20000
0003	AND	00002
0004	AND NOT	00003
0005	Instruksi	

## 2. Instruksi-instruksi Blok Logika

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

a. AND LOAD (AND LD)

Gambar 2.8 menunjukkan contoh penggunaan AND LD yang terdiri atas dua blok logik, yang akan menghasilkan kondisi ON jika blok logik kiri dalam kondisi ON (salah satu dari IR000.00 atau IR000.01 yang ON) **dan** blok logik kanan juga dalam keadaan ON (IR000.02 dalam kondisi ON atau IR000.03 dalam kondisi OFF).



Gambar 2.8 Contoh penggunaan instruksi AND LD

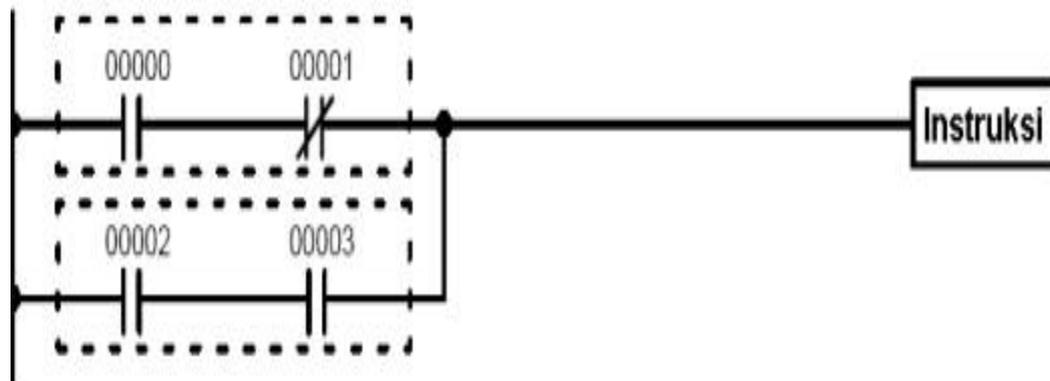
Tabel 2.6 Contoh penggunaan instruksi AND LD

Alamat	Instruksi	Operan
0000	LD	00000
0001	OR	00001
0002	OR NOT	00002
0003	AND LD	00003
0004		---

b. OR LOAD (OR LD)

Instruksi ini digunakan untuk meng-OR-logik-kan dua blok. Gambar 2.9 menunjukkan contoh penggunaan blok logik OR LD yang terdiri atas dua blok

logik. Kondisi eksekusi ON akan dihasilkan jika blok logik atas **atau** blok logik bawah dalam kondisi ON. Artinya, IR000.00 dan kondisi ON dan IR000.01 dalam kondisi OFF **atau** IR000.02 dan IR000.03 dalam kondisi ON).



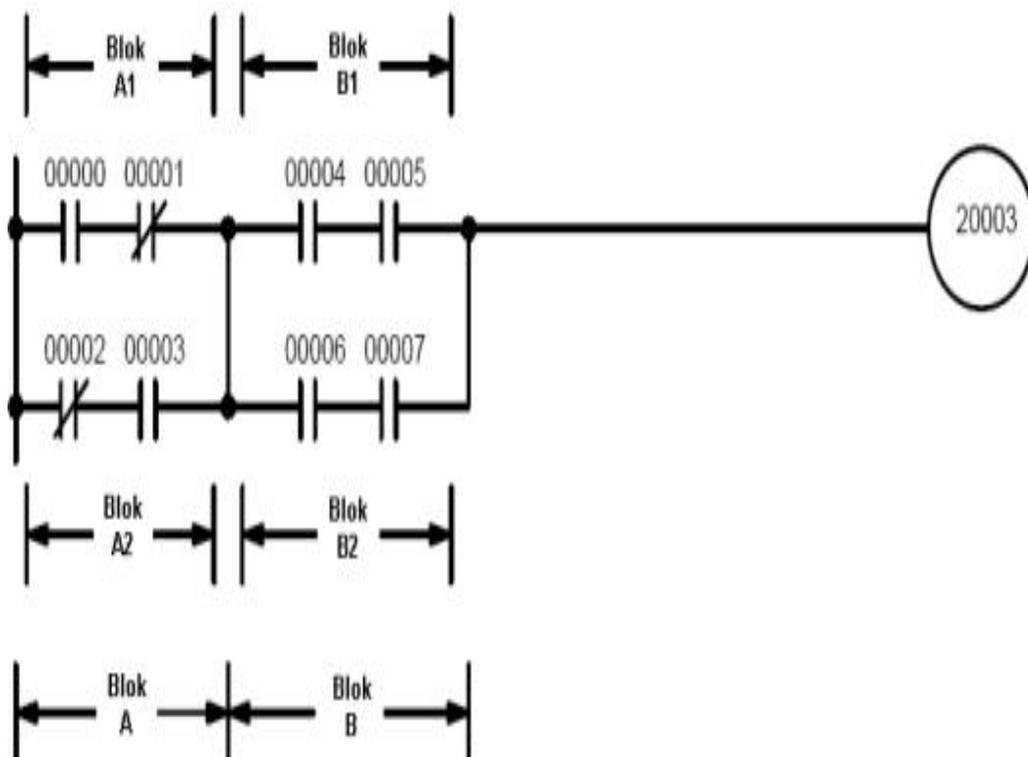
Gambar 2.9 Contoh penggunaan instruksi OR LD

Tabel 2.7 Contoh penggunaan instruksi OR LD

Alamat	Instruksi	Operan
0000	LD	00000
0001	AND NOT	00001
0002	LD	00002
0003	AND	00003
0004	OR LD	---

Untuk membuat kode mnemonic diagram tangga yang kompleks, caranya dengan cara membagi membagi diagram tersebut ke dalam blok-blok logik yang besar, kemudian membagi lagi blok yang besar tersebut menjadi blok-blok logik 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 logik **AND LD** dan **OR LD** hanya digunakan untuk menggabungkan *dua blok logik saja* (blok logik yang digabungkan berupa hasil penggabungan sebelumnya, atau hanya sebuah kondisi tunggal). Gambar 2.10 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 logik 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 logik **AND LD** (alamat 00010).



Gambar 2.10 Contoh diagram tangga yang kompleks

Tabel 2.8 Contoh diagram tangga yang kompleks

Alamat	Instruksi	Operan	Keterangan
00000	LD	00000	
00001	AND NOT	00001	
00002	LD NOT	00002	
00003	AND	00003	
00004	OR LD	-	Blok A1 dan A2
00005	LD	00004	
00006	AND	00005	
00007	LD	00006	
00008	AND	00007	
00009	OR LD	-	Blok B1 dan B2
00010	AND LD	-	Blok A dan B
00011	OUT	20003	

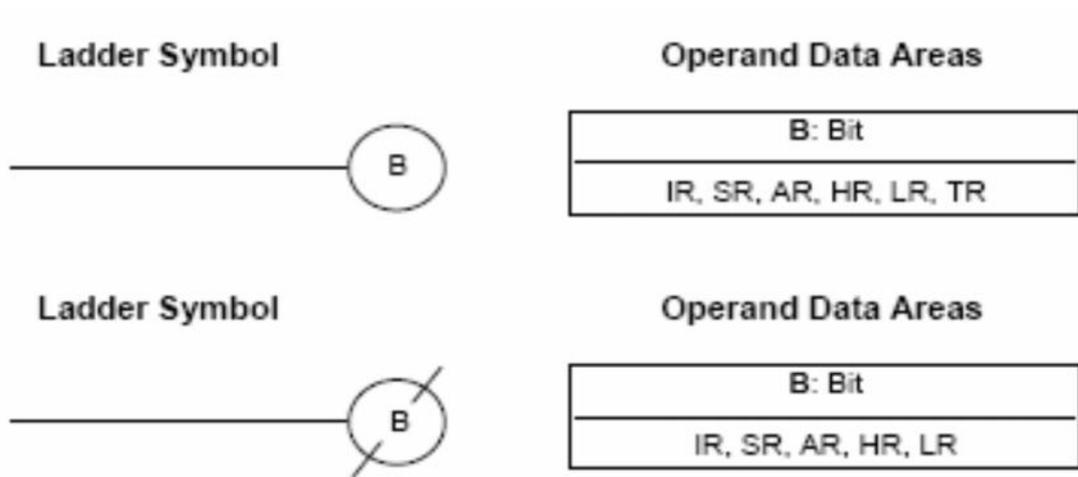
### 3. Instruksi Kendali Bit

Terdapat intruksui 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.

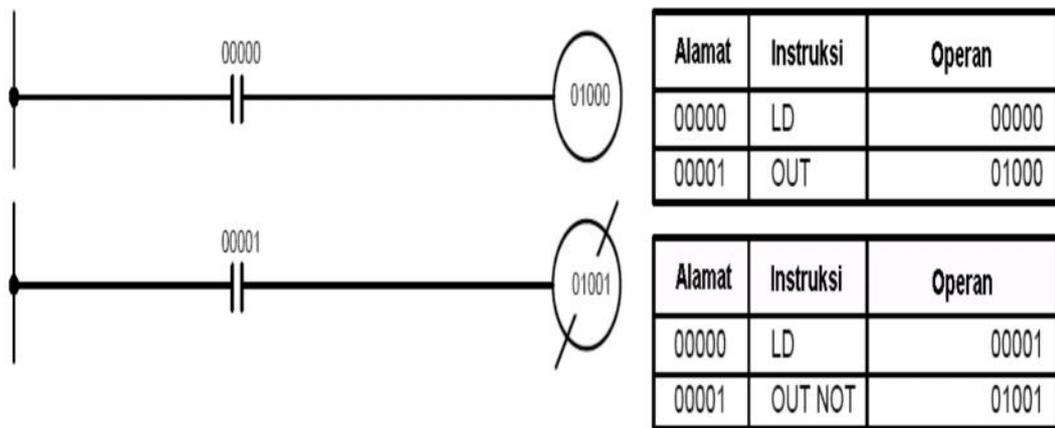
### 4. Instruksi OUTPUT (OUT) dan OUTPUT NOT (OUT NOT)

Instruksi ini digunakan untuk mengontrol operan yang berkaitan dengan kondisi eksekusi (apakah ON atau OFF). Dengan menggunakan instruksi OUT, maka bit operan akan menjadi ON jika kondisi eksekusinya juga ON, sedangkan OUT NOT akan menyebabkan bit operan menjadi ON jika kondisi eksekusinya OFF. Gambar 2.11 memperlihatkan simbol tangga dan area data

operan dari instruksi OUT dan OUT NOT, sedangkan Gambar 2.12 memperlihatkan contoh implementasi kedua instruksi tersebut.



Gambar 2.11 Simbol tangga dan area data operan instruksi OUT dan OUT NOT.



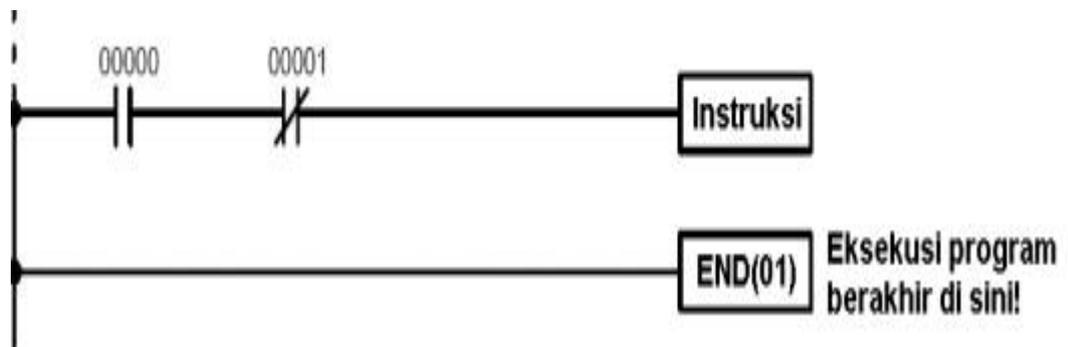
Gambar 2.12 Contoh penggunaan instruksi OUT dan OUT NOT.

### 5. 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 instruksi-instruksi 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 2.13 memperlihatkan contoh penggunaan instruksi END.



Gambar 2.13 contoh penggunaan instruksi END.

Tabel 2.9 contoh penggunaan instruksi END.

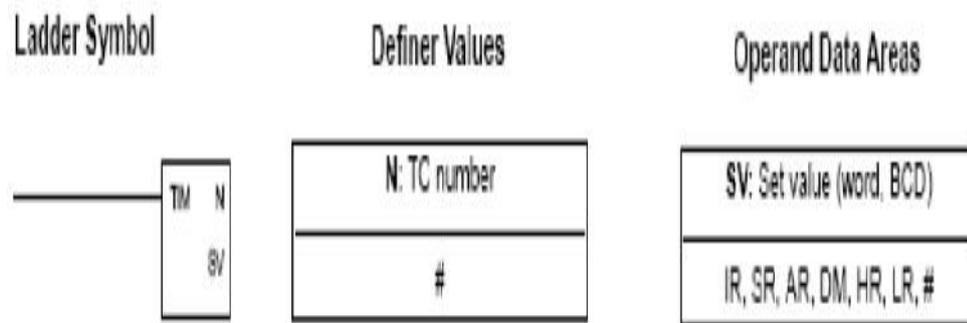
Alamat	Instruksi	Operan
0000	LD	00000
0001	AND NOT	00001
0002	Instruksi	
0003	END (01)	-

## 6. Instruksi Timer dan Counter

### a. Instruksi Timer (TIM)

Instruksi TIM dapat digunakan sebagai timer (pewaktu) ON-delay pada rangkaian relai. Gambar 2.14 memperlihatkan simbol tangga dan area data

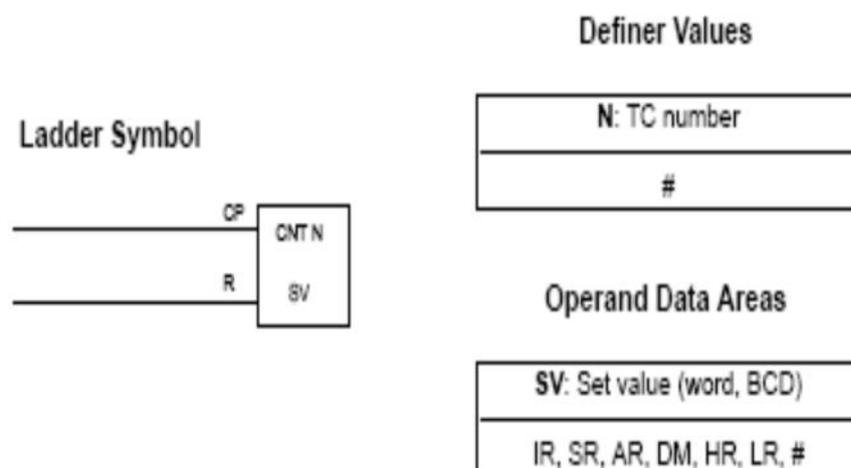
operan dari instruksi TIM. Instruksi TIM membutuhkan angka timer (N), dan nilai set (SV) antara 0000 sampai 9999 (artinya 000,0 sampai 999,9 detik).



Gambar 2.14 Simbol tangga dan area data operan dari instruksi TIMER (TIM)

b. 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 value (SV) Nilai set ini adalah 0000 sampai 9999. Gambar 2.15 memperlihatkan simbol tangga dan area data operan dari instruksi CNT.



Gambar 2.15 Simbol tangga dan area data operan dari instruksi COUNTER (CNT).

## 2.2 Input PLC

Pada dasarnya PLC memerlukan sebuah input untuk melakukan perintah yang sudah diprogram dan di upload ke PLC. Umumnya PLC hanya memerlukan input oleh sebuah sensor dan sebuah tombol tekan.

### 2.2.1 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 :

- a. Kontak NO (Normally Open) Biasanya berwarna hijau.
- b. Kontak NC (Normally Close) Biasanya berwarna Merah.



Gambar 2.16 Kontak NO



Gambar 2.17 kontak NC

## 2.3 Output PLC

Output yang digunakan nanti adalah lampu tanda yang untuk menandakan bahwa rangkaian atau motor itu bekerja.

### 2.3.1 Lampu Tanda

Lampu tanda atau indikator 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 suatu peralatan, 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 2.6 dibawah ini merupakan warna - warna yang menunjukkan fungsi dari lampu tanda.

Tabel 2.10 Fungsi Warna Lampu Tanda

<b>Kondisi peralatan/Beban</b>	<b>Warna lampu</b>
Kondisi tidak normal, beban lebih bahaya/emergency.	Merah
Hati-hati, perhatian	Kuning
Posisi siap, mulai beroperasi	Hijau
Beroperasi normal	Putih

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 yang biasa dilihat pada gambar 2.18 dibawah ini.



Gambar 2.18 Lampu Tanda