

DIAGRAM LADDER

Dr. Fatchul Arifin, MT
fatchul@uny.ac.id



Simbol

- 1. Load / LD = Star pada normally open input



- Instruksi ini seperti relay yang NO

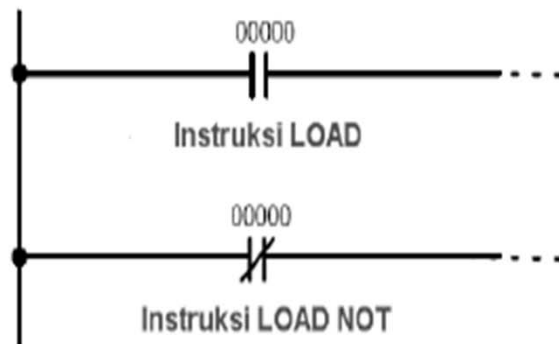
- 2. Load Not / LD NOT = Star pada normally close input



- Instruksi ini seperti relay yang NO

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

Instruksi ini dibutuhkan jika urutan kerja pada suatu sistem kontrol hanya membutuhkan satu kondisi logika saja dan sudah dituntut untuk mengeluarkan satu output. Logikanya seperti *contact NO relay* untuk instruksi LOAD dan seperti *contact NC relay* untuk instruksi LOAD NOT.

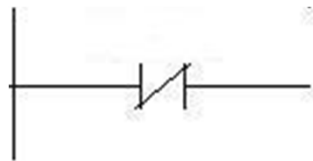


Alamat	Instruksi	Operan
00000	LD	00000
00001	Instruksi	
00002	LD NOT	00000
00003	Instruksi	

- 3. AND = menghubungkan dua atau lebih input dalam bentuk normally open secara seri.

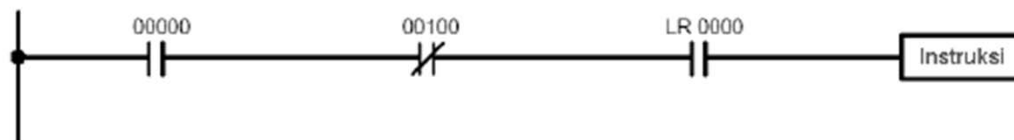


- 4. AND NOT = menghubungkan 2 atau lebih input dalam bentuk normally close.



2) AND dan AND NOT

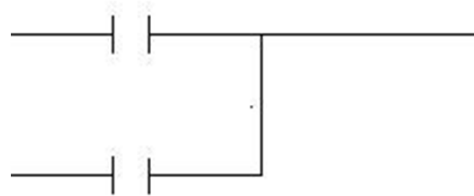
Jika terdapat dua atau lebih kondisi yang dihubungkan secara seri pada garis instruksi yang sama, maka kondisi yang pertama menggunakan instruksi LD atau LD NOT dan sisanya menggunakan instruksi AND atau AND NOT. Pada gambar di bawah ditunjukkan sebuah penggalan diagram tangga yang mengandung tiga kondisi yang dihubungkan secara seri pada garis instruksi yang sama berkaitan dengan LD, AND NOT, dan AND. Instruksi yang digambarkan paling kanan sendiri akan memiliki kondisi ON jika ketiga kondisi di kiri semuanya ON, dalam hal ini IR000.00 dalam kondisi ON, IR010.00 dalam kondisi OFF, dan LR000.00 dalam kondisi ON.



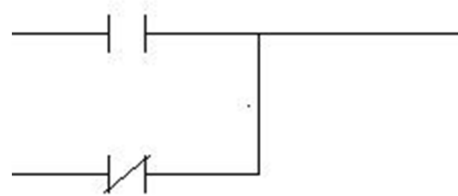
Alamat	Instruksi	Operan
00000	LD	00000
00001	AND NOT	00100
00002	AND	LR 0000
00003	Instruksi	

OR = menghubungkan 2 atau lebih input dalam bentuk normally open secara paralel.

- 5. OR = menghubungkan 2 atau lebih input dalam bentuk normally open secara paralel.



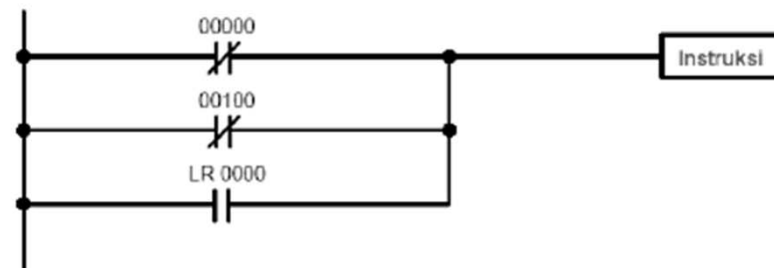
- 6. OR NOT = menghubungkan 2 atau lebih input dalam bentuk normally close secara paralel.



3) OR dan OR NOT

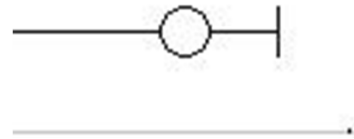
Jika dua atau lebih kondisi dihubungkan secara paralel, artinya dalam garis instruksi yang berbeda kemudian bergabung lagi dalam satu garis instruksi yang sama, maka kondisi yang pertama terkait dengan instruksi LD atau LD NOT dan sisanya berkaitan dengan instruksi OR atau OR NOT.

Blok instruksi ini akan memiliki kondisi eksekusi ON jika cukup salah satu dari ketiga kondisi dalam keadaan ON. Dalam hal ini kondisi OR dapat dibayangkan akan selalu menghasilkan kondisi ON jika salah satu dari dua atau lebih kondisi yang terhubung dengan instruksi ini dalam kondisi ON.



Alamat	Instruksi	Operan
00000	LD NOT	00000
00001	OR NOT	00100
00002	OR	LR 0000
00003	Instruksi	

- OUTPUT / OUT = menyalakan output.



- END = mengakhiri program

4) OUTPUT dan OUTPUT NOT

Cara yang paling mudah untuk mengeluarkan kondisi eksekusi adalah dengan menghubungkan langsung dengan keluaran melalui instruksi OUT atau OUT NOT. Kedua instruksi ini digunakan untuk mengontrol bit operan yang bersangkutan berkaitan dengan kondisi eksekusi. 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. Pada gambar di bawah terlihat jika IR010.00 akan ON selama IR000.00 juga ON, sedangkan IR010.01 akan ON selama IR000.01 dalam kondisi OFF.

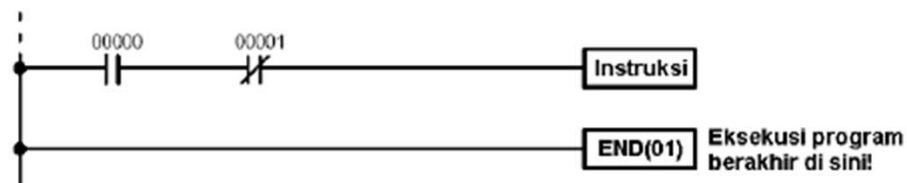


Alamat	Instruksi	Operan
00000	LD	00000
00001	OUT	01000

Alamat	Instruksi	Operan
00000	LD	00001
00001	OUT NOT	01001

5) END

Instruksi terakhir yang harus dituliskan atau digambarkan dalam diagram tangga adalah instruksi END. CPU pada PLC akan menyebabkan semua instruksi dalam program dari awal hingga ditemui instruksi END yang pertama, sebelum kembali lagi mengerjakan instruksi dalam program dari awal lagi, artinya instruksi yang ada di bawah atau setelah instruksi END akan diabaikan. Angka yang dituliskan pada instruksi END pada kode mnemonic merupakan kode fungsinya. Instruksi END tidak memerlukan operan serta tidak boleh diawali dengan suatu kondisi. Jika suatu program PLC tidak dilengkapi dengan instruksi END maka program tidak akan dijalankan sama sekali.

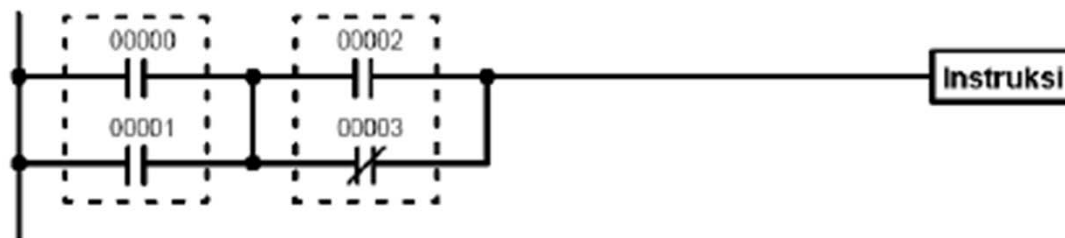


Alamat	Instruksi	Operan
00500	LD	00000
00501	AND NOT	00001
00502	Instruction	
00503	END(01)	---

And Load

6) AND LOAD (AND LD)

Untuk kondisi ladder diagram yang khusus seperti di bawah ini :

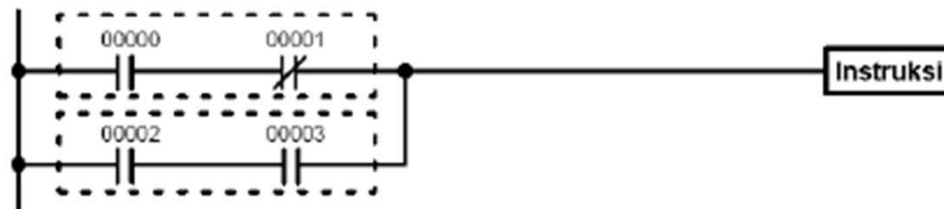


Alamat	Instruksi	Operan
00000	LD	00000
00001	OR	00001
00002	LD	00002
00003	OR NOT	00003
00004	AND LD	---

Or Load

7) OR LOAD (OR LD)

Untuk kondisi diagram tangga yang khusus seperti di bawah ini , kondisi eksekusi ON akan dihasilkan jika blok logic atas atau blok logic bawah dalam kondisi ON.



Alamat	Instruksi	Operan
00000	LD	00000
00001	AND NOT	00001
00002	LD	00002
00003	AND	00003
00004	OR LD	---

Titik Cabang

8) Garis Percabangan Instruksi

Pada pemrograman yang relatif kompleks, banyak dijumpai diagram tangga dengan banyak titik percabangan.



Diagram A: Penggambaran Benar

Alamat	Instruksi	Operan
00000	LD	00000
00001	Instruksi 1	
00002	AND	00002
00003	Instruksi 2	

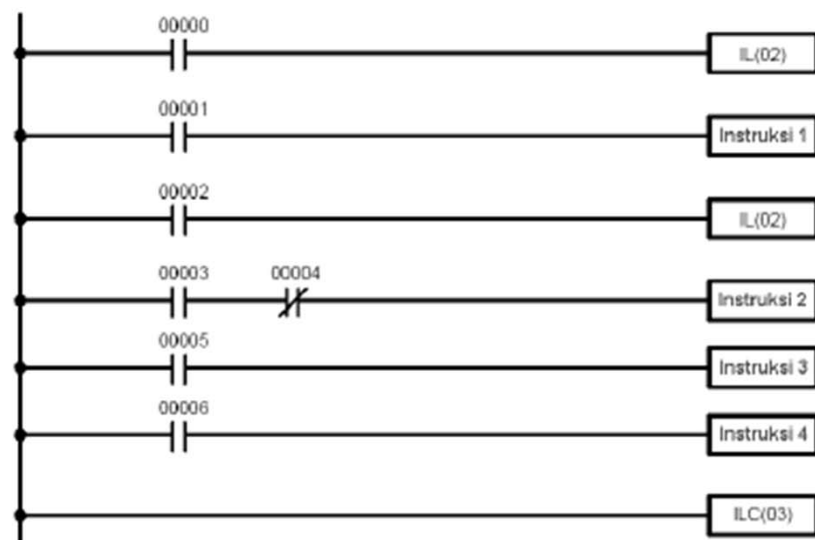


Diagram B: Penggambaran Salah

Alamat	Instruksi	Operan
00000	LD	00000
00001	AND	00001
00002	Instruksi 1	
00003	AND	00002
00004	Instruksi 2	

9) INTERLOCKS IL (02) dan INTERLOCKS CLEAR ILC (03)

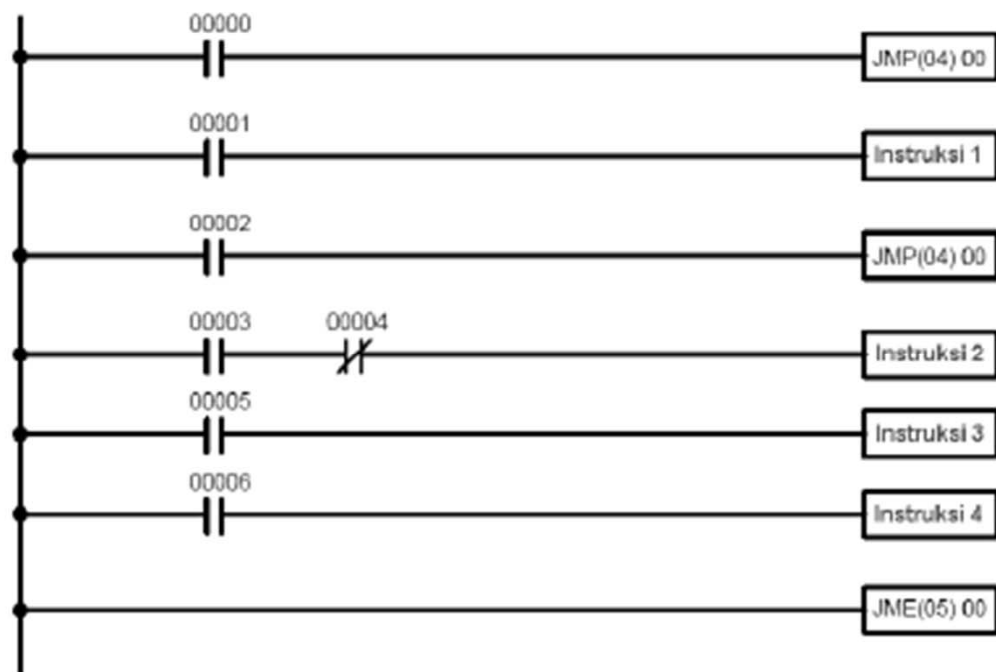
Interlocks dan Interlocks clear merupakan satu pasang instruksi. Jika ada Interlocks maka harus ada instruksi penutupnya yaitu Interlocks clear. Diagram tangga yang berada dalam wilayah IL (02) dan ILC (03) tidak akan bekerja jika IL (02) belum bekerja. Instruksi ini dapat menggantikan diagram tangga yang ada titik percabangannya sehingga diagram tangganya menjadi lebih sederhana.



Alamat	Instruksi	Operan
00000	LD	00000
00001	IL(02)	---
00002	LD	00001
00003	Instruksi 1	
00004	LD	00002
00005	IL(02)	---
00006	LD	00003
00007	AND NOT	00004
00008	Instruksi 2	
00009	LD	00005
00010	Instruksi 3	
00011	LD	00006
00012	Instruksi 4	
00013	ILC(03)	---

10) JUMP (JMP) dan JUMP END (JME)

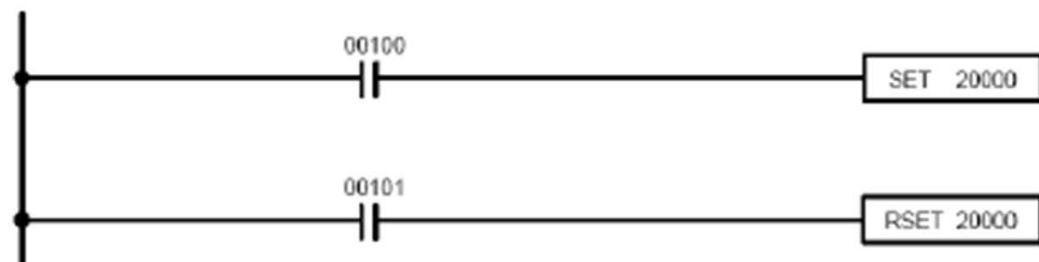
Instruksi ini mirip dengan IL (02) dan ILC (03). Bedanya jika kondisi logika untuk instruksi JMP sudah OFF, kondisi logika output diagram tangga yang berada diantara instruksi JMP dan JME yang mempunyai logic ON akan tetap ON (*latching*), walaupun kondisi input logic-nya sudah OFF.




Alamat	Instruksi	Operan
00000	LD	00000
00001	JMP(04)	00
00002	LD	00001
00003	Instruksi 1	
00004	LD	00002
00005	JMP(04)	00
00006	LD	00003
00007	AND NOT	00004
00008	Instruksi 2	
00009	LD	00005
00010	Instruksi 3	
00011	LD	00006
00012	Instruksi 4	
00013	JME(05)	00

11) SET dan RESET

Instruksi SET dan RESET ini hampir sama dengan instruksi OUT dan OUT NOT, hanya saja instruksi SET dan RESET ini mengubah kondisi status bit operan saat kondisi eksekusinya ON. Kedua instruksi ini tidak akan mengubah kondisi status bit jika kondisi eksekusinya OFF.



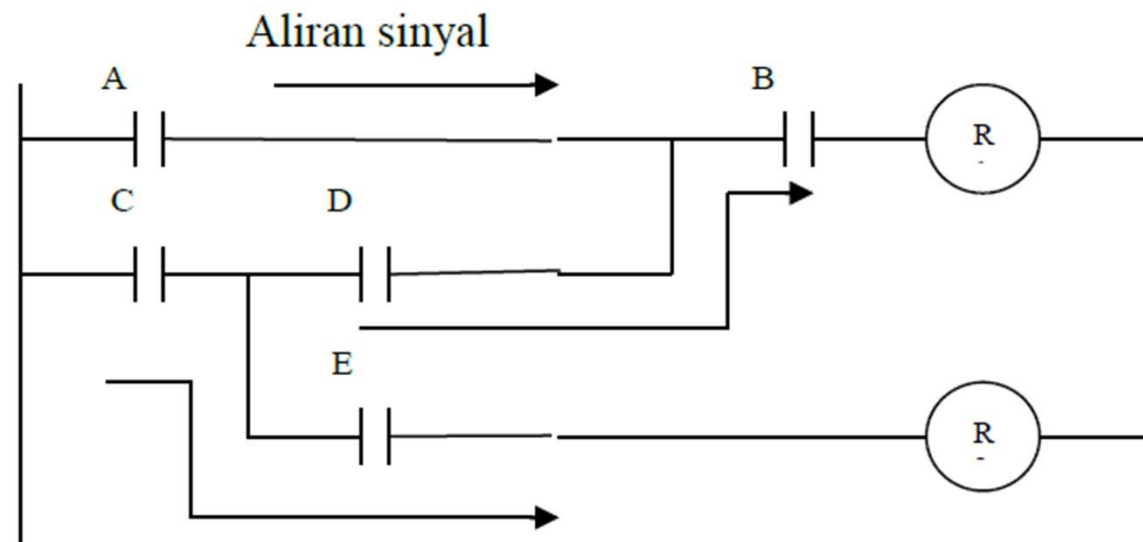
Alamat	Instruksi	Operan
00000	LD	00100
00001	SET	20000
00002	LD	00101
00003	RSET	20000

- 
- Konsep pembuatan program dilakukan dengan diagram tangga,
 - Ada kriteria kriteria tersendiri yang disesuaikan dengan kondisi kerja dari rangkaian logik yang telah dirancang pada PLC.
 - Semua hubungan kontak kontak pada diagram tangga terangkai secara elektronik, jadi tidak memerlukan kaebel-kabel penghubung

- Untuk itu pula agar rangkaian kontrol yang telah dibuat dengan diagram tangga dapat diprogram pada PLC maka beberapa ketentuan yang harus diikuti antara lain :
- 1. Jumlah kontak untuk relay-relay input output, relay-relay bantu, timer, counter, dapat digunakan sesuai dengan kapasitas maksimum yang disediakan oleh PLC.
 - lakukan dengan sesederhana mungkin, sehingga efisiensi kerja PLC dapat lebih optimal.
 - Untuk pembuatan rangkaian control yang relative agak kompleks hendaknya diusahakan menggunakan jumlah kontak seminimal mungkin. sehingga tidak melebihi dari kapasitas memori dari spesifikasi PLC yang hendak digunakan.

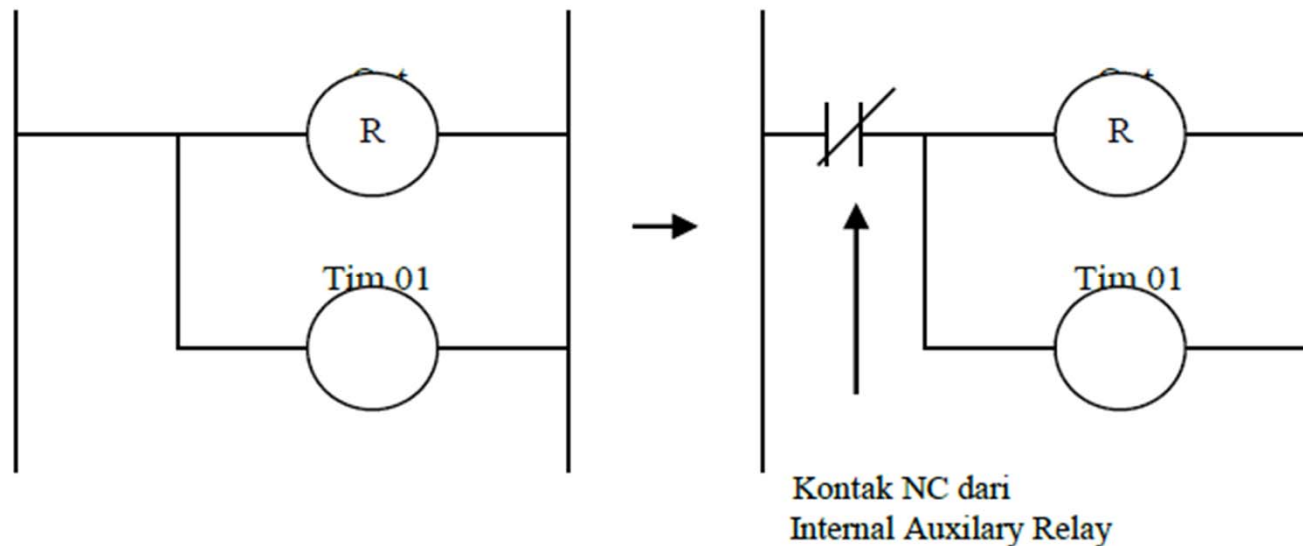
2. Kondisi sinyal yang mengalir pada rangkaian logic yang dirancang pada PLC,

- mengalir dari arah kiri ke kanan

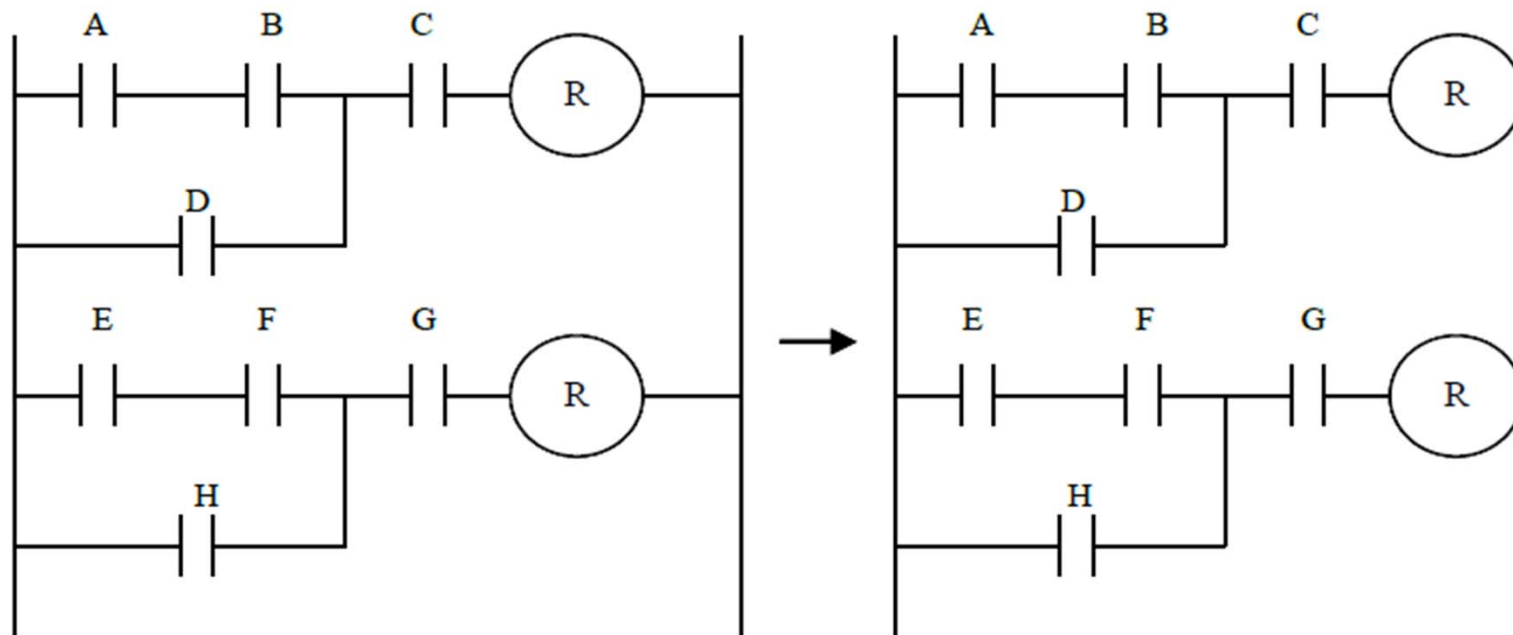


- apabila kontak A dan B dalam kondisi ON, maka sinyal akan mengalir dari A terus ke B dan akan mengaktifkan relay R1. Apabila pada saat tersebut kontak D dan E ON pula, maka relay R2 tidak akan bekerja, karena sinyal tidak akan mengalir dari kontak A, D, dan E.

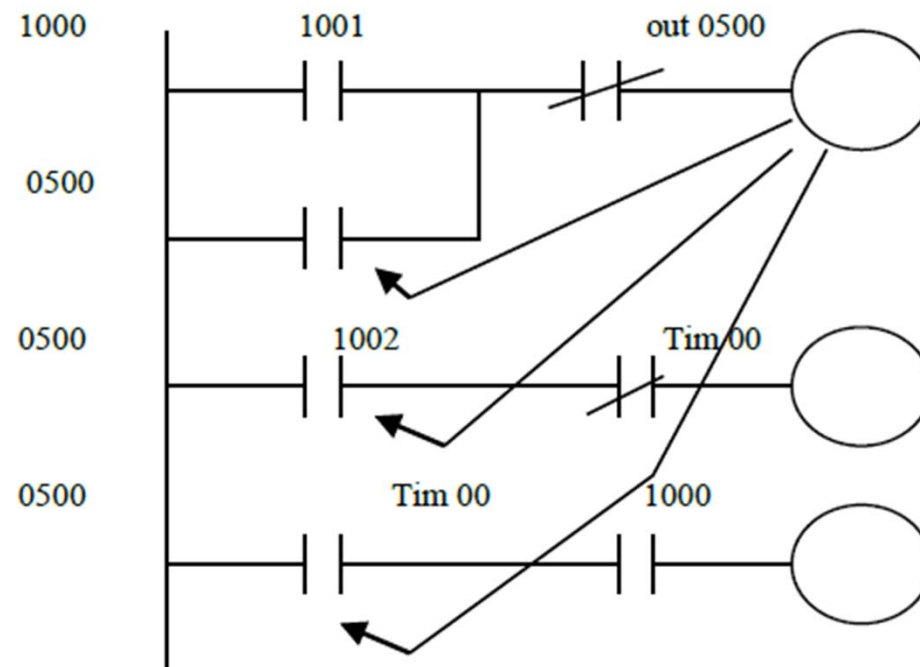
- 3. Tidak satupun koil atau relay output yang dapat dihubungkan langsung pada busbar bagian kiri. Apabila diperlukan bahwa relay output harus bekerja terus menerus, maka diantara busbar kiri dengan relay output diberi kontak NC.



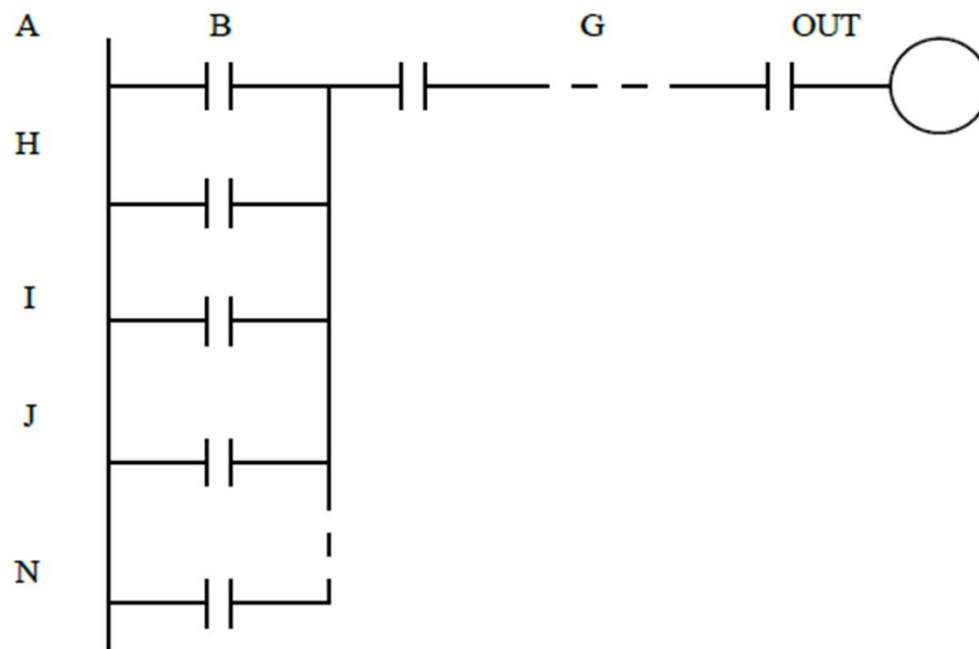
- 4. Busbar sebelah kanan dari diagram tangga boleh tidak digambar, karena hubungan. Busbar tersebut telah tersambung secara otomatis pada PLC.



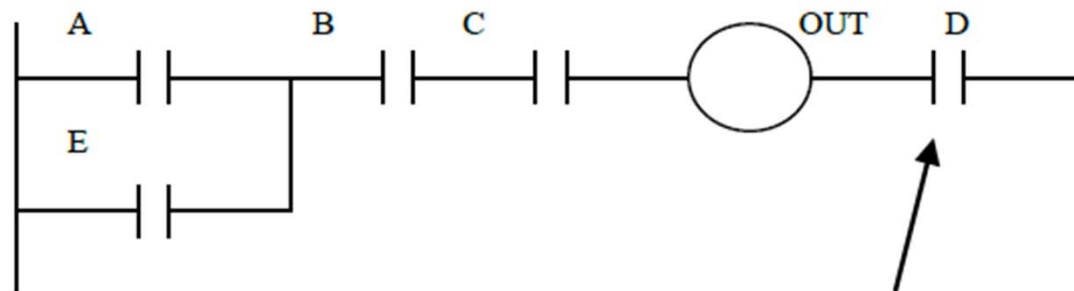
- 5. Semua output dilengkapi dengan kontak-kontak Bantu, yang dapat digunakan dalam jumlah yang tak terbatas dalam program baik dalam hubungan seri maupun paralel.



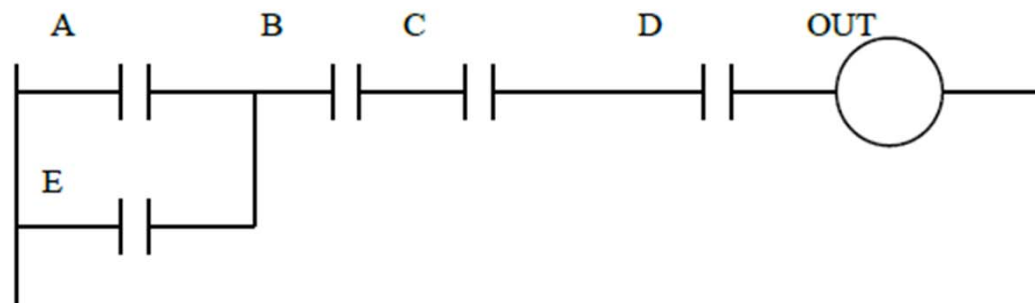
- 6. Jumlah kontak-kontak NO dan NC dapat dihubungkan secara seri maupun paralel dengan tak terbatas.



- 7. Tidak ada kontak yang dapat deprogram atau disisipkan setelah hasil output, atau dengan kata lain tidak ada koil yang diperbolehkan untuk disambung setelah kontak output



Penyisipan kontak setelah koil output tidak dikenal dalam program.



Listing Program

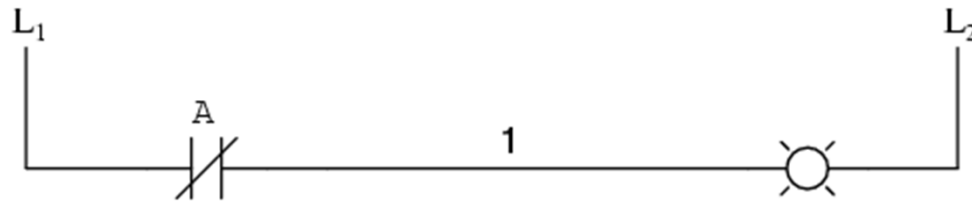
- Diagram Ladder yang telah dipunyai selanjutnya ditulis dalam Listing program

ALAMAT	MNEMONIK	DATA
0000	LD	0000
0001	OR NOT	0002
0002	AND	0001
0003	OUT	1000
0004	FUN 01	-

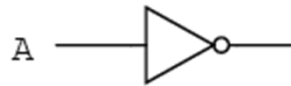
Latihan

- Buatlah Ladder Diagram Untuk Gerbang Logika
 - AND
 - OR
 - Not
 - NAND
 - NOR
 - EX OR
 - EX NOR

Ladder diagram gerbang NOT

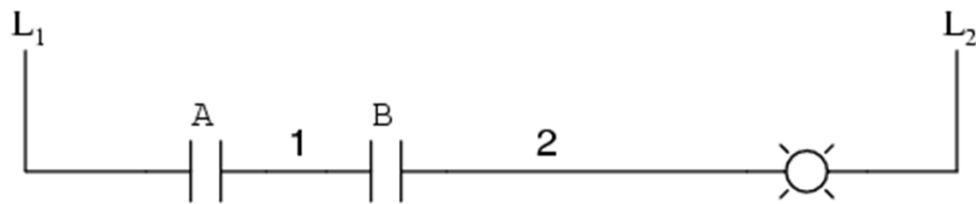


A	Output
0	1
1	0

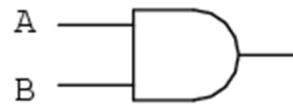


LD NOT A
OUT B
FUN 01

Ladder diagram gerbang AND

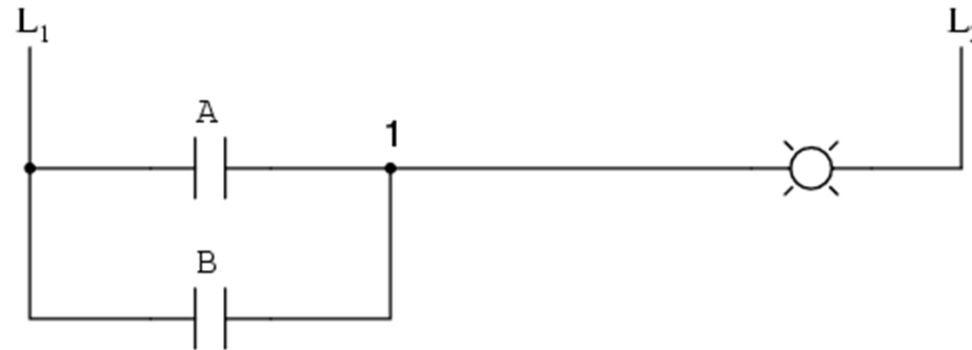


A	B	Output
0	0	0
0	1	0
1	0	0
1	1	1



LD 0000
AND 0001
OUT 1000
FUN 01

Ladder diagram gerbang OR

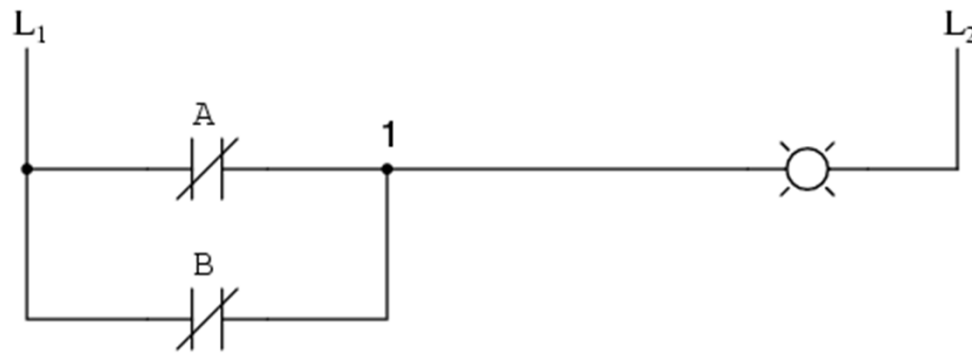


A	B	Output
0	0	0
0	1	1
1	0	1
1	1	1

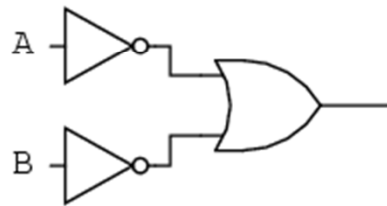


LD 0000
OR 0001
OUT 1000
FUN 01

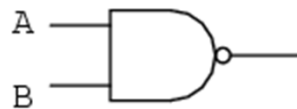
Ladder diagram gerbang NAND



A	B	Output
0	0	1
0	1	1
1	0	1
1	1	0

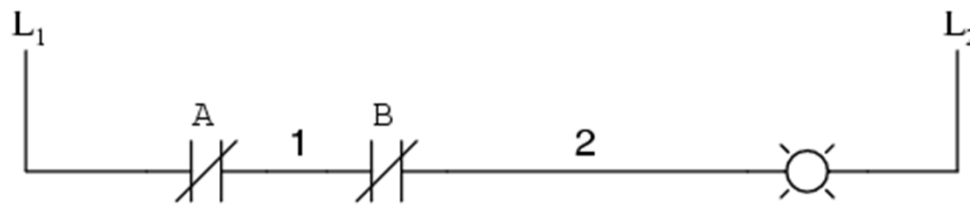


or

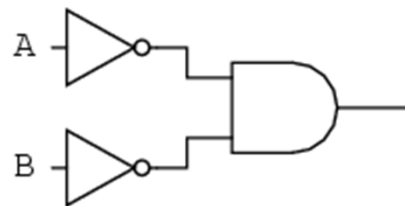


LD NOT 0000
 OR NOT 0001
 OUT 1000
 FUN 01

Ladder diagram gerbang NOR



A	B	Output
0	0	1
0	1	0
1	0	0
1	1	0

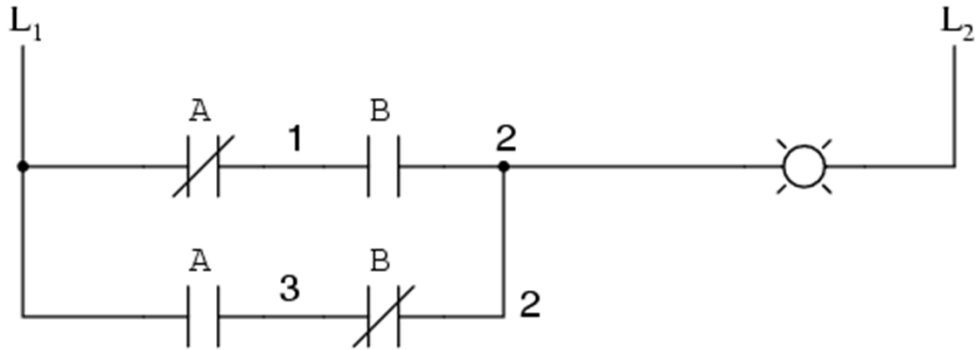


or

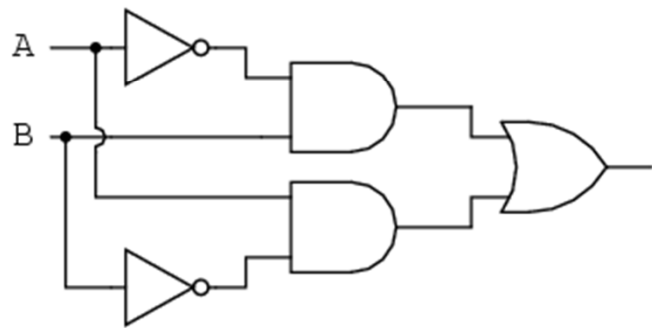


LD NOT 0000
 AND NOT 0001
 OUT 1000
 FUN 01

Ladder diagram gerbang EX-OR



A	B	Output
0	0	0
0	1	1
1	0	1
1	1	0



or



LD NOT A
 AND B
 LD A
 AND NOT B
 OR LOAD
 OUT 1000
 FUN 01

Kasus

- Sebuah motor akan hidup hila dikendalikan oleh suatu sumber dengan pola variasi pada port masukan 0001, 0101, 1101, 1001, 1011, 1111. Hal ini mengandung pengertian bahwa masukan yang diperlukan dalam mode variasi masukan adalah 4 port masukan dan keluaran 1 buah port keluaran.

Solusi

- (1) buat tabel kebenaran dari pola variasi masukan,
- (2) Buat persamaan dari hasil tabulasi yang dimaksud dg *Sum of product* atau *Product of Sum*
- (3) Sederhanakan persamaan yang telah didapatkan
- (4) Gambar rangkaian logika yang telah didapatkan,
- (5) Tentukan dan gambar *ladder diagram nya*
- (6) Tulislah *statement list (listing Programnya)*
-