

PLC

Dasar-dasar Pemrograman

Kursus ini dirancang bagi peserta yang akan membuat program kontrol untuk pengontrol terprogram untuk pertama kalinya.

Pendahuluan Tujuan kursus

Kursus ini menjelaskan tentang pemrograman, yang dapat digunakan untuk pengontrol terprogram MELSEC.

Salah satu bahasa pemrograman utama adalah Ladder Diagram (LD). Kursus ini mencakup poin-poin penting pemrograman ladder termasuk berbagai perintah utamanya.

Beberapa bagian dalam kursus ini didasarkan pada kursus-kursus dasar pengontrol terprogram MELSEC. Sebaiknya ambil kursus dasar yang relevan sebelum Anda mengikuti kursus ini.

Pendahuluan Struktur kursus

Berikut adalah daftar isi kursus.

Bab 1 - Pemrograman PLC

Bab ini menjelaskan poin-poin penting pemrograman ladder.

Bab 2 - Perintah device bit

Bab ini menjelaskan tentang perintah yang memuat device bit (ON/OFF).

Bab 3 - Perintah device word

Bab ini menjelaskan tentang perintah yang memuat device word (numerik).

Bab 4 - Perintah pencabangan program

Bab ini menjelaskan tentang perintah yang memuat program bercabang.

Tes Akhir

Nilai lulus: 60% atau lebih tinggi.

Pendahuluan**Cara menggunakan alat e-Learning ini**

| | | |
|------------------------------------|--|---|
| Buka halaman berikutnya | | Buka halaman berikutnya. |
| Kembali ke halaman sebelumnya | | Kembali ke halaman sebelumnya. |
| Beralih ke halaman yang diinginkan | | "Daftar Isi" akan ditampilkan, memungkinkan Anda untuk menavigasi ke halaman yang diinginkan. |
| Keluar dari kursus | | Keluar dari kursus. |

Pendahuluan **Perhatian selama penggunaan**

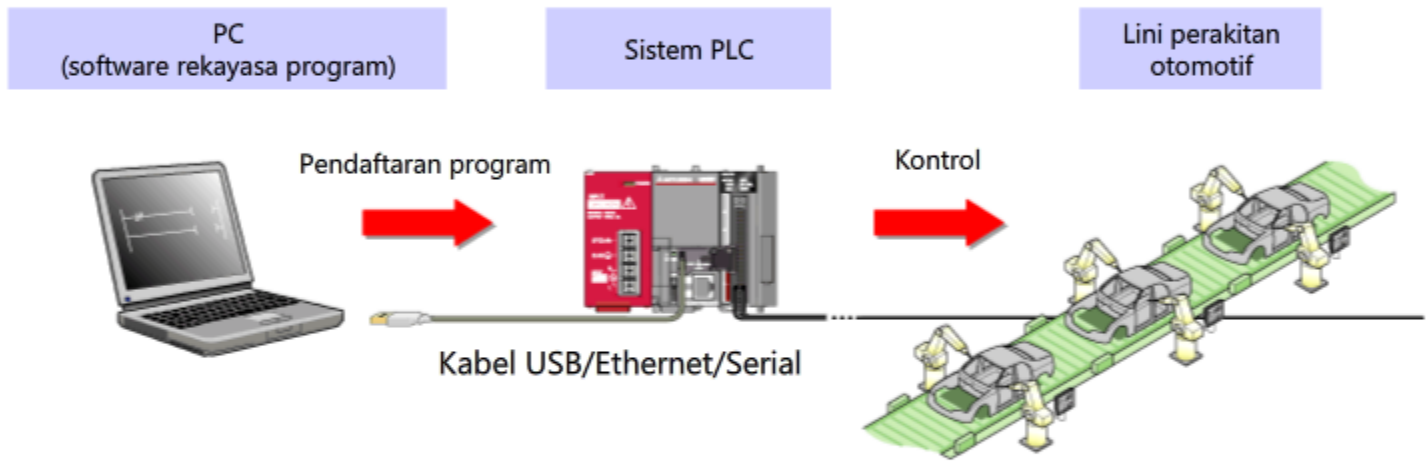
Petunjuk keselamatan

Saat Anda belajar dengan menggunakan produk sebenarnya, bacalah dengan cermat petunjuk keselamatan pada panduan yang sesuai.

Bab 1 Program kontrol

Operasi yang dijalankan oleh pengontrol terprogram ditulis sebagai program kontrol. Program ini didaftarkan ke dalam modul CPU, yang mengendalikan berbagai sinyal input dan output (I/O). Bahasa pemrograman yang digunakan untuk pengontrol terprogram meliputi Ladder, Instruction List (IL), dan Sequential Function Chart (SFC).

Kursus ini menjelaskan poin-poin penting pemrograman ladder termasuk berbagai perintah utamanya.



Dalam kursus ini, software rekayasa pengontrol terprogram, yakni GX Works2 atau GX Works3, digunakan untuk membuat program.

Untuk mempelajari cara menggunakan software rekayasa pengontrol terprogram ini, silakan ikuti kursus "GX Works2 Dasar" atau "Engineering Software MELSOFT GX Works3 (Ladder)" (Software Rekayasa MELSOFT GX Works3 (Ladder)).

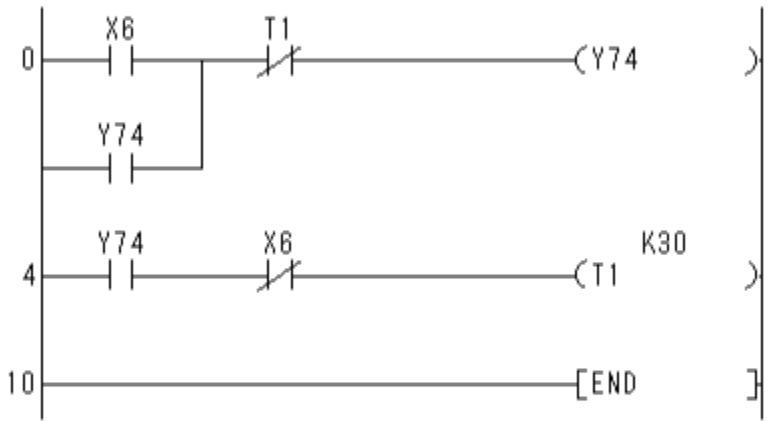
GX Works2 mendukung MELSEC Seri Q/L/F.

GX Works3 mendukung MELSEC Seri iQ-R/iQ-F.

1.1 Bahasa pemrograman

Bahasa pemrograman yang digunakan untuk pengontrol terprogram meliputi Ladder, Instruction List (IL), dan Sequential Function Chart (SFC).

Program ladder adalah diagram logika grafis yang didasarkan pada sirkuit elektrik. Pada program ladder, simbol yang menunjukkan perintah dihubungkan dengan garis, mirip pada diagram sirkuit, danalur operasi dapat dikenali dengan mudah. Selain itu, pemrograman ladder tidak memerlukan pengetahuan pemrograman khusus seperti pada bahasa C dan BASIC, dan dapat dipahami dengan mudah oleh siapa saja yang berpengalaman bekerja dengan relai dan sirkuit elektrik.



Tabel di bawah ini menunjukkan program yang sama pada IL. IL memerlukan sedikit pengetahuan pemrograman untuk menyatakan operasi dalam bentuk perintah.

| No. langkah | Perintah | Device |
|-------------|----------|--------|
| 0 | LD | X6 |
| 1 | OR | Y74 |
| 2 | ANI | T1 |
| 3 | OUT | Y74 |
| 4 | LD | Y74 |
| 5 | ANI | X6 |
| 6 | OUT | T1 K30 |
| 10 | END | |

Program untuk pengontrol terprogram dapat menangani dua tipe nilai.

| | |
|------|---|
| Bit | Bit dinyatakan dalam dua tipe sinyal elektrik, ON dan OFF. Nilai ini bisa juga dinyatakan sebagai "1" (ON) dan "0" (OFF). Nilai bit sering digunakan untuk menunjukkan status device I/O seperti sakelar dan lampu. |
| Word | Angka dan karakter. Nilai angka sering digunakan untuk menunjukkan kuantitas dan waktu. *Kursus ini hanya akan menjelaskan tentang angka. Untuk detail karakter yang digunakan sebagai nilai word, lihat panduan produk yang terkait. |

Format numerik berikut digunakan untuk menunjukkan nilai.

- Desimal
- Biner
- Heksadesimal
- Oktal

1.2.1

Notasi desimal

Dalam notasi desimal, besarnya (kuantitas) sebuah angka dinyatakan dengan format pokok 10 dari "0 hingga 9".

Pada pengontrol terprogram MELSEC, angka desimal didahului dengan huruf "K".
Contoh, "K153" menyatakan angka desimal "153".

1.2.2

Notasi biner

Bila notasi desimal biasanya digunakan untuk menyatakan kuantitas dan waktu, pengontrol terprogram dan pengontrol personal menggunakan data biner, yang merupakan kombinasi dari "0" dan "1".

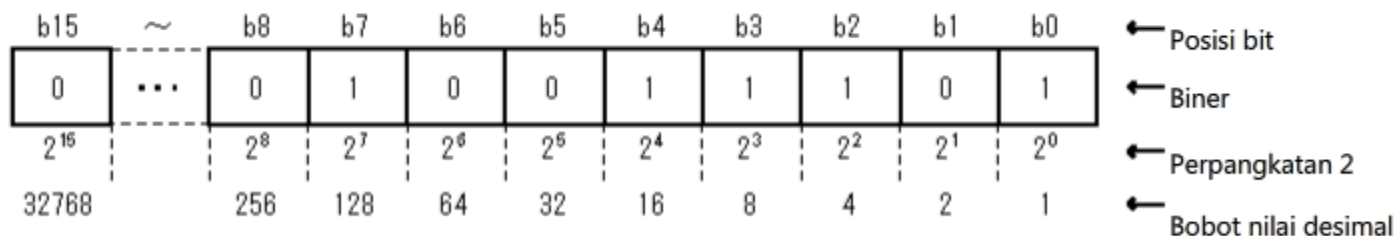
Tabel ini menunjukkan korespondensi antara nilai desimal dan nilai biner, hingga angka desimal "8".

| Desimal | Biner |
|---------|-------|
| 0 | 0 |
| 1 | 1 |
| 2 | 10 |
| 3 | 11 |
| 4 | 100 |
| 5 | 101 |
| 6 | 110 |
| 7 | 111 |
| 8 | 1000 |
| ⋮ | ⋮ |

Setiap kali perintah 1 word digunakan dalam program, perintah tersebut akan disimpan dan diproses sebagai data biner 16 bit oleh pengontrol terprogram sebenarnya. Data biner 16 bit ini adalah istilah lain untuk "1 word".

Contoh, desimal "157" dinyatakan sebagai "0000000010011101" dalam data biner.

Pada notasi desimal, bit ditulis dari kanan. (Bit paling kanan adalah bit awal.)



Untuk mengonversi nilai biner menjadi desimal, kalikan setiap status bit ("0" atau "1") dengan bobot yang terkait, dan jumlahkan semua hasilnya.

$$\begin{aligned}
 &= \underline{1} \times 128 + 0 \times 64 + 0 \times 32 + \underline{1} \times 16 + 1 \times 8 + \underline{1} \times 4 + 0 \times 2 + \underline{1} \times 1 \\
 &= 128 + 16 + 8 + 4 + 1 \\
 &= 157
 \end{aligned}$$

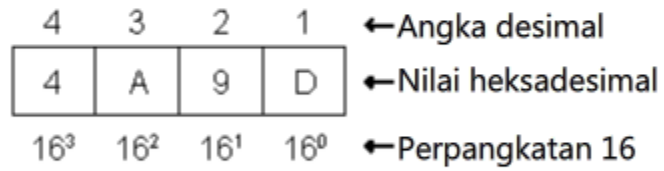
Dari ilustrasi di atas, biner dapat dianggap sebagai format angka yang didasarkan pada bobot.

1.2.3 Notasi heksadesimal

Pada notasi heksadesimal, besarnya (kuantitas) sebuah angka dinyatakan dengan pokok 16 atau ditunjukkan dengan 16 karakter alfanumerik: 0 hingga 9 dan A hingga F. Setiap digit pada notasi heksadesimal naik sebagai 0, 1...9, A...E, lalu F. Apabila nilai tersebut melebihi radiks "F", satu ditambahkan ke kiri dan menjadi "10". Pada pengontrol terprogram MELSEC, angka heksadesimal didahului dengan huruf "H".
 Contoh, "H4A9D" menyatakan angka heksadesimal "4A9D".

Notasi biner bisa panjang dan sulit digunakan pada program dan di tampilan monitor.
 Dalam kasus semacam ini, notasi heksadesimal berguna.
 Satu digit nilai heksadesimal dapat menyatakan 4 bit (4 digit) nilai biner.

Gambar di bawah menunjukkan bagaimana nilai heksadesimal dinyatakan sebagai nilai desimal.



$$= 4 \times 16^3 + A \times 16^2 + 9 \times 16^1 + D \times 16^0$$

(4096) (10) (256) (16) (13) (1)

$$= 19101$$

*Satu digit nilai heksadesimal dapat menyatakan 4 bit biner.

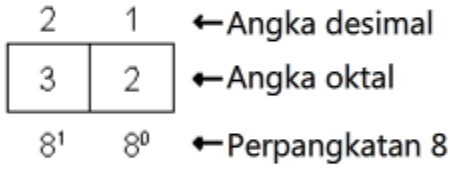
| Desimal | Biner | Heksadesimal |
|---------|---|--------------|
| 0 | 0 | 0 |
| 1 | 1 | 1 |
| 2 | 10 | 2 |
| 3 | 11 | 3 |
| 4 | 100 | 4 |
| 5 | 101 | 5 |
| 6 | 110 | 6 |
| 7 | 111 | 7 |
| 8 | 1000 | 8 |
| 9 | 1001 | 9 |
| 10 | 1010 | A |
| 11 | 1011 | B |
| 12 | 1100 | C |
| 13 | 1101 | D |
| 14 | 1110 | E |
| 15 | 1111 | F |
| ----- | | |
| 16 | 10000 | 10 |
| 17 | 10001 | 11 |
| 18 | 10010 | 12 |
| ⋮ | ⋮ | ⋮ |
| ⋮ | ⋮ | ⋮ |
| ⋮ | ⋮ | ⋮ |
| 19101 | $\overbrace{0100}^4$ $\overbrace{1010}^A$ $\overbrace{1001}^9$ $\overbrace{1101}^D$ | 4 A 9 D |

1.2.4 Notasi oktal

Dalam notasi oktal, besarnya (kuantitas) sebuah angka dinyatakan dengan format pokok 8 ("0 hingga 7"). Apabila nilai naik dari "0", "1", "2" hingga "7", satu ditambahkan ke kiri dan menjadi "10".

Notasi oktal ini digunakan untuk nomor I/O pada MELSEC Seri iQ-F/F.

Gambar di bawah menunjukkan bagaimana nilai oktal dinyatakan sebagai nilai desimal.



$$= 3 \times 8^1 + 2 \times 8^0$$

$$= 26$$

*Satu digit nilai oktal dapat menyatakan 3 bit biner.

| Desimal | Biner | Heksadesimal | Oktal |
|---------|---------|--------------|-------|
| 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 |
| 2 | 10 | 2 | 2 |
| 3 | 11 | 3 | 3 |
| 4 | 100 | 4 | 4 |
| 5 | 101 | 5 | 5 |
| 6 | 110 | 6 | 6 |
| 7 | 111 | 7 | 7 |
| <hr/> | | | |
| 8 | 1000 | 8 | 10 |
| 9 | 1001 | 9 | 11 |
| 10 | 1010 | A | 12 |
| 11 | 1011 | B | 13 |
| 12 | 1100 | C | 14 |
| 16 | 1101 | D | 15 |
| 14 | 1110 | E | 16 |
| 15 | 1111 | F | 17 |
| <hr/> | | | |
| 16 | 1000 | 10 | 20 |
| 17 | 10001 | 11 | 21 |
| 18 | 10010 | 12 | 22 |
| . | . | . | . |
| . | . | . | . |
| . | . | . | . |
| 26 | 011 010 | 1A | 32 |

1.3

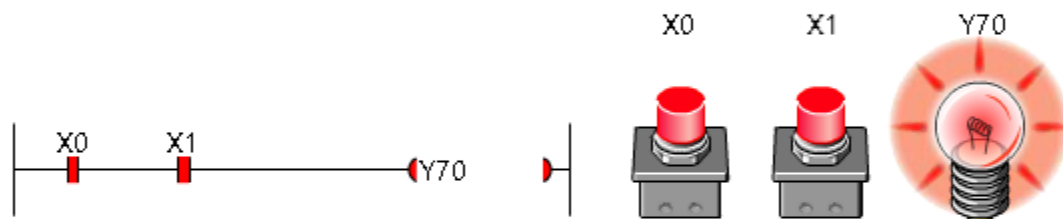
Dasar-dasar pemrograman PLC

Pada kontrol sekuensial, serangkaian operasi dijalankan berdasarkan sinyal ON/OFF yang diterima dari peralatan yang terhubung ke modul input, lalu hasil operasinya dioutput ke peralatan yang terhubung ke modul output.

Untuk menjalankan kontrol semacam itu, program kontrol harus memiliki kondisi input dan output, yang akan dijalankan ketika kondisi input terpenuhi.

Program di bawah ini memerintahkan operasi berikut:

- Ketika sakelar tombol-tekan yang terhubung ke terminal X1 dan X2 keduanya ON, nyalakan terminal Y70
- Hasil operasi ini dioutput ke terminal Y70 untuk menyalakan lampu yang terhubung



Jika sakelar X0 dan X1 ditekan bersamaan, lampu Y70 ON.

1.4 Nomor dan device I/O

Program yang dijelaskan pada Bab 1.3 menggunakan simbol alfanumerik seperti X0, X1, dan Y70 untuk mengidentifikasi peralatan I/O. Karakter alfanumerik ini disebut nomor I/O.

Bab ini menjelaskan tentang device dan nomor I/O, yang diperlukan untuk membuat program kontrol.

MELSEC Seri iQ-R/Q/L/iQ-F dan MELSEC Seri F menggunakan format berbeda untuk menyatakan nomor device.

Tabel di bawah merangkum perbedaannya.

| Seri MELSEC | Bit | | | Word |
|---------------|-----------------|------------------|--------------------|-------------------|
| | X (angka input) | Y (angka output) | M (relai internal) | D (register data) |
| Seri iQ-R/Q/L | Heksadesimal | Heksadesimal | Desimal | Desimal |
| Seri iQ-F/F | Oktal | Oktal | Desimal | Desimal |

1.4.1 Nomor I/O dan sinyal I/O (MELSEC Seri iQ-R/Q/L)

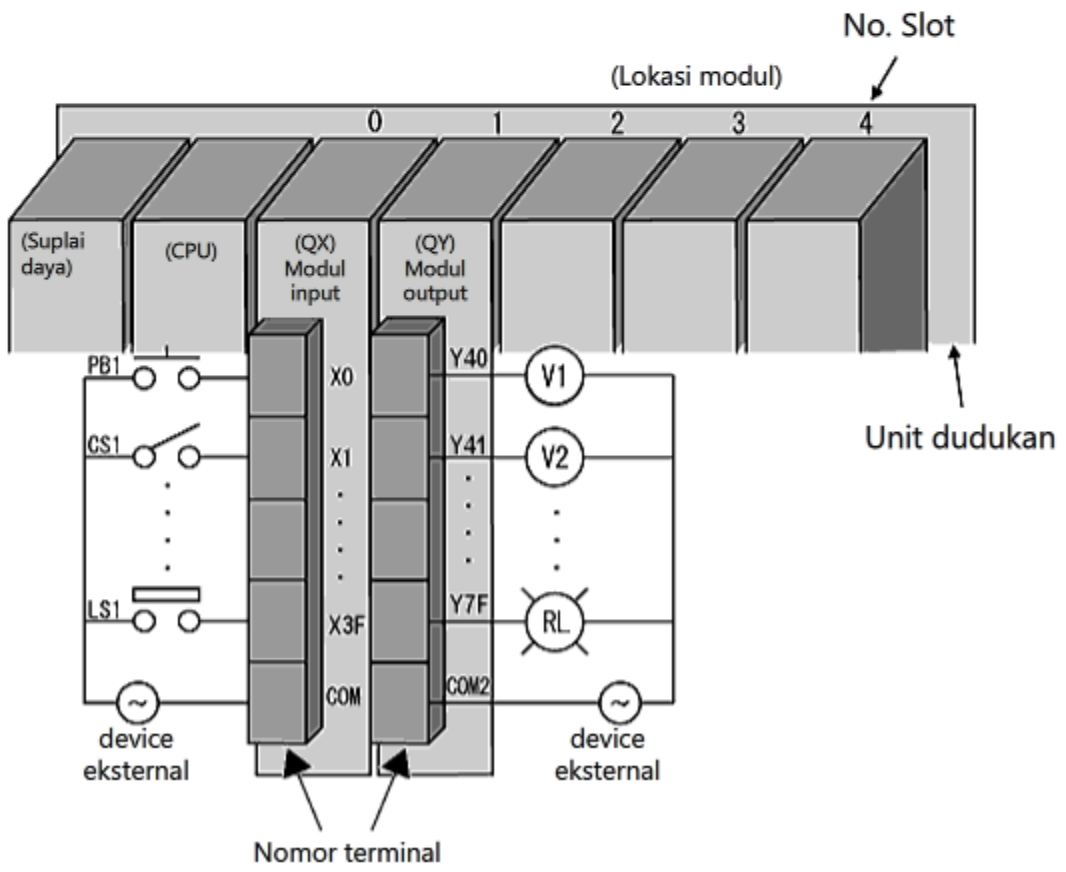
MELSEC Seri iQ-R/Q/L

Nomor I/O terdiri atas sebuah abjad, yang menunjukkan input (X) atau output (Y), dan nilai heksadesimal yang menyatakan nomor terminal.

Nomor I/O mula-mula ditentukan berdasarkan posisi instalasi modul.

Rentang nomor I/O selanjutnya ditentukan berdasarkan jumlah point I/O yang digunakan pada modul tersebut. (Jumlah point I/O yang digunakan sebanding dengan jumlah terminal I/O modul.)

Gambar di bawah menunjukkan bagaimana nomor I/O ditetapkan ke modul input 64 point dan modul output 64 point, yang dipasang berturut-turut pada slot No. 0 dan No.1.



1.4.2 Nomor I/O dan sinyal I/O (MELSEC Seri iQ-F/F)

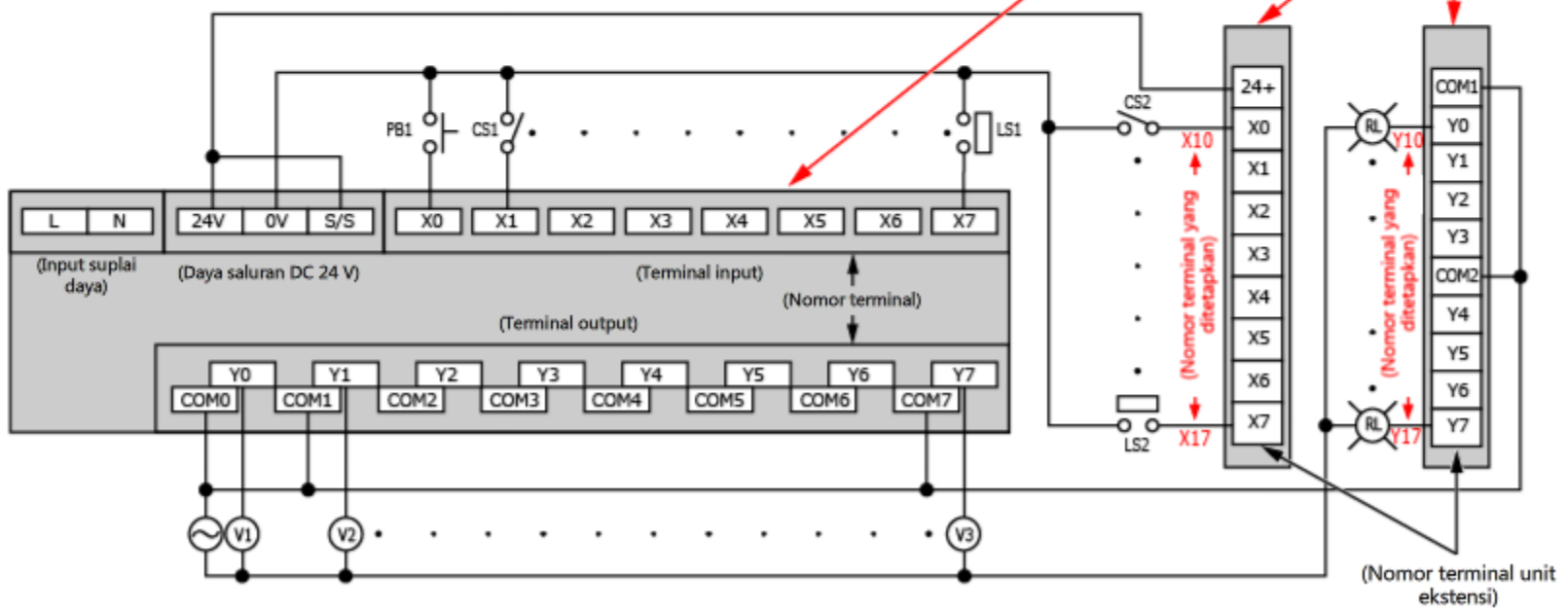
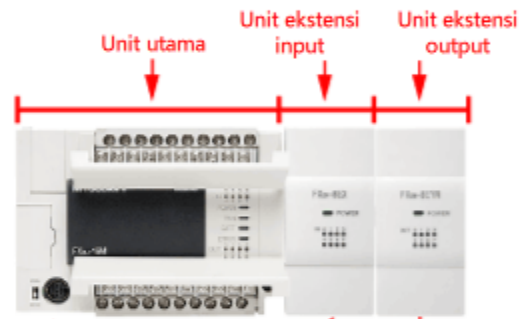
MELSEC Seri iQ-F/F

Nomor I/O terdiri atas sebuah abjad, yang menunjukkan input (X) atau output (Y), dan nilai oktal yang menyatakan nomor terminal.

Nomor I/O mula-mula ditentukan berdasarkan nomor I/O akhir unit utama atau blok ekstensi I/O yang mendahului. Rentang nomor I/O selanjutnya ditentukan berdasarkan jumlah point I/O yang digunakan pada unit tersebut. (Jumlah point I/O yang digunakan sebanding dengan jumlah point I/O yang ada pada unit ekstensi I/O.)

Digit pertama sebuah nomor I/O selalu dimulai dari "0" untuk unit ekstensi baru. Misalnya, jika nomor I/O unit yang mendahului berakhir di X7, maka nomor I/O unit berikutnya dimulai dari X10.

Gambar di bawah menunjukkan bagaimana nomor I/O ditetapkan ke sebuah unit ekstensi input 8 point dan sebuah unit ekstensi output 8 point, yang ditambahkan ke unit utama MELSEC Seri F.



1.4.3 Nomor dan device I/O

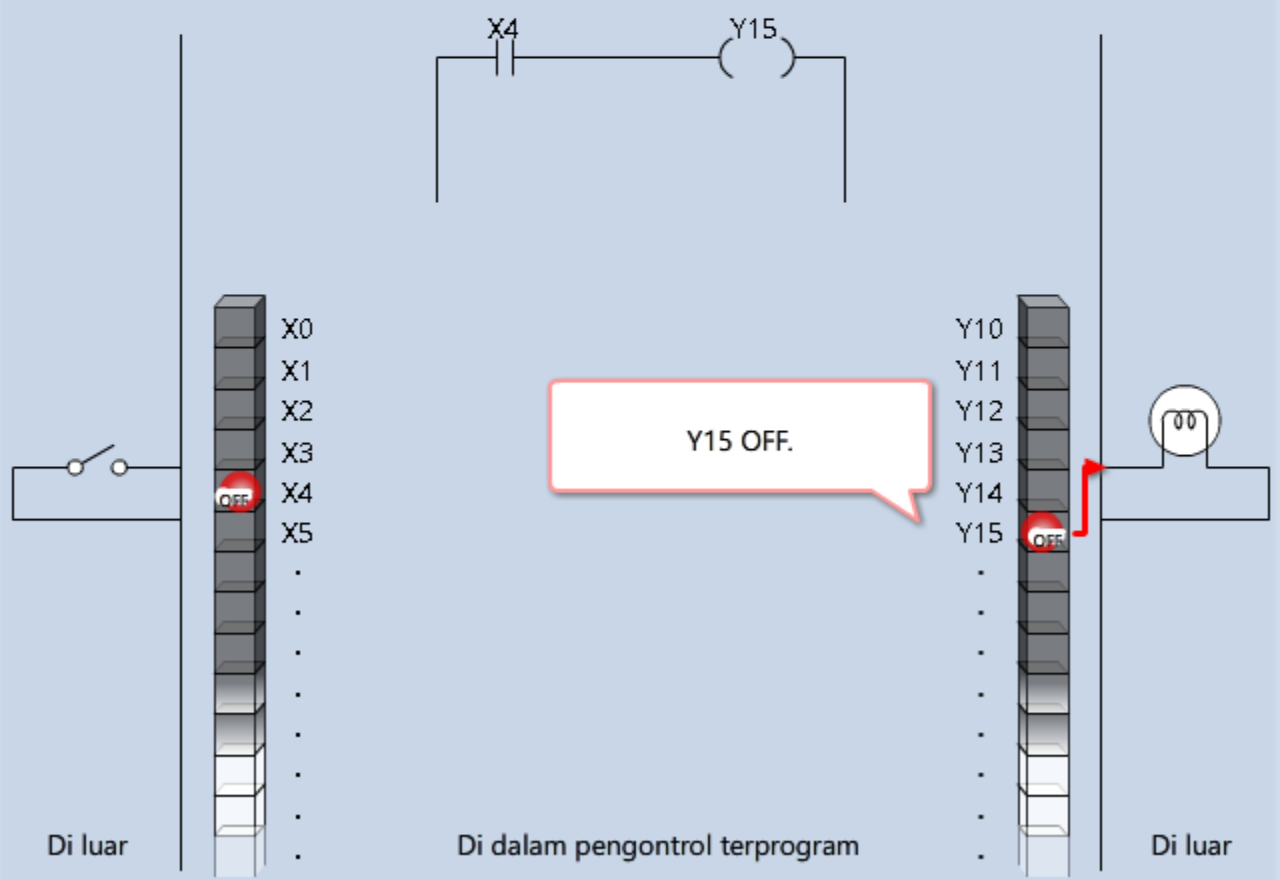
Status peralatan yang terhubung ke unit disimpan di area memori pengontrol terprogram yang disebut "device". Demikian pula untuk output, peralatan output beroperasi sesuai dengan status device. Seperti dijelaskan di atas, program kontrol sering dijalankan berdasarkan status device.

Device yang menyimpan informasi bit (ON/OFF) seperti input (X) dan output (Y) disebut "device bit".

Nomor device ini berhubungan dengan nomor I/O.

Contoh, status terminal yang ditetapkan dengan nomor I/O X0 disimpan di device X0.

Dengan cara yang sama, status device Y10 berhubungan dengan terminal yang ditetapkan dengan nomor I/O Y10.



1.4.4

Relai internal

Kita telah mempelajari bahwa device bit, seperti X (input) dan Y (output), berhubungan dengan nomor yang ditetapkan ke terminal I/O modul fisik.

Ada sekelompok lain device bit yang tidak berhubungan dengan terminal I/O modul, dan salah satunya disebut "relai internal (M)".

Relai internal (M) dinyatakan dalam format desimal, meskipun device input (X) dan output (Y) dinyatakan dalam heksadesimal untuk MELSEC Seri iQ-R/Q/L, dan dalam oktal untuk MELSEC Seri iQ-F/F.

Relai internal (M) digunakan terutama untuk menyimpan data bit sementara.

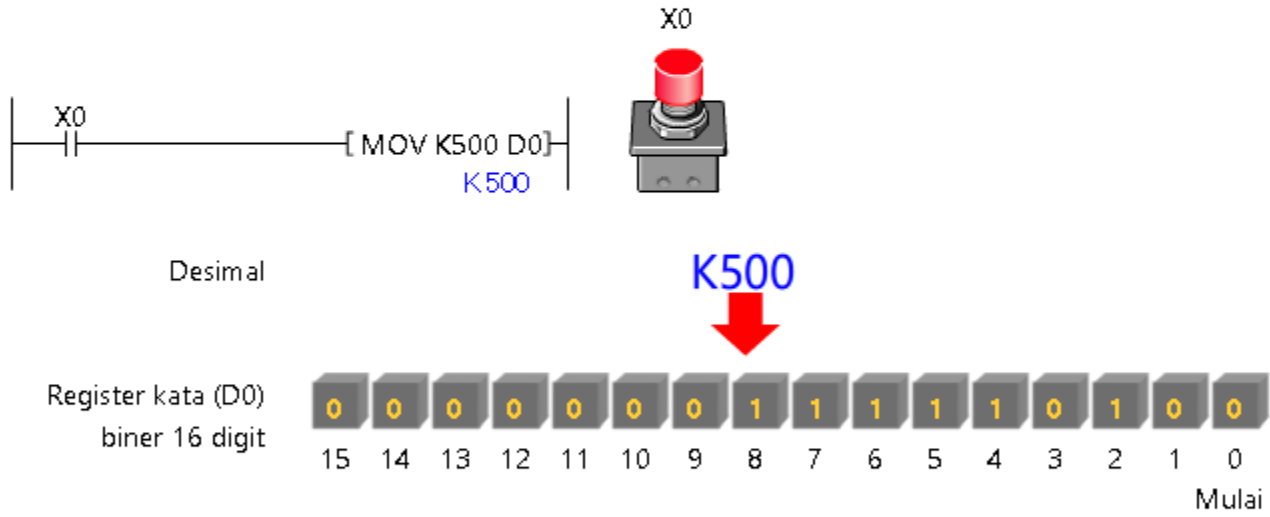
Misalnya, relai internal (M) dapat digunakan untuk menyimpan hasil komputasi sebuah operasi, sehingga hasil itu dapat digunakan pada ladder rung lain.

1.4.5 Device word

Kita telah mempelajari bahwa device yang menyimpan informasi bit (ON/OFF) disebut "device bit" dan yang menyimpan informasi kata disebut "device word".

"Register data" (D) adalah salah satu device word yang lazim digunakan. Satu register data (D) dapat menyimpan data 1 word (16 bit).

Animasi di bawah ini menunjukkan cara menggunakan register data (D). Pada program ini, ketika X0 ON, "K500" (nilai desimal) akan disimpan ke D0. Perintah MOV menyalin nilai ke device yang ditentukan. (Detail selengkapnya akan dijelaskan pada Bab 3.1.) Pada bagian ini, angka disimpan di register data.



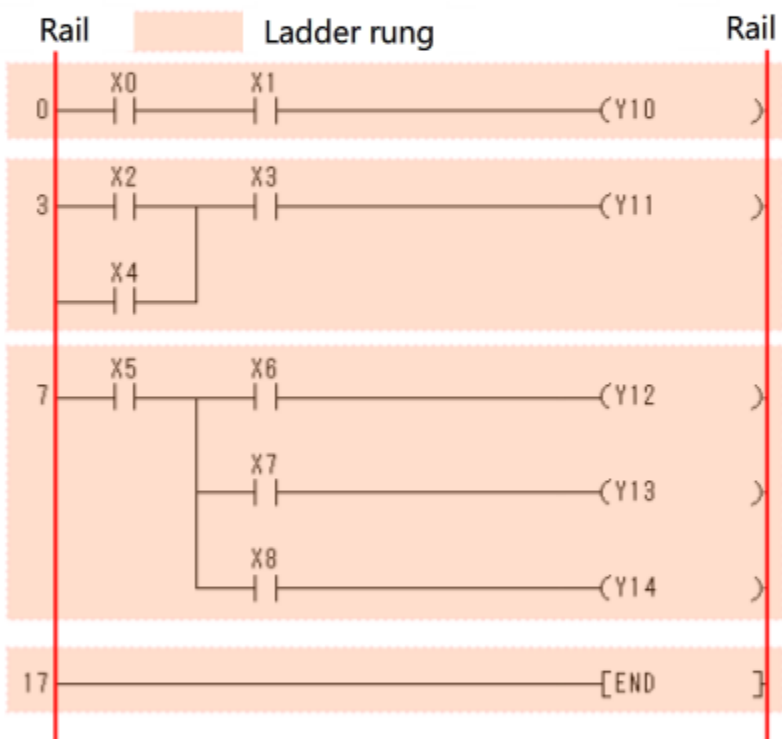
Sekalipun tombol dilepaskan dan X0 mati, "K500" yang disimpan di register data D0 dipertahankan.

1.5 Membuat program kontrol

Program kontrol terdiri atas rail pada tepi kanan dan kiri dan simbol perintah yang dihubungkan dengan garis.

Satu area dari perintah -| |- yang terhubung ke rail kiri hingga perintah -()- atau -[]- yang terhubung ke rail kanan disebut satu ladder rung.

Beberapa ladder rung membentuk sebuah program kontrol, yang berakhir dengan perintah -[END]- atau - [FEND] -.



Perbedaan antara software rekayasa

Perintah -()- dan -[]- antara software rekayasa GX Works2 dan GX Works3 berbeda. Dalam kursus ini, deskripsi menggunakan perintah GX Works2.

| | GX Works2 | GX Works3 |
|-------|----------------|-----------|
| -()- | <Y10 > | Y10 |
| -[]- | {MOV K500 D0 } | |

1.5.1

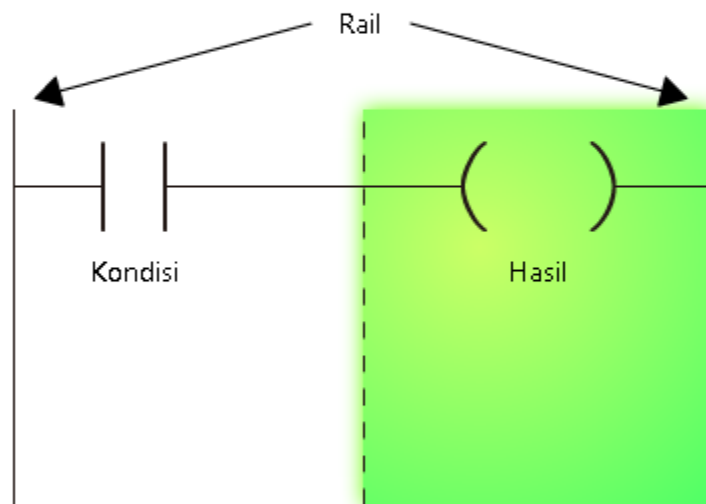
Simbol perintah

Seperti dijelaskan pada Bab 1.3, sebuah pengontrol terprogram harus diperintahkan untuk menjalankan operasi tertentu ketika kondisi input yang ditetapkan terpenuhi. Untuk perintah semacam itu, simbol perintah digunakan untuk menyatakan kondisi input dan detail output.

Simbol perintah biasanya memuat nomor device.

Nomor device menentukan area (device) yang menyimpan status, yang digunakan untuk penilaian kondisi, atau sebagai hasil output.

Ladder rung memuat kondisi dan hasil output. Kondisi ditempatkan di kiri, dan hasil output ditempatkan di kanan. Hasil output dapat berupa sinyal ON/OFF sederhana atau perintah khusus seperti operasi kalkulasi atau penyalinan.



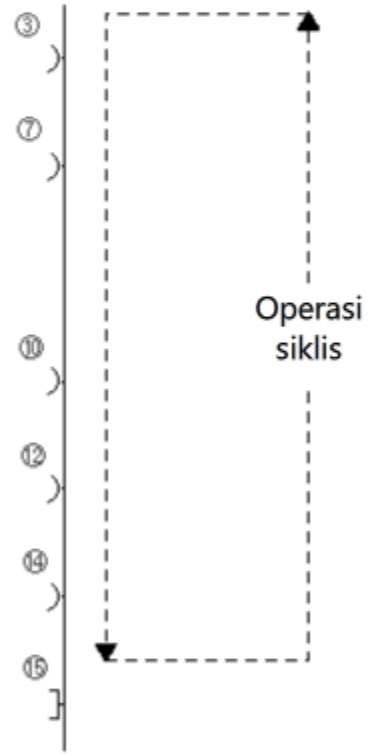
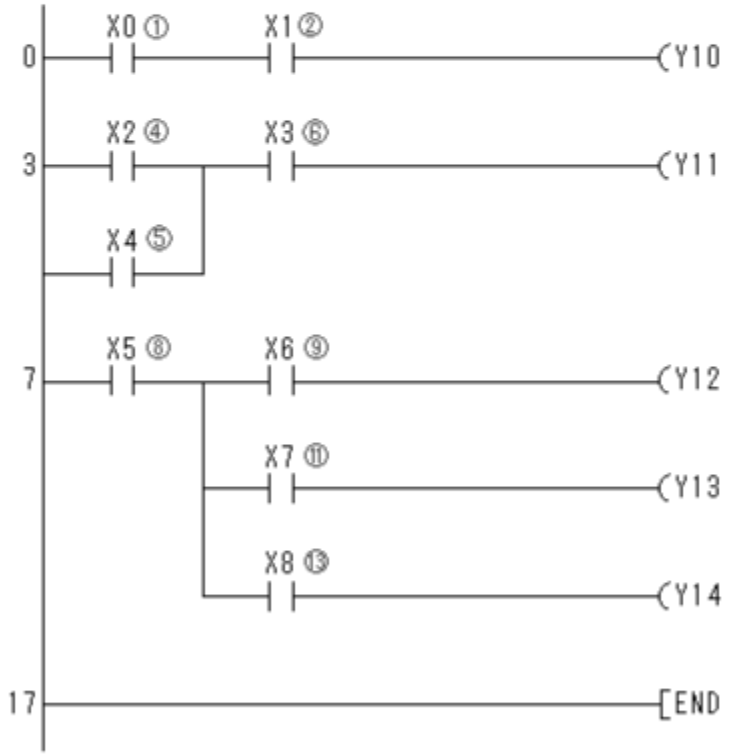
Pada diagram ladder, ada dua rail paralel.
Di sebelah kiri, kondisi ditulis.
Di sebelah kanan, hasil ditulis.

1.6 Prosedur eksekusi program

Sebuah program dimulai dari perintah awal di kiri dan berakhir dengan perintah -[END]-. Setelah mencapai perintah -[END]-, eksekusi program dimulai lagi dari perintah awal. Eksekusi berulang ini disebut "operasi siklis".

Satu siklus operasi siklis disebut satu "pemindaian", dan lamanya waktu yang diperlukan untuk memproses satu pemindaian disebut "waktu pindai".

Gambar di bawah ini menunjukkan prosedur eksekusi perintah. Perintah dijalankan dari kiri ke kanan pada setiap ladder rung, lalu dari ladder rung paling atas ke paling bawah (No. 1, 2,...15 -> 1...).



1.7

Waktu refresh

Seperti dijelaskan sebelumnya, waktu pindai adalah lamanya waktu yang diperlukan untuk menjalankan serangkaian program sebanyak satu kali. Waktu pindai juga dapat dinyatakan sebagai:

$$\text{Waktu pindai} = \text{waktu refresh} + \text{waktu eksekusi program} + \text{waktu pemrosesan END}$$

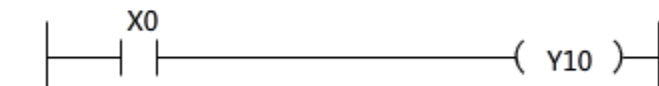
Waktu refresh adalah lamanya waktu yang diperlukan untuk membaca data dari modul input ke dalam device input (X) plus waktu yang diperlukan untuk menulis data dari device output (Y) ke modul output.



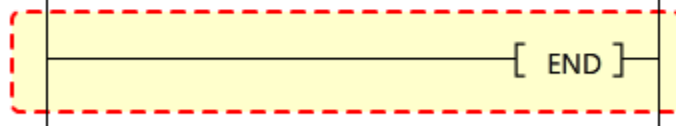
(1) Refresh I/O

- Mengirimkan status ON/OFF perangkat output ke peralatan output yang terhubung
- Menyimpan status ON/OFF yang diterima dari peralatan input yang terhubung ke dalam perangkat input

(2) Eksekusi program



(3) Pemrosesan END



Pemrosesan END oleh perintah END
(Detail dihilangkan di sini.)

Catat bahwa status ON/OFF sakelar yang sebenarnya akan dibaca satu kali dan disimpan ke device input (X), dan data baru menimpa data yang ada selama refresh.

Begitu juga data di device output (Y) akan ditulis satu kali ke modul output ketika sebuah perintah dijalankan.

Itu berarti bahwa jika sebuah sinyal berubah dari OFF ke ON, lalu ke OFF lagi, maka sinyal tersebut tidak akan dikenali sebagai ON. Namun, waktu pindai sangat singkat dibandingkan panjang sinyal. Akibatnya, pengontrol terprogram sangat jarang melewati perubahan status sinyal.

Bab 2 Perintah device bit

Bab ini menjelaskan tentang perintah yang menggunakan device bit (ON/OFF).

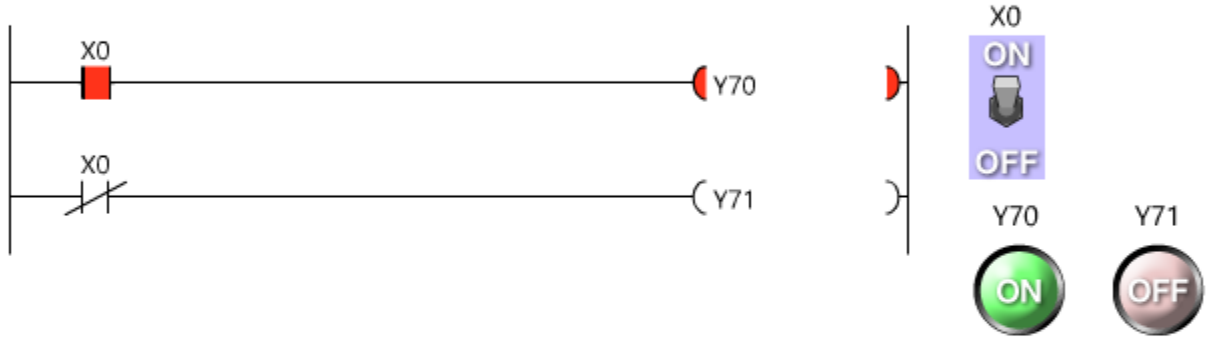
Operasi yang menggunakan device bit adalah operasi paling dasar dalam program kontrol. Input dari peralatan input digunakan sebagai kondisi untuk mengontrol peralatan output.

2.1 Kondisi input dan output

Kontak normal-terbuka (NO) dan normal-tertutup (NC) digunakan sebagai kondisi input.
 Ketika kondisi input terpenuhi, perintah output kumparan (perintah OUT) dioutput.
 Ketika kondisi input tidak terpenuhi, perintah output kumparan tidak dioutput.
 Perintah kontak NO/NC dan perintah OUT adalah kombinasi perintah utama yang digunakan pada program kontrol.

Program ladder dan operasi

Simulasikan pengoperasian perintah NO, NC, dan OUT dengan mengklik sakelar input di sebelah kanan.



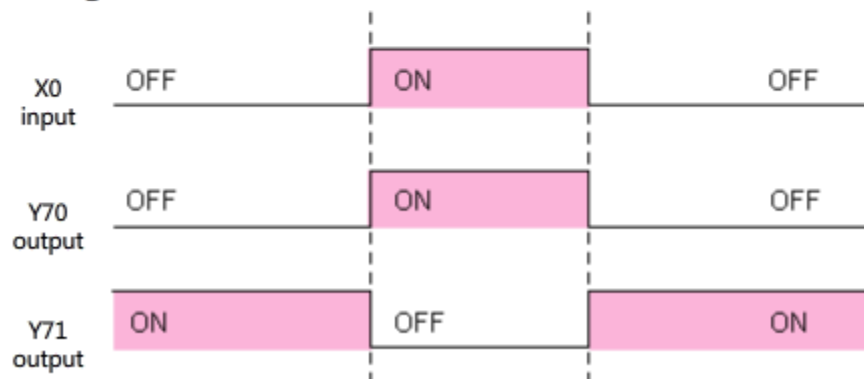
2.1

Kondisi input dan output

■ Kode perintah dan fungsi

| Simbol | Fungsi |
|--------|---|
| | Kontak NO Dijalankan ketika status device ON. |
| | Kontak NC Dijalankan ketika status device OFF (kebalikan dari kontak NO). |
| | Output kumparan (OUT) Ketika kondisi input yang mendahului terpenuhi, data pada device preset akan dioutput. |
| | Perintah End (END) Menunjukkan akhir program. Sebuah program memerlukan sebuah perintah END. |

■ Diagram waktu



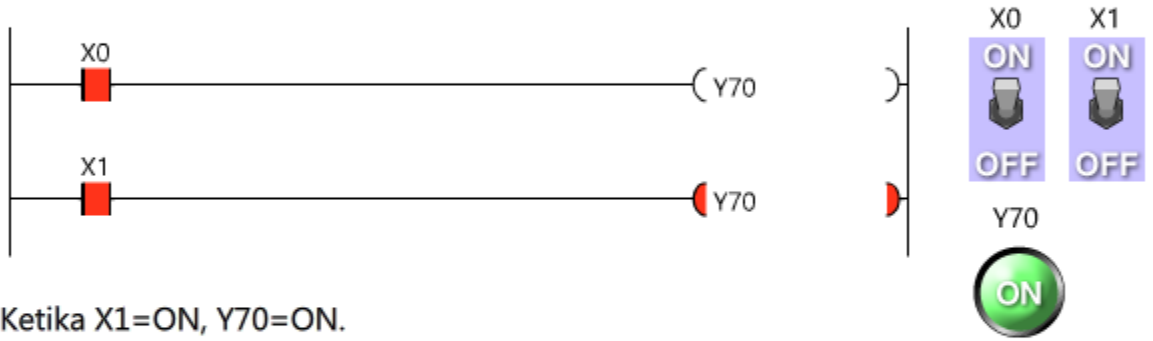
2.1.1 Menggunakan nomor device yang sama untuk beberapa perintah

Pada satu ladder rung, satu perintah OUT hanya boleh digunakan dengan satu nomor device. Jika beberapa perintah OUT digunakan dengan nomor device yang sama, maka hanya perintah OUT terakhir yang valid, sedangkan perintah OUT pertama tidak valid.

Program ladder

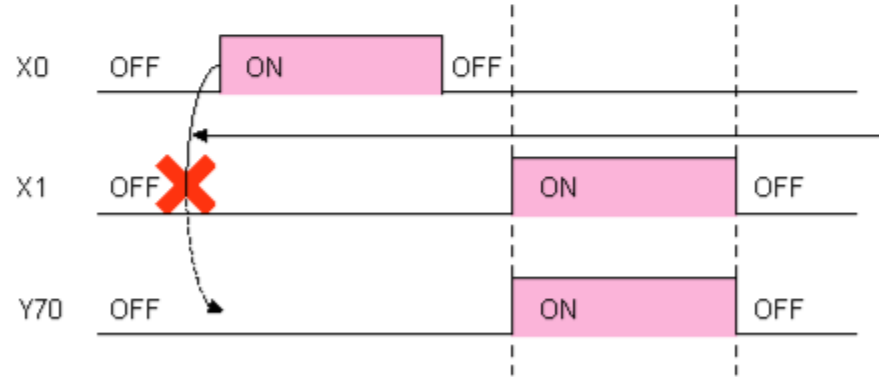
Simulasikan pengoperasian dua perintah dengan nomor device yang sama dengan mengeklik sakelar input di sebelah kanan.

Tipe penggunaan ini (menggunakan OUT Y70 untuk dua perintah) disebut "kumparan rangkap".



Ketika X1=ON, Y70=ON.

Diagram waktu

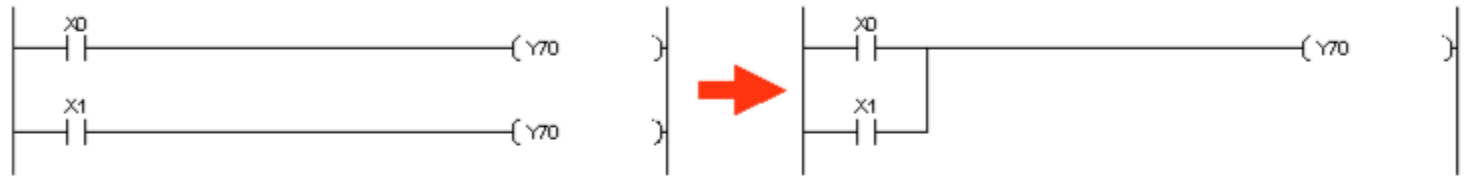


Kondisi input pertama X0 diabadikan karena kondisi input yang lebih akhir diprioritaskan.

2.1.1 Menggunakan nomor device yang sama untuk beberapa perintah

Contoh koreksi

Dalam contoh ini, kondisi input "X1" memiliki prioritas lebih tinggi, dan "X0" diabaikan.
 Dengan mengoreksi ladder rung ke kondisi seperti pada Gambar B, device Y70 ON ketika salah satu dari kedua kondisi input terpenuhi, sehingga menghindari konflik antara kedua perintah OUT.



(Gambar A)

(Gambar B)

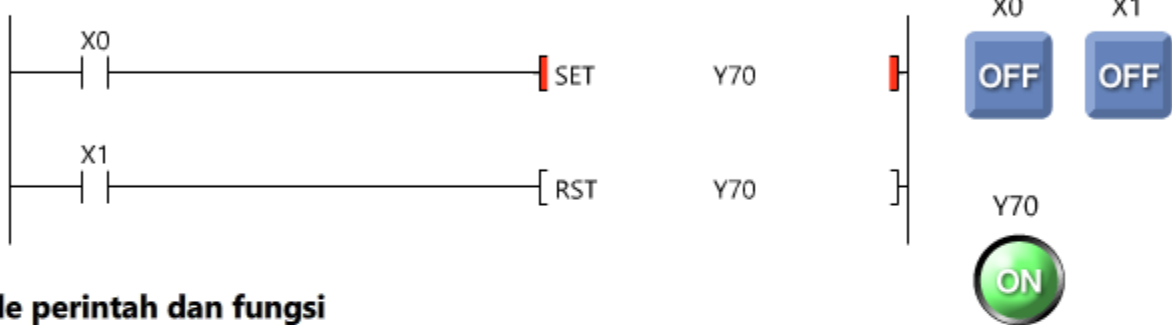
2.2 Mempertahankan/membatalkan output

Tidak seperti perintah OUT, perintah retensi operasi (perintah SET) mempertahankan status output sekalipun kondisi input terpenuhi.

Untuk membatalkan output (OFF), perintah pembatalan retensi operasi (perintah RST) dapat dijalankan.

Program ladder dan operasi

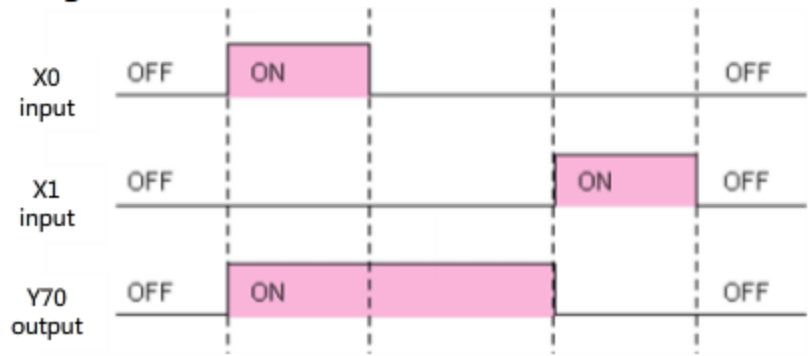
Simulasikan pengoperasian perintah SET dan RST dengan mengklik sakelar input di sebelahan kanan.



Kode perintah dan fungsi

| Simbol | Fungsi |
|--------|---|
| | Perintah retensi operasi (SET) Perintah ini menyalakan device, dan mempertahankan status ON (output). Output dipertahankan sekalipun kondisi input terpenuhi. |
| | Perintah pembatalan retensi operasi (RSET) Membatalkan status ON, dan membatalkan output ke device yang ditentukan. |

Diagram waktu



2.2.1 Perbedaan antara perintah OUT dan SET

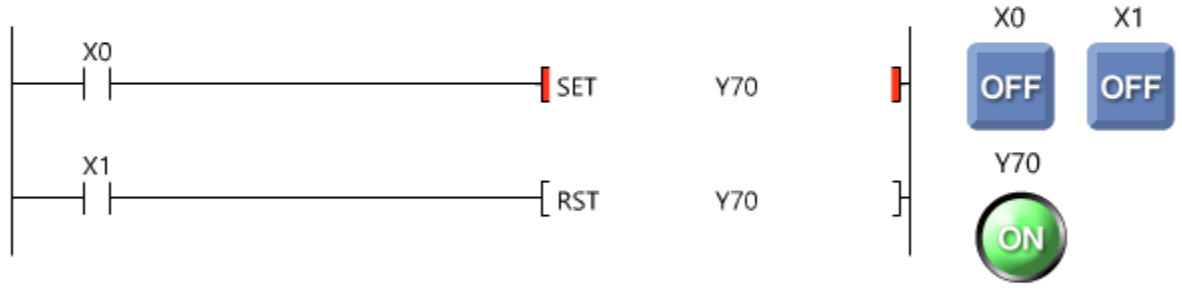
Simulasikan perbedaan operasi antara perintah OUT dan SET dengan mengklik sakelar input di sebelah kanan.

Perintah OUT



Y70 ON ketika kondisi input terpenuhi.

Perintah SET/RST



Setelah kondisi input terpenuhi, Y70 ON hingga perintah RST dijalankan.

2.2.2 Mengganti ladder retensi dengan perintah SET

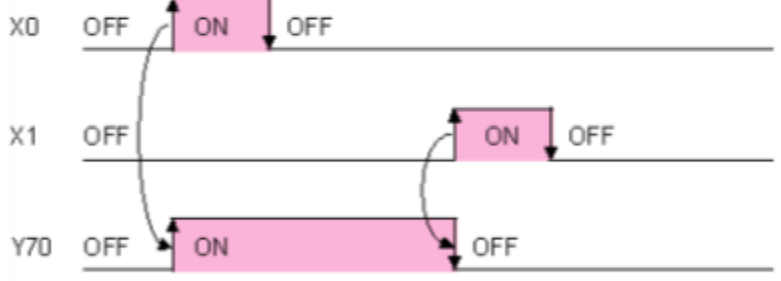
Program ladder dan operasi

Simulasikan pengoperasian ladder retensi dengan mengklik sakelar input di sebelah kanan.



Ketika X0=ON dan X1=OFF, Y70=ON.
Y70 = ON (retensi) hingga X1=ON.

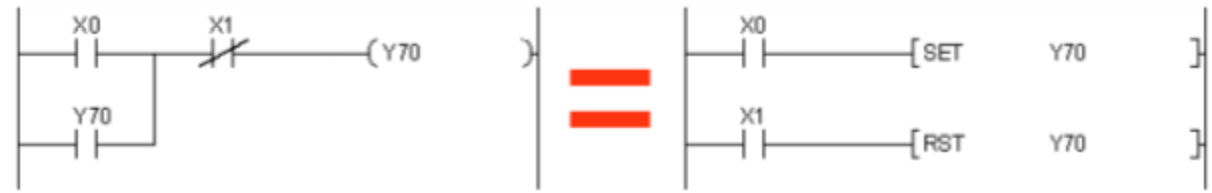
Diagram waktu



Bahkan setelah X10 OFF, Y70 (kumparan) tetap ON (retensi)

Mengganti dengan perintah SET

Program ladder retensi dapat ditulis ulang sebagai program ladder dengan perintah SET. Dengan perintah SET, program ladder dapat disederhanakan.



2.3 Menambahkan kondisi (logika AND)

Untuk menggunakan logika AND, kontak NO/NC dipasang secara seri.
Pada logika AND, kondisi terpenuhi ketika beberapa kontak NO/NC, yang terhubung secara seri, ON.

Program ladder dan operasi

Simulasikan pengoperasian logika AND dengan mengklik sakelar input di sebelah kanan.



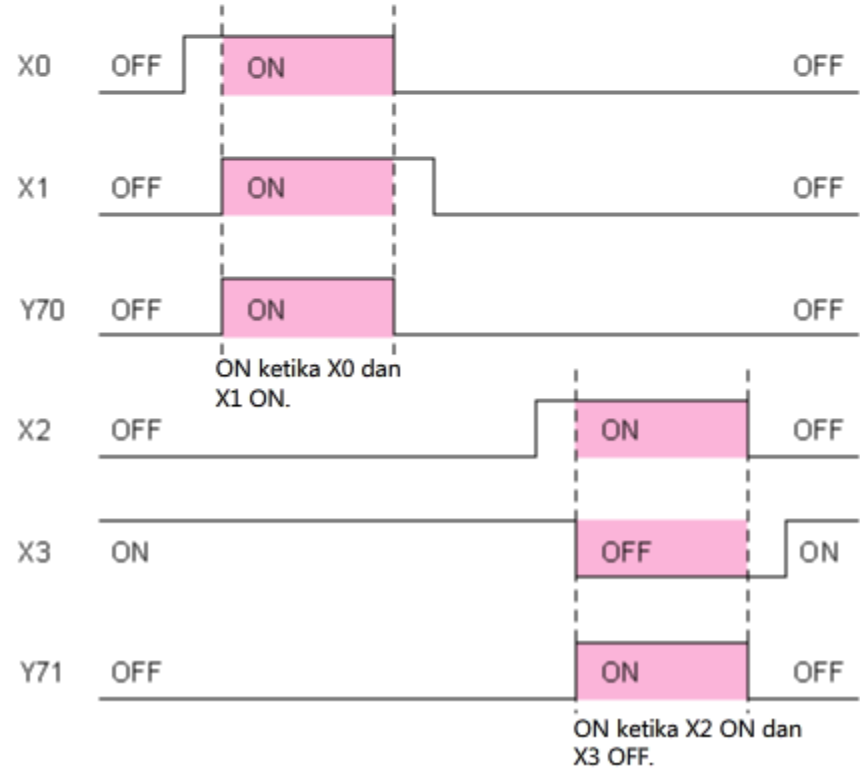
Ketika X0 dan X1 ON, Y70 ON.
Ketika X2 ON dan X3 OFF, Y71 ON.

2.3 Menambahkan kondisi (logika AND)

Kode perintah dan fungsi

| Simbol | Fungsi |
|--------|---|
| | Koneksi serial kontak NO Kontak NO terhubung secara seri (horizontal). |
| | Koneksi serial kontak NC Kontak NC terhubung secara seri (horizontal). |

Diagram waktu



2.4 Menambahkan kondisi (logika OR)

Untuk menggunakan logika OR, kontak NO/NC dipasang secara paralel. Pada logika OR, kondisi terpenuhi ketika salah satu dari kontak NO/NC, yang terhubung secara paralel, ON.

Program ladder dan operasi

Simulasikan pengoperasian logika OR dengan mengklik sakelar input di sebelah kanan.



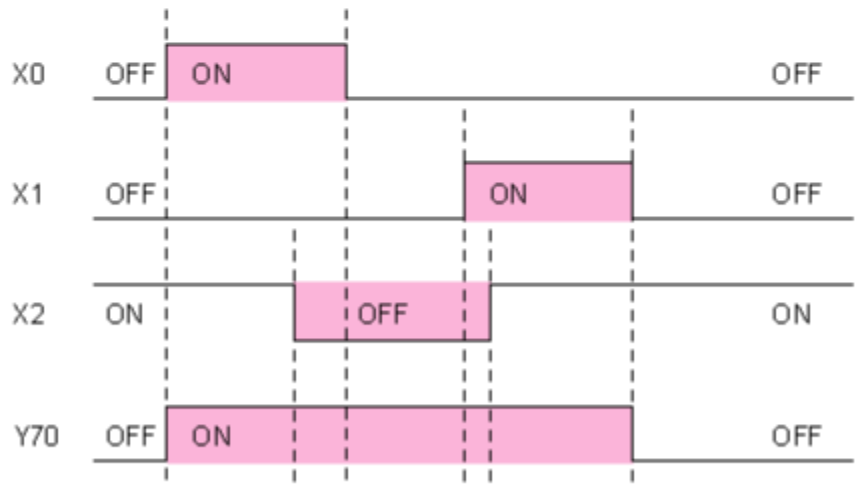
Y70 ON ketika salah satu dari kondisi berikut terpenuhi: X0 ON, X1 ON, atau X2 OFF.

2.4 Menambahkan kondisi (logika OR)

■ Kode perintah dan fungsi

| Simbol | Fungsi |
|--------|---|
| | <p>Koneksi paralel kontak NO Kontak NO terhubung secara paralel (vertikal).</p> |
| | <p>Koneksi paralel kontak NC Kontak NC terhubung secara paralel (vertikal).</p> |

■ Diagram waktu



2.5 Output sebagai pulsa

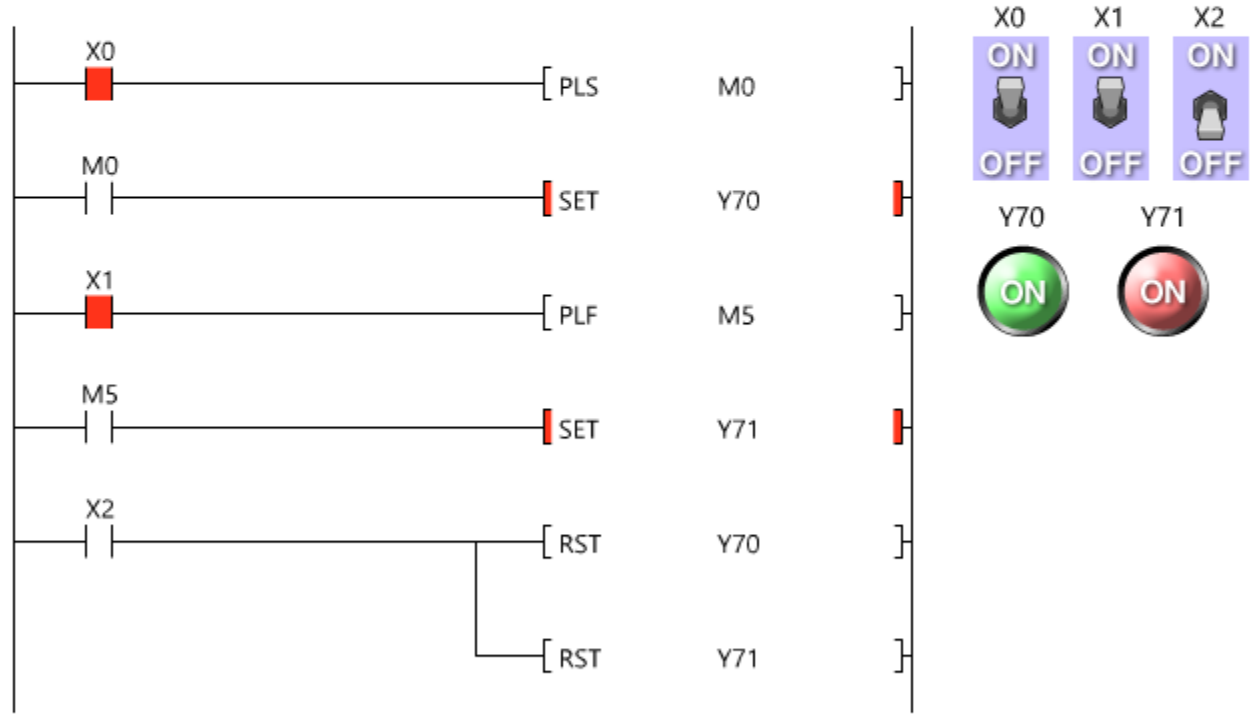
Tidak seperti perintah OUT, perintah tepi naik (perintah PLS) membuat kumparan ON selama satu pemindaian setelah kondisi input terpenuhi.

Berkebalikan dengan perintah PLS, perintah tepi turun (perintah PLF) membuat kumparan ON selama satu pemindaian setelah kondisi input tidak terpenuhi.

Kumparan yang ON karena perintah PLS/PLF kembali OFF setelah satu pemindaian.

Program ladder dan operasi

Simulasikan pengoperasian perintah PLS dan PLF dengan mengklik sakelar input di sebelah kanan.



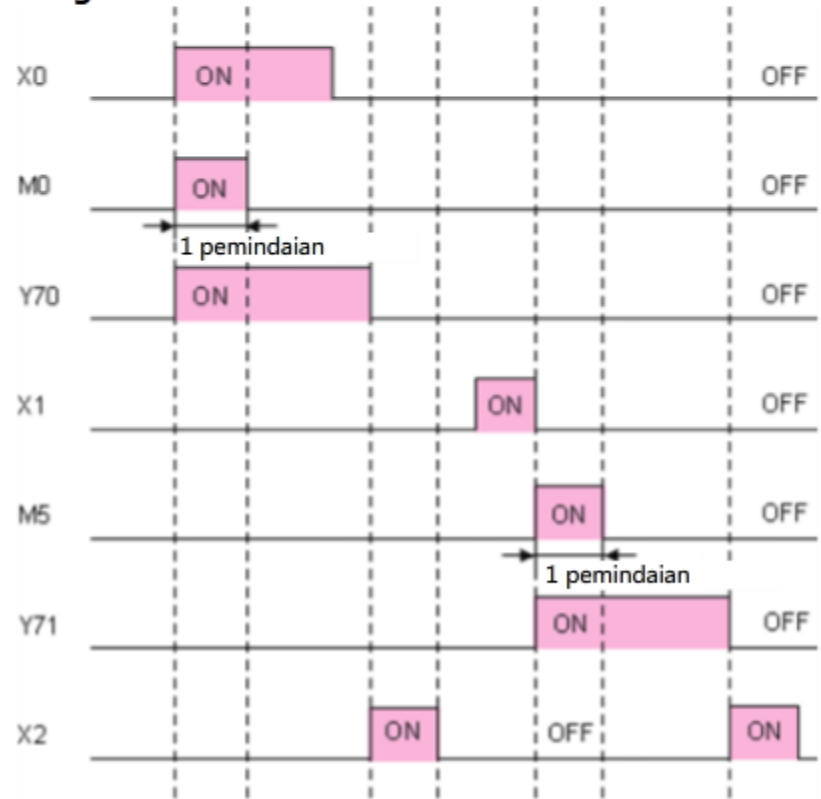
Pada tepi naik X0 (OFF ke ON), M0 ON selama 1 pemindaian
Pada tepi turun X1 (ON ke OFF), M5 ON selama 1 pemindaian

2.5 Output sebagai pulsa

Kode perintah dan fungsi

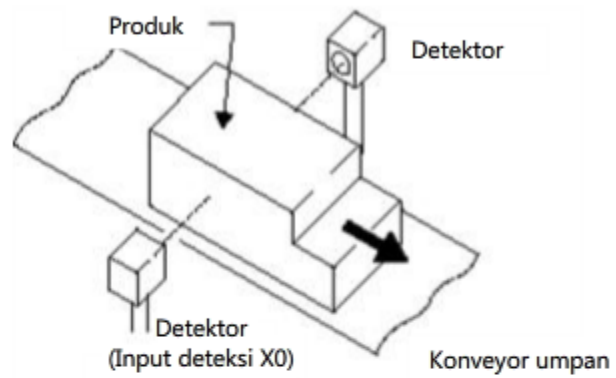
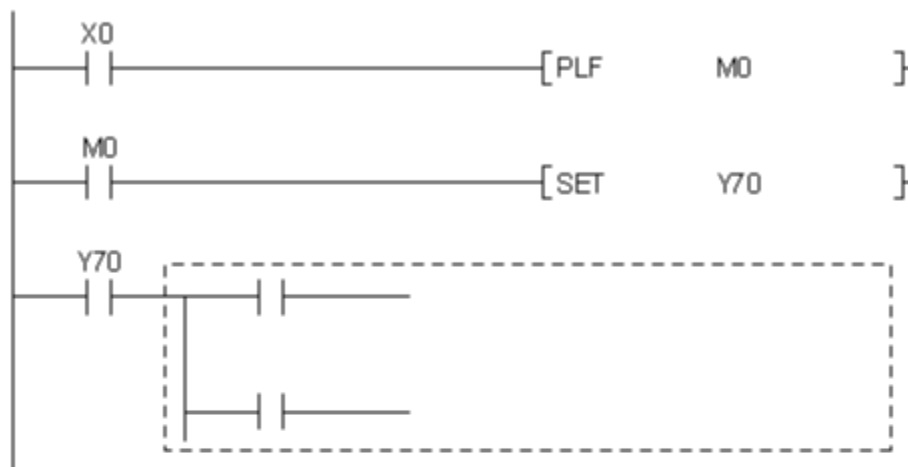
| Simbol | Fungsi |
|--------|---|
| | Output pada tepi naik (PLS) Data dioutput ke device yang ditentukan pada pemindaian ke-1 setelah kondisi input terpenuhi. |
| | Output pada tepi turun (PLF) Data dioutput ke device yang ditentukan pada pemindaian ke-1 setelah kondisi input tidak terpenuhi. |

Diagram waktu



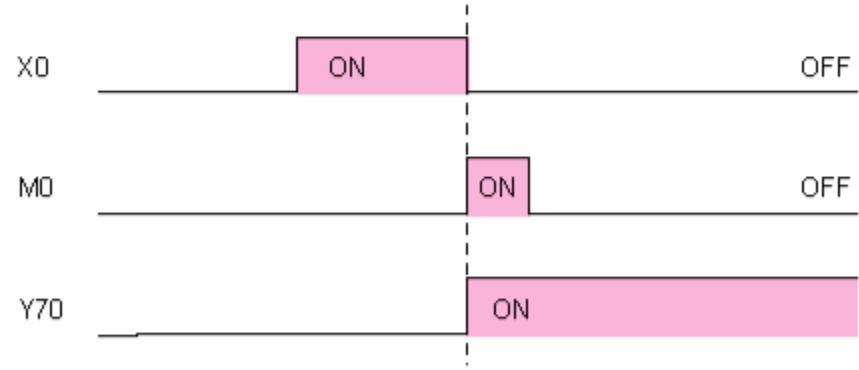
2.5.1 Contoh aplikasi output pulsa

Program ladder



Output pulsa digunakan untuk mendeteksi objek bergerak yang lewat. Ketika sebuah produk yang lewat terdeteksi, proses berikutnya dimulai.

Diagram waktu



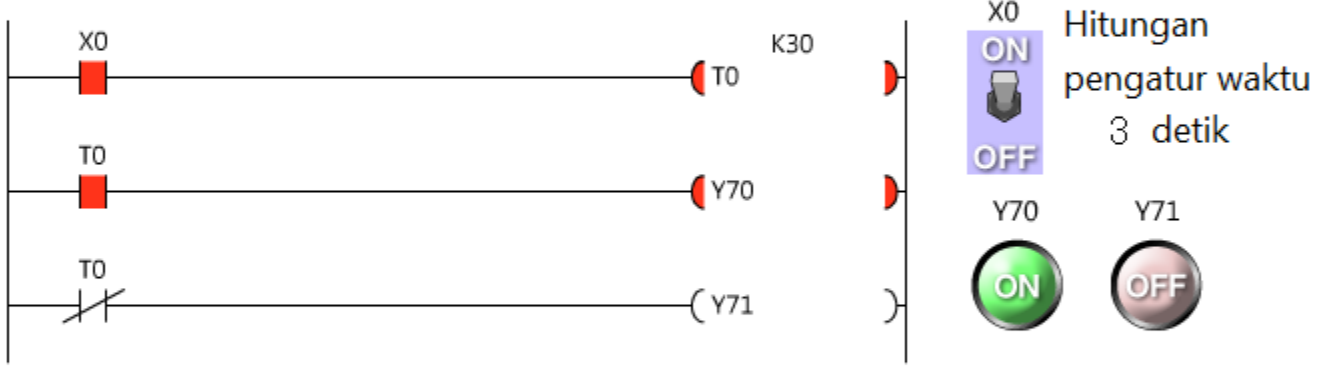
2.6 Pengukuran waktu

Perintah OUT dan device pengatur waktu (T) digunakan untuk pengukuran waktu. Ketika kondisi input terpenuhi (ON), pengukuran waktu dimulai. Ketika periode waktu mencapai nilai yang ditentukan, device pengatur waktu (T) ON. Jika kondisi input tidak terpenuhi (OFF) atau device pengatur waktu (T) direset dengan perintah RST, waktu yang berlalu dan output diinisiasi.

Status device pengatur waktu (T) dapat digunakan sebagai kondisi input pada bagian lain program.

Program ladder dan operasi

Simulasikan pengoperasian pengatur waktu dengan mengklik sakelar input di sebelah kanan.



X0 ON, kemudian setelah 3 detik, Y70 ON dan Y71 OFF.

2.6 Pengukuran waktu

Kode perintah dan fungsi

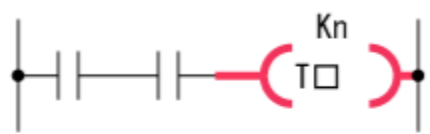
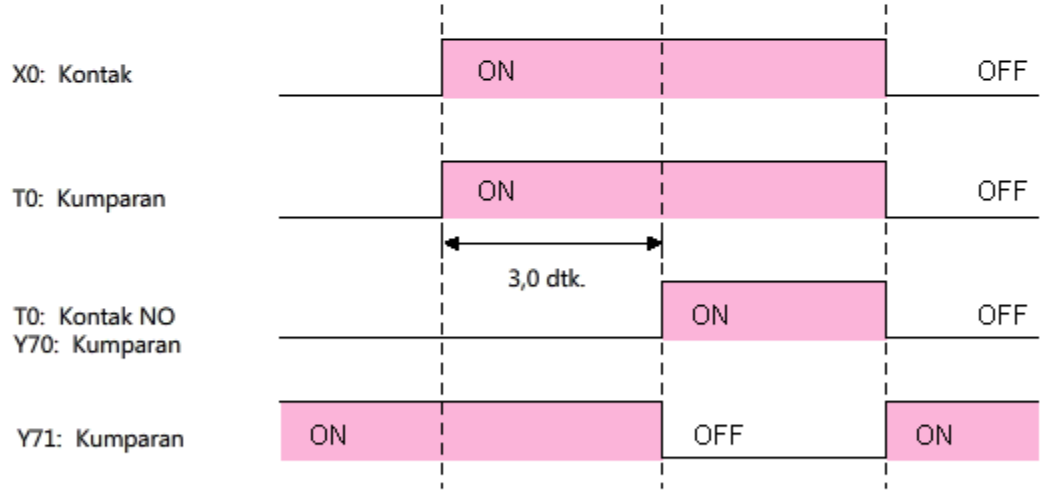
| Simbol | Fungsi |
|---|---|
|  <p>□ : Nomor device</p> | <p>Operasi pengatur waktu</p> <p>Device pengatur waktu (T) digunakan dengan output kumparan (OUT) untuk mengukur berapa lama kondisi tersebut terpenuhi (menjadi ON).</p> <p>Waktu-habis terjadi setelah periode waktu yang ditentukan.</p> <p>Bersamaan dengan waktu-habis, pengatur waktu (T0) ON.</p> <p>Nilai yang ditetapkan pengatur waktu ditunjukkan dengan "Kn" (n: desimal).</p> <p>Pengatur waktu sering digunakan sebagai penghitung waktu tunda-aktif, yang menentukan waktu setelah kondisi tertentu terpenuhi.</p> |

Diagram waktu

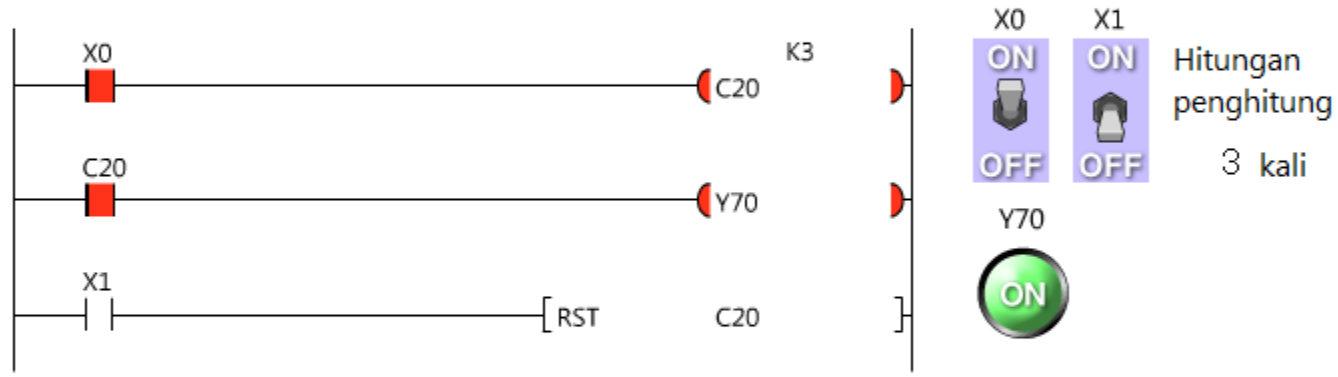


2.7 Penghitungan

Perintah OUT dan device penghitung (C) digunakan untuk penghitungan.
 Ketika kondisi input terpenuhi, hitungan bertambah, dan ketika hitungan mencapai nilai yang ditentukan, device penghitung yang ditentukan (C) ON.
 Jika device penghitung (C) direset dengan perintah RST, hitungan dan status device diinisiasi.
 Status device penghitung (C) dapat digunakan sebagai kondisi input pada bagian lain program.

Program ladder dan operasi

Simulasikan pengoperasian penghitung dengan mengklik sakelar input di sebelah kanan.



Nilai di C20 bertambah setiap kali X0 ON. Ketika hitungan mencapai 3 (hitungan-selesai), Y70 ON.

2.7 Penghitungan

Kode perintah dan fungsi


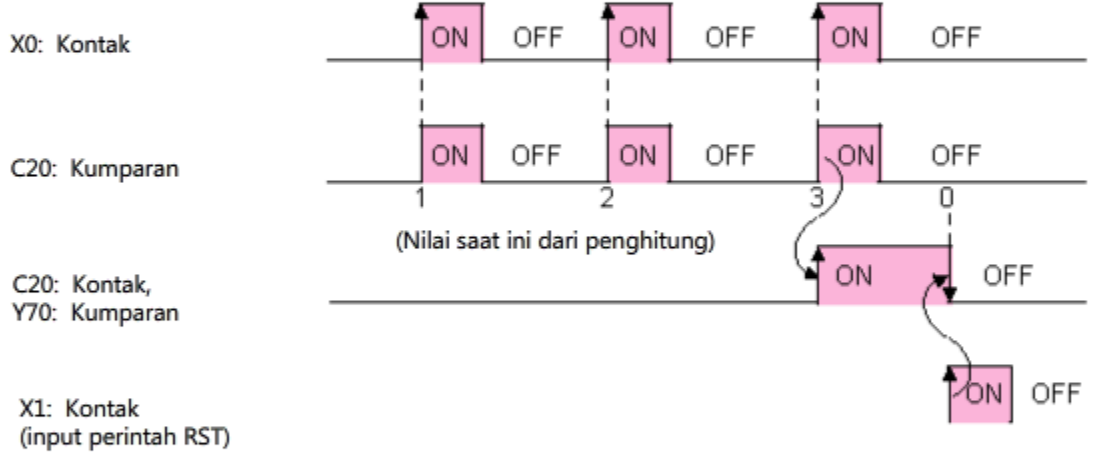
| Simbol | Fungsi |
|---|---|
|  <p>Kn C □</p> <p>□ : Nomor device</p> | <p>Penghitung</p> <p>Bersama dengan output kumparan (OUT), penghitung menghitung naik (satu demi satu) berapa kali kondisi terpenuhi.</p> <p>Hitungan-selesai terjadi ketika hitungan mencapai angka yang ditentukan, dan kontak penghitung ON.</p> <p>Nilai yang ditetapkan penghitung ditunjukkan dengan "Kn" (n: desimal).</p> |

Diagram waktu



Bab 3 Perintah device word

Bab ini menjelaskan tentang perintah yang menggunakan device word.

Device word berguna untuk mengontrol waktu, hitungan, dan nilai yang diinput dari peralatan eksternal.

Device word dapat menjadikan program kontrol lebih responsif terhadap operasi sebenarnya.

- Simulasikan pengoperasian program dasar untuk memahami operasi berbagai perintah utama
- Dari simulasi, pahami kegunaan berbagai perintah dan pemrosesan yang dijalankan di pengontrol terprogram

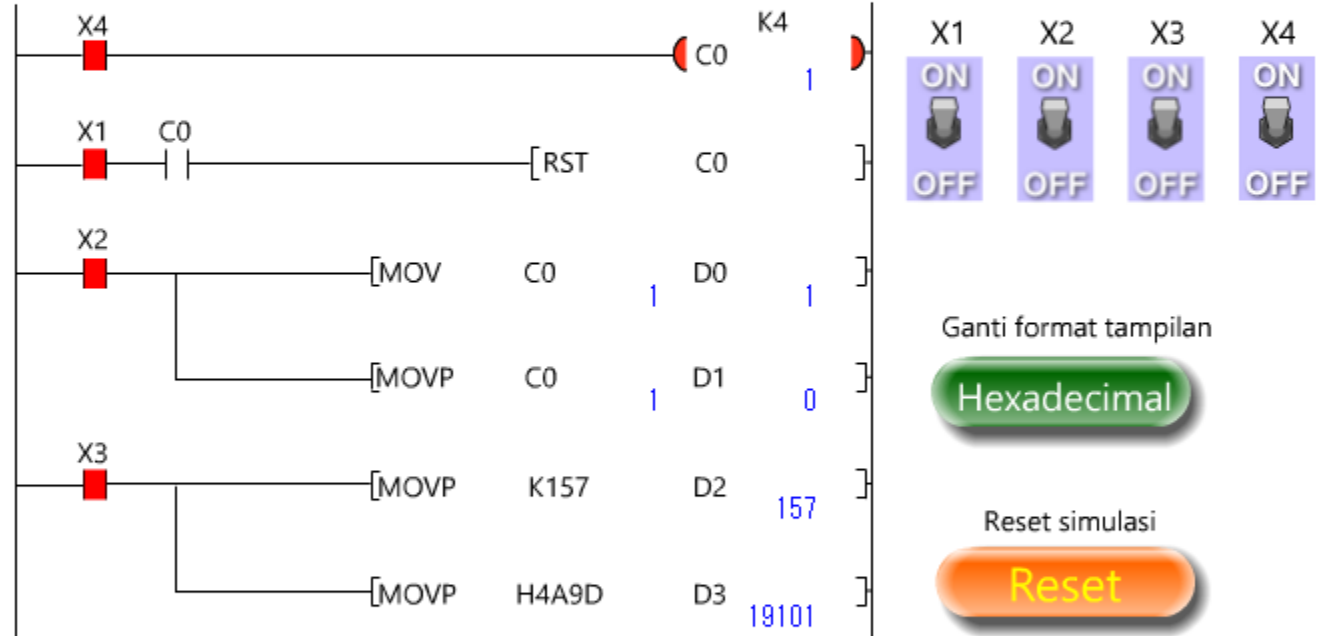
3.1 Memindahkan data ke device word

Perintah transfer data 16 bit (MOV) memindahkan (menyalin) data satuan 1 word (16 bit) ke device word yang ditentukan. Data yang dapat dipindahkan bisa berupa nilai yang ada di device atau bisa pula ditentukan. Data yang dapat dipindahkan bisa berformat desimal atau heksadesimal.

Program ladder dan operasi

Simulasikan pengoperasian perintah berikut dengan mengklik sakelar input di sebelah kanan.

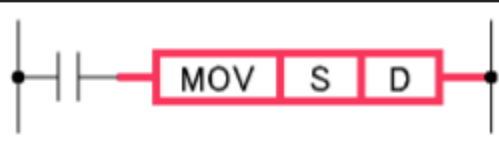
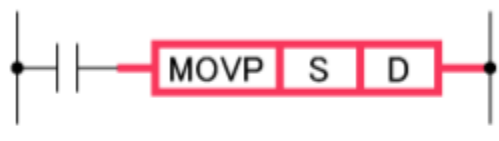
Setiap angka berwarna biru menunjukkan nilai (nilai saat ini) yang disimpan di device.



Saat Anda mengatur X4 ON/OFF berulang-ulang, nilai saat ini dari C0 bertambah. (0, 1...4->0...).

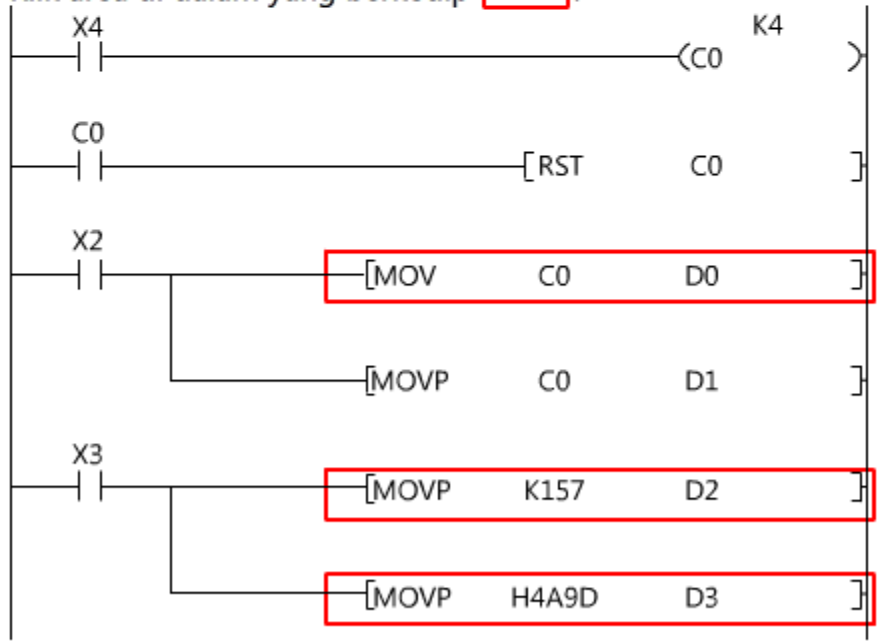
3.1 Memindahkan data ke device word

Kode perintah dan fungsi

| Simbol | Fungsi |
|---|--|
|  | Transfer data 16 bit (MOV) Selama kondisi input terpenuhi, data yang ditentukan pada sumber (S) ditransfer (disalin) ke device yang ditentukan pada destinasi (D). |
|  | Transfer data 16 bit (pulsa) (MOVP) Pada tepi naik kondisi (OFF ke ON), data yang ditentukan pada sumber (S) ditransfer (disalin) ke device yang ditentukan pada destinasi (D). |

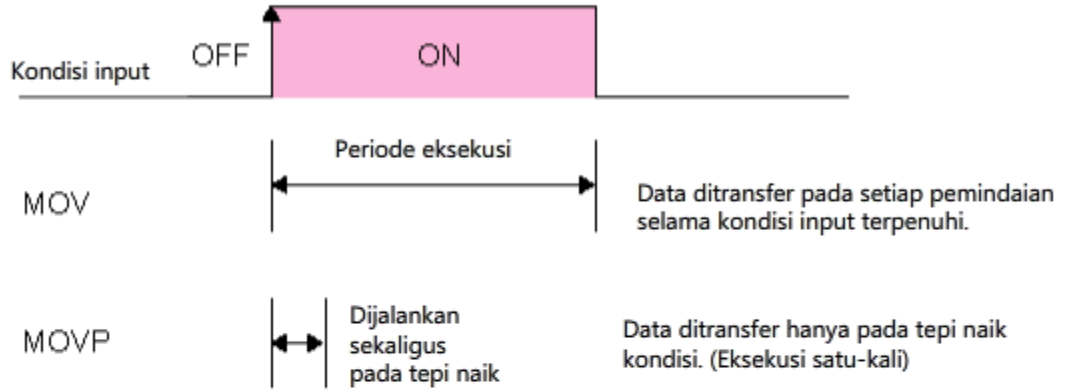
Program ladder

Klik area di dalam yang berkedip .

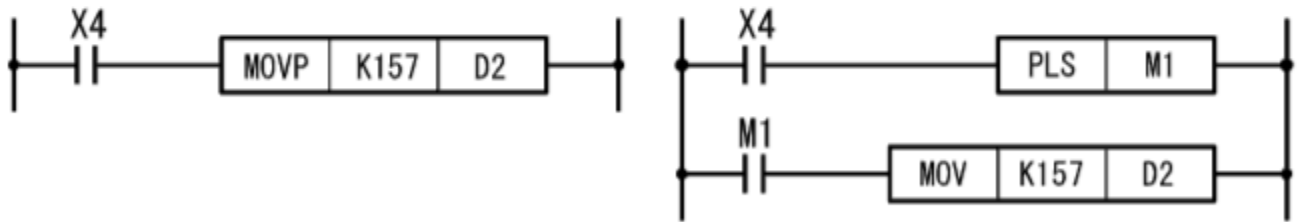


3.1.1 Perbedaan antara MOV dan MOVP

Perintah MOV digunakan untuk terus-menerus membaca data yang berubah-ubah. Sementara itu, perintah MOVP digunakan untuk mentransfer data sekaligus, misalnya untuk mengatur data atau membaca data ketika terjadi kesalahan.



Gambar di bawah ini menunjukkan dua program yang akan menghasilkan operasi yang sama, dengan perintah MOV dan MOVP. Pada kedua ladder rung, transfer data dijalankan begitu X4 ON. Dengan perintah MOVP, operasi dapat dijalankan tanpa perintah PLS, yang menentukan eksekusi operasi pada tepi naik.

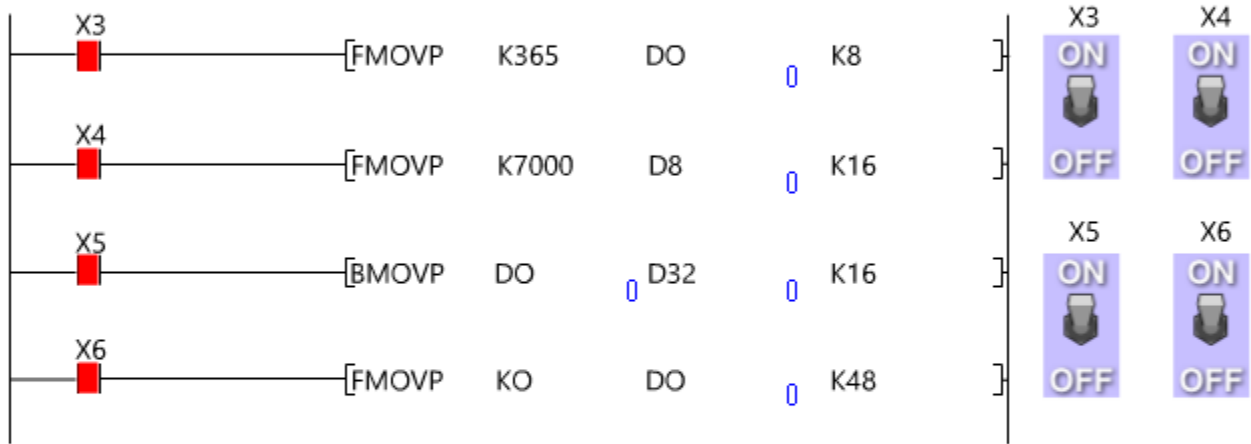


3.1.2 Memindahkan data ke beberapa device word sekaligus

Perintah MOV/MOVP digunakan untuk mentransfer data ke device. Untuk mentransfer data ke beberapa device dengan nomor yang kontinu, "perintah transfer serempak data identik" (FMOV) atau "perintah transfer blok data identik" (BMOV) dapat digunakan.

Program ladder dan operasi

Simulasikan pengoperasian perintah berikut dengan mengklik sakelar input di sebelah kanan. Setiap angka berwarna biru menunjukkan nilai (nilai saat ini) yang disimpan di device.



Monitor perangkat

| | | | | | |
|----|--------------------------------|-----|--------------------------------|-----|--------------------------------|
| DO | <input type="text" value="0"/> | D8 | <input type="text" value="0"/> | D32 | <input type="text" value="0"/> |
| D1 | <input type="text" value="0"/> | D9 | <input type="text" value="0"/> | D33 | <input type="text" value="0"/> |
| D7 | <input type="text" value="0"/> | D23 | <input type="text" value="0"/> | D47 | <input type="text" value="0"/> |

Ketika setiap sinyal input ON, data yang ditentukan ditransfer sekaligus.

CATATAN: Pada ladder rung ketiga yang dimulai dengan X5, data ditransfer oleh perintah BMOV.

3.1.2 Memindahkan data ke beberapa device word sekaligus

Kode perintah dan fungsi

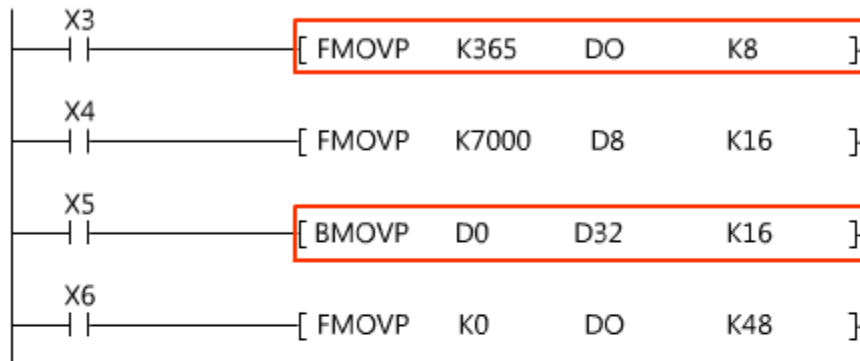
| Simbol | Fungsi |
|--------|--|
| | <p>Transfer serempak data identik (FMOV) Selama kondisi input terpenuhi, data yang ditentukan pada sumber (S) ditransfer (disalin) ke device yang ditentukan oleh destinasi (D) dan ke sejumlah "n" device setelah D.</p> |
| | <p>Transfer serempak data identik (pulsa) (FMOV P) Pada tepi naik kondisi, data yang ditentukan pada sumber (S) ditransfer (disalin) ke device yang ditentukan oleh destinasi (D) dan ke sejumlah "n" device setelah D.</p> |
| | <p>Transfer serempak data blok (BMOV) Selama kondisi input terpenuhi, data pada device yang ditentukan oleh sumber (S) dan sejumlah "n" device setelahnya ditransfer ke device yang ditentukan oleh device (D) dan ke sejumlah "n" device setelahnya.</p> |
| | <p>Transfer serempak data blok (pulsa) (BMOV P) Pada tepi naik kondisi, data pada device yang ditentukan oleh sumber (S) dan sejumlah "n" device setelahnya ditransfer ke device yang ditentukan oleh device (D) dan ke sejumlah "n" device setelahnya.</p> |

3.1.2

Memindahkan data ke beberapa device word sekaligus

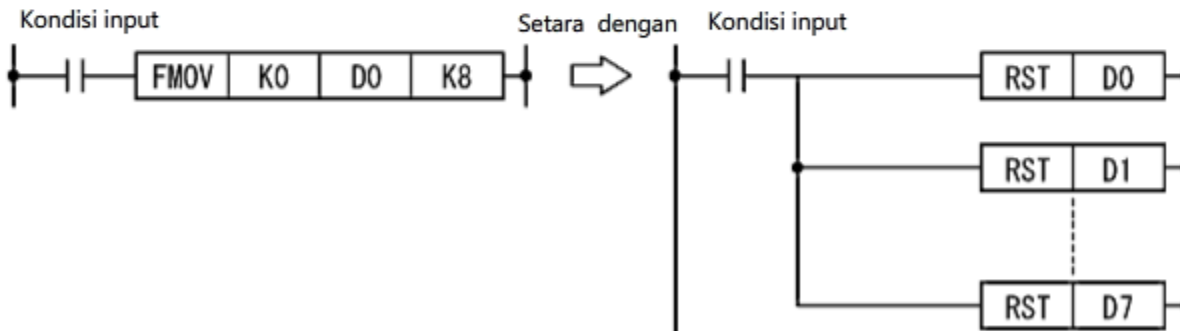
■ Program ladder dan operasi

Klik area di dalam yang berkedip .



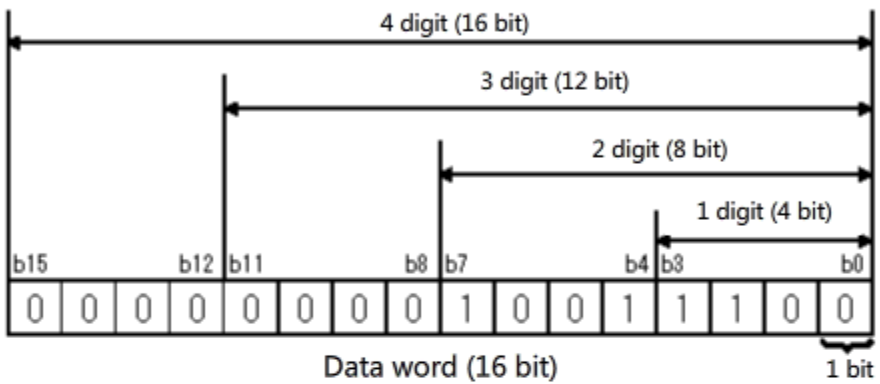
■ Aplikasi perintah FMOV dan BMOV

Perintah FMOV memudahkan untuk membersihkan volume data yang besar sekaligus.



3.1.3 Digit device bit

Empat device bit dikelompokkan menjadi satu digit device bit untuk mengontrol informasi bit pada rentang tertentu (transfer data, dll.).



■ Cara menentukan digit device bit

Digit device bit dinyatakan sebagai "bilangan digit" + "nomor device mulai". Bilangan digit adalah kelipatan 4. Tabel berikut menunjukkan beberapa contohnya. Berikut ini adalah contoh ketika nomor device mulai adalah "M0".

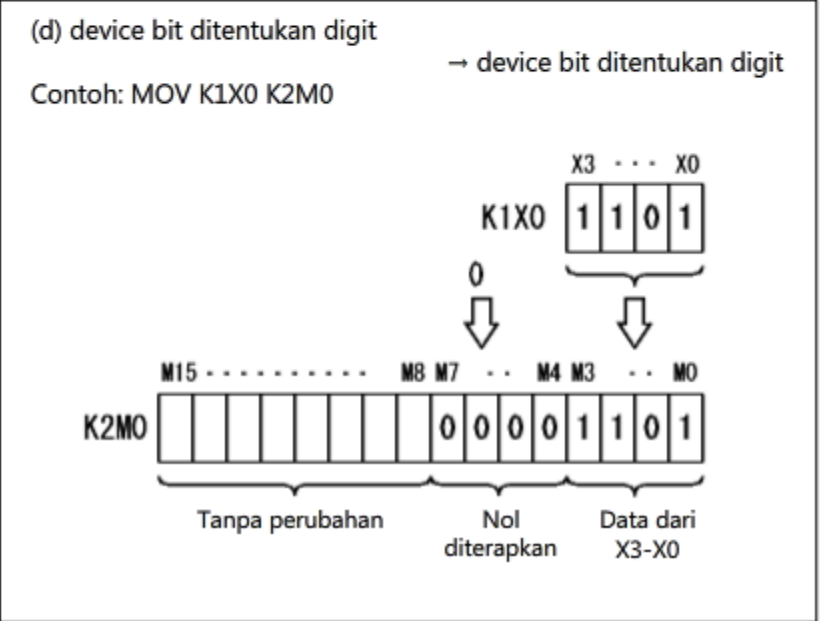
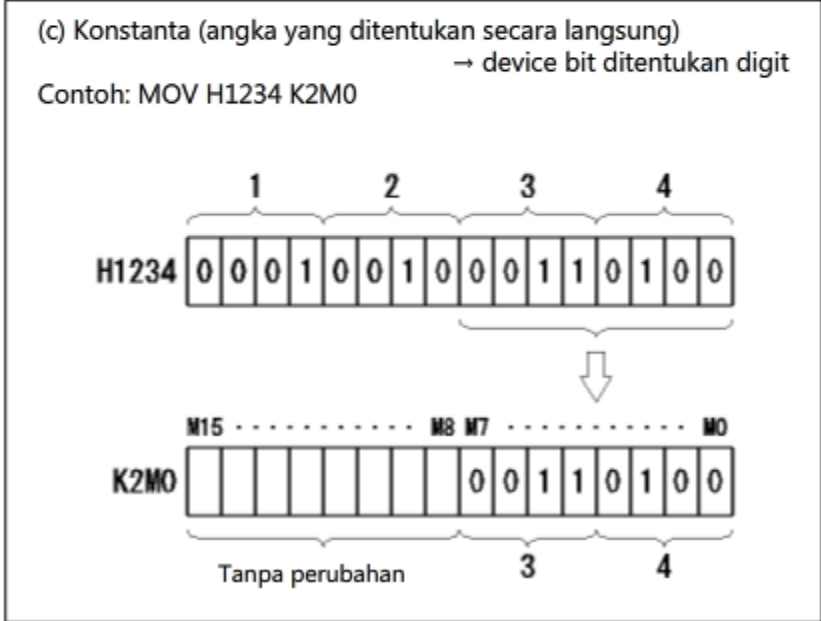
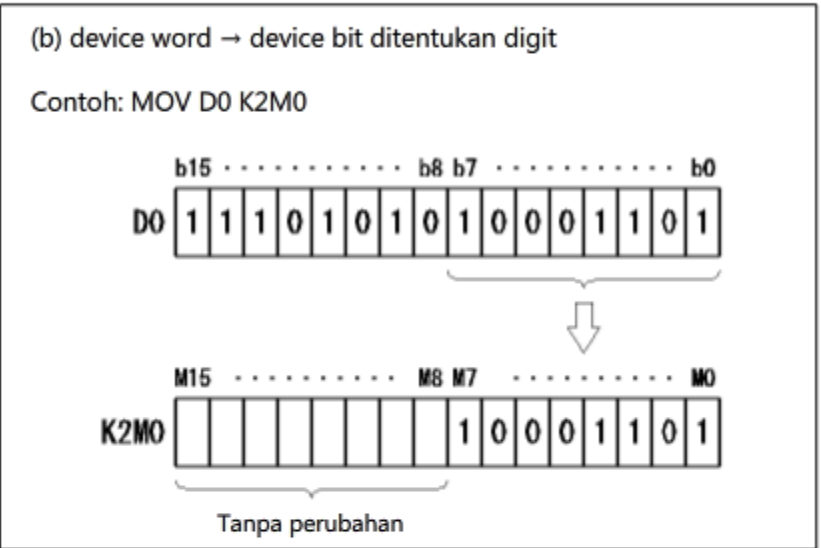
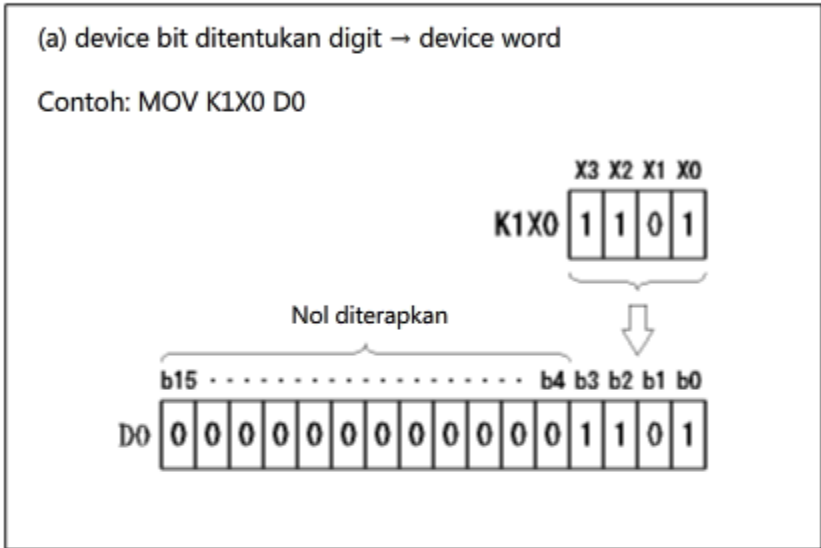
| Rentang bit | Metode spesifikasi |
|-------------|------------------------------|
| Data 16 bit | K4M0 (16 bit, M0 hingga M15) |
| Data 32 bit | K8M0 (32 bit, M0 hingga M31) |

Digit device bit (jumlah bit) menentukan rentang nilai numerik yang dapat digunakan.

| Digit device bit | Rentang nilai numerik yang dapat digunakan |
|------------------|--|
| K1 (4 bit) | 0 hingga 15 |
| K2 (8 bit) | 0 hingga 255 |
| K3 (12 bit) | 0 hingga 4095 |
| K4 (16 bit) | -32768 hingga 32767 Bit ke-16 dapat digunakan sebagai tanda positif/negatif untuk menyatakan nilai negatif. |

3.1.3 Contoh transfer digit device bit

Perintah transfer data digunakan untuk mentransfer (menyalin) angka-angka dari device sumber ke device destinasi. Contoh berikut ini menunjukkan bagaimana data yang ditentukan ditransfer.

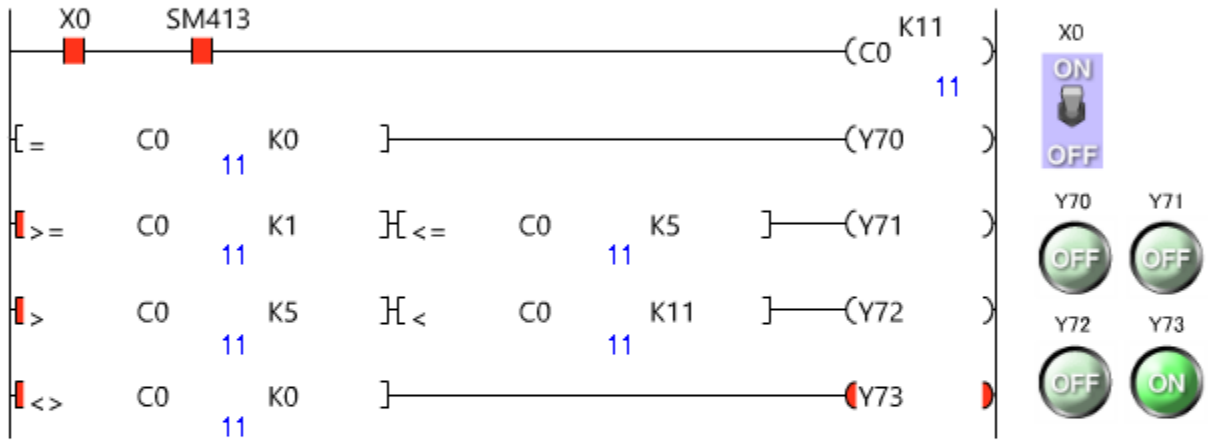


3.2 Membandingkan nilai numerik

Perintah operasi perbandingan digunakan untuk membandingkan data satuan-word dengan data yang tersimpan di device word. Ketika kondisi () terpenuhi, perintah berikutnya dijalankan.

Program ladder dan operasi

Simulasikan pengoperasian perintah berikut dengan mengklik sakelar input di sebelah kanan. Setiap angka berwarna biru menunjukkan nilai (nilai saat ini) yang disimpan di device.



Reset simulasi
Reset





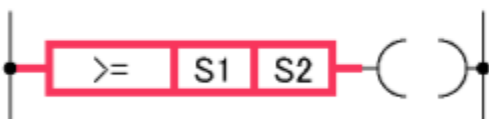

Y70 hingga Y73 ON/OFF tergantung nilai saat ini dari C0.

SM413 adalah relai khusus yang di-ON/OFF-kan pada interval 1 detik oleh modul CPU. (jam 2-detik)
Selama X0 ON, C0 menghitung naik setiap 2 detik.

* SM413 adalah relai khusus yang mengubah ON/OFF pada interval 1 detik (jam 2-detik). SM403 dapat digunakan untuk MELSEC Seri iQ-R/Q/L/iQ-F. MELSEC Seri F tidak memiliki relai jam 2-detik, tetapi memiliki M8011 (jam 0,01 dtk), M8012 (jam 0,1 dtk), M8013 (jam 1 dtk), dan M8014 (jam 1 mnt).

3.2 Membandingkan nilai numerik

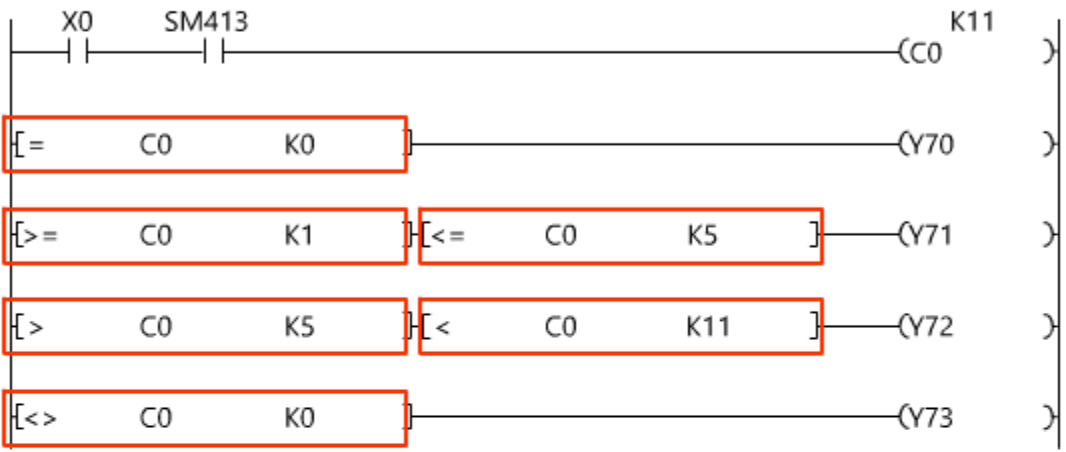
Kode perintah dan fungsi

| Simbol | Fungsi |
|---|---|
|  | Membandingkan data biner 16 bit (=) Kondisi terpenuhi ketika SOURCE 1 (SUMBER 1) sama dengan SOURCE2 (SUMBER2). |
|  | Membandingkan data biner 16 bit (<) Kondisi terpenuhi ketika SOURCE 1 (SUMBER 1) lebih kecil daripada SOURCE2 (SUMBER2). |
|  | Membandingkan data biner 16 bit (>) Kondisi terpenuhi ketika SOURCE 1 (SUMBER 1) lebih besar daripada SOURCE2 (SUMBER2). |
|  | Membandingkan data biner 16 bit (<=) Kondisi terpenuhi ketika SOURCE 1 (SUMBER 1) sama dengan atau lebih kecil daripada SOURCE2 (SUMBER2). |
|  | Membandingkan data biner 16 bit (>=) Kondisi terpenuhi ketika SOURCE 1 (SUMBER 1) sama dengan atau lebih besar daripada SOURCE2 (SUMBER2). |
|  | Membandingkan data biner 16 bit (<>) Kondisi terpenuhi ketika SOURCE 1 (SUMBER 1) tidak sama dengan SOURCE2 (SUMBER2). |

3.2 Membandingkan nilai numerik

Program ladder dan operasi

Klik area di dalam yang berkedip



SM413 adalah relai khusus yang di-ON/OFF-kan pada interval 1 detik oleh modul CPU (jam 2-detik). Relai khusus (SM) adalah device relai pada modul CPU. Setiap relai khusus memiliki kegunaan tertentu.

Bagian ini menjelaskan operasi aritmetika dasar pada device word (numerik).

■ Penambahan dan pengurangan

Operasi aritmetika yang menggunakan simbol penambahan (+) dan pengurangan (-).

■ Perkalian dan pembagian

Operasi aritmetika yang menggunakan simbol perkalian (*) dan pembagian (/).

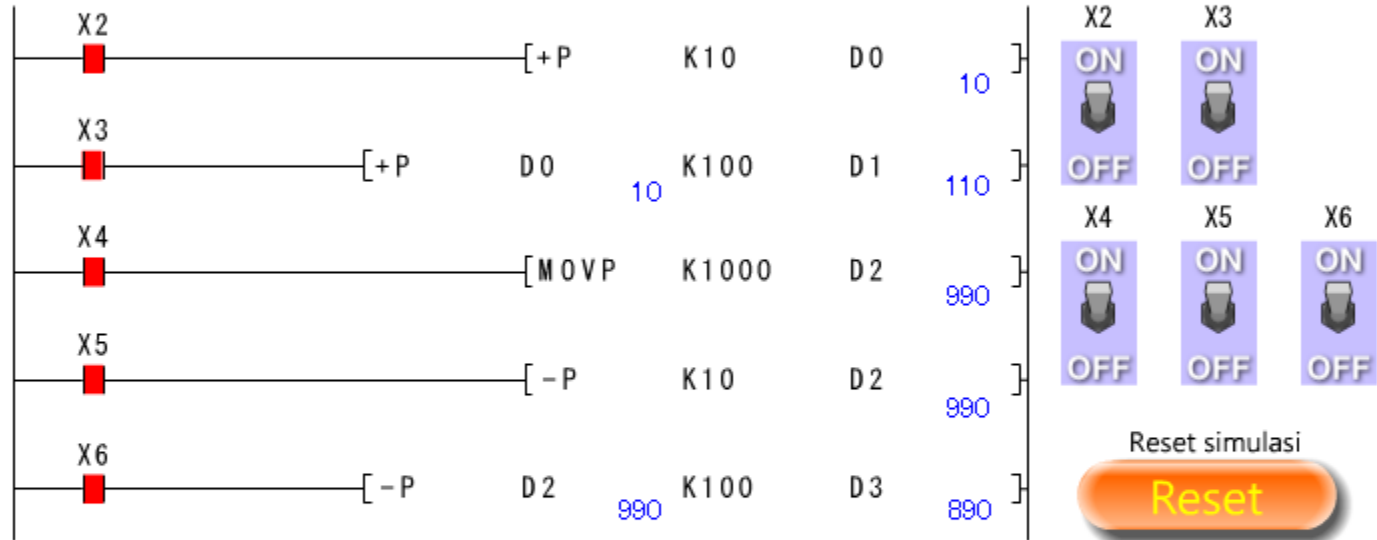
MELSEC Seri iQ-R/Q/L/iQ-F dan MELSEC Seri F menggunakan perintah yang berbeda, tetapi konsep dasarnya sama. Penjelasan dalam bagian ini didasarkan pada perintah yang digunakan pada MELSEC Seri iQ-R/Q/L/iQ-F.

3.3.1 Penambahan dan pengurangan

Diagram di bawah ini menunjukkan perintah yang menjalankan penambahan dan pengurangan dan menyimpan nilai yang diperoleh di dalam device yang ditentukan.

Program ladder dan operasi

Simulasikan pengoperasian perintah berikut dengan mengklik sakelar input di sebelah kanan. Setiap angka berwarna biru menunjukkan nilai (nilai saat ini) yang disimpan di device.


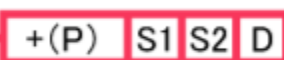
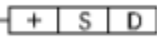



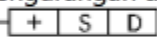
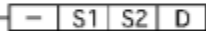


Ketika setiap sinyal input ON, operasi aritmetika dijalankan.

* Contoh didasarkan pada MELSEC Seri iQ-R/Q/L/iQ-F.

3.3.1 Penambahan dan pengurangan

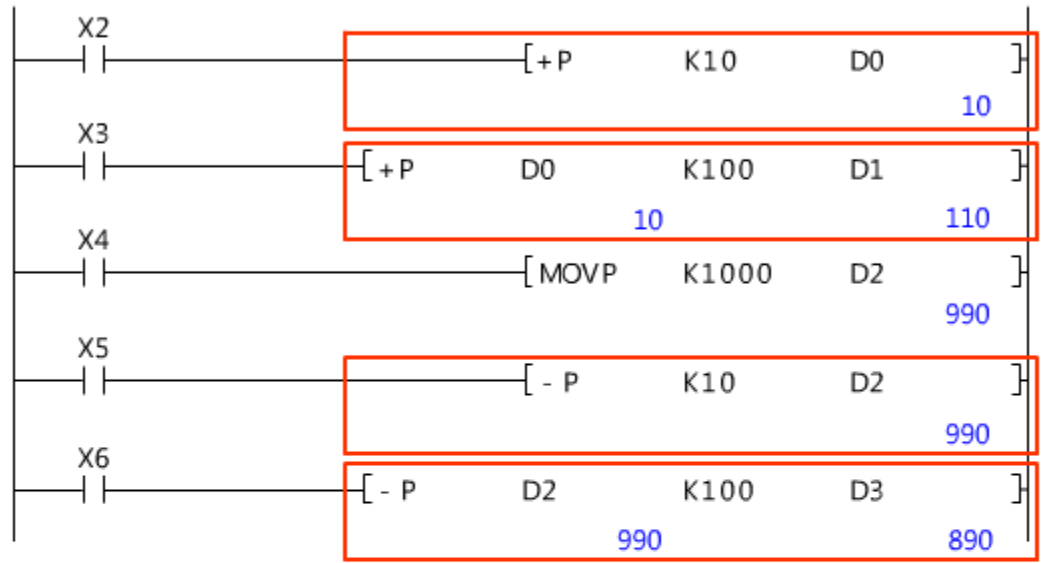
Kode perintah dan fungsi

| Simbol | Fungsi |
|---|---|
|  | <p>Penambahan data biner 16 bit</p> |
|  | <p>-  : Operasi "D + S = D" dijalankan. -  : Operasi "S1 + S2 = D" dijalankan.</p> |
|  | <p>Pengurangan data biner 16 bit</p> |
|  | <p>-  : Operasi "D - S = D" dijalankan. -  : Operasi "S1 - S2 = D" dijalankan.</p> |

3.3.1 Penambahan dan pengurangan

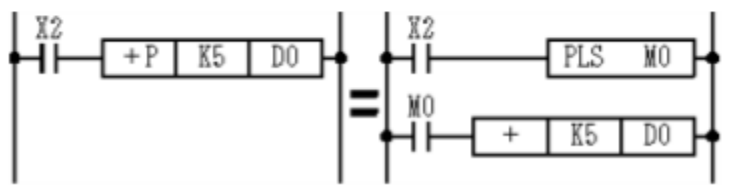
Program ladder dan operasi

Klik area di dalam yang berkedip .



Catatan tentang perintah penambahan dan pengurangan

Dalam keadaan normal, gunakan perintah +P/-P untuk menjalankan penambahan/pengurangan.
 Jika perintah +/- digunakan, penambahan/pengurangan dijalankan secara berulang selama kondisi input terpenuhi.
 Pada salah satu ladder rung berikut, penambahan dijalankan hanya sekali ketika X2 ON.



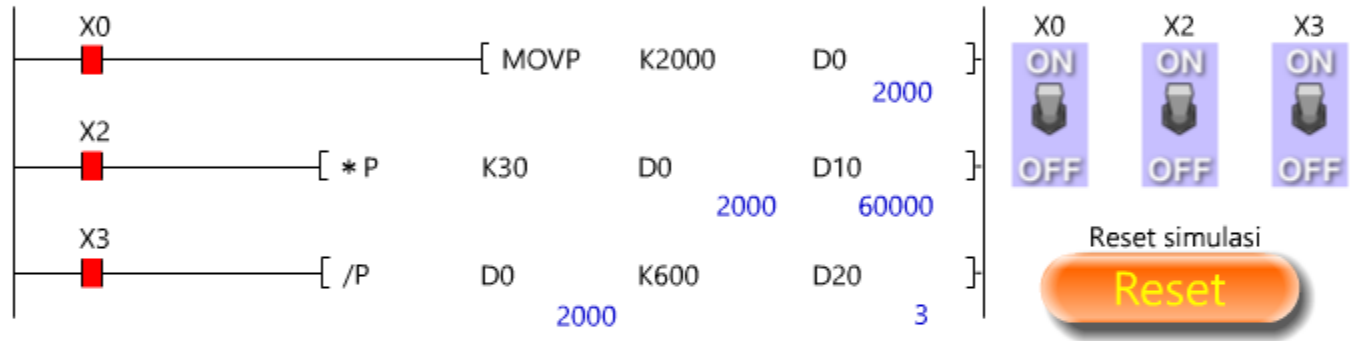
* Contoh didasarkan pada MELSEC Seri iQ-R/Q/L/iQ-F.

3.3.2 Perkalian dan pembagian

Diagram di bawah ini menunjukkan perintah yang menjalankan perkalian dan pembagian dan menyimpan nilai yang diperoleh di dalam device yang ditentukan.

Program ladder dan operasi

Simulasikan pengoperasian perintah berikut ini dengan mengeklik sakelar input di sebelah kanan. Setiap angka berwarna biru menunjukkan nilai (nilai saat ini) yang disimpan di device.



Ketika setiap sinyal input ON, operasi aritmetika dijalankan.

* Contoh didasarkan pada MELSEC Seri iQ-R/Q/L/iQ-F.

3.3.2 Perkalian dan pembagian

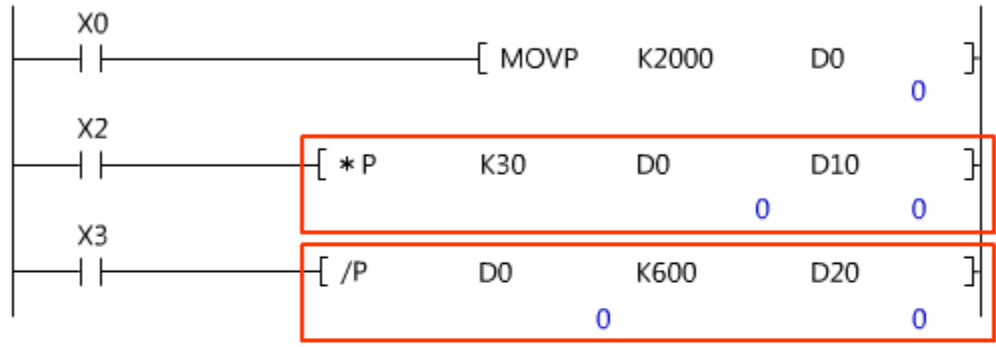
Kode perintah dan fungsi

| Simbol | Fungsi |
|--------|--|
| | <p>Perkalian data biner 16 bit (*) Operasi "S1 x S2 = (D+1 D)" dijalankan. ("D+1" adalah device yang mengikuti D. Jika D adalah D100, maka "D+1" adalah D101.) Hasil operasi adalah data 32 bit, yang terdiri atas 2 satuan word ("D" dan "D+1").</p> |
| | <p>Pembagian data biner 16 bit Operasi "S1/S2 = (D [kuosien], D + 1 [sisal])" dijalankan. ("D + 1" adalah device yang mengikuti D. Jika D adalah D100, maka "D + 1" adalah D101.) Hasil operasi berupa bilangan bulat.</p> |

3.3.2 Perkalian dan pembagian

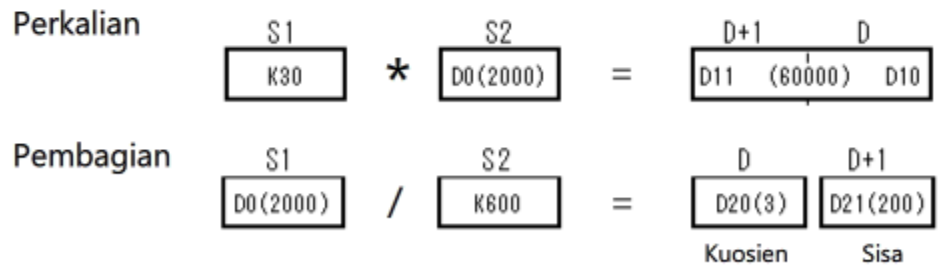
Program ladder dan operasi

Klik area di dalam yang berkedip .



Catatan tentang perintah perkalian dan pembagian

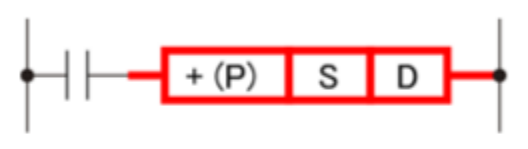

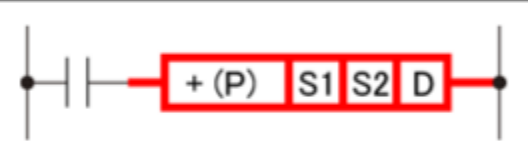
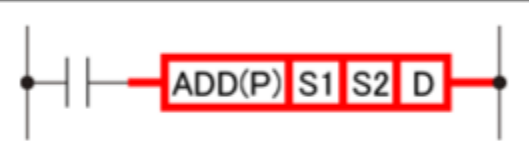
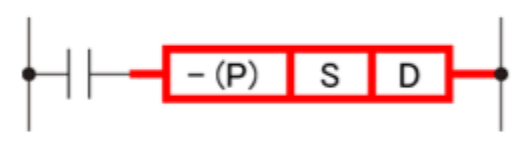
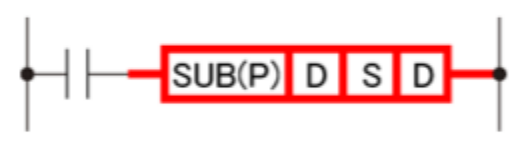
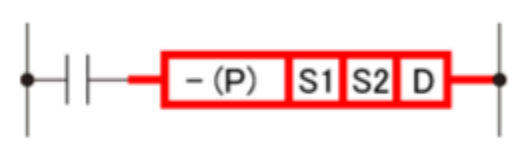
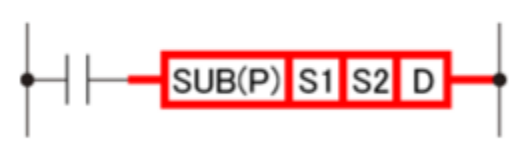
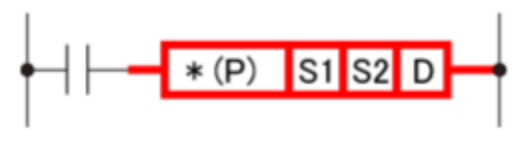
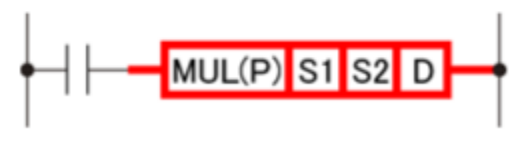
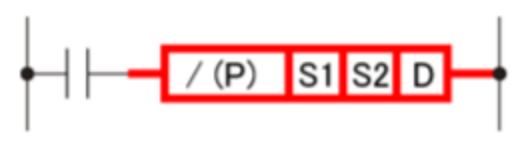
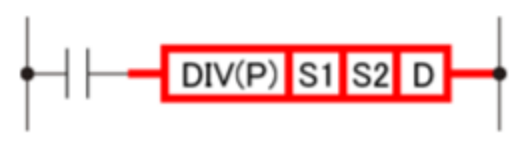
Untuk menjalankan perintah perkalian atau pembagian, dua device word yang berurutan (D, D+1) harus ada untuk destinasi (D).



* Contoh didasarkan pada MELSEC Seri iQ-R/Q/L/iQ-F.

3.3.3 Perbedaan antara MELSEC Seri iQ-R/Q/L/iQ-F dan MELSEC Seri F

MELSEC Seri iQ-R/Q/L/iQ-F dan MELSEC Seri F menggunakan simbol yang berbeda. Tabel di bawah ini menunjukkan perbedaan utamanya.

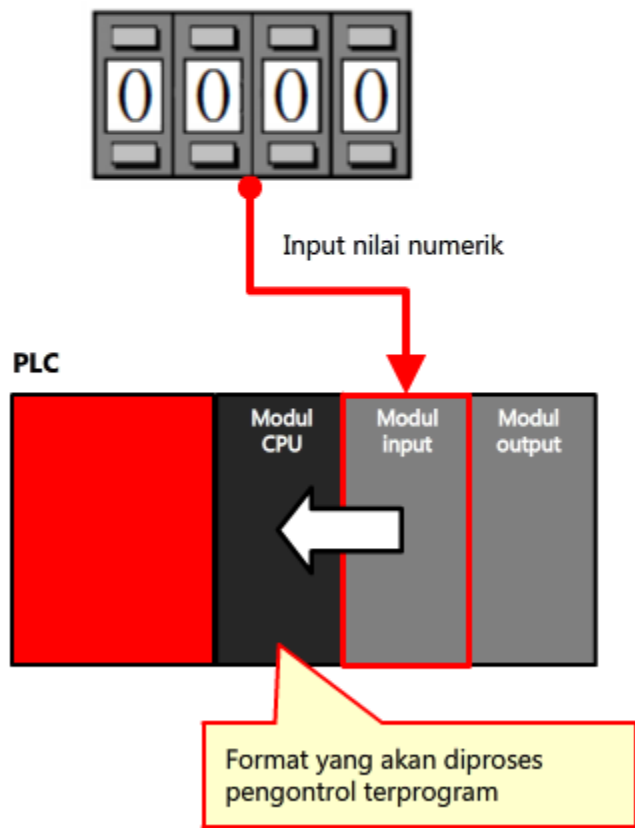
| Operasi aritmetika | Perintah yang digunakan di MELSEC Seri iQ-R/Q/L/iQ-F | Perintah yang digunakan di MELSEC Seri F | Perbedaan |
|--------------------|---|--|--|
| Penambahan (+) |  |  | MELSEC Seri iQ-R/Q/L/iQ-F: +(P) MELSEC Seri F: ADD(P) |
| |  |  | |
| Pengurangan (-) |  |  | MELSEC Seri iQ-R/Q/L/iQ-F: -(P) MELSEC Seri F: SUB(P) |
| |  |  | |
| Perkalian (*) |  |  | MELSEC Seri iQ-R/Q/L/iQ-F: *(P) MELSEC Seri F: MUL(P) |
| Pembagian (/) |  |  | MELSEC Seri iQ-R/Q/L/iQ-F: /(P) MELSEC Seri F: DIV(P) |

3.4 Transmisi/penerimaan data antara PLC dan device I/O

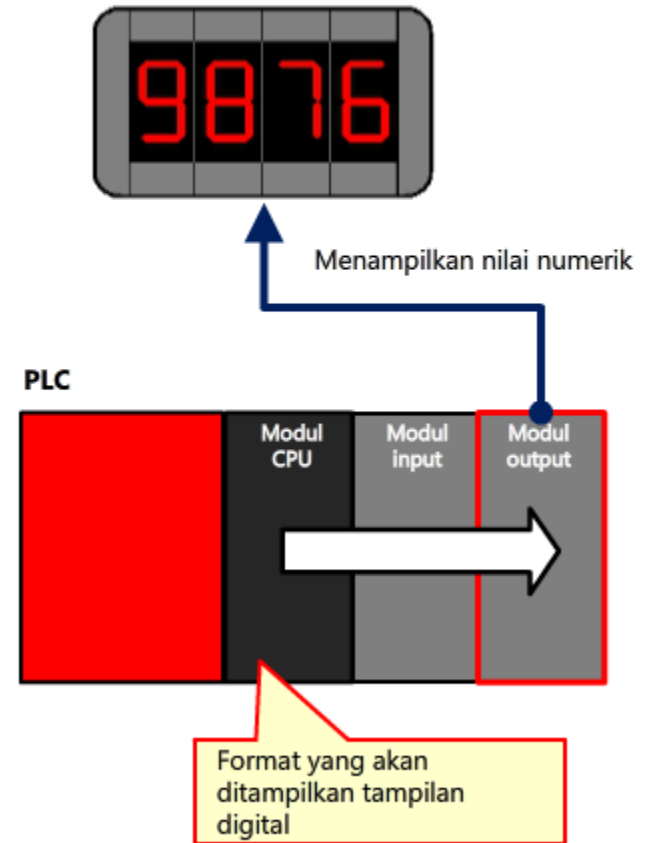
Sakelar input digital adalah device input yang menginput data ke pengontrol terprogram dalam bentuk nilai numerik. Tampilan digital adalah device output yang menampilkan data yang diterima dari pengontrol terprogram dalam bentuk nilai numerik.

Data yang diterima dari sakelar input digital harus diformat agar dapat diproses oleh pengontrol terprogram. Demikian juga data yang dioutput ke tampilan digital harus diformat ke dalam format yang dapat dibaca oleh tampilan digital.

Sakelar input digital



Tampilan digital



3.4.1 Menerima input sakelar input digital

Agar pengontrol terprogram dapat menerima input sakelar input digital, perintah BIN digunakan.

Kode perintah dan fungsi

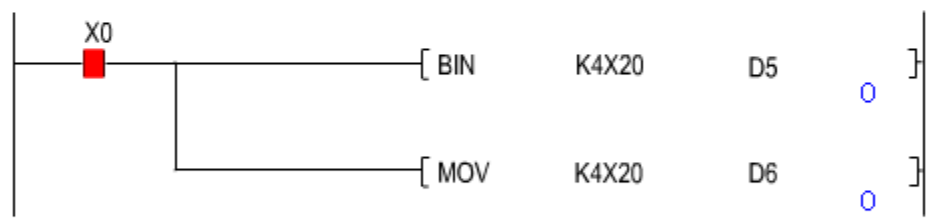
| Simbol | Fungsi |
|--------|---|
| | Data pada device (S) diformat ke dalam sebuah format yang dapat diproses oleh pengontrol terprogram, lalu disimpan di dalam device (D). |

Program ladder dan operasi

Simulasikan pengoperasian perintah berikut dengan mengklik sakelar input di sebelah kanan. Setiap angka berwarna biru menunjukkan nilai (nilai saat ini) yang disimpan di device.

D5 menyimpan data yang diterima dari sakelar input digital setelah diformat oleh perintah BIN.

D6 menyimpan data yang tidak diformat yang diterima dari sakelar input digital.



Sakelar input digital

<- Penurunan

0 0 0 0

<- Kenaikan

X2F hingga X20

Jika perintah MOV digunakan, angkanya tidak cocok.

Reset simulasi



3.4.2 Menampilkan data PLC pada tampilan digital

Untuk menampilkan data pengontrol terprogram pada tampilan digital, perintah BCD digunakan.

Kode perintah dan fungsi

| Simbol | Fungsi |
|--------|---|
| | Data pada device (S) diformat ke dalam sebuah format yang dapat ditampilkan pada tampilan digital, lalu disimpan di dalam device (D). |

Program ladder dan operasi

Simulasikan pengoperasian perintah berikut dengan mengklik sakelar input di sebelah kanan.

Tampilan digital atas menampilkan data yang diformat oleh perintah BCD.

Tampilan digital bawah menampilkan data yang tidak diformat.

The simulation interface includes a ladder logic diagram with an input X6 connected to two parallel branches: one with a BCD instruction (K9876 to K4Y50) and one with a MOV instruction (K9876 to K4Y60). A switch labeled X6 is currently in the 'ON' position. To the right, two digital displays are shown: the top one (Y5F to Y50) displays '9876' and the bottom one (Y6F to Y60) displays '2694'. A 'Reset' button is located at the bottom of the interface.

Jika perintah MOV digunakan, angkanya tidak cocok.

Dalam kursus ini Anda telah mempelajari:

- Konsep input dan output ke/dari pengontrol terprogram
- Perintah utama yang mengontrol pengontrol terprogram
- Informasi yang diterima oleh pengontrol terprogram MELSEC dijalankan di program ladder pada pengontrol terprogram, dan hasil eksekusi dikirimkan ke luar sebagai output
- Perbedaan format data bit dan data word
- Dasar-dasar program kontrol

Silakan ikuti kursus "Engineering Software MELSOFT GX Works3 (Ladder)" (Software Rekayasa MELSOFT GX Works3 (Ladder)) untuk mempelajari cara mengedit dan mendaftarkan program ke modul CPU MELSEC Seri iQ-R/iQ-F.

Silakan ikuti kursus "GX Works2 Dasar" untuk mempelajari cara mengedit dan mendaftarkan program ke modul CPU MELSEC Seri Q/L/F.

Tes**Tes Akhir**

Setelah menyelesaikan semua pelajaran pada kursus **Dasar-dasar pemrograman PLC**, kini Anda siap mengerjakan tes akhir. Jika Anda masih kurang memahami salah satu topik yang dibahas, gunakan kesempatan ini untuk mengulas topik tersebut.

Total terdapat **11 pertanyaan (54 pilihan)** dalam Tes Akhir ini.

Anda dapat mengikuti tes akhir sesering mungkin.

Cara menilai tes

Setelah memilih jawaban, pastikan mengeklik tombol **Jawab**. Jawaban akan hilang jika Anda melanjutkan tanpa mengeklik tombol Jawab. (Dianggap sebagai pertanyaan belum dijawab.)

Hasil penilaian

Jumlah jawaban yang benar, jumlah pertanyaan, persentase jawaban yang benar, dan hasil lulus/gagal akan ditampilkan pada halaman nilai.

Jawaban yang benar: **5**

Jumlah total pertanyaan: **5**

Persentase: **100%**

Agar lulus tes, Anda harus menjawab **60%** pertanyaan dengan benar.

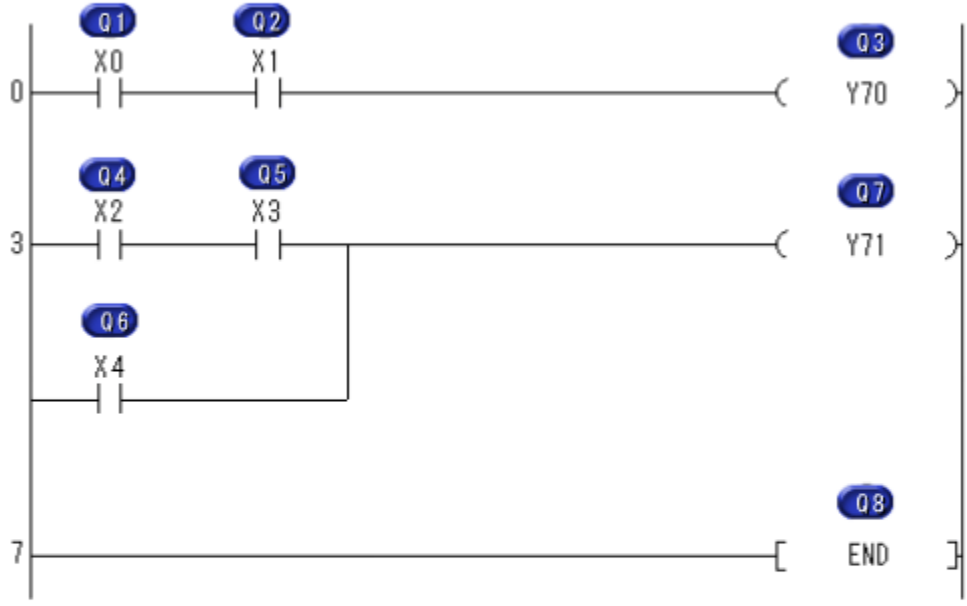
Lanjutkan

Tinjau

- Klik tombol **Lanjutkan** untuk keluar dari tes.
- Klik tombol **Tinjau** untuk mengulas tes. (Jawaban yang benar dicentang)
- Klik tombol **Coba lagi** untuk mengulang tes.

Tes Tes Akhir 1

Nomori perintah berikut sesuai urutan pemrosesannya.



- P1
- P2
- P3
- P4
- P5
- P6
- P7
- P8

Jawab

Kembali

Tes Tes Akhir 2

Kalimat di bawah ini menguraikan tentang peralatan I/O eksternal dan sinyal I/O ke/dari pengontrol terprogram. Lengkapi kalimat dengan memilih kata yang tepat.

1) Angka input dan output untuk pengontrol terprogram Seri Q dimulai dari

(--Select--) dan menggunakan nilai (--Select--).

2) Angka yang sama digunakan untuk sinyal input dan output. Karena itu,

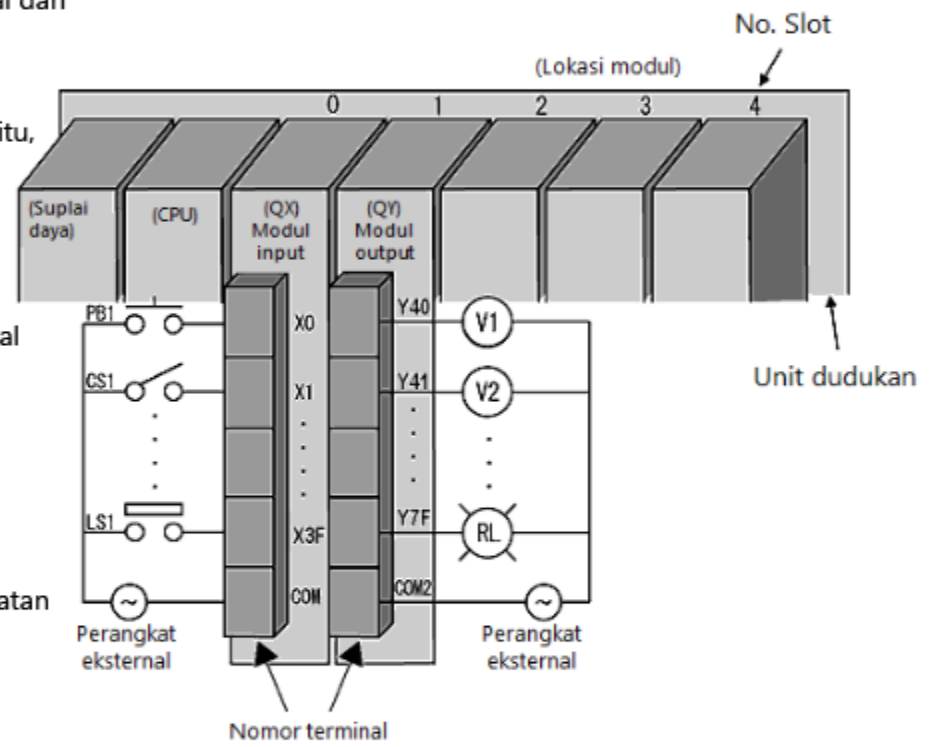
input didahului oleh (--Select--) dan output didahului oleh (--Select--).

3) Angka yang ditetapkan ke sinyal yang diinput dari peralatan eksternal ditentukan oleh kondisi berikut:

- Di mana (--Select--) dipasang pada unit dudukan
- Nomor terminal

4) Angka yang ditetapkan ke sinyal yang dioutput (kumparan) ke peralatan eksternal ditentukan oleh kondisi berikut:

- Di mana (--Select--) dipasang pada unit dudukan
- Nomor terminal



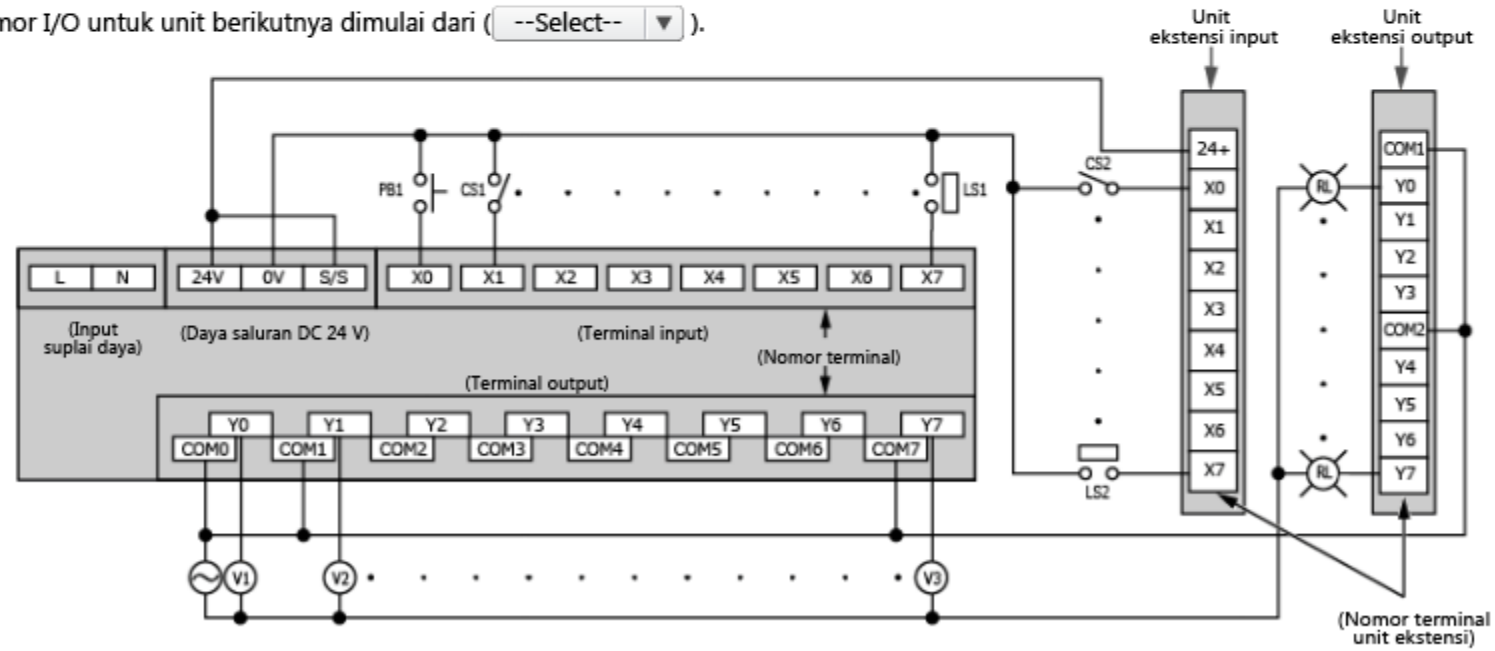
Jawab

Kembali

Tes Tes Akhir 3

Kalimat di bawah ini menguraikan tentang peralatan I/O eksternal dan nomor I/O yang ditetapkan ke pengontrol terprogram. Lengkapi kalimat dengan memilih kata yang tepat. (MELSEC Seri F)

- 1) Nomor I/O untuk pengontrol terprogram MELSEC Seri Q dimulai dari (--Select--) dan menggunakan nilai (--Select--).
- 2) Angka yang sama digunakan untuk sinyal input dan output. Karena itu, input didahului oleh (--Select--) dan output didahului oleh (--Select--).
- 3) Jika unit ekstensi I/O ditambahkan, unit tersebut akan diberi nomor yang ada setelah nomor yang ditetapkan ke (--Select--) yang mendahuluinya.
- 4) Nomor I/O sebuah unit selalu dimulai dari nomor yang memuat "0" dalam digit pertamanya. Jika nomor I/O dari unit yang mendahului berakhir di X17, maka nomor I/O untuk unit berikutnya dimulai dari (--Select--).

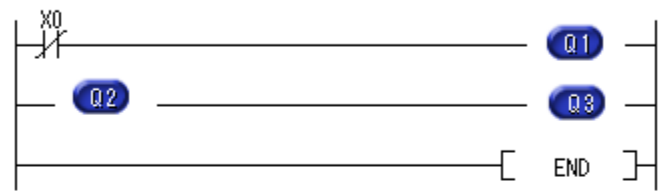


Tes Tes Akhir 4

Tarik dan taruh perintah yang benar untuk menyelesaikan program yang menjalankan operasi berikut ini:

Ketika sakelar X0 OFF, lampu A ON. (Y70 ON)

Ketika sakelar ini ON, lampu B ON. (Y71 ON)



P1 P2 P3

Jawab

Kembali

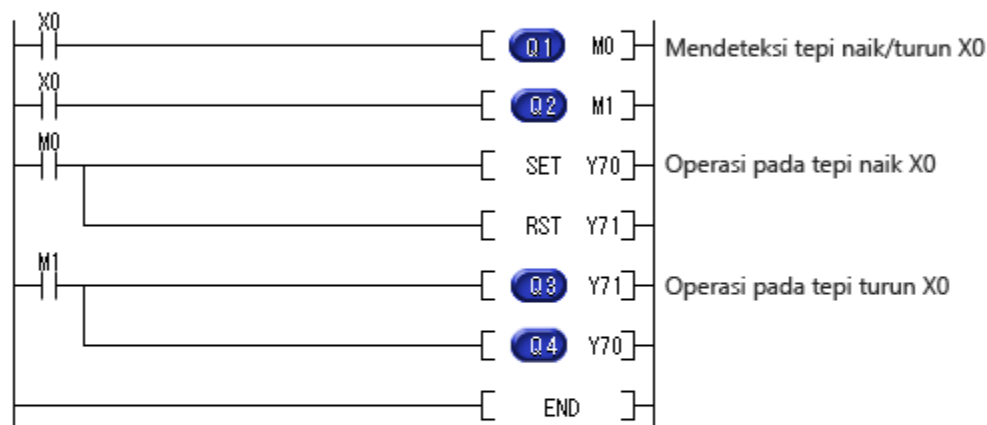
Tes Tes Akhir 5

Tarik dan taruh perintah yang benar untuk menyelesaikan program yang menjalankan operasi berikut ini:

Selama material diproses, "sinyal dalam-proses" (X0) ON

Pada tepi naik "sinyal dalam-proses" (X0), lampu A ON (Y70 ON), dan lampu B OFF (Y71 OFF).

Pada tepi turun "sinyal dalam-proses" (X0), lampu B ON (Y70 ON), dan lampu A OFF (Y71 OFF).



P1 --Select-- P2 --Select-- P3 --Select-- P4 --Select--

Jawab

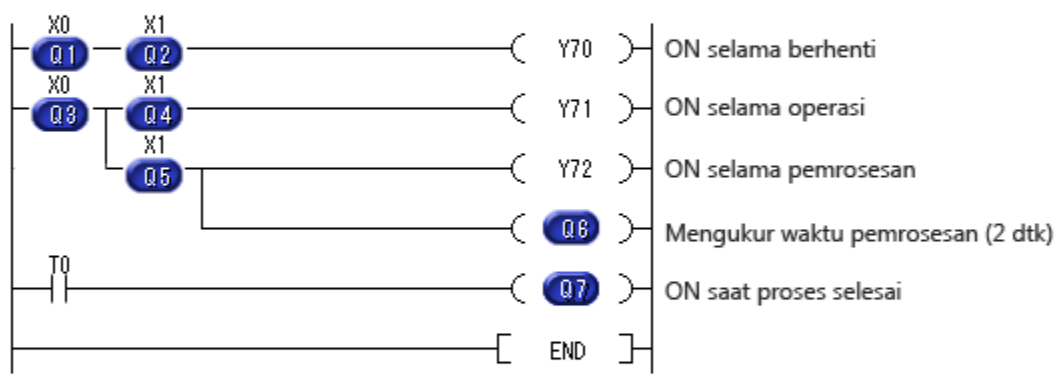
Kembali

Tes Tes Akhir 6

Tarik dan taruh perintah yang benar untuk menyelesaikan program yang menjalankan operasi berikut ini:

Lampu dinyalakan dengan meng-ON/OFF-kan sakelar mulai operasi (X0) dan sakelar mulai pemrosesan (X1).
2 detik setelah kedua sakelar ON, lampu D ON.

| [Operasi mulai (X0)] | [Sakelar mulai pemrosesan (X1)] | [Lampu] |
|----------------------|---------------------------------|---|
| OFF | OFF | Lampu A (Y70 ON) |
| ON | OFF | Lampu B (Y71 ON) |
| ON | ON | Lampu C (Y72 ON), dan setelah 2 dtk, Lampu D (Y73 ON) |



P1 P2 P3 P4

P5 P6 P7

Tes Tes Akhir 7

Tarik dan taruh perintah yang benar untuk menyelesaikan program yang menjalankan operasi berikut ini:

Saat produk lewat di konveyor, sinyal X0 ON.

Setelah produk lewat (setelah 3 dtk), lampu A ON. (Y70 ON selama 1 dtk.)

Setelah 5 produk lewat, lampu B ON. (Y71 ON)

Setelah lampu B ON, sakelar konfirmasi (X1) ON.



P1 --Select-- P2 --Select-- P3 --Select-- P4 --Select--

P5 --Select-- P6 --Select--

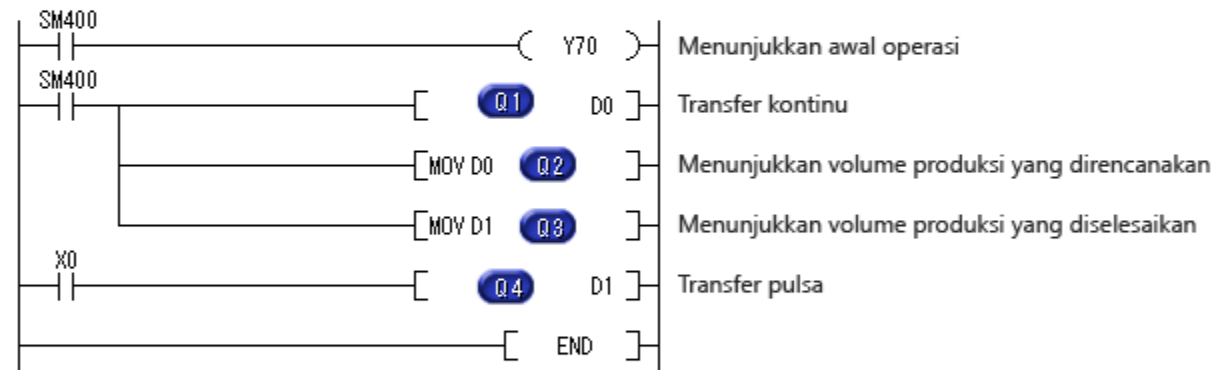
Jawab

Kembali

Tes Tes Akhir 8

Tarik dan taruh perintah yang benar untuk menyelesaikan program yang menjalankan operasi berikut ini:

- 1) Ketika operasi dimulai, lampu A ON. (Y70 ON)
- 2) Volume produksi yang direncanakan diinput menggunakan sakelar digital (X20-X2F). Volume tersebut ditransfer ke register data D0 setiap kali diinput.
- 3) Nilai yang disimpan di register data (D0, D1) terus-menerus ditransfer dan diperbarui di tampilan digital seperti di bawah.
 Y40-Y4F: Menunjukkan volume produksi yang direncanakan (D0)
 Y50-Y5F: Menunjukkan volume produksi yang diselesaikan (D1)
- 4) Sakelar digital X30 hingga X3F digunakan untuk menginput volume produksi yang diselesaikan. Ketika sakelar penyelesaian pengaturan (X0) ON, volume produksi yang diselesaikan ditransfer ke data register D1.

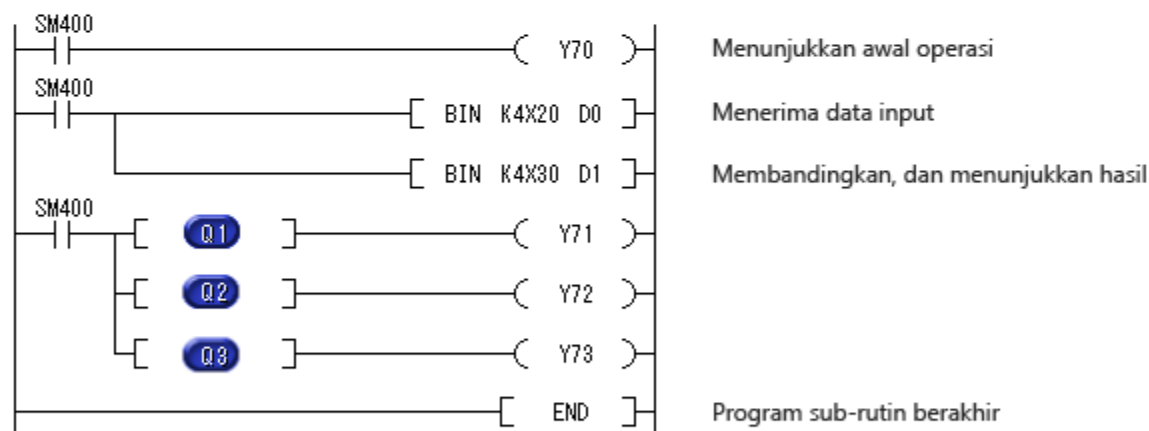


* Dalam program ini, perintah MOV digunakan untuk transfer data.
 * Untuk memonitor D0 dan D1, gunakan nilai heksadesimal.

P1 P2 P3 P4

Tarik dan taruh perintah yang benar untuk menyelesaikan program yang menjalankan operasi berikut ini:

- 1) Ketika operasi dimulai, lampu A ON. (Y70 ON)
- 2) Operasi berikut dijalankan.
 - Volume produksi yang direncanakan A, yang telah diinput dengan sakelar digital (X20-X2F), diformat dan ditransfer ke register data D0.
 - Volume produksi yang direncanakan B, yang telah diinput dengan sakelar digital (X30-X3F), diformat dan ditransfer ke register data D1.
 - Register data D0 dan D1 dibandingkan satu sama lain, dan hasilnya ditunjukkan oleh lampu.
 - D0>D1: Lampu B (Y71 ON/OFF)
 - D0=D1: Lampu C (Y72 ON/OFF)
 - D0<D1: Lampu D (Y73 ON/OFF)

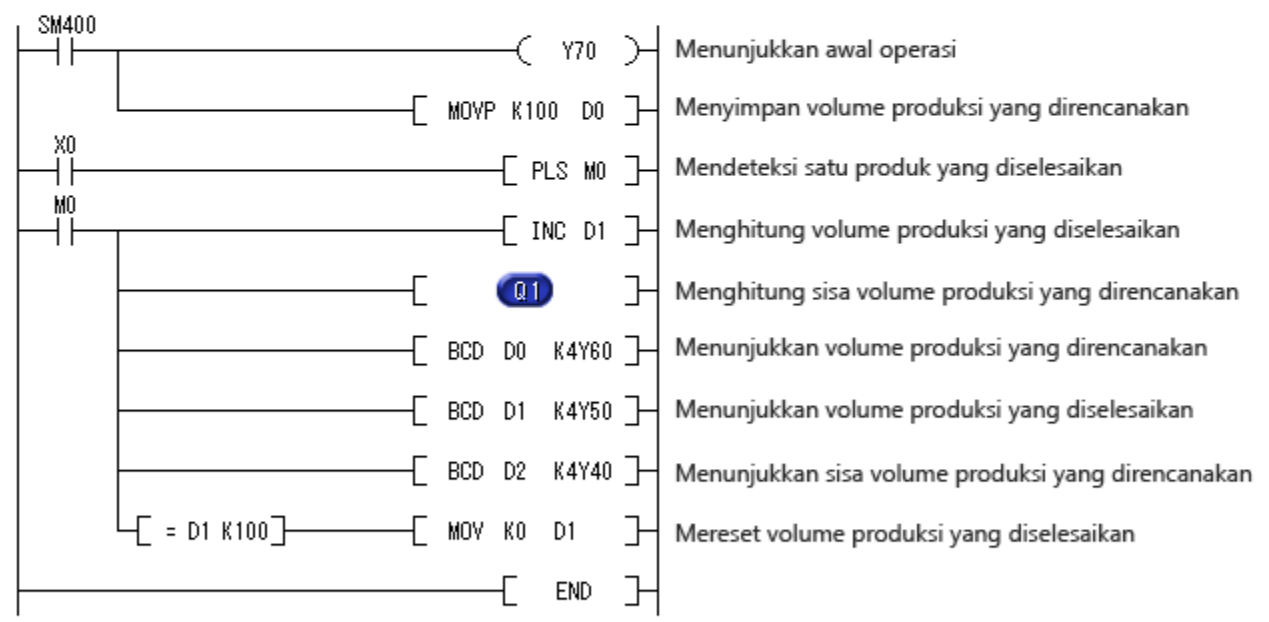


P1 P2 P3

Tes Tes Akhir 10

Tarik dan taruh perintah yang benar untuk menyelesaikan program yang menjalankan operasi berikut ini:

- 1) Ketika operasi dimulai, lampu A ON. (Y70 ON)
 - 2) Saat mulai, volume produksi yang direncanakan 100 disimpan di register data D0.
 - 3) Setiap kali sebuah produk diselesaikan, informasi berikut disimpan di register data.
 - D1: Volume produksi yang diselesaikan (dihitung pada tepi naik X0)
 - D2: Sisa volume produksi yang direncanakan ($D2 = D0 - D1$)
- Tampilan digital menampilkan data berikut:
- Y40-Y4F: Nilai di D2 (sisa volume produksi yang direncanakan (0 hingga 100))
 - Y50-Y5F: Nilai di D1 (volume produksi yang diselesaikan (0 hingga 100))
 - Y60-Y6F: Nilai D0 (volume produksi yang direncanakan (100))



Q1 --Select--

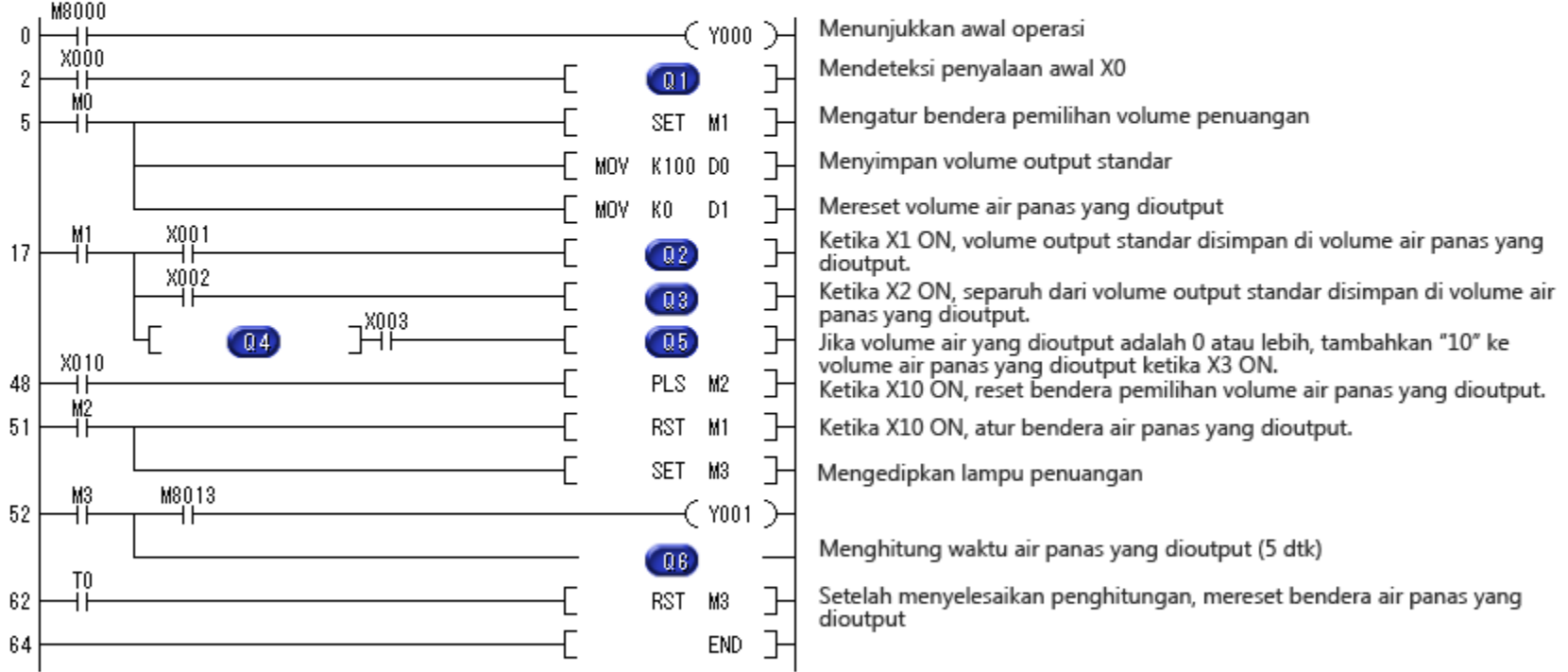
Jawab

Kembali

Tes Tes Akhir 11

Program kontrol yang ditunjukkan di bawah adalah untuk MELSEC Seri F dan memuat perintah dan relai khusus. Tarik dan taruh perintah yang tepat untuk menyelesaikan program yang menuangkan air panas dari dispenser air panas.

- 1) Ketika operasi dimulai, lampu ON. (Y0 ON)
- 2) Pada tepi naik dari awal operasi dispenser air panas (X0 ON), "100" disimpan di volume penyaluran air standar D0, dan "0" disimpan di volume air panas yang dioutput D1.
(Reset data)
- 3) Pilih volume air panas yang dioutput.
Pada tepi naik X1, volume output standar D0 disimpan di volume air panas yang dioutput D1.
Pada tepi naik X2, separuh dari volume output standar D0 disimpan di volume air panas yang dioutput D1.
- 4) Jika volume air panas yang dioutput D1 dipilih dan nilainya 0 atau lebih, maka "+10" ditambahkan ke volume air panas yang dioutput D1 pada tepi naik X3. lalu nilai tambah ini disimpan di volume air panas yang dioutput.
- 5) Pada tepi naik output air panas (X10), lampu penerangan berkedip dalam interval 1 dtk (Y1 mengulangi ON/OFF), dan waktu output air panas 5 dtk dihitung (T0).
- 6) Hentikan lampu penerangan yang berkedip setelah menghitung waktu output air panas (T0).



P1 P2 P3 P4

P5 P6

Tes Skor Tes

Anda telah menyelesaikan Tes Akhir. Bidang hasil Anda adalah sebagai berikut.
Untuk menutup Tes Akhir, lanjutkan ke halaman berikutnya.

Jawaban yang benar: **11**

Jumlah total pertanyaan: **11**

Persentase: **100%**

Lanjutkan

Tinjau

Selamat. Anda berhasil lulus tes.

Anda telah menyelesaikan kursus **Dasar-dasar Pemrograman PLC**.

Terima kasih telah mengikuti kursus ini.

Kami harap Anda menikmati pelajaran, dan kami harap informasi yang diperoleh dalam kursus ini dapat bermanfaat di masa mendatang.

Anda dapat mengulas kursus ini sesering yang Anda inginkan.

Tinjau

Tutup