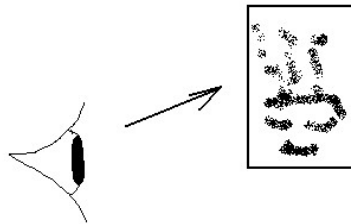


# Waktu Tanggap

**OHT 3**

## ILUSTRASI WAKTU TANGGAP PADA MANUSIA

**INPUT-** Mata melihat Obyek  
( Baca Input sinyal)



**EXECUTION-** (Pengolahan sinyal input di dalam CPU/PLC)

Otak memerlukan waktu untuk mencerna sinyal input dari mata.

Diperlukan waktu untuk melakukan proses informasi.

Pertimbangan untuk mengeluarkan output ke Mulut.



**OUTPUT-** (Ouput menerima sinyal dari CPU/PLC)

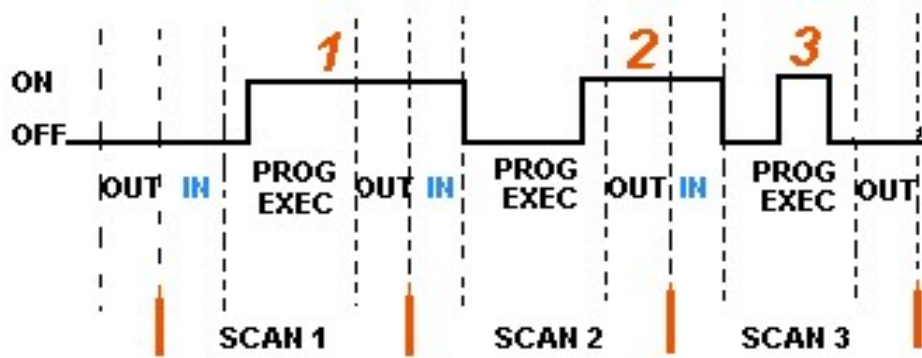
Keluar berupa kata-kata dari mulut.



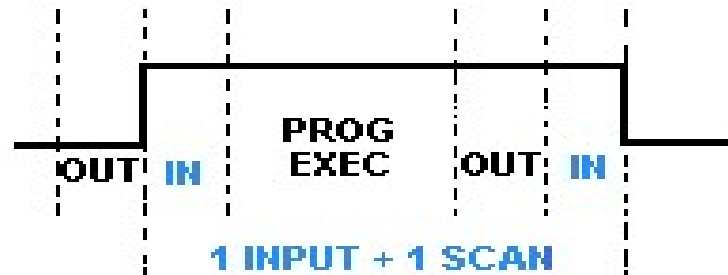
## Permasalahan Pada Waktu Tanggap

OHT 4

MENGUJI STATUS INPUT SEBAGAI BAGIAN DARI SCAN.



Sinyal pada 1 INPUT + 1 SCAN



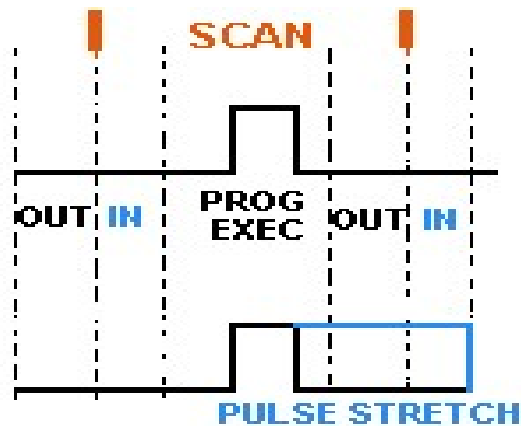
Kondisi yang diinginkan:

Kondisi ON input = 1 input waktu tunda + 1 waktu scan

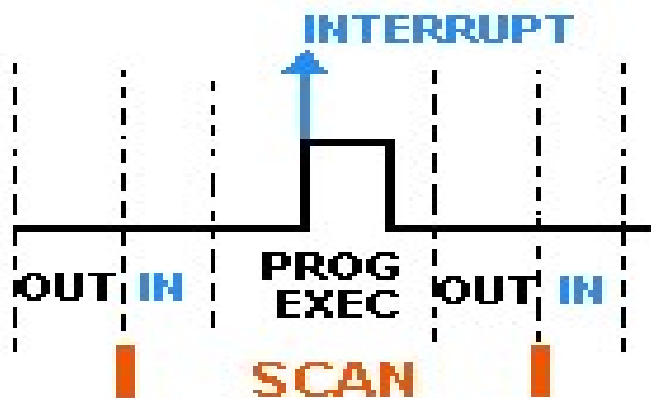
## Permasalahan Pada Waktu Tanggap (Lanjutan)

OHT 5

MEMPERPANJANG WAKTU BACA INPUT (Pulse stretch).



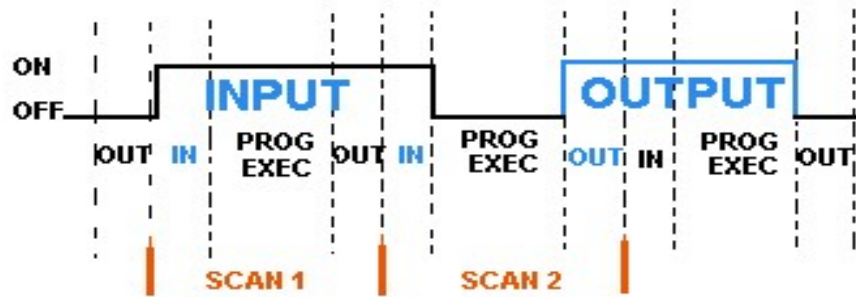
Menginterupsi scan untuk melaksanakan rutin khusus.



## Permasalahan Pada Waktu Tanggap (Lanjutan)

OHT 6

**TUNDA WAKTU PANJANG UNTUK MEMBUAT OUTPUT ON SETELAH INPUT ON**

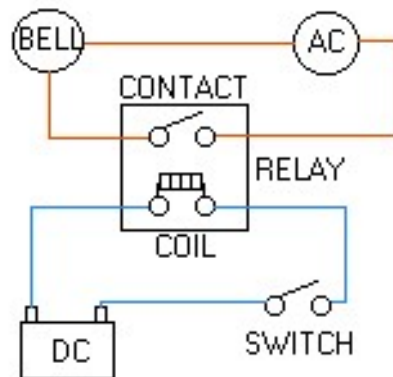


**Tunda waktu maksimum adalah 2 sklus scan - 1 tunda waktu input.**

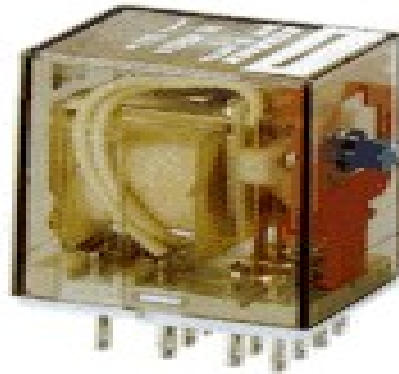
# Relay

**OHT 7**

## Rangkaian Listrik Pada Relay



## Contoh Relay

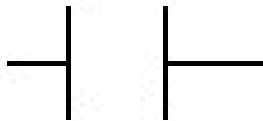


# Penganti Relay

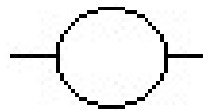


## PLC TIDAK KENAL TERMINOLOGY

SAKLAR, RELAY, BELL, INPUT, OUTPUT, KUMPARAN, KONTAK DLL..



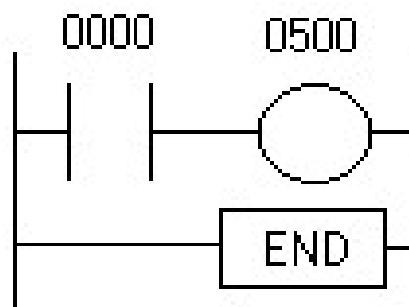
Simbol kontak (input)



Simbol kumparan (output)

Sumber AC merupakan eksternal sumber, sehingga kita tidak menggunakannya pada ladder diagram. PLC hanya melakukan output mana yang akan diaktifkan atau tidak

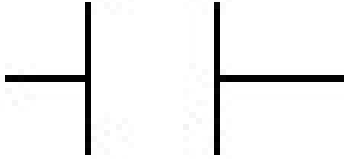
## Ladder Diagram



# Instruksi Dasar PLC

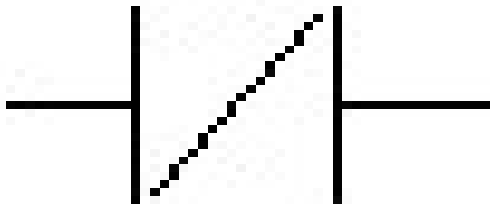
**OHT 9**

Instruksi Load (LD):



**Simbol Load (kontak)**

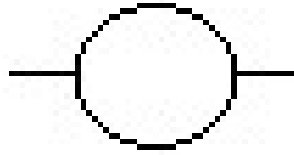
Instruksi LoadBar



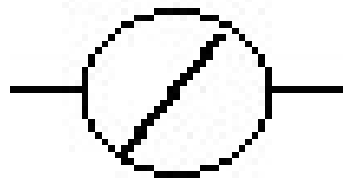
**Simbol LoadNot (normally closed)**

Kondisi Logika	Load	LoadBar
0	False	True
1	True	False

## Instruksi Dasar (Lanjutan)



**Simbol OUT (kumparan)**



**Simbol OUTBar (kumparan normally closed)**

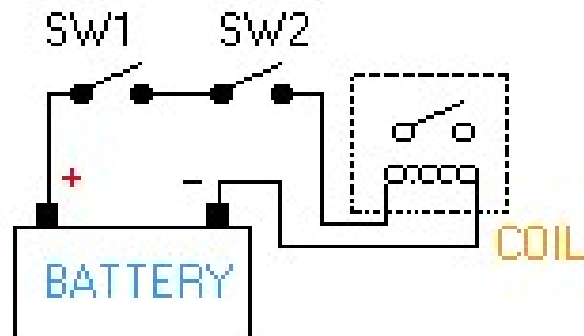
Kondisi Logika	Out	OutBar
0	False	True
1	True	False



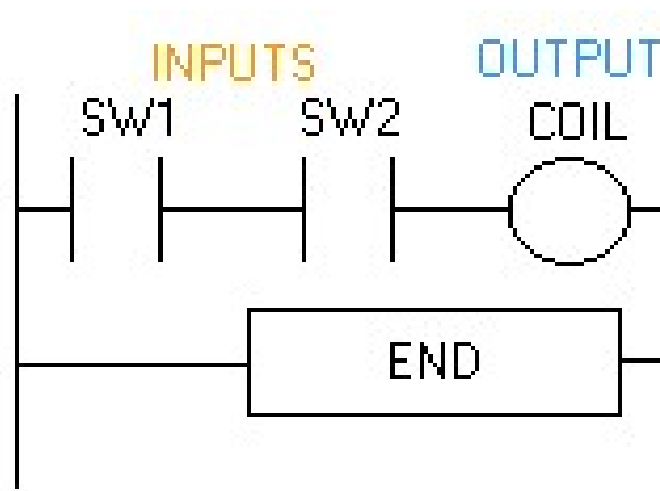
# Contoh Sederhana

**OHT 11**

## Rangkaian Sebuah Relay.



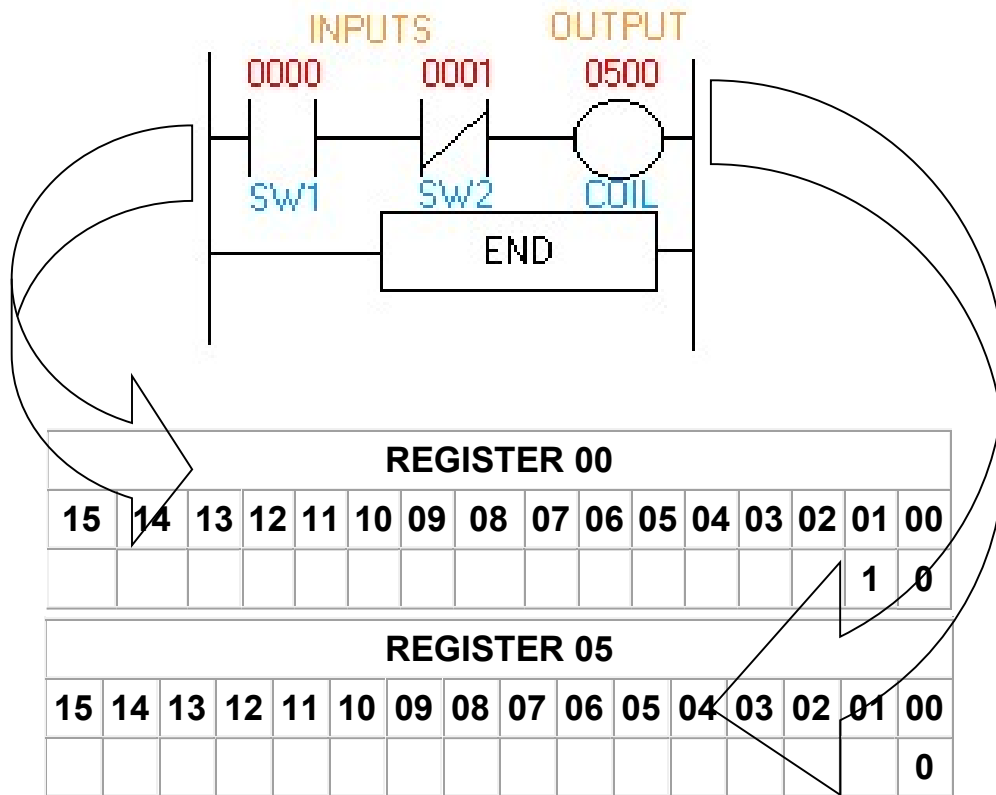
## Ladder Diagram



# Register PLC

OHT 12

## Ladder Diagram



SIMBOL KONDISI LOGIKA			
LOGIKA BIT	LD	LDB	OUT
Logika 0	False	True	False
Logika 1	True	False	True

# Register PLC (lanjutan)



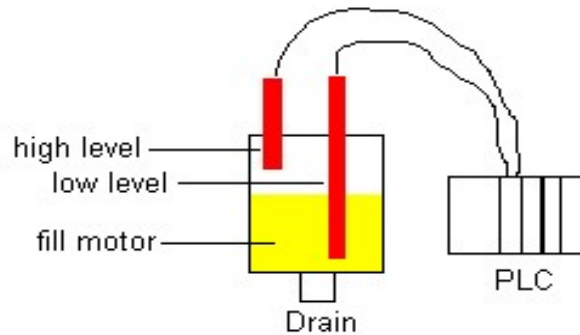
**Tabel Kebenaran**

Input		Output	Register Logika Bit		
SW1(LD)	SW2(LDB)	KUMPARAN(OUT)	SW1(LD)	SW2(LDB)	KUMPARAN(OUT)
False	True	False	0	0	0
False	False	False	0	1	0
True	True	True	1	0	1
True	False	False	1	1	0

# Scan Program

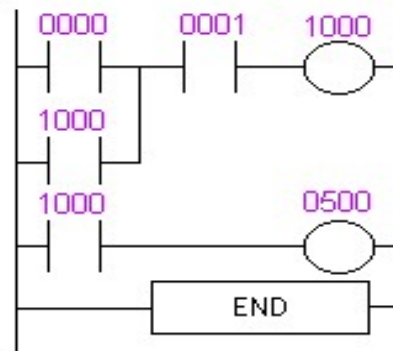
OHT 14

Pengisian minyak pelumas dari tangki, untuk ini digunakan 2 sensor yang diletakan satu di dasar dan di atas.

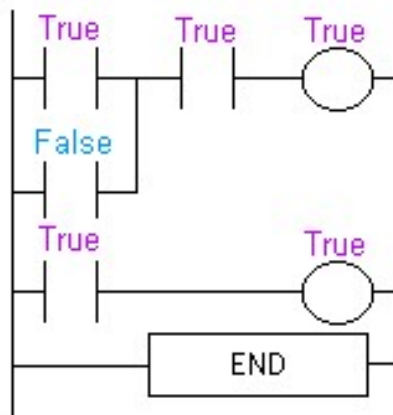


Dispensing oil from a tank

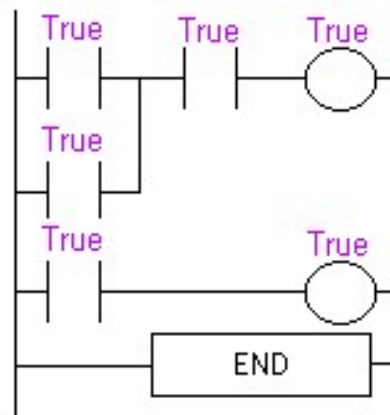
Ladder Diagram dari soal diatas



Saat Tanki kosong, input 0000 True dan input 0001 True juga.



Scan 1

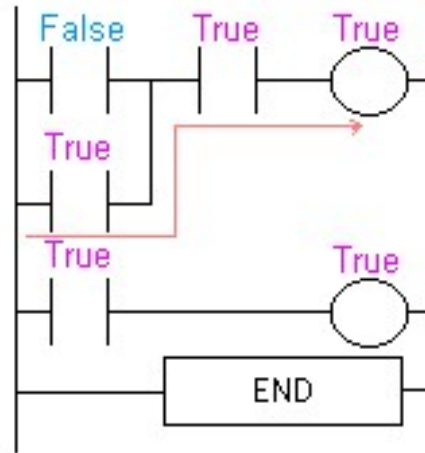


Scan 2

## Scan Program (Lanjutan)

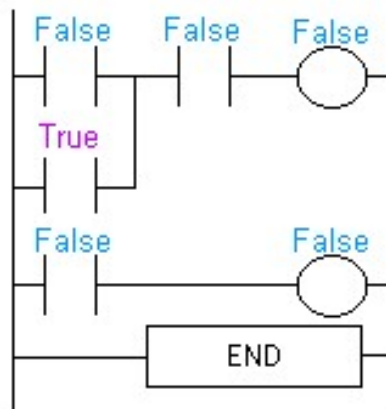
OHT 15

Setelah 100 kali scan level naik dan sensor bawah menjadi False.

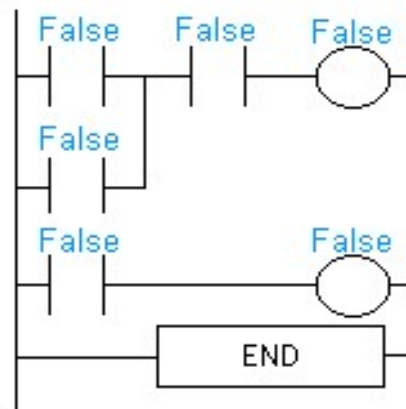


Scan 101

Setelah 1000 kali scan level mencapai sensor atas (True)



Scan 1001

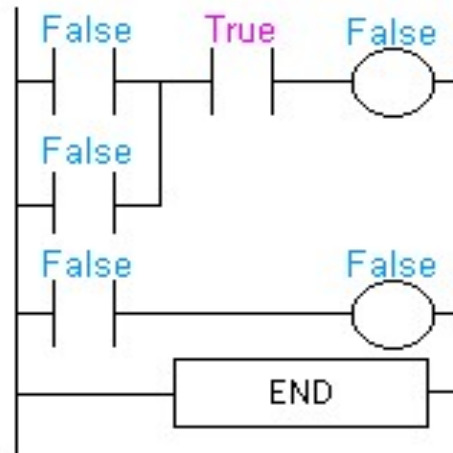


Scan 1002

## Scan Program (Lanjutan)

**OHT 16**

Setelah 1050 kali scan level turun di bawah sensor atas, dengan demikian sensor atas menjadi berkondisi True lagi.



**Scan 1050**

Walaupun sensor level atas berkondisi True, jalur (rung) tetap False karena sensor level bawah terendah minyak berarti False dan kumparan 1000 masih mengingat logika False.

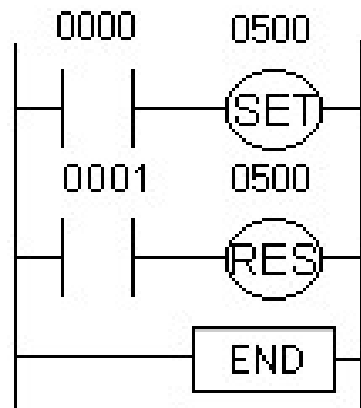
Setelah 2000 kali scan level minyak berada di bawah sensor level bawah, yang berarti sensor berkondisi True lagi. Dengan demikian program scan akan diulangi dari scan 1 kembali dimana motor pompa berkondisi True dan seterusnya.

# Instruksi Latch

**OHT 17**

Instruksi Latch sering disebut dengan SET atau OTL (output latch), kebalikan dari itu adalah instruksi Unlatch yang sering disebut dengan RES (reset), OUT ( output unlatch) atau RST (reset).

Contoh Ladder Diagram:



# Penghitung (Counter)



Penghitung naik yaitu menghitung mulai 1, 2, 3 .....dst, untuk ini disebut dengan CTU (count up) CNT,C , atau CTR.

Penghitung turun yang menghitung turun 9, 8, 7,.....dst. yang disebut CTD (count down) dan penghitung naik/turun yang disebut UDC(up-down counter).

Penghitung dengan kecepatan tinggi, yang sering disebut dengan HSC (high-speed counter), CTH (CounTer High-speed?), penghitung khusus yang berupa "hardware", atau penghitung berupa simulasi "software".

- Aturan yang baik dan mudah adalah menggunakan penghitung normal yaitu dari software kecuali pulsa yang kita hitung lebih cepat 2x dari waktu yang dibutuhkan scan, sebagai
- Jika waktu scan 2 ms dan pulsa yang masuk untuk dihitung setiap kali 4 ms atau lebih lama dibanding penggunaan penghitung software.
- Jika masuk lebih cepat dibanding setiap 4 ms (misal 3 ms) maka gunakan penghitung hardware dengan kecepatan tinggi (high-speed counter). ( $2 \times \text{waktu scan} = 2 \times 2\text{ms} = 4\text{ms}$ )

Untuk menggunakannya kita harus tahu 3 hal:

1. Darimana datangnya pulsa yang ingin kita hitung, biasanya berasal dari satu input atau sensor yang dihubungkan padanya misal pada input 0000.
2. Berapa banyak pulsa yang ingin dihitung sebelum kita mereaksinya, taruhlah hitungan 5 sebelum kita memasukan sesuatu ke dalam kotak.
3. Kapan dan bagaimana kita akan mereset penghitung sehingga dapat memulai menghitung kemabali, misal setelah selesai hitungan 5 maka penghitung direset.

## Batas Hitungan:

Penghitung desimal 16 bit dari 0 sampai 9999, penghitung 16 bit BCD dari -32.768 sampai +32.767 atau dari 0 sampai 65535. dan 16 bit biner menghitung dari -32,768 - 32767 dan 0 to 65535.



# Penghitung (Lanjutan)

OHT 19

Simbol Penghitung dengan Reset:

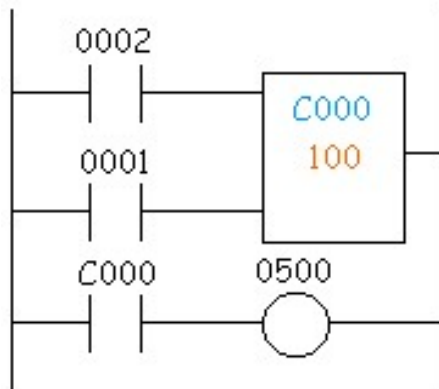


Cxxx adalah nama dari penghitung (counter), jika kita ingin menyebutnya penghitung 000 maka padanya kita tuliskan “C000”.

yyyyy adalah jumlah pulsa yang ingin kita hitung sebelum melakukan sesuatu. Jika penghitung telah selesai misal menghitung yyyy maka akan memberikan kondisi ON kontak-kontak yang berinisial Cxxx pada seluruh ladder diagram.

Catatan bahwa penghitung (counter) menjumlahkan nilai dan penghitungan berlangsung manakala terjadi perubahan transisi dari OFF ke ON pada pulsa input.

Contoh Ladder Diagram:

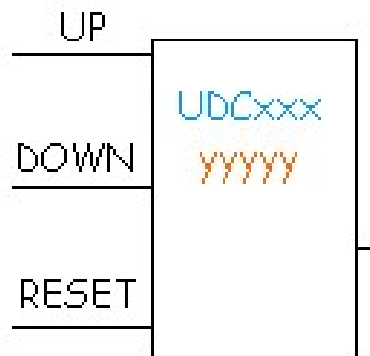


Nama penghitung C000, menghitung dari 0 sampai 100 yang dipicu pulsa yang bersal dari input 0001 sebelum mengaktifkan output 500 serta meresetnya dari sensor yang beralamat input 0002.

## Penghitung (Lanjutan)

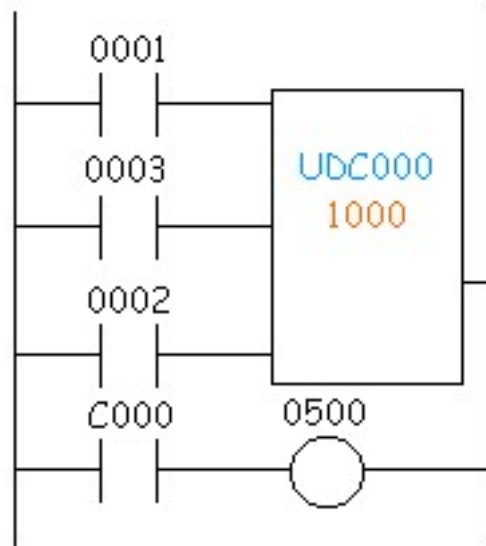
OHT 20

Simbol penghitung naik/turun (up/down counter), dengan nama penghitung UDCxxx dan nilai akumulasi hitungan yyyyyy.



Penghitung (counter) menjumlahkan nilai dan penghitungan berlangsung manakala terjadi perubahan transisi dari OFF ke ON pada pulsa input.

**Contoh Ladder Diagram:**



# Timer



Tipe timer diproduksi oleh pabrik disesuaikan dengan metoda dan cara mereka masing-masing, yaitu.

- **On-Delay timer**-Tipe sangat sederhana, yaitu setelah sensor melalui input berkondisi ON, maka tidak langsung memberikan reaksi ON pada output karena harus menunggu selama waktu yang telah ditentukan ( misal x detik). Timer yang demikian sering disebut dengan TON (timer on-delay), TIM (timer) atau TMR(timer).
- **Off-Delay timer**- Timer ini kebalikan dari timer diatas, setelah sensor melalui input berkondisi ON, maka tidak langsung memberikan reaksi OFF pada output karena harus menunggu selama waktu yang telah ditentukan ( misal x detik). Timer yang demikian sering disebut dengan TOF (timer off-delay).
- **Retentive or Accumulating timer**- Tipe ini merupakan timer yang memerlukan 2 input, satu input untuk start dan yang lain untuk meresetnya. Saat input sensor ON timer mulai berdetak dan bila timer telah mencapai nilai yang telah ditentukan akan mengaktifkan kontaknya sehingga ON, untuk meresetnya yaitu dengan memberikan ON pada reset sehingga nilai kembali pada nilai awal dan kontak dari timer kembali pada kondisi OFF.Timer ini akan mempertahankan nilai detak bialamana timer dihentikan selama waktu detak dan akan melanjutkan nilai tersebut bila timer diaktifkan kembali dan untuk meresetnya adalah melalui input reset. Timer ini sering disebut dengan RTC (retentive timer) atau (accumulating timer).

Timer berdetak mulai 0 sampai 9999 kali atau 0 sampai 65535 kali.

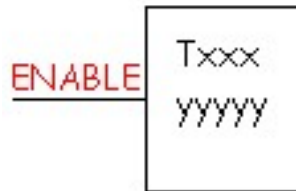
Kebanyakan PLC memiliki timer 16 bit, dimana 0-9999 adalah 16 bit BCD (binary coded decimal) dan 0-65535 adalah 16 bit biner. Setiap detak adalah memakan waktu x detik.

Perbedaan timer pada waktu dasarnya (timebase), yaitu TMH (high speed timer), TMS (super high speed timer) atau TMRAF (accumulating fast timer)

# Timer (Lanjutan)

**OHT 22**

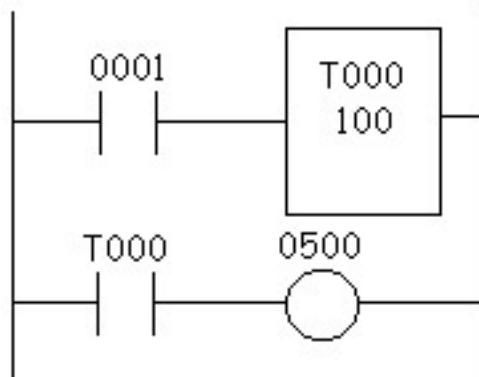
## Simbol Instruksi Timer



## Cara kerja Timer:

- Timer ini masuk dalam kelompok tipe on-delay dengan nama Txxx, ketika input enable berkondisi ON maka timer mulai bekerja (berdetak).
- Ketika ia berdetak yyyyy kali (waktu yang ditentukan), maka akan membuat kontakanya ON.
- Lama waktu detak (naik) antar satu pabrik dengan pabrik pembuat PLC lain adalah bervariasi ( misal 1 ms atau 1 s ).
- Dibawah ini merupakan simbol yang diaplikasikan pada ladder diagram:

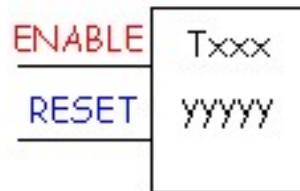
## Contoh Ladder Diagram



# Timer (Lanjutan)

OHT 23

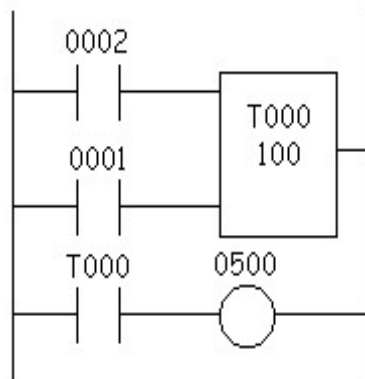
Simbol Timer dengan reset:



Prinsip kerja:

- Timer ini diberi nama Txxx, ketika input enable berkondisi ON maka timer mulai berdetak dan ketika detaknya mencapai yyyyy kali (sebagai nilai yang ditentukan), maka akan menyebabkan kontakanya berkondisi ON.
- Bilamana enable input kembali berkondisi OFF maka timer berhenti berdetak dan nilai timer akan dipertahankan, bila enable input kembali ON maka timer akan melanjutkan dengan nilai yang terakhir.
- Hanya satu jalan untuk membuat timer kembali pada nilai yang ditentukan yaitu dengan memberikan kondisi ON pada input reset.

Contoh Ladder Diagram



**Satu hal penting** adalah antara penghitung dan timer tidak boleh memiliki alamat yang sama, karena keduanya menggunakan alamat register dalam PLC yang sama.

# Ketepatan Timer



Untuk timer dengan durasi dalam mili detik (1ms = 0,001 detik) harus diperhitungkan kepresisiannya.

Terdapat dua macam error pada penggunaan timer:

- pertama input error
- kedua output error

**Input error-** Ini muncul saat input timer berubah on selama siklus scan dan input dibaca sekali selama satu scan, jika kondisi input off ketika PLC membacanya dan terlambat berubah on dalam scan maka akan didapat error. Untuk baca kondisi on tersebut harus menunggu sampai instruksi timer dijalankan selama pelaksanaan program scan berikutnya. Bahayanya jika instruksi timer merupakan instruksi terakhir pada rung maka akan didapat kesalahan yang besar.

**Output error-** Ini muncul pada saat timer pada posisi "times out" dan ketika PLC selesai melaksanakan program sebagai bagian dari scan dimana seharusnya PLC melakukan pembaharuan kondisi output. Hal ini disebabkan timer selesai selama pelaksanaan program sedangkan PLC belum sempat membuat output on karena harus menyelesaikan pelaksanaan program lanjutan.

## Ketepatan Timer (Lanjutan)

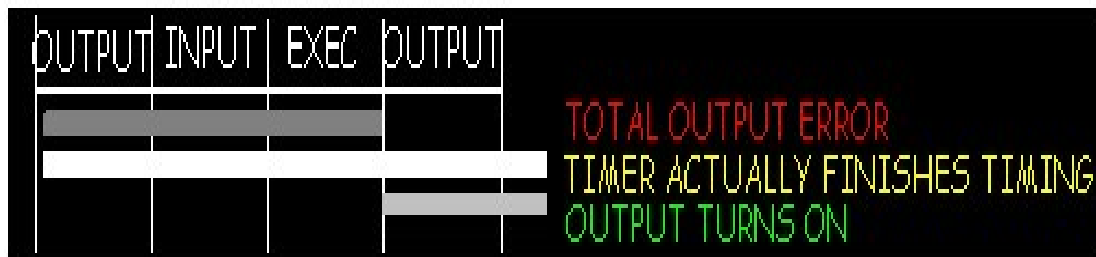
OHT 25

Kemungkinan adanya input error adalah:

- 1 waktu scan penuh + 1 waktu pelaksanaan program.
- Waktu pelaksanaan program bervariasi tergantung berapa banyak instruksi dalam program



Kemungkinan adanya output error adalah 1 waktu scan penuh



Total kemungkinan terjelek error pada timer yaitu:

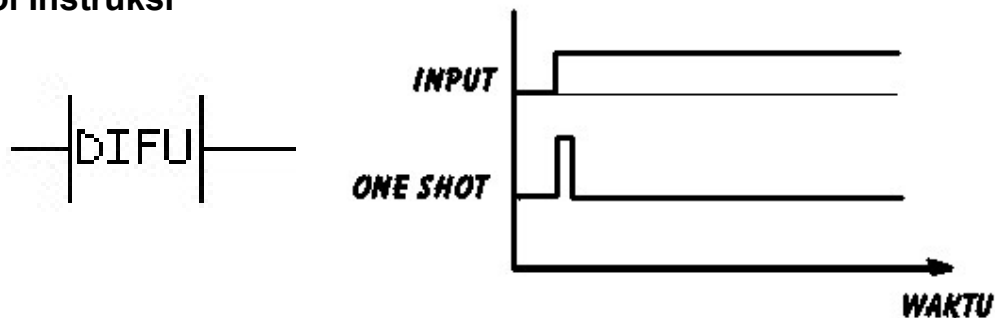
**1 scan + 1 eksekusi + 1 scan = 2 scan + 1 eksekusi program**

# One-shot

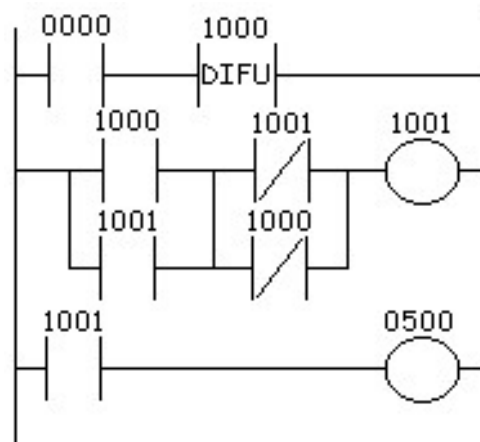
OHT 26

One-shot adalah sebuah alat pemrograman (programming tool), yang digunakan untuk membuat suatu program dengan hanya satu kali scan. Instruksi ini dikenal dengan nama DIFU atau DIFD (differentiate up/down), sotu/sotd (single output up/down), osr (one-shot rising).

## Simbol Instruksi



## Contoh ladder diagram:



## Prinsip kerja ladder

- **Rung 1-** Ketika NO (normally open) input 0000 menjadi true DIFU 1000 menjadi true.
- **Rung 2-** NO 1000 berkondisi true, NO 1001 mencatat false, NC 1001 mencatat true, NC 1000 berubah false. Dengan kita dapatkan jalur true, (NO 1000 & NC 1001) OUT 1001 menjadi true.
- **Rung 3-** NO 1001 berkondisi true oleh karena itu OUT 500 berkondisi true.



## One-shot (Lanjutan)



### Scan berikutnya

- **Rung 1**- NO 0000 mencatat true. DIFU 1000 menjadi false. Hal ini karena instruksi DIFU true hanya untuk satu scan. (misal kenaikan tebing logika padanya sebelum rung)
- **Rung 2**- NO 1000 berkondisi false, NO 1001 mencatat true, NC 1001 berkondisi false, NC 1000 berubah true. Sehingga kita tetap mendapat jalur true, (NO 1001 & NC 1000) OUT 1001 mencatat true.
- **Rung 3**- NO 1001 berkondisi true oleh karena itu OUT 500 mencatat true.

Setelah 100 kali scan, NO 0000 berubah off (menjadi false). Logika mencatat pada kondisi yang sama pada “Scan berikutnya” seperti ditunjukkan diatas, difu tidak memberikan reaksi oleh karena itu logika tetap sama pada rung 2 dan 3.

Pada scan 101 NO 0000 berubah kembali on. (menjadi true)

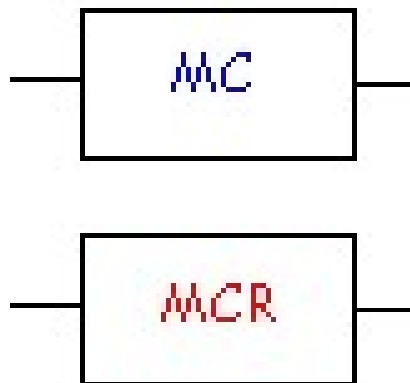
- **Rung 1**- Ketika NO (normally open) input 0000 menjadi true DIFU 1000 menjadi true.
- **Rung 2**- NO 1000 berkondisi true, NO 1001 mencatat true, NC 1001 menjadi false, NC 1000 juga menjadi false. Sehingga kita tidak lagi memiliki jalur true, OUT 1001 menjadi false.
- **Rung 3**- NO 1001 berkondisi false oleh karena itu OUT 500 menjadi false.

# Pengendali Master

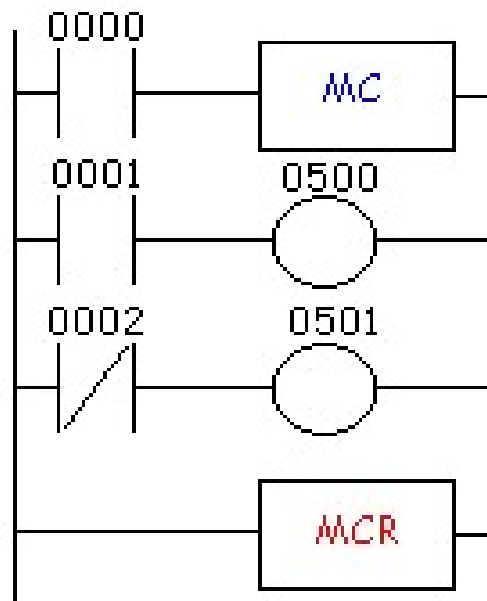
**OHT 28**

Instruksi pengendali master umumnya digunakan secara pasangan dengan pengendali master reset, umumnya disimbolkan dengan MC/MCR (master control set/master control reset) atau MCS/MCR secara sederhana MCR (master control reset).

## Simbol pengendali master



## Ladder Diagram



## Pengendali Master (Lanjutan)



**Timers** sebaiknya tidak digunakan dalam blok mc/mcr karena beberapa diproduksi memberikan reset ke 0 ketika blok false dimana produksi lain akan mempertahankan status waktu saat itu.

**Counters** umumnya mempertahankan nilai hitungannya saat itu.

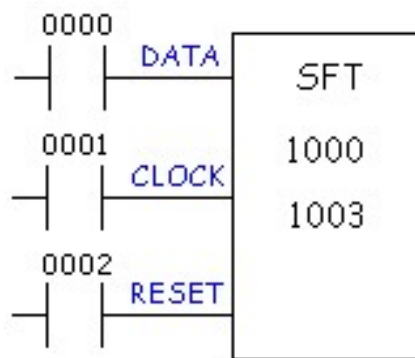
# Register Geser



**Fungsi Register** Digunakan untuk menyimpan status atau kondisi dari suatu keadaan yang terjadi sebelumnya dan merupakan deretan bit untuk menyimpan status on/off. Setiap kali perubahan status akan disimpan pada bit pertama dan bit-bit yang tersimpan di register digeser ke bit berikutnya.

**Nama register geser** SFT (Shift), BSL (Bit Shift Left), SFR (Shift Forward Register), pada register-register ini data digeser ke kiri. Sedangkan yang lain yaitu BSR (Bit Shift Right) dan SFRN (Shift Forward Register Not) digeser ke kanan.

## Sintak Instruksi Register



**Data-** Penampung input data dengan status true/false yang akan digeser pada register. Bilamana data input true maka bit pertama dalam register akan berisi 1. Data tersebut dapat masuk ke register hanya pada saat ada clock dan memanfaatkan sisi tebing (edge) naik dari input clock.

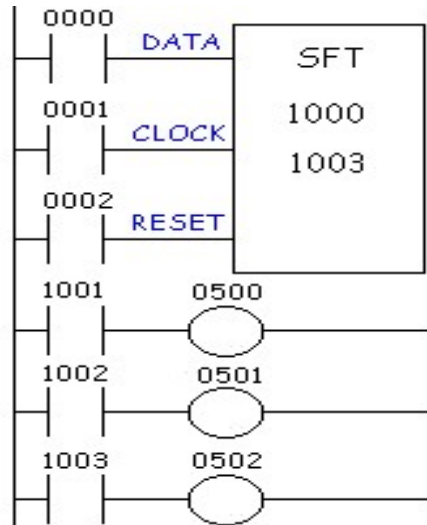
**Clock-** Input clock input merupakan pemberitahuan ke register geser untuk melakukan sesuatu pada sisi tebing (edge) naik dari input clock, register geser akan menggeser data satu lokasi ke lokasi berikutnya dan input data akan menempati bit pertama dari register geser. Setiap kali clock proses input data akan diulangi.

**Reset-** Input reset berfungsi untuk menghapus sehingga isi register geser menjadi 0.

# Register Geser (Lanjutan)

OHT 31

## Contoh Ladder diagram



## Struktur Register pada PLC

10xx Register															
15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
												0	0	0	0

Register 1000, 1001, 1002, dan 1003 berisi status false

10xx Register															
15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
												1	0	0	1

Register 1000, dan 1003 berisi status true

Register 1001 dan 1002 berisi status false

## Ambil dan Pemindahan Data

OHT 32

**Mengapa kita menginginkan mendapat data?**

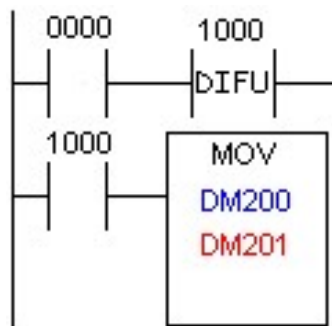
Jawabnya adalah sederhana, data perlu dipindahkan atau dihapuskan, terkadang kita berkeinginan menyimpan data konstan, mengambil data biner dari terminal input, melakukan proses matematika dan menyimpan hasilnya ditempat yang berbeda.

**Simbol instruksi MOV**



- **Sumber (xxxx)** Tempat dimana data ingin dipindahkan, jika kita tulis DM100 maka data yang ingin dipindahkan berada di lokasi memori 100.
- **Tujuan(yyyy)** Tempat dimana data akan ditempatkan, jika tuliskan DM201 maka data akan dituliskan di lokasi memori 201. Dan jika ditulis 0500 berarti data dipindahkan secara fisik ke output 0500 (LSB), sampai output 0515 (MSB).

**Contoh Ladder diagram**



## Ambil dan Pemindahan Data (Lanjutan)

OHT 33

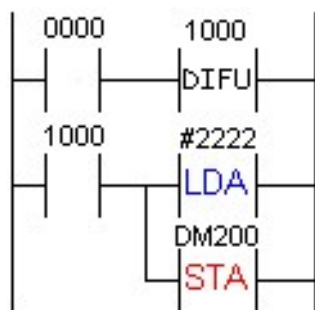
Instruksi pasangan

LDA (LoaD Accumulator) dan STA (STore Accumulator).



- **LDA-** Instruksi mirip dengan instruksi MOV. Tempat dimana data akan dipindahkan, jika kita tulis DM100 maka data yang diinginkan dipindahkan berada di lokasi memori 100.
- **STA-** Instruksi mirip dengan instruksi MOV. Tempat dimana data akan dipindahkan, jika kita tulis DM201 maka data yang diinginkan dipindahkan berada di lokasi memori 201. Jika dituliskan 0500 berarti data dipindahkan secara fisik ke output 0500 (LSB) sampai output 0515 (MSB).

Contoh Ladder Digram:



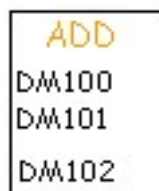
# Instruksi Matematika

OHT 34

Hampir semua PLC memiliki fungsi berikut:

- Penjumlahan - Kemampuan menambahkan data, instruksi ini disebut instruksi ADD.
- Pengurangan - Kemampuan mengurangi data, instruksi ini disebut instruksi SUB.
- Multiplikasi - Kemampuan mengalikan data, instruksi ini disebut instruksi MUL.
- Pembagian - Kemampuan membagi, instruksi ini disebut instruksi DIV.

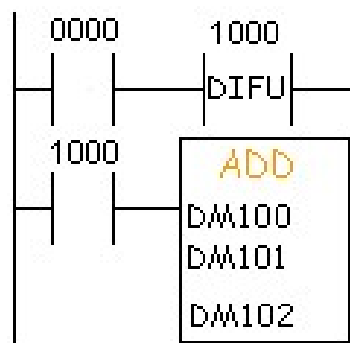
Sintak Instruksi:



- Sumber DM100- Alamat memori dimana data pertama berada
- Sumber DM101- Alamat memori dimana data kedua berada
- Tujuan – Alamat memori dimana hasil logika ditempatkan.

**DM100 ADD DM101 = DM102**

Contoh Ladder Diagram Instruksi ADD

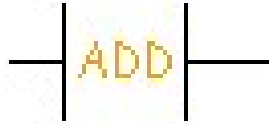




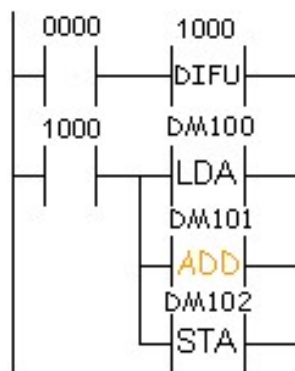
# Instruksi Matematika (Lanjutan)

OHT 35

Simbol ADD (dual method)



Contoh Ladder Diagram



Jika diperoleh hasil lebih besar dari kemampuan lokasi memori

Memori umumnya memiliki 16 bit lokasi

bilangan lebih besar 65535 ( $2^{16}=65535$ ) akan terjadi over flow

Beberapa PLC menggunakan 32 bit fungsi matematika

Pembagian dengan nol (illegal) akan terjadi overflow

Operasi matematika pada PLC

- Akar kuadrat
- Skala
- Nilai mutlak
- Sinus
- Cosinus
- Tangen
- Logaritma Natural
- Logaritma berbasis 10
- $X^Y$  (X pangkat Y)
- Arcsinus (tan, cos)

Beberapa PLC menggunakan Floating point

10 dibagi 3 akan didapatkan 3,33333

# Fungsi Boolean

OHT 36

## Tabel Kebenaran AND

Hasil = A AND B		
A	B	Hasil
0	0	0
1	0	0
0	1	0
1	1	1

## Tabel Kebenaran OR

Hasil = A OR B		
A	B	Hasil
0	0	0
1	0	1
0	1	1
1	1	1

## Tabel Kebenaran EXOR

Hasil = A XOR B		
A	B	Hasil
0	0	0
1	0	1
0	1	1
1	1	0

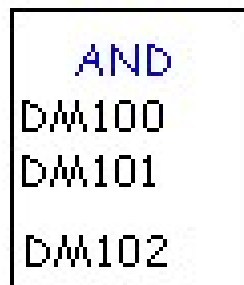
## Fungsi Boolean (Lanjutan)

OHT 37

Instruksi logika dinyatakan dengan

- **AND, ANDA, ANDW**
- **OR, ORA, ORW**
- **XOR, EORA XORW.**

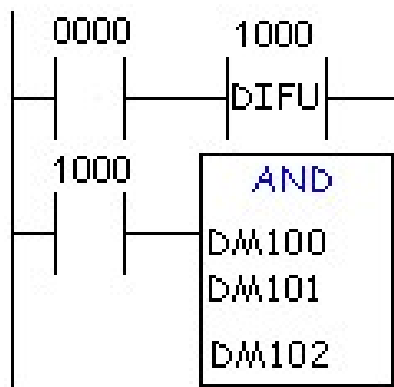
### Sintak instruksi Logika



- Sumber DM100- Alamat memori dimana data pertama berada
- Sumber DM101- Alamat memori dimana data kedua berada
- Tujuan – Alamat memori dimana hasil logika ditempatkan.

**DM100 AND DM101 = DM102**

### Contoh Ladder Diagram Logika AND



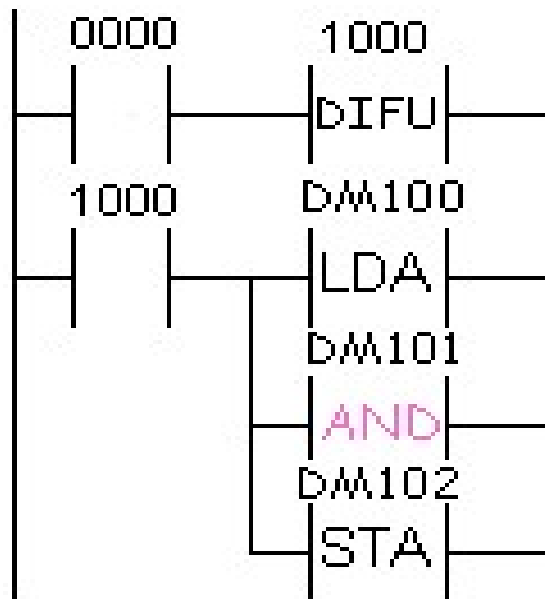
# Fungsi Boolean (Lanjutan)

**OHT 38**

Simbol AND (instruksi dual)



Contoh Ladder diagram Instruksi AND



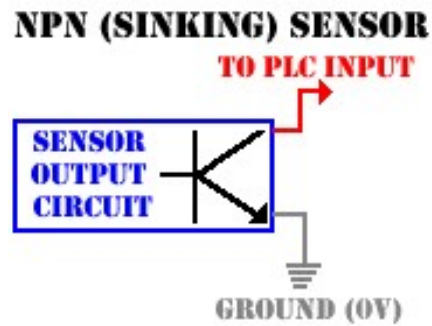
Contoh ini berlaku juga untuk fungsi OR atau EXOR

# Input DC

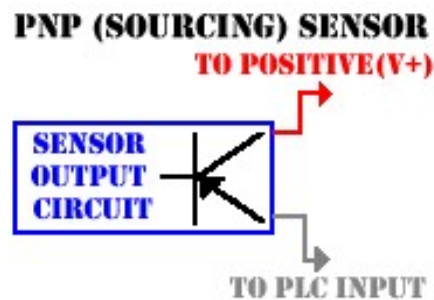
OHT 39

Sambungan output sensor ke input NPN dan PNP.

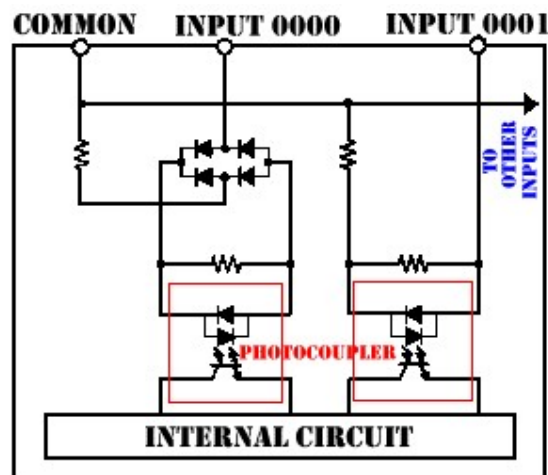
Pemberian tegangan positif.



Pemberian tegangan positif sumber



## Rangkaian Input DC



# Input AC

**OHT 40**

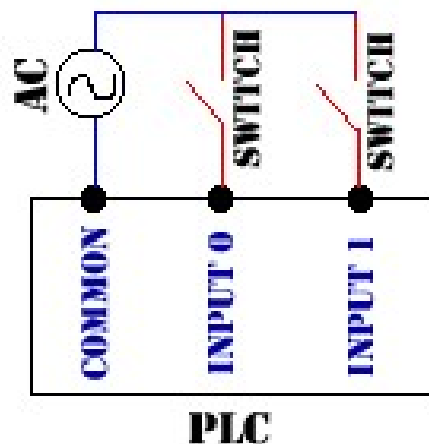
**Hati-hati AC cukup berbahaya.**



UCH...

- Modul input AC jarang digunakan dibandingkan input DC
- Transistor tidak dapat bekerja dengan tegangan AC
- Tegangan AC digunakan saklar atau saklar limit dsb.
- Operasi sensor dengan tegangan DC.
- Digunakan kawat dengan isolasi berwarna coklat (aktif), biru (netral) dan hijau/kuning (ground).

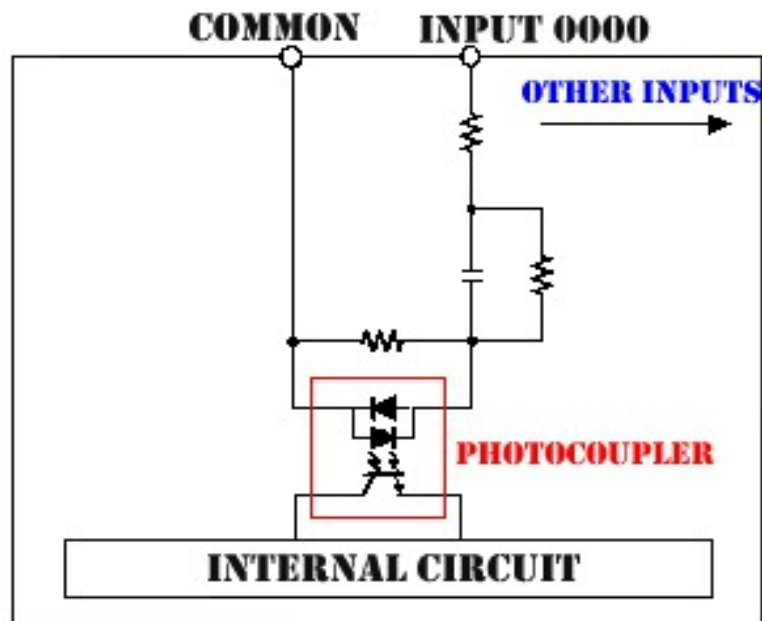
## Rangkaian Input AC pada PLC



# Input AC

OHT 41

## Rangkaian modul input AC pada PLC



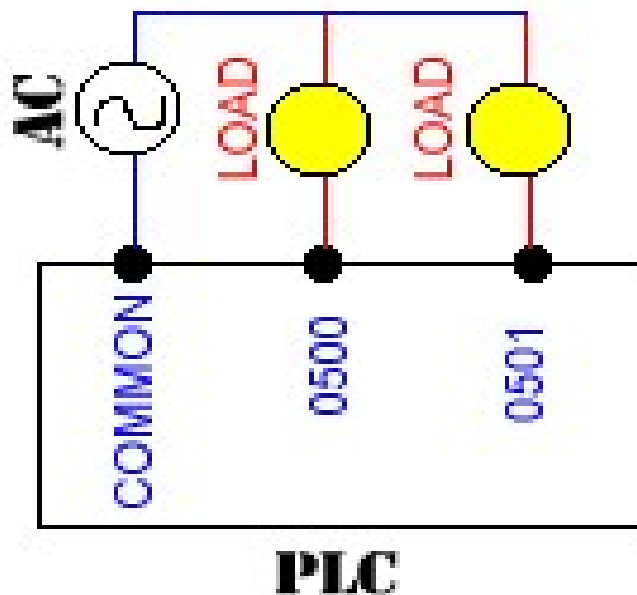
- Rangkaian photo-coupler bekerja berdasarkan konversi sinyal input listrik melalui sinar (cahaya) diubah kembali menjadi sinyal listrik untuk diproses lebih lanjut oleh rangkaian internal.
- Pada input AC jauh lebih lambat dibanding input DC.
- Saklar mekanik sangat lambat, membutuhkan 25 mili detik atau lebih, sehingga dibutuhkan rangkaian internal delay

## Output Relay

**OHT 42**

- Beberapa beban yang sering digunakan adalah selenoid, lampu, motor listrik dll.
- Sebelum menghubungkan beban pada PLC sebaiknya lihat terlebih dahulu spesifikasi beban, perlu arus berapa bekerja dengan tegangan berapa sebab bila arus dan tegangan kerja melebihi tegangan atau arus yang diijinkan pada output relay maka akan berakibat relay terbakar.

### Cara Menghubungkan Output Relay

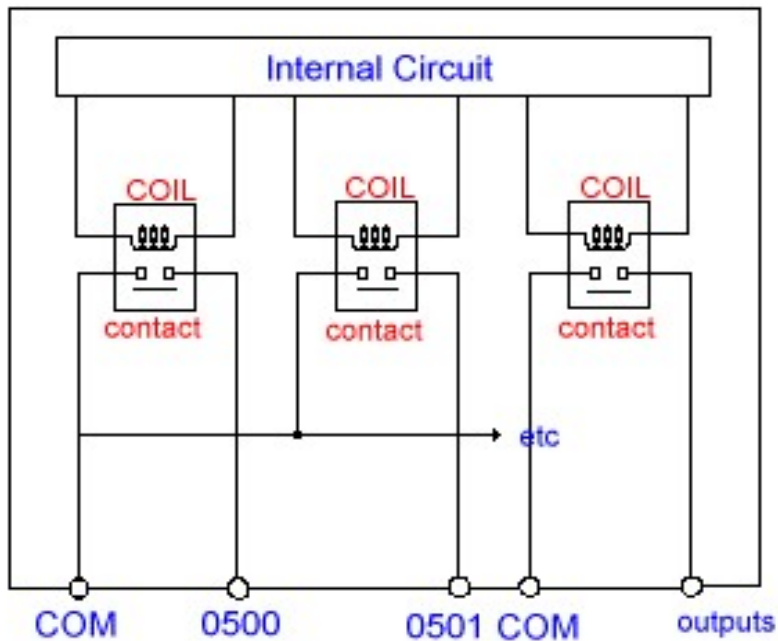




# Output Relay

**OHT 43**

**Internal relay PLC berfungsi menghubungkan antara terminal output PLC dengan rangkaian internal PLC**



# Output Transistor

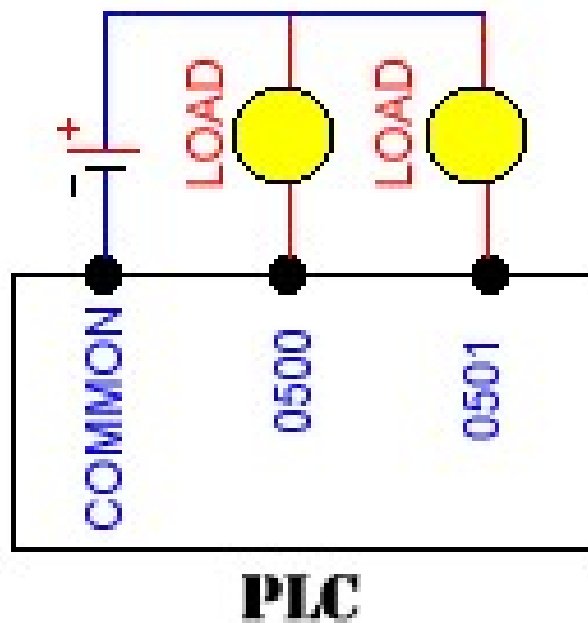
**OHT 44**

Output transistor hanya dapat bekerja dengan arus atau tegangan DC

Terdapat dua tipe

- Transistor bipolar yaitu PNP dan NPN
- MOSFET (Metal Oxide Semiconductor- Field Effect Transistor)

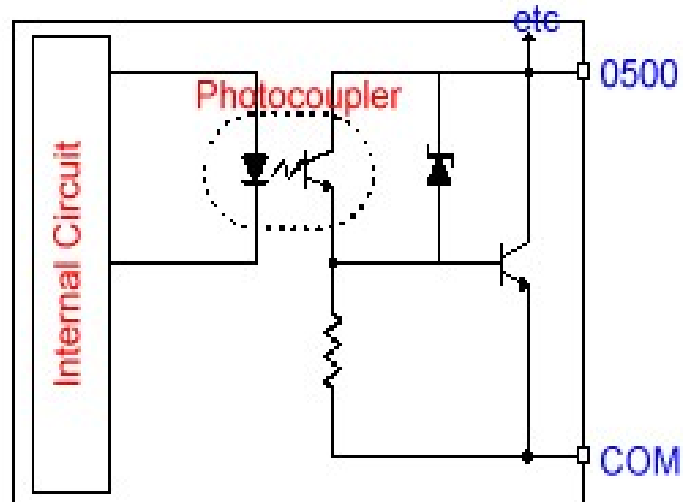
Rangkaian Beban Pada Output Transistor Npn,



# Output Transistor

**OHT 45**

## Rangkaian Output Photocoupler



### Output Transistor:

Tidak dapat memberikan beban yang lebih besar dibanding relay, ON/OFF jauh lebih cepat dibanding relay, saklar untuk arus yang kecil, dapat bertahan lama dibanding relay karena pada kontak muncul percikan api dan hanya bekerja dengan tegangan DC.

## **BAB 5 CARA MENILAI UNIT INI**

### **Apa yang dimaksud dengan penilaian?**

Penilaian adalah proses pengumpulan petunjuk dan pembuatan penilaian atas kemajuan kearah ketercapaian kriteria unjuk kerja yang dimaksud dalam Standar Kompetensi. Pada poin yang tepat, penilaian dilakukan dengan mengetahui apakah kompetensi sudah dicapai atau belum. Penilaian cenderung mengidentifikasi prestasi-prestasi peserta pelatihan dibanding menampilkan unjuk kerja relatif antara peserta dengan peserta lain.

### **Apakah yang kita maksud dengan kompeten?**

Tanyakan pada diri anda, "Apa yang benar-benar dibutuhkan oleh karyawan untuk melakukan sesuatu?". Jawaban terhadap pertanyaan kepada anda yaitu apa yang kita maksudkan dengan sebuah kata "kompeten". Untuk menjadi kompeten dalam suatu pekerjaan yang berkaitan dengan ketrampilan berarti bahwa orang tersebut harus mampu untuk:

- unjuk kerja pada tingkat ketrampilan yang dapat diterima
- mengorganisir tugas-tugas yang dibutuhkan
- merespon dan mereaksi secara layak bila sesuatu salah
- menjalankan suatu peranan dalam skema sesuatu pada pekerjaan
- mentransfer ketrampilan dan pengetahuan pada situasi baru.

Bila anda menilai kompetensi ini anda harus mempertimbangkan seluruh issue-issue diatas untuk mencerminkan kerja sebenarnya dan alami.

### **Pengakuan Kompetensi Pengakuan kemampuan yang dimiliki**

Prinsip penilaian nasional terpadu memberikan pengakuan terhadap kompetensi yang ada tanpa memandang dimana kompetensi tersebut diperoleh. Penilaian mengakui bahwa individu-individu dapat mencapai kompetensi dalam berbagai cara:

- kualifikasi terdahulu
- belajar secara informal.

Pengakuan terhadap Kompetensi yang ada dengan mengumpulkan petunjuk untuk menilai setiap individu terhadap standar kompetensi agar dapat menentukan apakah mereka telah

### **Kualifikasi penilai**

Dalam kondisi lingkungan kerja, yaitu seorang penilai industri yang diakui dapat menentukan apakah seorang pekerja mampu melakukan tugas yang terdapat dalam unit kompetensi ini. Jika anda diakui untuk menilai unit ini kemungkinan anda dapat memilih metode yang ditawarkan dalam pedoman ini, atau mengembangkan metode anda sendiri untuk melakukan penilaian. Para penilai harus memperhatikan petunjuk bukti dalam standar kompetensi sebelum memutuskan metode penilaian yang akan dipakai.

## Ujian yang disarankan

Unit Kompetensi ini, secara umum mengikuti format berikut:

- (a) menampilkan ketrampilan dan pengetahuan penunjang untuk setiap elemen kompetensi/kriteria unjuk kerja, dan
- (b) berhubungan dengan sesi praktek atau tugas untuk memperkuat teori atau layanan praktek dalam suatu ketrampilan.

Ini penting sekali bahwa peserta dinilai (penilaian formatif) pada setiap elemen kompetensi. Mereka tidak dapat mengikuti progress unit berikutnya sampai mereka benar-benar berkemampuan pada materi yang melingkupi sesi pelatihan.

Sebagai patokan keharusan disini adalah paling sedikit satu penilaian tugas untuk pengetahuan pendukung pada setiap elemen kompetensi. Setiap sesi praktek atau tugas disarankan dinilai secara individu untuk sub kompetensi. Sesi praktek diharuskan untuk diulang sampai tingkat yang disyaratkan dari sub kompetensi dapat dicapai.

Tes pengetahuan penunjang biasanya digunakan tes obyektif. Sebagai contoh, pilihan ganda, komparasi, mengisi/melengkapi kalimat. Penggunaan Tes Essay berupa pertanyaan biasanya tidak cocok untuk tipe unit ini.

Penilaian untuk unit ini, berdasar pada dua hal yaitu:

- pengetahuan dan ketrampilan pendukung
- hubungan dengan ketrampilan praktek.

Untuk unit Penggunaan Pelatihan Berdasar Kompetensi pada tempat kerja penilaian berikut disarankan untuk digunakan:

### Penilaian Ketrampilan dan Pengetahuan Penunjang

Elemen satu ...Mengidentifikasi dan menjelaskan bagian komponen, dan operasi PLC sederhana

#### Pengujian satu

1. Gambarkan secara blok PLC dan jelaskan fungsi setiap komponen.
2. Jelaskan tiga dasar siklus scanning.
3. Jelaskan metode untuk mengatasi problem Tanggap waktu PLC, dengan menggunakan timing diagram.
4. Sebutkan keuntungan penggunaan PLC dibanding kontrol industrial dari

Elemen dua: ..... Menginterpretasi ladder diagram dan menampilkan pemrograman PLC sederhana

#### Pengujian dua

1. Jelaskan instruksi I/O dasar – Load, Load Not, Out, Out Not dan terapkan dalam aplikasi sederhana.
2. Jelaskan penggunaan Latch, Timers, One-shots, Master Control dan Shift Registers dan terapkan dalam aplikasi sederhana.
3. Jelaskan apa yang dimaksud dengan akuisisi dan perpindahan data dan terapkan pada aplikasi sederhana.