

Lampiran

Lampiran 1. Bahasa Pemrograman PLC

PLC tidak dapat bekerja sendiri tanpa adanya program yang mengatur langkah kerja didalamnya. Program yang dibuat ditujukan untuk mengatur kerja antara sensor dan aktuator. Program dirancang untuk memberikan perintah ke aktuator berdasarkan keadaan sensor ataupun juga berdasarkan perhitungan aritmatika (Endaryoko, 2011).

Berkaitan dengan pemrograman PLC, terdapat lima model atau metode yang telah distandardisasi penggunaannya oleh IEC (*International Electrotechnical Commission*) 1131-3 sebagai berikut (Sonjaya, 2011).

1. List Instruksi (*Instruction List*)

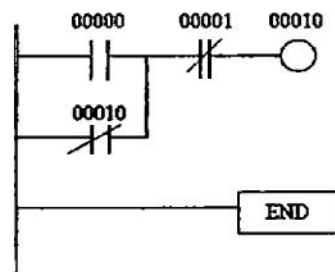
Pemrograman dengan menggunakan instruksi-instruksi bahasa level rendah (*Mnemonic*), seperti LD, AND, OR dan lain sebagainya (Sonjaya, 2011).

Tabel 1. Contoh *List Instruksi*

ALAMAT	INSTRUKSI	OPERAN
0	LD	00000
1	AND NOT	00001
2	OUT	00002
3	END	

2. Diagram Tangga (*Ladder Diagram*)

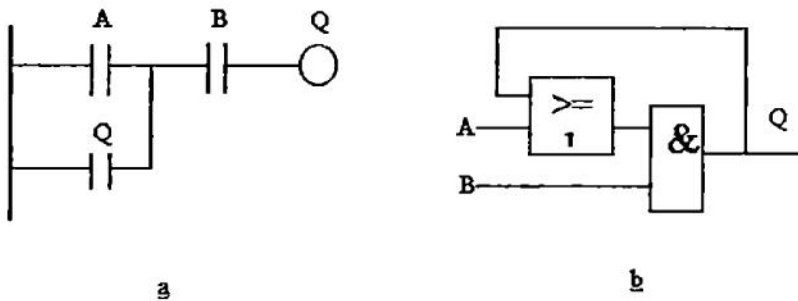
Pemrograman berbasis logika *relay*, cocok digunakan untuk persoalan-persoalan Kontrol Diskret yang *input/output* hanya memiliki dua kondisi *on* atau *off* seperti pada sistem kontrol konveyor, lift, dan motor-motor industri (Sonjaya, 2011).



Gambar 1. Diagram Tangga

3. Diagram Blok Fungsi (*Function Block Diagram*)

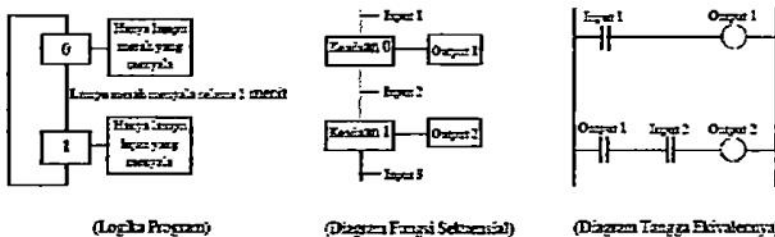
Pemrograman berbasis aliran data secara grafis. Banyak digunakan untuk tujuan kontrol proses yang melibatkan perhitungan-perhitungan kompleks dan akuisisi data analog (Sonjaya, 2011).



Gambar 2. (a) Diagram Tangga (b) Diagram Blok Fungsional Ekuivalennya

4. Diagram Fungsi Sekuensial (*Sequential Function/Flow Charts/Grafcet*)

Metode grafis untuk pemrograman terstruktur yang banyak melibatkan langkah-langkah rumit, seperti pada bidang robotika, perakitan kendaraan, *Batch Control*, dan lain sebagainya (Sonjaya, 2011).



Gambar 3. Diagram Fungsi Sekuensial

5. Teks Terstruktur (*Structured Text*)

Tidak seperti keempat metode sebelumnya, pemrograman ini menggunakan statemen-statement yang umum dijumpai pada bahasa *level tinggi* (*high level programming*) seperti *If/Then*, *Do/While*, *Case*, *For/Next*, dan lain sebagainya. Dalam aplikasinya, model ini cocok digunakan untuk perhitungan-perhitungan matematis yang kompleks, pemrosesan tabel, serta fungsi-fungsi kontrol yang memerlukan algoritma khusus (Sonjaya, 2011).

```

ST program:

(* score > selover 1-500)      (* If score > selover, *)
underNG := FALSE;            (* Turn off underNG *)
OK := FALSE;                  (* Turn off OK *)
overNG := TRUE;               (* Turn on overNG *)

ELSIF score < selunder < 100  (* If score < selover and score < selunder then *)
overNG := FALSE;             (* Turn on overNG *)
OK := TRUE;                   (* Turn off OK *)
underNG := TRUE;             (* Turn on underNG *)

ENDIF
underNG := FALSE;            (* If selover < score < selunder then *)
overNG := FALSE;            (* Turn off overNG *)
OK := TRUE;                   (* Turn off OK *)

END;                          (* end of IF section *)

```

Gambar 4. Teks Terstruktur

Namun Sumbodo(2008) menyederhanakan pemrograman PLC menjadi dua macam yaitu dengan diagram *ladder* dan bahasa *mnemonic*. Pemrograman biasanya membuat diagram *ladder* terlebih dahulu dan kemudian baru menerjemahkannya dalam bahasa *mnemonic*, atau bisa juga langsung menggambar *ladder diagram* pada layar monitor (Sumbodo, 2008).

Lampiran 2. Program-program *Ladder* Dasar yang Digunakan

1. Program *Input-Output*

Ini merupakan program paling dasar dari diagram tangga PLC. Penulis menilai terdapat beberapa program yang cukup fundamental untuk dibahas. Berikut program-program tersebut.

a) *Input-Output* Dasar

Pada hakekatnya terdapat dua buah program *input* dan *output* yang paling mendasar pada PLC. Dua program ini yaitu *input-output* dengan *Normally Open Contact* dan *input-output* dengan *Normally Close Contact*. Keduanya dibedakan berdasarkan jenis kontak yang digunakan.

i. *Input-Output* dengan N.O. *Contact*



Gambar 5. *Input-Output* dengan N.O. *Contact*

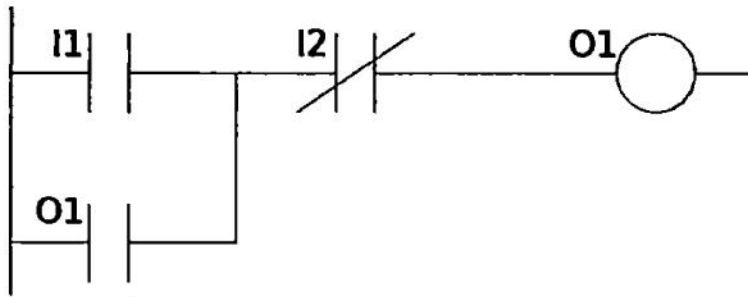
ii. *Input-Output* dengan N.C. *Contact*

Penjelasan program ini sama dengan "Program Gerbang Logika NOT".

2. *Latching/Set* dan *Unlatching/Reset*

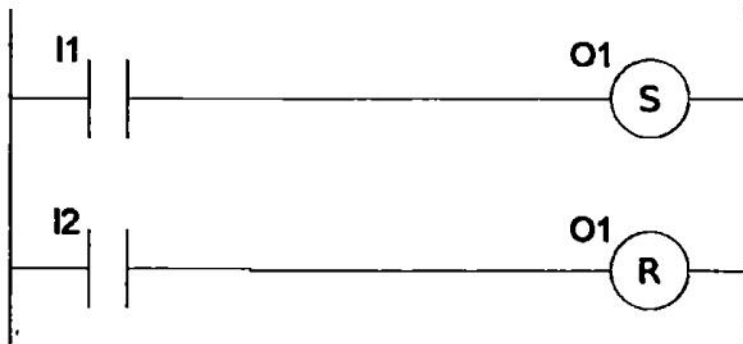
Dikutip dari <http://www.plcs.net/chapters/latch14.htm>, "The latch instruction is often called a SET or OTL (output latch). The unlatch instruction is often called a RES (reset), OUT (output unlatch) or RST (reset). The diagram below shows how to use them in a program". Ini berarti *SET* = *Output Latch* dan *RESET* = *Input Latch*. Dengan kata lain, program *latch-unlatch* sama dengan program *SET-RESET*.

Terdapat dua model dari program ini. Yang pertama adalah dengan menggunakan salah satu konvensi yang berlaku pada *coil* PLC, yaitu *self holding*.



Gambar 6. SET-RESET dengan Self Holding

Yang kedua adalah dengan menggunakan coil tipe SET dan RESET.



Gambar 7. SET-RESET dengan Coil Khusus

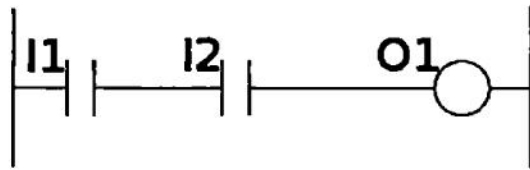
Secara logika, kedua program *ladder* tersebut memiliki kesamaan. Ketika I1 aktif atau bernilai 1, O1 akan diatur bernilai 1. O1 akan bernilai 1, walaupun I1 dinonaktifkan atau bernilai 0. O1 hanya akan kembali bernilai 0 jika I2 diaktifkan.

3. Program Operasi Logika

Pada sistem digital dikenal beberapa tipe dasar gerbang logika. Gerbang merupakan suatu rangkaian dengan satu atau beberapa masukan yang akan menghasilkan satu buah keluaran. Pada diagram *ladder* yang dipakai pada PLC gerbang-gerbang logika tersebut dapat dianalogikan sebagai suatu saklar. Saklar tersebut mempunyai dua keadaan yaitu ON(terhubung) atau OFF(terputus) (Anonymous, 2009). Berikut penjelasan tentang beberapa program *ladder* gerbang logika dasar untuk PLC.

a) AND

Program AND merupakan program yang berbasis logika gerbang AND. Jadi semua kondisi yang dihasilkan akan memenuhi logika AND. Berikut gambar dari program *ladder* dengan logika AND.



Gambar 8. Program *AND*

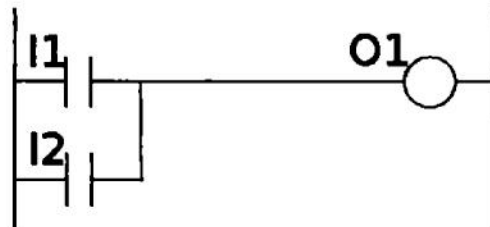
Program *ladder* tersebut bekerja secara teknis memenuhi tabel kebenaran berikut.

Tabel 2. Tabel Kebenaran Program *Ladder AND*

I1	I2	O1
0	0	0
1	0	0
0	1	0
1	1	1

b) *OR*

Program *OR* merupakan program yang berbasis logika gerbang *OR*. Jadi semua kondisi yang dihasilkan akan memenuhi logika *OR*. Berikut gambar dari dari program *ladder* dengan logika *OR*.



Gambar 9. Program *OR*

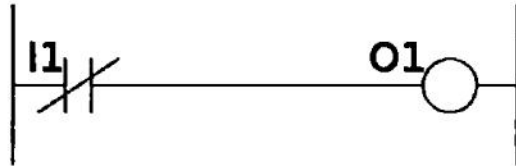
Program *ladder* tersebut bekerja secara teknis memenuhi tabel kebenaran berikut.

Tabel 3. Tabel Kebenaran Program *Ladder OR*

I1	I2	O1
0	0	0
1	0	1
0	1	1
1	1	1

c) *NOT*

Program *NOT* merupakan program yang berbasis logika gerbang *NOT*. Jadi semua kondisi yang dihasilkan akan memenuhi logika *NOT*. Berikut gambar dari dari program *ladder* dengan logika *NOT*.



Gambar 10. Program *NOT*

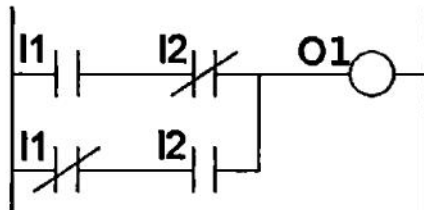
Program *ladder* tersebut bekerja secara teknis memenuhi tabel kebenaran berikut.

Tabel 4. Tabel Kebenaran Program *Ladder NOT*

I1	O1
0	1
1	0

d) *EXOR*

Program *EXOR* merupakan program yang berbasis logika gerbang *EXOR*. Jadi semua kondisi yang dihasilkan akan memenuhi logika *EXOR*. Berikut gambar dari dari program *ladder* dengan logika *EXOR*.



Gambar 11. Program *EXOR*

Program *ladder* tersebut bekerja secara teknis memenuhi tabel kebenaran berikut.

Tabel 5. Tabel Kebenaran Program *Ladder EXOR*

I1	I2	O1
0	0	0
1	0	1
0	1	1
1	1	0