

PLC

Aplikasi Pemrograman

Kursus ini dirancang bagi peserta yang telah menyelesaikan Kursus Dasar-dasar MELSEC Seri Q dan siap untuk mempelajari langkah selanjutnya, yaitu pemrograman.

Pendahuluan Tujuan Kursus

Kursus ini dirancang bagi peserta yang telah menyelesaikan Kursus Dasar atau yang memiliki cukup pengetahuan untuk mempelajari lebih lanjut tentang fungsi dan kegunaan pengontrol terprogram MELSEC Seri Q.

Dengan mengikuti kursus ini, Anda akan mempelajari tentang kegunaan berbagai perangkat pengontrol terprogram Seri Q, konfigurasi sistem CPU dan diagnostik, dan kegunaan fungsi-fungsi pokok pengontrol terprogram Seri Q.

Pendahuluan Struktur Kursus

Berikut adalah daftar isi kursus.
Sebaiknya Anda mulai dari Bab 1.

Bab 1 Pengaturan dan Modifikasi Perangkat

Mempelajari cara mengatur dan mengubah pengaturan perangkat, dan fungsi pengancing.

Bab 2 Cara Menggunakan Perangkat dengan Berbagai Fungsi

Mempelajari cara menggunakan pengatur waktu retentif, register indeks, relai khusus, dan register khusus.

Bab 3 Memori untuk Modul CPU dan Register File

Mempelajari tipe memori yang dapat digunakan dengan modul CPU dan cara menggunakan register file.

Bab 4 Program dengan Bilangan Asli

Mempelajari penanganan bilangan asli dan operasi yang menggunakan bilangan asli.

Bab 5 Konsep Nomor I/O dan Cara Menggunakan Fungsi Penetapan I/O

Mempelajari konsep nomor I/O dan cara menggunakan fungsi penetapan I/O.

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, yang memungkinkan Anda menavigasi ke halaman yang diinginkan.
Keluar dari kursus		Keluar dari kursus. Jendela seperti jendela "Daftar Isi" dan kursus akan ditutup.

Pendahuluan **Perhatian Selama Penggunaan**



Petunjuk keselamatan

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

Petunjuk keselamatan dalam kursus ini

- Layar yang ditampilkan pada versi perangkat lunak yang Anda gunakan mungkin berbeda dengan yang ada di dalam kursus ini. Kursus ini menggunakan versi perangkat lunak berikut:
 - GX Works2 Version 1.91V

Bab 1 Pengaturan dan Modifikasi Device

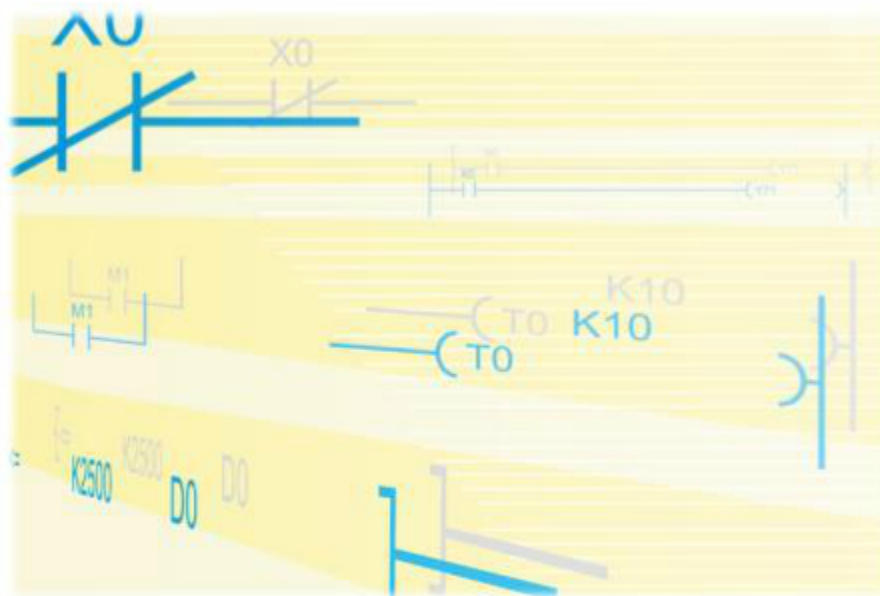
Bab ini menjelaskan cara mengubah pengaturan device.

Bagian 1.1: Menentukan Device

Bagian 1.2: Cara Menyesuaikan Jumlah device point

Bagian 1.3: Menyimpan Status device saat Daya Mati atau Reset

Bagian 1.4: Ringkasan



1.1 Menentukan Device

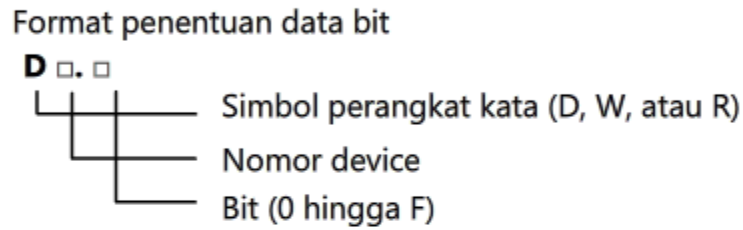
1.1.1 Penentuan Bit Word Device

Word Device biasanya ditentukan menggunakan data word, tetapi device ini juga dapat ditentukan menggunakan data bit (seperti bilangan bulat, dll.).

Data bit dapat digunakan pada word device seperti register data (D) dan register file (R).

Contoh: Register data (D)

0	0	1	0	0	1	1	0	1	1	0	0	0	1	0	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---



Contoh program

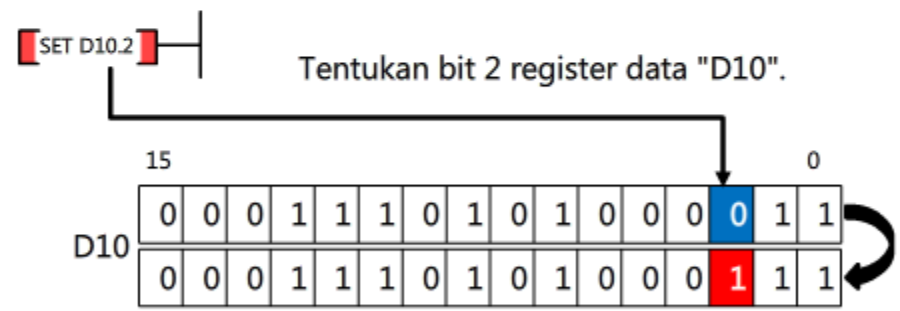
Contoh 1 Bila bit 5 register data "D0" adalah 1.



Bila bit 5 register data "D0" adalah 0.



Contoh 2

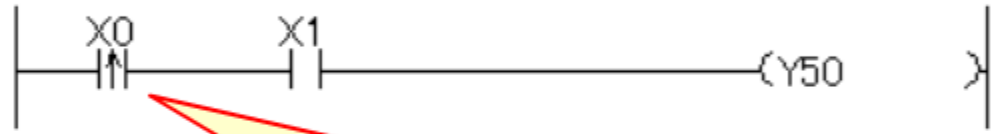


1.1.2 Penentuan kontak leading-edge atau trailing-edge

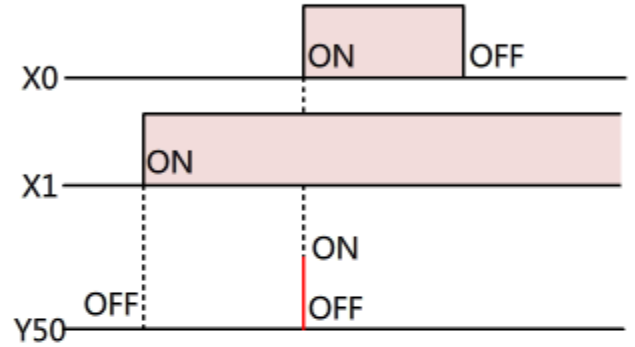
Untuk operasi ON/OFF kontak, sinyal dapat diatur agar menyala hanya selama 1 pemindaian pada kontak leading-edge atau trailing-edge.

Ini berguna untuk memrogram kondisi input sinyal leading-edge atau trailing-edge.

Contoh program untuk kontak leading-edge



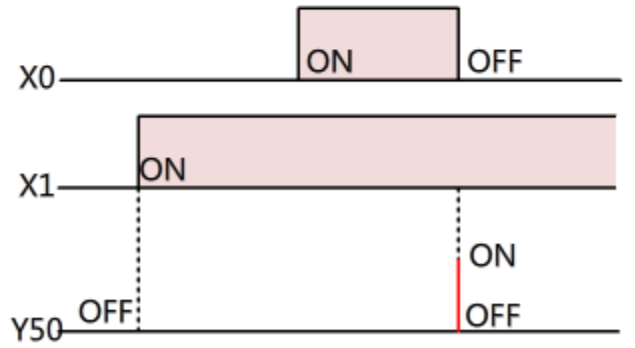
Bila kontak "X0" berubah dari OFF ke ON, tetap ON selama satu pemindaian.



Contoh program untuk kontak trailing-edge



Bila kontak "X0" berubah dari ON ke OFF, tetap ON selama satu pemindaian.



1.2

Cara Menyesuaikan Jumlah device point

Modul CPU yang berbeda memiliki jumlah device point yang berbeda. Nomor device mereka mula-mula dialokasikan sesuai kapasitas modul CPU yang digunakan.

Bila jumlah point yang dialokasikan ke device yang sering digunakan tidak mencukupi, kurangi point yang dialokasikan ke device lain dan gunakan point tersebut untuk device yang sering digunakan. Buka tab Device pada jendela PLC Parameter (Parameter PLC) untuk mengubah pengaturan.

Contoh layar pengaturan perangkat

	Sym.	Dig.	Device Points	Local Device Start	Local Device End
Input Relay	X	16	8K		
Output Relay	Y	16	8K		
Internal Relay	M	10	8K		
Latch Relay	L	10	8K		
Link Relay	B	16	8K		
Annunciator	F	10	2K		
Link Special	SB	16	2K		
Edge Relay	V	10	2K		
Step Relay	S	10	8K		
Timer	T	10	2K		
Retentive Timer	ST	10	0K		
			1K		
			12K		
			8K		
			2K		
Index	Z	10	20		

Device Points :

- Secara default, nilai awal telah diatur.
- Nilai pada sel yang berwarna putih dapat diubah.
- Atur device point dalam satuan 16 point.
- 1K point berarti 1024 point data sebenarnya.

Jika kapasitas device point yang diatur melebihi kapasitas modul CPU, pesan yang meminta agar pengaturan diubah akan ditampilkan.

Total Device Point:
Secara otomatis dikonversi ke dalam satuan kata.



Please set the total number of devices used in the sequence program so that it is 29 K words or less.

OK

Device Total	28.8	K Words
Word Device	25.0	K Words
Bit Device	44.0	K Bits

The total number of device points is up to 29 K words.

Latch(1) : Able to clear the value by using a latch clear

Jumlah maksimum device point = kapasitas modul CPU
Contoh, kapasitas modul CPU Q06UDEHCPU adalah 29K word.

When using the local devices, please do the file setting at PLC file setting parameter.

1.3

Menyimpan Status Device saat Daya Mati atau Reset

Fungsi Latch

Dengan menggunakan fungsi latch, modul CPU mempertahankan nilai device saat operasi berhenti. Misalnya, jika kegagalan daya sementara yang terjadi lebih lama daripada batas yang dibolehkan, modul CPU akan mempertahankan data saat operasi berhenti dan menggunakan data tersebut ketika me-restart kontrol sekuens.

Jika fungsi latch tidak digunakan, nilai device akan di-reset ke nilai default (perangkat bit ke OFF dan word device ke "0") pada kejadian berikut:

- (1) Daya diputus
- (2) Di-reset dengan sakelar "RUN/STOP/RESET" (JALAN/BERHENTI/RESET)
- (3) Kegagalan daya sementara yang lebih lama daripada batas yang dibolehkan pada modul suplai daya

Pengaturan jangkauan latch

Pilih tab Device pada jendela PLC Parameter (Parameter PLC) GX Works2 untuk mengatur jangkauan latch. Di bawah ini adalah contoh pengaturan untuk mengancing latch relay L0 ke L1024 dan register data D0 ke D128.

				A	B	C	D	
	Sym.	Dig.	Device Points	Latch (1) Start	Latch (1) End	Latch (2) Start	Latch (2) End	Loc
Input Relay	X	16	8K					
Output Relay	Y	16	8K					
Internal Relay	M	10	8K					
Latch Relay	L	10	8K			0	1024	
Link Relay	B	16	8K					
Annunciator	F	10	2K					
Link Special	SB	16	2K					
Edge Relay	V	10	2K					
Step Relay	S	10	8K					
Timer	T	10	2K					
Retentive Timer	ST	10	0K					
Counter	C	10	1K					
Data Register	D	10	12K			0	128	
Link Register	W	16	8K					
Link Special	SW	16	2K					
Index	Z	10	20					

A	Latch (1) Start	Tentukan nomor mulai jangkauan latch yang akan diatur.
C	Latch (2) Start	
B	Latch (1) End	Tentukan nomor selesai jangkauan latch yang akan diatur.
D	Latch (2) End	

* Lihat halaman berikutnya untuk perbedaan antara Latch (1) dan (2).

1.3 Menyimpan Status Device saat Daya Mati atau Reset

Cara menghapus data latch

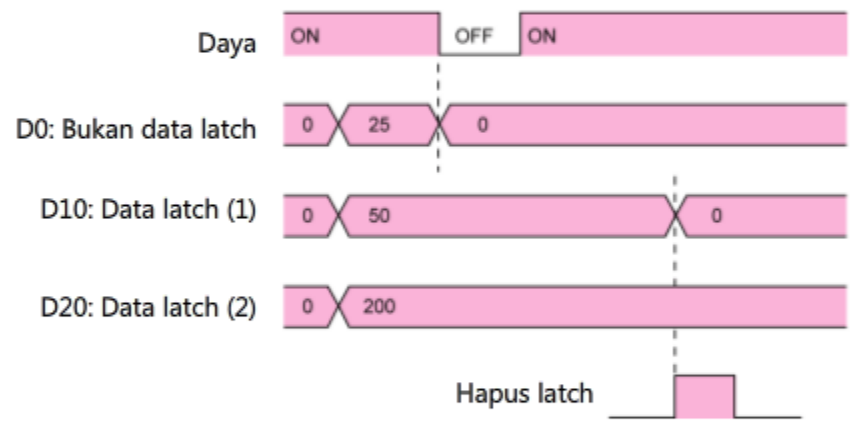
Metode penghapusan untuk latch (1) dan (2) berbeda.

Pengancing (1): Menghapus data yang dipertahankan dari jendela Remote Operation (Operasi Jarak-Jauh) GX Works2. Gunakan latch 1 bila data yang dipertahankan harus dihapus di lokasi instalasi.

Latch (2): Menghapus data yang dipertahankan melalui instruksi khusus pada program. Gunakan latch 2 bila data yang dipertahankan tidak perlu dihapus di lokasi instalasi.

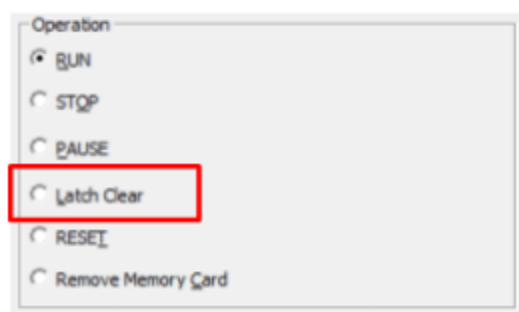
	Sym.	Dig.	Device Points	Latch (1) Start	Latch (1) End	Latch (2) Start	Latch (2) End	Loca
Input Relay	X	16	8K					
Output Relay	Y	16	8K					
Internal Relay	M	10	8K					
Latch Relay	L	10	8K			0	1024	
Link Relay	B	16	8K					
Annunciator	F	10	2K					
Link Special	SB	16	2K					
Edge Relay	V	10	2K					
Step Relay	S	10	8K					
Timer	T	10	2K					
Retentive Timer	ST	10	0K					
Counter	C	10	1K					
Data Register	D	10	12K	0	128			
Link Register	W	16	8K					
Link Special	SW	16	2K					
Index	Z	10	20					

Diagram waktu



Cara menghapus data yang dipertahankan lewat operasi jarak-jauh

Pilih Online pada bilah menu GX Works2, lalu pilih Remote Operation (Operasi Jarak-Jauh).



Dalam bab ini, Anda telah mempelajari:

- Menentukan Device
- Cara menyesuaikan jumlah device point
- Menyimpan status device saat daya mati atau reset

Hal-hal penting

Mengubah jumlah device point	<ul style="list-style-type: none">• Modul CPU yang berbeda memiliki jumlah device point yang berbeda, dan nomor device mereka mula-mula dialokasikan sesuai kapasitas modul CPU.• Bila jumlah point yang dialokasikan ke device yang sering digunakan tidak mencukupi, kurangi point yang dialokasikan ke device lain dan gunakan point tersebut untuk device yang sering digunakan.
Fungsi latch	Fungsi latch modul CPU mempertahankan nilai device saat daya terputus atau di-reset dan menggunakan data tersebut ketika operasi restart. Nilai yang dipertahankan dihapus dengan perintah hapus latch.

Bab 2**Cara Menggunakan Device dengan Berbagai Fungsi**

Bab ini menjelaskan tentang device dengan berbagai fungsi tertanam.

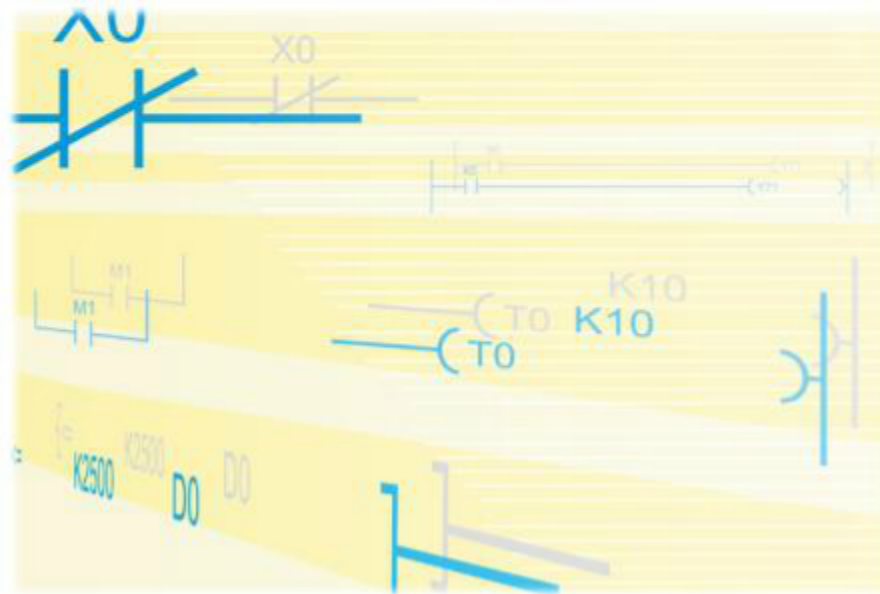
Berbeda dengan device seperti register data, yang hanya dapat menyimpan nilai, device seperti pengatur waktu retentif atau register indeks memiliki fungsi sendiri-sendiri.

Bagian 2.1: Cara Menggunakan Pengatur Waktu Retentif

Bagian 2.2: Cara Menggunakan Register Indeks

Bagian 2.3: Cara Menggunakan Relai Khusus dan Register Khusus

Bagian 2.4: Ringkasan



2.1 Cara Menggunakan Pengatur Waktu Retentif

2.1.1 Perbedaan antara pengatur waktu dan pengatur waktu retentif

Pengatur waktu dan pengatur waktu retentif sama-sama digunakan dalam program sekuens untuk operasi yang melibatkan pengukuran waktu.

* Detail tentang pengatur waktu dijelaskan dalam Kursus Dasar-dasar MELSEC Seri Q.

(a) Pengatur waktu

Pengatur waktu mengaktifkan kontak pada waktu tertentu setelah menyalakan kumparan. Setelah kumparan mati, nilai pengatur waktu di-reset ke "0". Simbol perangkat untuk pengatur waktu adalah "T".

Program ladder dan operasi

Atur sakelar ke ON/OFF untuk melihat cara kerja pengatur waktu.

3 detik setelah X0 berubah ke ON, Y70 juga akan berubah ke ON dan Y71 akan berubah ke OFF.

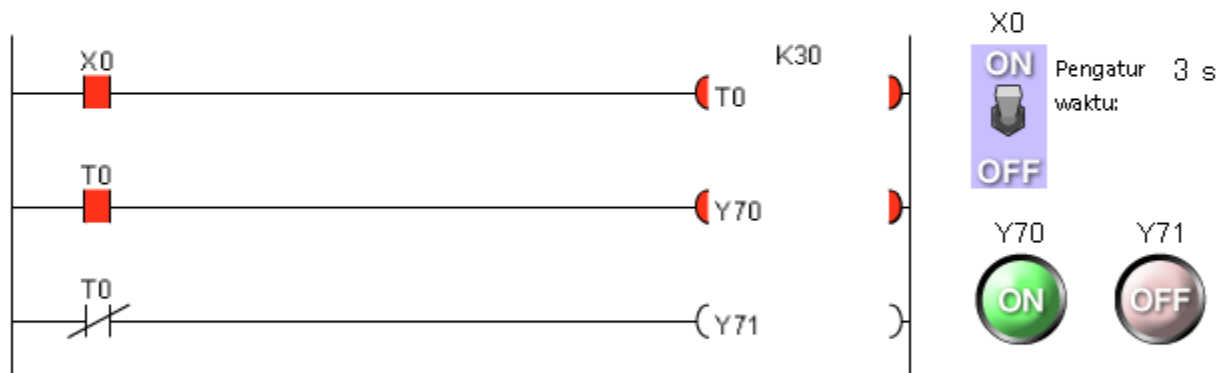
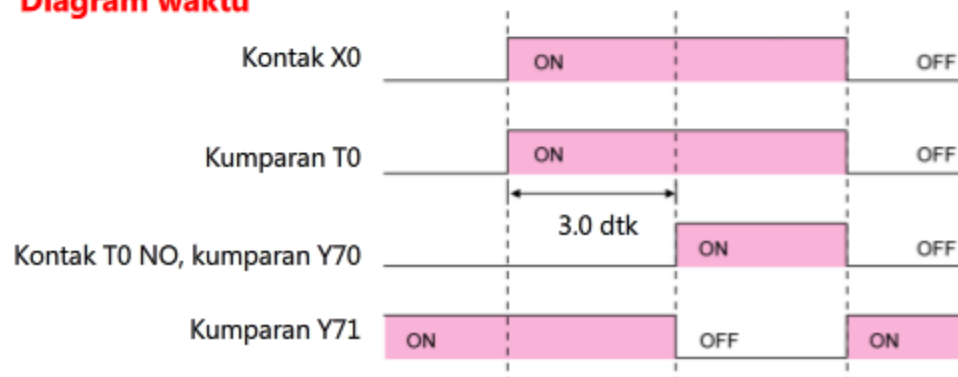


Diagram waktu



2.1.1 Perbedaan antara pengatur waktu dan pengatur waktu retentif

(b) Pengatur waktu retentif

Pengatur waktu retentif berguna untuk mengukur total waktu operasi. Pengatur waktu retentif mengaktifkan kontak (OFF ke ON) pada waktu tertentu setelah menyalakan kumparan. Setelah kumparan mati, nilai pengatur waktu tidak di-reset dan dipertahankan. Saat kumparan aktif kembali, pengatur waktu me-restart pengukuran dari nilai yang dipertahankan. Simbol device untuk pengatur waktu retentif adalah "ST".

Program ladder

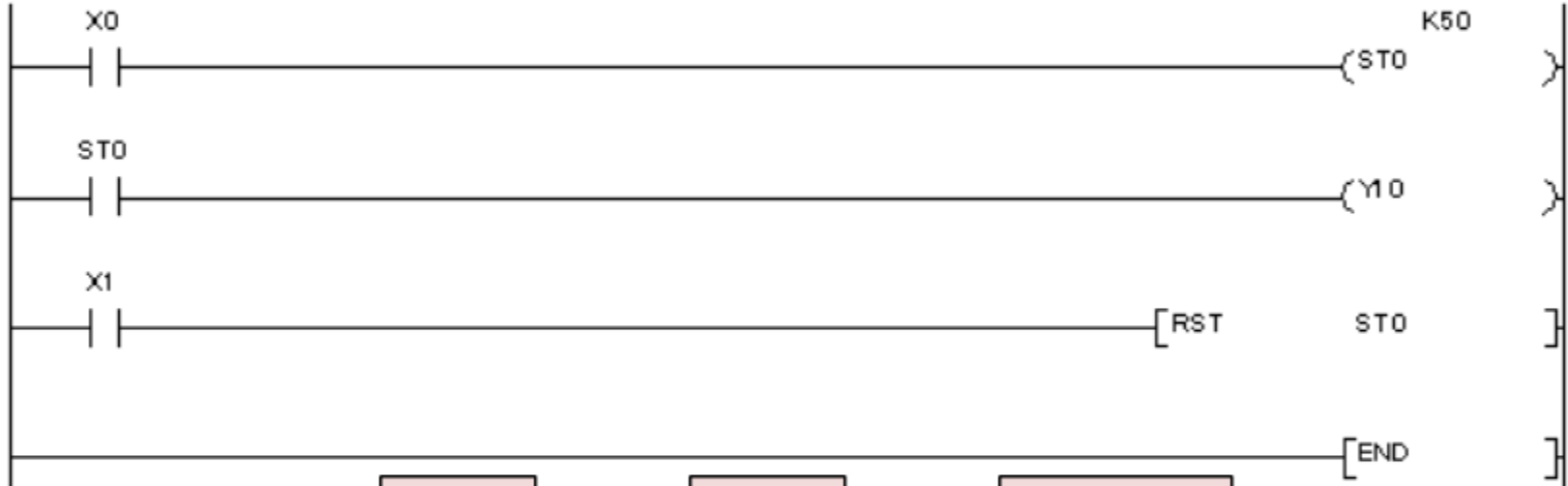
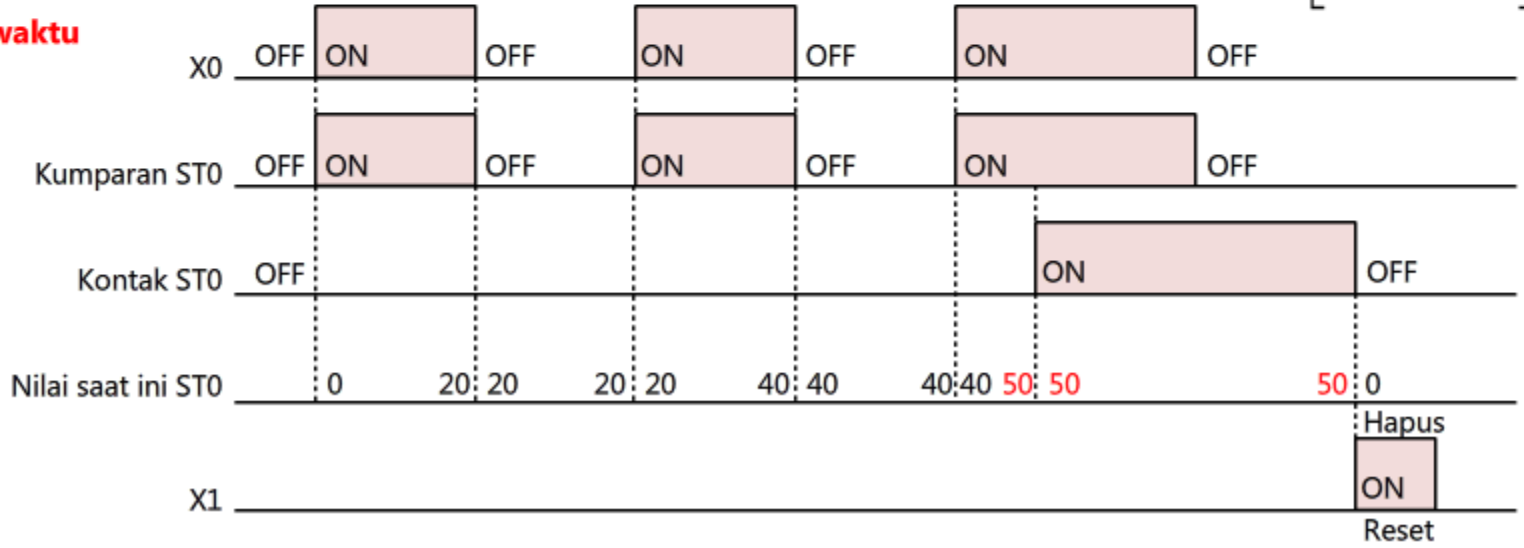


Diagram waktu



2.1.1 Perbedaan antara pengatur waktu dan pengatur waktu retentif

(b) Pengatur waktu retentif

Pengatur waktu retentif berguna untuk mengukur total waktu operasi. Pengatur waktu retentif mengaktifkan kontak (OFF ke ON) pada waktu tertentu setelah menyalakan kumparan. Setelah kumparan mati, nilai pengatur waktu tidak di-reset dan dipertahankan. Saat kumparan aktif kembali, pengatur waktu me-restart pengukuran dari nilai yang dipertahankan. Simbol device untuk pengatur waktu retentif adalah "ST".

Program ladder

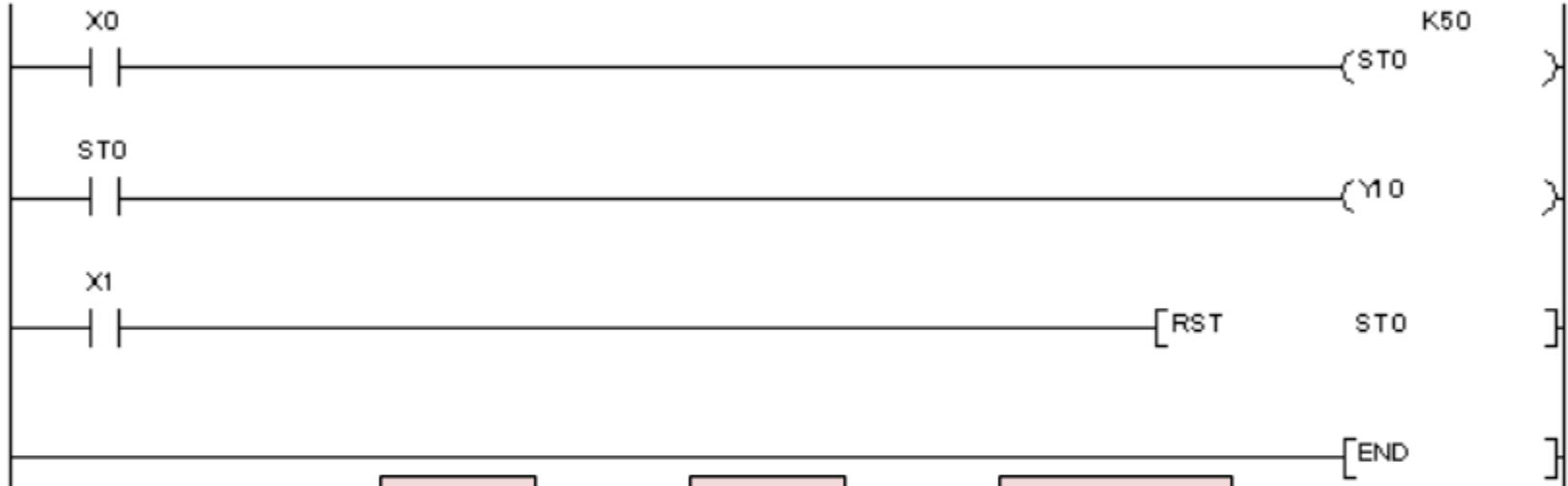
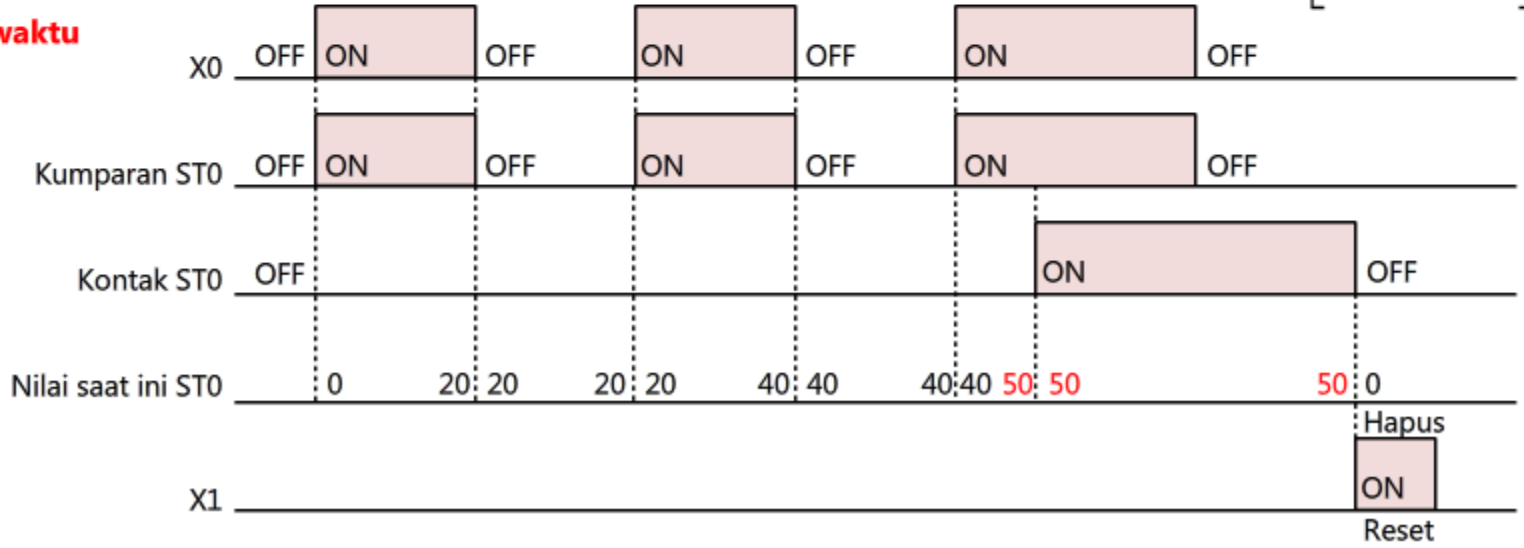


Diagram waktu



2.1.3 Persiapan menggunakan pengatur waktu retentif

Jumlah point yang digunakan oleh pengatur waktu retentif mula-mula adalah "0". Untuk menggunakan pengatur waktu retentif, beberapa point perlu dialokasikan.
 Buka jendela PLC Parameter (Parameter PLC) pada GX Works2, pilih tab Device (Perangkat), dan atur jumlah device point yang digunakan oleh pengatur waktu retentif.
 Di bawah ini adalah contoh pengaturan untuk menggunakan ST0 hingga ST63 (64 point) untuk pengatur waktu retentif.

	Sym.	Dig.	Device Points	Latch (1) Start	Latch (1) End	Latch (2) Start	Latch (2) End	Local Device Start	Local Device End
Input Relay	X	16	8K						
Output Relay	Y	16	8K						
Internal Relay	M	10	8K						
Latch Relay	L	10	8K						
Link Relay	B	16	8K						
Annunciator	F	10	2K						
Link Special	SB	16	2K						
Edge Relay	V	10	2K						
Step Relay	S	10	8K						
Timer	T	10	2K						
Retentive Timer	ST	10	64						
Counter	C	10	1K						
Data Register	D	10	12K						
Link Register	W	16	8K						
Link Special	SW	16	2K						
Index	Z	10	20						

Device Total	<input type="text" value="28.9"/>	K Words
Word Device	<input type="text" value="25.1"/>	K Words
Bit Device	<input type="text" value="44.2"/>	K Bits

The total number of device points is up to 29 K words.
 Latch(1) : Able to clear the value by using a latch clear.
 Latch(2) : Unable to clear the value by using a latch clear. Clearing will be executed by remote operation or program.
 Scan time is extended by the latch range setting (including L).
 If the latch is necessary, please set the required minimum latch range.
 When using the local devices, please do the file setting at PLC file setting parameter.

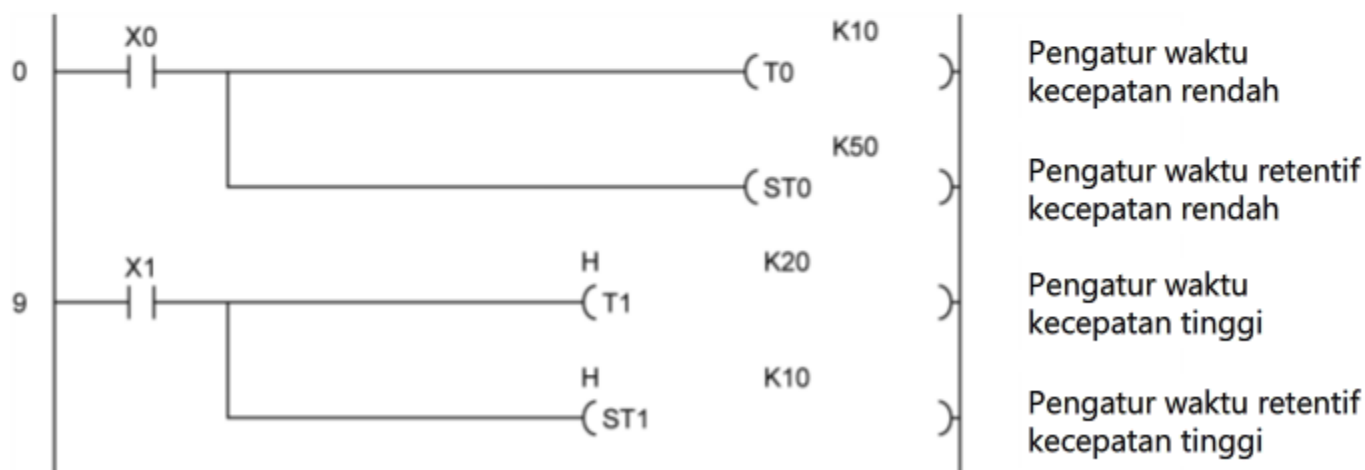
2.1.4

Perbedaan antara pengatur waktu kecepatan rendah dan pengatur waktu kecepatan tinggi

	Satuan	Contoh program	Operasi
Pengatur waktu kecepatan rendah	100 ms (default)	$\{ T0 \}^{K50}$	Pengatur waktu kecepatan rendah T0 menghitung 5 detik.
Pengatur waktu kecepatan tinggi	10 ms (default)	$\{ T1 \}^{K50}$	Pengatur waktu kecepatan tinggi T1 menghitung 0,5 detik.
Pengatur waktu retentif kecepatan rendah	100 ms (default)	$\{ ST0 \}^{K50}$	Pengatur waktu retentif kecepatan rendah ST0 menghitung 5 detik.
Pengatur waktu retentif kecepatan tinggi	10 ms (default)	$\{ ST1 \}^{K50}$	Pengatur waktu retentif kecepatan tinggi ST1 menghitung 0,5 detik.

Satuan awal untuk pengukuran waktu adalah 100 ms untuk pengatur waktu kecepatan rendah dan 10 ms untuk pengatur waktu kecepatan tinggi. Lihat halaman berikutnya untuk cara mengubah satuan.

Di bawah ini contoh program ladder yang menggunakan pengatur waktu.



2.1.4

Perbedaan antara pengatur waktu kecepatan rendah dan pengatur waktu kecepatan tinggi

Cara mengubah satuan pengatur waktu

Ubah Timer Limit Setting (Pengaturan Batas Pengatur Waktu) pada tab PLC System (Sistem PLC) di jendela PLC Parameter (Parameter PLC).

Di bawah ini adalah contoh pengaturan pada layar PLC System (Sistem PLC).

Timer Limit Setting	
Low Speed	<input type="text" value="100"/> ms (1ms--1000ms)
High Speed	<input type="text" value="10.00"/> ms (0.01ms--100ms)

2.2

Cara Menggunakan Register Indeks

Register indeks "Z", dalam kombinasi dengan device lain, menentukan (mengindeks) nomor device dari device yang akan dikontrol. Register indeks berguna untuk menyederhanakan program karena dapat mendeskripsikan beberapa device secara serempak.

•Apabila digunakan, register indeks akan ditulis setelah simbol device dan nomor device seperti terlihat di bawah untuk menunjukkan device target kontrol sebenarnya.

Device target kontrol sebenarnya = simbol device (nomor device + register indeks)

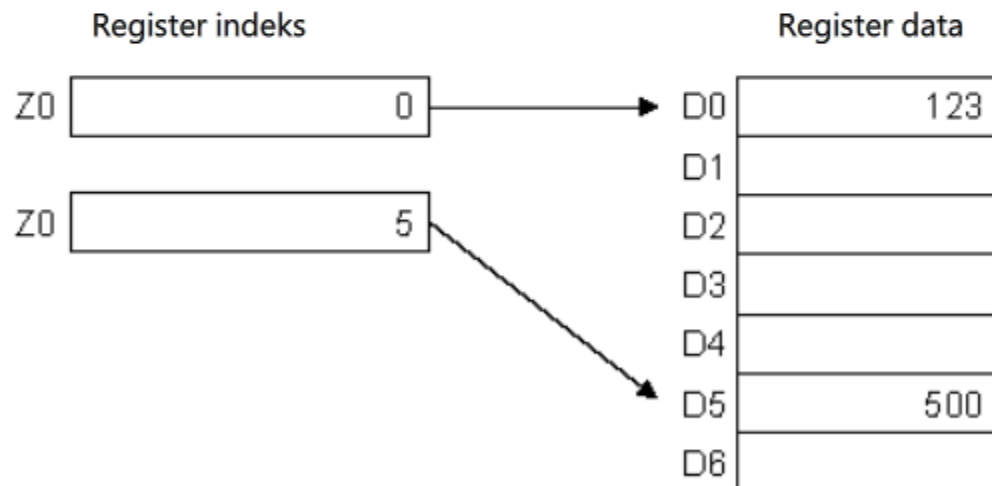
•16 point, dari Z0 hingga Z15, dapat digunakan untuk register indeks.

Contoh register indeks

Apabila device dinyatakan sebagai "D0Z0", itu berarti D (0 + Z0), sehingga nomor device adalah "0 + (nilai Z0)".

Contoh: Bila Z0 = 0, nomor device adalah D0.

Bila Z0 = 5, nomor device adalah D5.



2.2

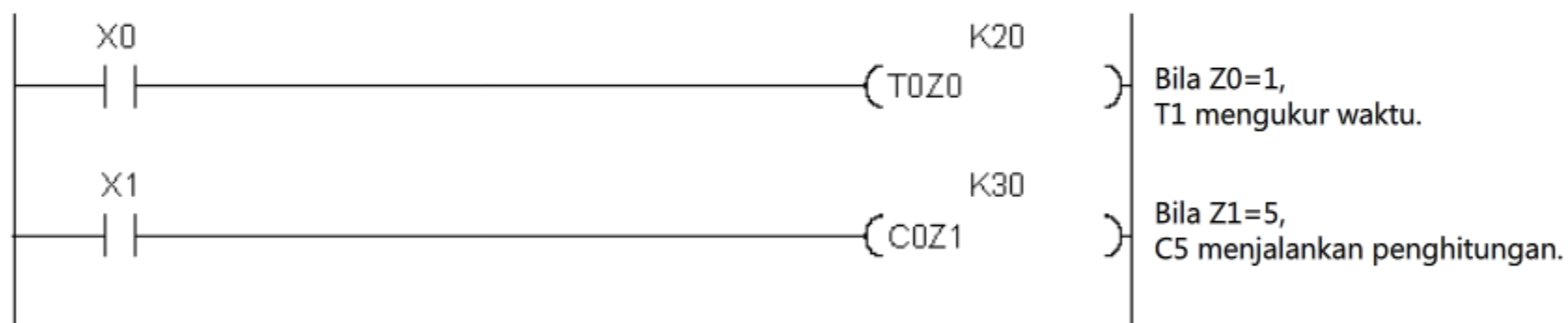
Cara Menggunakan Register Indeks

Device yang dapat diindeks dengan register indeks

Device berikut ini dapat diindeks menggunakan register indeks:

Bit Device	X, Y, M, L, S, B, F
Word Device	T, C, D, R, W
Konstanta	K, H
Penunjuk	P

Catatan: Untuk kontak dan kumparan yang digunakan pada pengatur waktu dan penghitung, hanya register indeks Z0 dan Z1 yang tersedia.



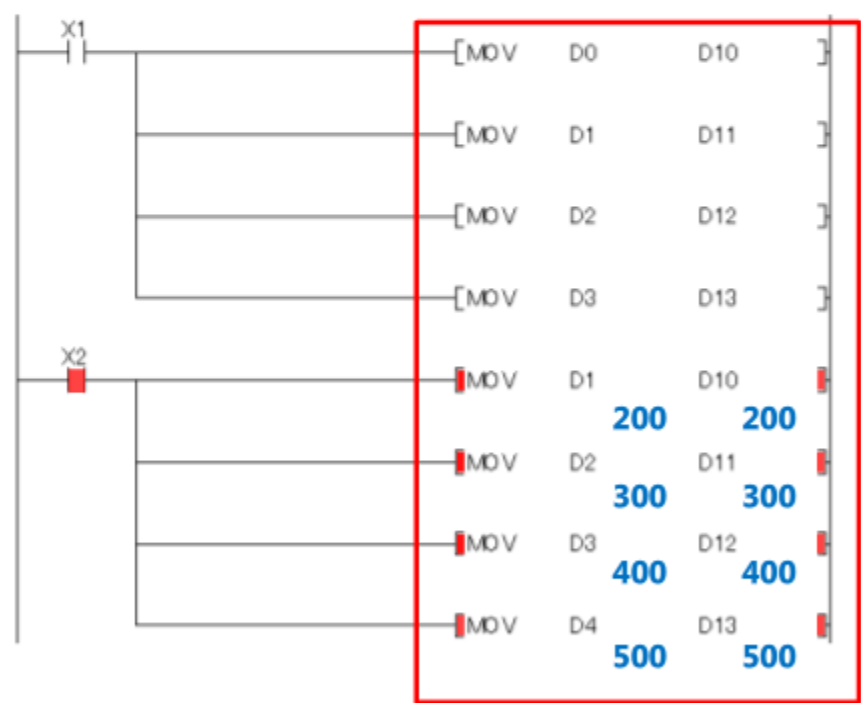
2.2 Cara Menggunakan Register Indeks

Penyederhanaan program menggunakan register indeks

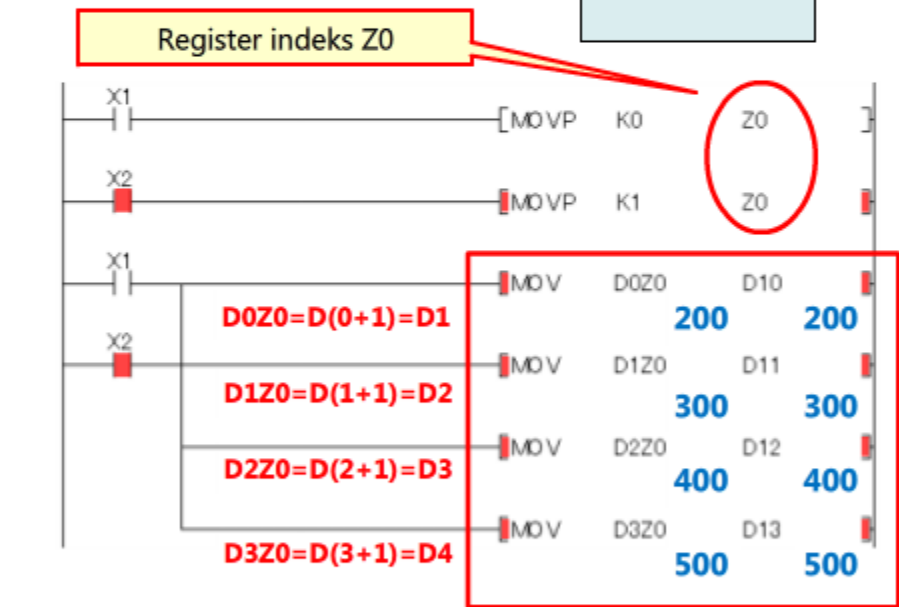
Program yang ditunjukkan di bawah ini mentransfer nilai pada "D0 hingga D4" ke "D10 hingga D13" bila X1 atau X2 berubah ke ON. Program (1) dan (2) akan memberikan hasil yang sama. Pada program (1), data ditransfer secara langsung. Pada program (2), data ditransfer lewat register indeks.

Nilai awal
 D0=100
 D1=200
 D2=300
 D3=400
 D4=500

(1) Contoh tanpa register indeks



(2) Contoh dengan register indeks

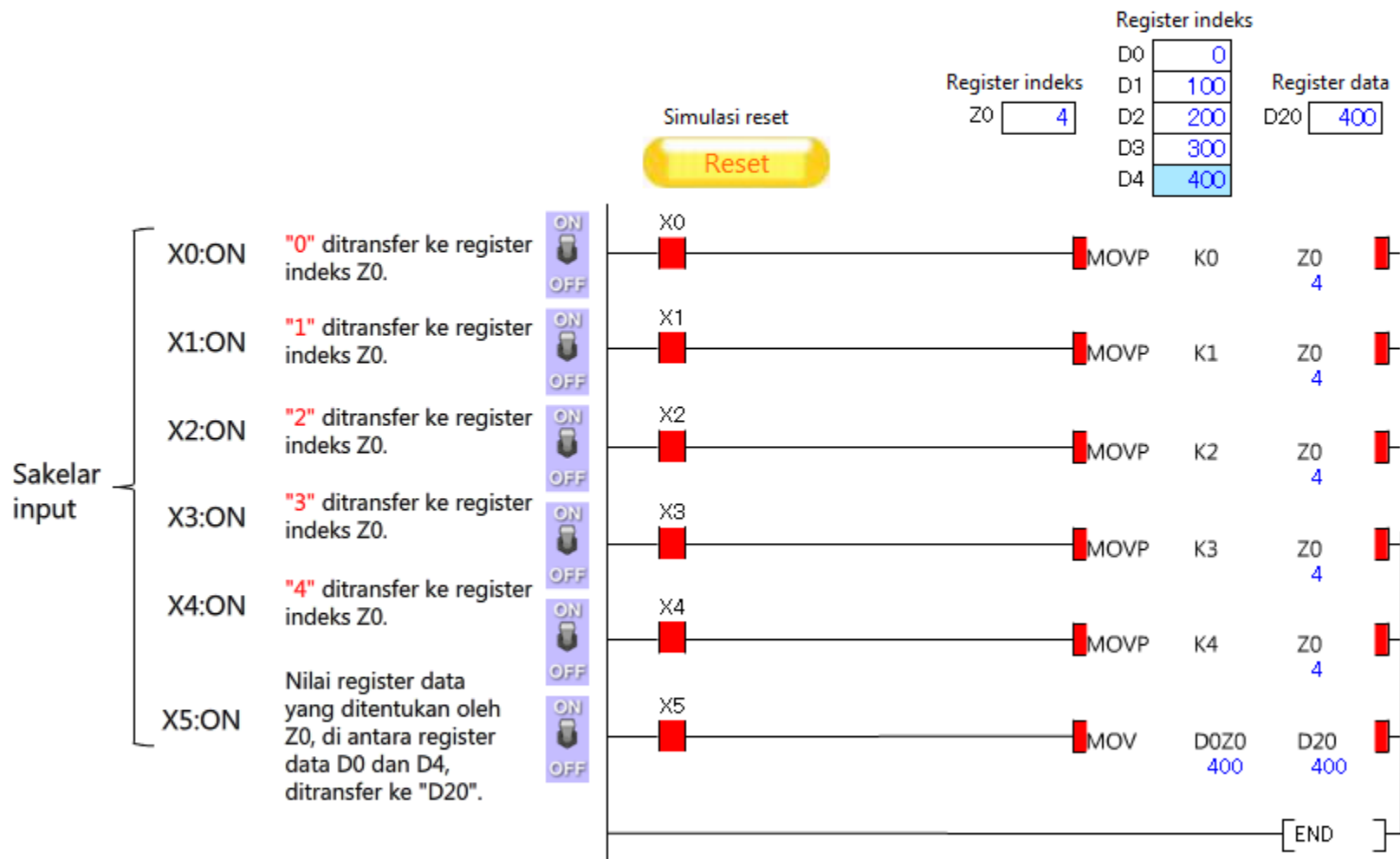


Penyederhanaan program

2.2.1 Operasi register indeks

Klik sakelar input X0 hingga X5 untuk melihat cara kerja register indeks Z0.

*K0 hingga K400 sudah disimpan di register data D0 hingga D4.



2.3

Cara Menggunakan Relai Khusus dan Register Khusus

Relai khusus dan register khusus yang digunakan dalam modul CPU memiliki fungsi dan operasi yang telah ditentukan. Relai internal yang digunakan untuk informasi bit (ON/OFF) disebut "relai khusus (SM)", dan register internal yang digunakan untuk informasi kata disebut "register khusus (SD)".

Pada program, keduanya digunakan sebagai kondisi penilaian untuk operasi. Keduanya juga digunakan sebagai kondisi pemantauan, yang dapat ditentukan pada monitor perangkat GX Works2.

Relai khusus dan register khusus dikelompokkan berdasarkan tipenya sebagai berikut.

Informasi diagnostik: SM0 hingga 199, SD0 hingga 199

Menyimpan hasil diagnostik modul CPU.

Berbagai kesalahan diagnostik dan kode kesalahan

Informasi sistem: SM200 hingga 399, SD200 hingga 399

Menyimpan informasi sistem modul CPU.

Informasi modul CPU, data jam, dll.

Jam/penghitung sistem: SM400 hingga 499, SD400 hingga 499

Menyimpan sinyal jam dan nilai hitungan yang digunakan sebagai dasar elemen pewaktuan.

Berbagai sinyal jam

Informasi pemindaian: SM500 hingga 599, SD500 hingga 599

Menyimpan informasi eksekusi pemindaian program.

Berbagai informasi waktu pindai

Informasi kartu memori: SM600 hingga 699, SD600 hingga 699

Menyimpan informasi kartu seperti penggunaan kartu memori dan register file.

Kartu memori aktif/nonaktif

Informasi instruksi: SM700 hingga 799, SD700 hingga 799

Menyimpan status eksekusi dan informasi kontrol terkait instruksi khusus.

Bendera eksekusi instruksi

Informasi debug: SM800 hingga 899, SD800 hingga 899

Menyimpan informasi terkait debugging.

Pemantauan status jejak

Dalam bab ini, Anda telah mempelajari:

- Cara menggunakan pengatur waktu retentif
- Cara menggunakan register indeks
- Cara menggunakan relai khusus dan register khusus

Hal-hal penting

Penggunaan pengatur waktu retentif	<ul style="list-style-type: none">• Untuk menggunakan pengatur waktu retentif, beberapa titik harus dialokasikan di jendela PLC Parameter (Parameter PLC).• Waktu yang diukur (nilai saat ini) dan status kontak (ON/OFF) pengatur waktu retentif tidak dihapus sekalipun kondisi berubah menjadi tidak memenuhi kondisi input setelah timeout (lewat waktu).• Program memerlukan ladder untuk me-reset pengatur waktu retentif. (Instruksi RST digunakan.)
Penggunaan register indeks	<ul style="list-style-type: none">• Register indeks "Z" ditunjukkan setelah perangkat yang digunakan pada program. Misalnya, "D0Z5".• 16 point, dari Z0 hingga Z15, tersedia untuk register indeks.
Fungsi relai khusus dan register khusus	Relai khusus dan register khusus digunakan untuk menunjukkan kondisi internal modul CPU, termasuk informasi diagnostik dan informasi sistem.

Bab 3 Memori untuk Modul CPU dan Register File

Bab ini menjelaskan tentang memori yang tersedia untuk modul CPU dan cara menggunakan register file.

Bagian 3.1: Memori untuk modul CPU

Bagian 3.2: Cara Menggunakan Register File

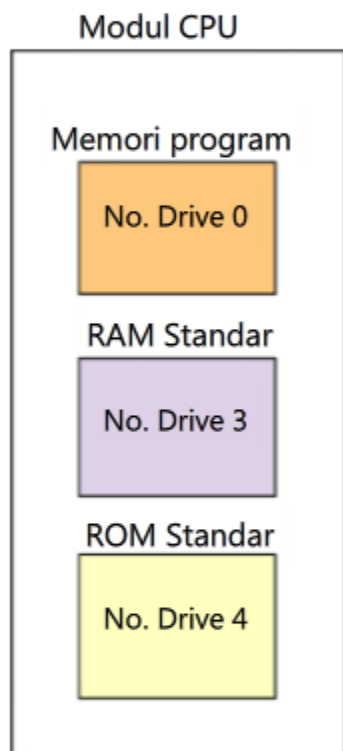
Bagian 3.3: Ringkasan



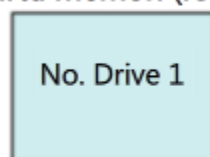
3.1

Memori untuk Modul CPU

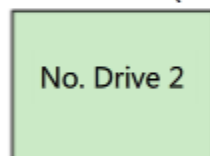
Modul CPU dapat menggunakan dua tipe memori, satu yang terpasang di dalam modul CPU, dan sebuah kartu memori yang dapat disisipkan ke dalam slot modul CPU. Untuk memastikan akses modul CPU oleh GX Works2, nomor drive, yang menunjukkan tipe memori target, harus ditentukan dengan benar.



Kartu memori (RAM)



Kartu memori (ROM)



Tipe memori		Tipe data disimpan	Kondisi data saat daya terputus	Pemformatan memori untuk penggunaan pertama
Modul CPU	Memori program	•Parameter •Program	Dipertahankan menggunakan baterai modul CPU	Diperlukan (Gunakan GX Works2)
	RAM Standar	•Register file •Perangkat lokal		
	ROM Standar	•Parameter •Program	Dipertahankan tanpa menggunakan baterai	Tidak diperlukan
Kartu memori	RAM	•Parameter •Program •Register file •Perangkat lokal	Dipertahankan menggunakan baterai kartu memori	Diperlukan (Gunakan GX Works2)
	ROM	•Parameter •Program •Register file	Dipertahankan tanpa menggunakan baterai	Kartu flash tidak diperlukan. Kartu ATA diperlukan (Gunakan GX Works2).

- Program yang disimpan di ROM standar atau kartu memori di-boot (dimuat) ke memori program modul CPU dan dijalankan ketika modul CPU dimulai.
- Bila register file disimpan di RAM standar, kecepatan akses ke register file sama cepatnya dengan mengakses register data (D).
- Bila RAM standar digunakan, mematikan daya tanpa baterai cadangan akan menghapus data yang disimpan di RAM.
- Secara umum, RAM baca/tulis kecepatan tinggi digunakan untuk menyalakan sistem dan ROM digunakan untuk operasi sistem kontinu.

3.2 Cara Menggunakan Register File

Bagian ini menjelaskan tentang pengaturan yang menetapkan register file lokal sebagai destinasi penyimpanan. Di jendela PLC Parameter (Parameter PLC), pilih tab PLC File (File PLC). Lalu, pilih "Use the same file name as the program" (Gunakan nama file yang sama dengan program) untuk File Register (Register File) dan tentukan memori destinasi penyimpanan.

Catat bahwa kartu memori diperlukan untuk pengaturan ini. (RAM standar hanya dapat menyimpan satu register file.)

File Register

Not Used

Use the same file name as the program

Corresponding Memory

Use the following file

Corresponding Memory

File Name

Capacity K Points
(1K--4086K Points)

Transfer to Standard ROM at Latch data backup operation.

Following settings are available in device setting when select "Use the following file" and specify capacity.

- Change of latch(2) of file register.
- Assignment to expanded data register/expanded link register of part of file register area.

Untuk "Corresponding Memory" (Memori yang Terkait), pilih "Memory card (RAM)" (Kartu memori (RAM)).

Pengaturan register file harus dibuat untuk setiap program. Pengaturan ini akan ditulis ke modul CPU di PLC write.

Dalam bab ini, Anda telah mempelajari:

- Memori untuk modul CPU
- Cara menggunakan register file

Hal-hal penting

Penggunaan register file

Untuk menggunakan register file, RAM standar modul CPU atau kartu memori harus dipilih sebagai destinasi penyimpanan data. Untuk pengaturan, buka tab PLC File (File PLC) pada jendela PLC Parameter (Parameter PLC). Register file mempertahankan data bahkan saat daya mati (OFF).

Bab 4 Program dengan Bilangan Asli

Bab ini menjelaskan bagaimana program menangani bilangan asli dan petunjuk pengoperasian.

- Bagian 4.1: Aplikasi dan Notasi Bilangan Asli
- Bagian 4.2: Petunjuk Pengoperasian Bilangan Asli
- Bagian 4.3: Perintah Konversi antara Bilangan Bulat dan Bilangan Asli
- Bagian 4.4: Ringkasan



Aplikasi bilangan asli

- "Bilangan asli" adalah nilai numerik yang meliputi angka desimal.
- Program sekuens biasanya dikonfigurasi menggunakan bilangan bulat. Namun, bilangan asli dengan desimal diperlukan dalam program untuk operasi aritmetika lanjutan seperti fungsi trigonometri dan operasi perpangkatan.
- Data numerik bilangan asli disebut "data titik mengambang".

Perhatian

- Satu bilangan asli **selalu menggunakan dua word device berurutan** (menempati ruang memori 32 bit), berapa pun ukuran bilangan tersebut.
- Pada program sekuens, **petunjuk pengoperasian khusus** (penambahan, pengurangan, perkalian, pembagian, fungsi khusus, dll.) yang menangani bilangan asli tersedia. Perintah konversi, misalnya antara bilangan asli dan bilangan bulat, juga tersedia.

Notasi untuk bilangan asli

"E" digunakan untuk menyatakan bilangan asli.

(1) Menyatakan konstanta dengan bilangan asli

Untuk menulis sebuah konstanta, mulailah dengan "E".

Bentuk normal	Tuliskan nilai numerik apa adanya. (Contoh) 10.2345 sebagai "E10.2345".
Bentuk perpangkatan	Tuliskan nilai numerik sebagai "(nilai numerik) x 10 ⁿ ". (Contoh) 1234.0 sebagai "E1.234+3".

(2) Perintah dengan bilangan asli

Tambahkan "E" di depan perintah.

Misalnya, perintah transfer adalah "EMOV", dan perintah penambahan atau pengurangan adalah "E+" atau "E-".

4.2

Petunjuk Pengoperasian Bilangan Asli

4.2.1

Perintah penambahan dan pengurangan

Kode perintah	Contoh ladder	
E+ (Penambahan)	 Operasi bilangan asli "D + S = D" dijalankan.	 Operasi bilangan asli "S1 + S2 = D" dijalankan.
E- (Pengurangan)	 Operasi bilangan asli "D - S = D" dijalankan.	 Operasi bilangan asli "S1 - S2 = D" dijalankan.

S (sumber): Data sebelum operasi (konstanta, nomor perangkat)

D (destinasi): Destinasi data setelah operasi (nomor perangkat)

P: Perintah untuk dijalankan pada leading-edge

S1 dan S2: Dua item data yang akan dioperasikan.

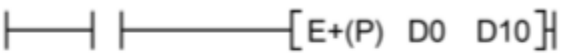
Catatan:

Pada operasi bilangan asli, S1, S2, dan D pada ladder semuanya harus berupa bilangan asli.

Bilangan bulat dan bilangan asli tidak dapat dicampur untuk sebuah operasi.

4.2.1 Perintah penambahan dan pengurangan

Contoh program dengan perintah penambahan

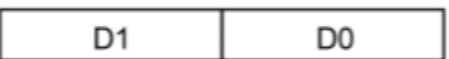


Bilangan asli floating point (32 bit)



2.54

Bilangan asli floating point (32 bit)

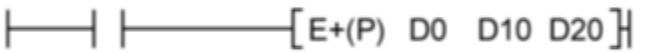


10.55

Bilangan asli floating point (32 bit)



13.09

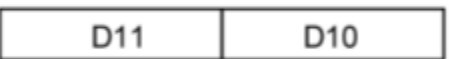


Bilangan asli floating point (32 bit)



1000.000

Bilangan asli floating point (32 bit)



3.140

Bilangan asli floating point (32 bit)

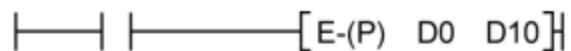


1003.140

4.2.1

Perintah penambahan dan pengurangan

Contoh program dengan perintah pengurangan



Bilangan asli floating point (32 bit)



1000.000

Bilangan asli floating point (32 bit)

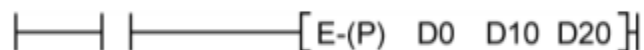


320.560

Bilangan asli floating point (32 bit)



679.440



Bilangan asli floating point (32 bit)



2.540

Bilangan asli floating point (32 bit)



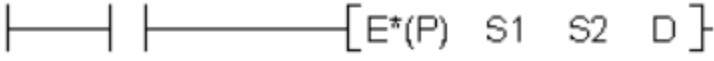
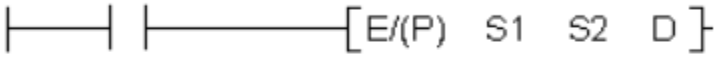
10.550

Bilangan asli floating point (32 bit)



-8.010

4.2.2 Perintah perkalian dan pembagian

Kode perintah	Contoh ladder
E* (Perkalian)	 Operasi bilangan asli "S1 * S2 = D" dijalankan.
E/ (Pembagian)	 Operasi bilangan asli "S1 / S2 = D" dijalankan.

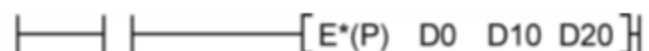
S1, S2 (sumber): Dua item data yang akan dioperasikan
 D (destinasi): Destinasi data setelah operasi (nomor perangkat)
 P: Perintah untuk dijalankan pada tepi depan

Catatan:

Pada operasi bilangan asli, S1, S2, dan D pada ladder semuanya harus berupa bilangan asli. Bilangan bulat dan bilangan asli tidak dapat dicampur untuk sebuah operasi.

4.2.2 Perintah perkalian dan pembagian

Contoh program dengan perintah perkalian



Bilangan asli floating point (32 bit)



1000.000

×

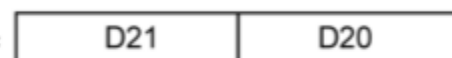
Bilangan asli floating point (32 bit)



25.590

=

Bilangan asli floating point (32 bit)



25590.000

Contoh program dengan perintah pembagian



Bilangan asli floating point (32 bit)



1000.000

÷

Bilangan asli floating point (32 bit)



25.590

=





Bilangan asli floating point (32 bit)



39.078

4.3

Perintah Konversi antara Bilangan Bulat dan Bilangan Asli

Kode perintah	Contoh ladder	
FLT (Konversi bilangan bulat ke bilangan asli)	Bilangan bulat (16 bit) dikonversi menjadi bilangan asli (32 bit). 	Bilangan bulat (32 bit) dikonversi menjadi bilangan asli (32 bit). 
INT (Konversi bilangan asli ke bilangan bulat)	Bilangan asli (32 bit) dikonversi menjadi bilangan bulat (16 bit). 	Bilangan asli (32 bit) dikonversi menjadi bilangan bulat (32 bit). 

S (sumber): Data sebelum operasi (konstanta, nomor perangkat)

D (destinasi): Destinasi data setelah operasi (nomor perangkat)

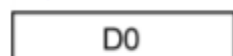
4.3

Perintah Konversi antara Bilangan Bulat dan Bilangan Asli

Contoh program dengan perintah konversi bilangan bulat (16 bit) / bilangan asli (32 bit)

┌───┐ ┌──────────┐ [FLT(P) D0 D10]

Bilangan bulat (16 bit)



30000



Bilangan asli floating point (32 bit)



30000.000

Contoh program dengan perintah konversi bilangan bulat (32 bit) / bilangan asli (32 bit)

┌───┐ ┌──────────┐ [DFLT(P) D0 D10]

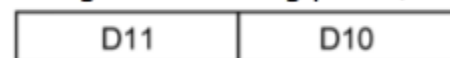
Bilangan bulat (32 bit)



90000



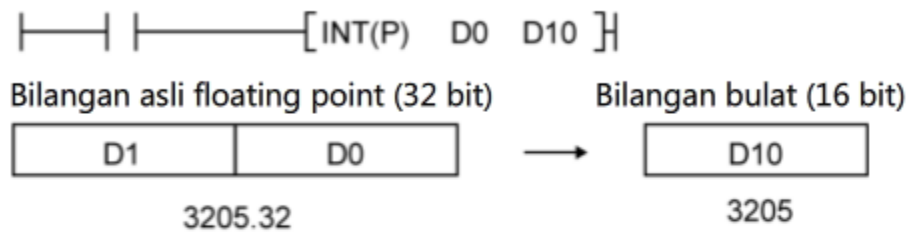
Bilangan asli floating point (32 bit)



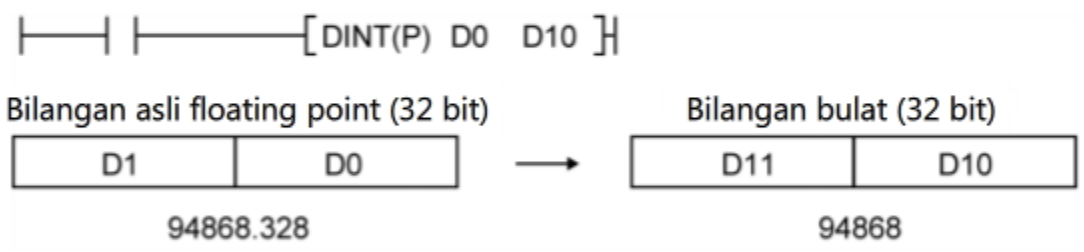
90000.000

4.3 Perintah Konversi antara Bilangan Bulat dan Bilangan Asli

Contoh program dengan perintah konversi bilangan asli (32 bit) / bilangan bulat (16 bit)



Contoh program dengan perintah konversi bilangan asli (32 bit) / bilangan bulat (32 bit)



4.4

Ringkasan



Dalam bab ini, Anda telah mempelajari:

- Aplikasi dan notasi bilangan asli
- Petunjuk pengoperasian bilangan asli
- Perintah konversi antara bilangan bulat dan bilangan asli

Hal-hal penting

Operasi bilangan asli

- Data bilangan asli menggunakan memori 2 word (32 bit).
- Tambahkan E di depan petunjuk pengoperasian bilangan asli, misalnya E* (perkalian).
- Bilangan bulat dan bilangan asli tidak dapat diproses bersama-sama. Bilangan bulat harus dikonversi menjadi bilangan asli sebelum memproses operasi aritmetika.

Bab 5**Konsep Nomor I/O dan Cara Menggunakan Fungsi Penetapan I/O**

Bab ini menjelaskan tentang konsep nomor I/O dan cara menggunakan fungsi penetapan I/O.

Bagian 5.1: Konsep Nomor I/O

Bagian 5.2: Nomor I/O untuk Unit Dudukan Ekstensi

Bagian 5.3: Pemeriksaan Penetapan Nomor I/O pada Monitor Sistem

Bagian 5.4: Cara Menggunakan Fungsi Penetapan I/O

Bagian 5.5: Ringkasan



5.1

Konsep Nomor I/O

Nomor I/O ditetapkan ke modul I/O pada base unit seperti ditunjukkan di bawah ini.

(Ada tiga tipe modul I/O: tipe 16, 32, dan 64 titik. Contoh di bawah ini menggunakan modul I/O tipe 16 point.)

	0	1	2	3	4	Nomor slot
Modul suplai daya	0	10	20	30	40	Nomor I/O
Modul CPU	hingga	hingga	hingga	hingga	hingga	
	F	1F	2F	3F	4F	

(Contoh) Unit dudukan Q35B dengan lima slot I/O

Nomor I/O (heksadesimal 0 hingga F) ditetapkan ke setiap slot (modul) secara berurutan, mulai dari slot yang paling dekat dengan modul CPU. Setiap slot (modul) secara default diberi 16 nomor I/O.

5.1 Konsep Nomor I/O

Bila modul I/O 16, 32, dan 64 point digunakan bersama-sama, nomor I/O ditetapkan sebagai berikut:

		0	1	2	3	4	Nomor slot
Modul suplai daya	Modul CPU	Tipe 16 point	Tipe 32 point	Tipe 64 point	Tipe 32 point	Tipe 16 point	
		0	10	30	70	90	Nomor I/O
		hingga	hingga	hingga	hingga	hingga	
		F	2F	6F	8F	9F	

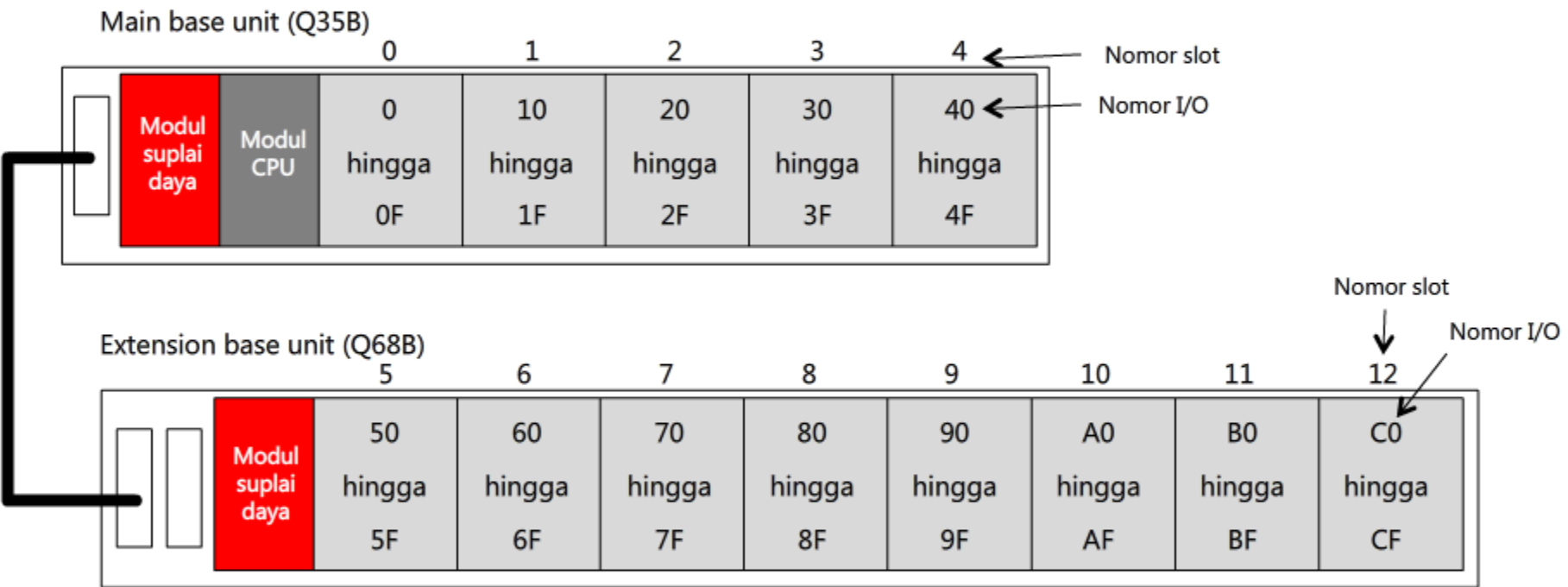
Jika ada slot kosong di tengah unit dudukan, nomor I/O juga ditetapkan ke slot tersebut. (Pada pengaturan awal.)

		0	1	2	3	4	Nomor slot
Modul suplai daya	Modul CPU	Tipe 16 point	Tipe 32 point	Tipe 64 point	Slot kosong	Tipe 16 point	
		0	10	30	70	80	Nomor I/O
		hingga	hingga	hingga	hingga	hingga	
		F	2F	6F	7F	8F	

CATANAN: 16 nomor I/O (heksadesimal) ditetapkan ke slot kosong secara default. Tetapi, pengaturan ini dapat diubah, dan nomor I/O dalam rentang 0 hingga 64 dapat diatur dalam satuan 16 point.

5.2 Nomor I/O untuk Base Unit Ekstensi

Setiap nomor I/O modul, yang terkait dengan relai input (X)/output (Y) modul CPU, ditetapkan secara otomatis dengan mendeteksi modul pada base unit.
 Nomor I/O modul pada base unit ekstensi juga ditetapkan secara otomatis, setelah nomor I/O terakhir pada main base unit.
 Gambar berikut menunjukkan bagaimana nomor I/O ditetapkan menggunakan modul 16 point.



5.3 Pemeriksaan Penetapan Nomor I/O pada Monitor Sistem

Untuk memeriksa penetapan nomor I/O, buka menu pada GX Works2, pilih Diagnostics (Diagnostik), lalu pilih System Monitor (Monitor Sistem).

The screenshot shows the 'System Monitor' window in GX Works2. It includes a 'Main Base' section with a rack view and an 'I/O ADR.' field showing '0000 0010 0050 0090 00A0 00B0 00C0 00D0'. A 'Base Information List' table is visible, along with a 'Module Information List (Main Base)' table. Three red callout boxes provide instructions: (1) 'Pilih base unit yang ingin Anda periksa.' pointing to the 'Main Base' row in the Base Information List; (2) 'Periksa nomor I/O mulai modul pada base unit yang dipilih.' pointing to the 'I/O ADR.' field; and (3) 'Periksa nomor I/O mulai modul pada base unit yang dipilih.' pointing to the 'I/O Address' column in the Module Information List.

(1) Pilih base unit yang ingin Anda periksa.

(2) Periksa nomor I/O mulai modul pada base unit yang dipilih.

(3) Periksa nomor I/O mulai modul pada base unit yang dipilih.

Base	Module	Base Model Name	Power Supply	Base Type	Slots	Installed Modules
		Main Base	Exist	Q	8	4
		Extension Base1				
		Extension Base2				
		Extension Base3				
		Extension Base4				
		Extension Base5				
		Extension Base6				
		Extension Base7				
Overall	1Base					4Module

Status	Base-Slot	Series	Model Name	Point	Parameter		I/O Address	Network No. Station No.	Master PLC
					Type	Point			
	-	-	Power	-	Power	-	-	-	-
		Q	Q06UDEHCPU	-	CPU	-	-	-	-
	0-0	-	Empty	-	Empty	16Point	0000	-	-
	0-1	Q	QX42	64Point	Input	64Point	0010	-	-
	0-2	Q	QY42P	64Point	Output	64Point	0050	-	-
	0-3	Q	Q64AD	16Point	Intelli.	16Point	0090	-	-
	0-4	Q	Q62DAN	16Point	Intelli.	16Point	00A0	-	-
	0-5	-	Empty	-	Empty	16Point	00B0	-	-
	0-6	-	Empty	-	Empty	16Point	00C0	-	-
	0-7	-	Empty	-	Empty	16Point	00D0	-	-

5.4

Cara Menggunakan Fungsi Penetapan I/O

Fungsi penetapan I/O menetapkan nomor I/O tetap ke slot base unit, bukan ke modul yang terpasang. Itu berarti penetapan kembali nomor I/O tidak diperlukan untuk modul yang telah ada sekalipun penataan sistem berubah (misalnya, ketika ada modul baru yang ditambahkan).

(1) Tanpa fungsi penetapan I/O

Penataan sistem tanpa modul baru

Modul suplai daya	Modul CPU	Modul input	Modul output	Modul fungsi cerdas	
		64 point	64 point	16 point	
		X00 hingga X3F	Y40 hingga Y7F	X/Y80 hingga X/Y8F	

Penataan sistem dengan modul baru (modul input 32 point dan modul output 16 point ditambahkan)

Modul suplai daya	Modul CPU	Modul input	Modul input	Modul output	Modul output	Modul fungsi cerdas
		64 point	32 point	64 point	16 point	16 point
		X00 hingga X3F	X40 hingga X5F	Y60 hingga Y9F	YA0 hingga YAF	X/YB0 hingga X/YBF

Nomor I/O perlu ditetapkan kembali karena ada modul baru.

5.4

Cara Menggunakan Fungsi Penetapan I/O

(2) Dengan fungsi penetapan I/O

Penataan sistem tanpa modul baru

Modul suplai daya	Modul CPU	Modul input	Modul output	Modul fungsi cerdas	
		64 point	64 point	16 point	
		X00 hingga X3F	Y40 hingga Y7F	X/Y80 hingga X/Y8F	

Penataan sistem dengan modul baru (modul input 32 point dan modul output 16 point ditambahkan)

Modul baru

Modul suplai daya	Modul CPU	Modul input	Modul input	Modul output	Modul output	Modul fungsi cerdas
		64 point	32 point	64 point	16 point	16 point
		X00 hingga X3F	X90 hingga XAF	Y40 hingga Y7F	YB0 hingga YBF	X/Y80 hingga X/Y8F

Karena nomor I/O untuk modul yang telah ada tidak berubah, hanya program untuk modul tambahan yang perlu dimodifikasi.

5.4 Cara Menggunakan Fungsi Penetapan I/O

Pengaturan penetapan I/O dapat dikonfigurasi dari GX Works2. Buka jendela PLC Parameter (Parameter PLC), lalu pilih tab I/O Assignment (Penetapan I/O). Sembarang nomor I/O dapat ditetapkan untuk setiap slot, terlepas dari urutan fisik slot tersebut.

I/O Assignment(*1)

No.	Slot	Type	Model Name	Points	Start XY
0	PLC	PLC			
1	0(*-0)	Input	QX42	64Points	0000
2	1(*-1)	Input	QX41	32Points	0090
3	2(*-2)	Output	QY42	64Points	0040
4	3(*-3)	Output	QY50	16Points	00B0
5	4(*-4)	Intelligent	Q62DA	16Points	0080
6	5(*-5)				
7	6(*-6)				

Switch Setting
Detailed Setting
Select PLC type
New Module

Assigning the I/O address is not necessary as the CPU does it automatically.
Leaving this setting blank will not cause an error to occur.

Atur nomor I/O mulai modul.

Nomor I/O tidak harus nomor kontinu. Beberapa nomor dapat diloncati. Jika sistem akan diperluas di waktu mendatang, sebaiknya beberapa nomor dicadangkan.

Module Selection

Module Type: Input module
Module Name: QX41

Mount Position

Base No.: - Mounted Slot No.: 1 Acknowledge I/O Assignment

Specify start XY address: 0090 (H) 1 Slot Occupy [32 points]

Title setting

Title:

Klik New Module (Modul Baru) untuk membuka jendela ini. Di sini, Anda dapat memilih dan mendaftarkan tipe modul dan nama modul menggunakan daftar dropdown.

5.4.1 Pengaturan slot unit dudukan

Setiap slot unit dudukan juga memiliki nomor yang disebut nomor slot, yang dapat ditetapkan dari dalam pengaturan penetapan I/O. Nomor slot ditetapkan secara otomatis (pada kebanyakan kasus). Nomor ini juga dapat ditetapkan secara manual menggunakan mode detail. Mode detail berguna untuk mencadangkan beberapa nomor slot untuk perluasan sistem di waktu mendatang.

Mode otomatis (default)

Nomor slot diatur secara otomatis sesuai jumlah slot fisik dudukan (utama atau ekstensi).

Bila extension base unit dihubungkan ke main base unit, nomor slot extension base ditetapkan setelah nomor slot terakhir main base unit.

(Contoh) Bila main base unit memiliki lima slot (slot nomor 0 hingga 4), maka slot untuk extension base unit yang dihubungkan akan dimulai dari nomor 5.

Mode detail

Atur jumlah slot untuk setiap base unit. Sembarang nomor dapat diatur. Bila menggunakan mode detail, pengaturan ini diperlukan untuk semua base unit yang digunakan. Untuk membuat pengaturan, buka jendela PLC Parameter (Parameter PLC), dan pilih tab I/O Assignment (Penetapan I/O).

I/O Assignment(*1)

No.	Slot
0	PLC
1	0(0-0)
2	1(0-1)
3	2(0-2)
4	3(0-3)
5	4(0-4)
6	5(1-0)
7	6(1-1)

Assigning the I/O address is not necessary as the CPU does it automatically.
Leaving this setting blank will not cause an error to occur.

Base Setting(*1)

Main	Ext.Base	Slots
Q3	Q6	5
		8

A (B-C)
A: Nomor slot berurutan
(nomor slot main base unit + nomor slot extension base unit)
B: Nomor base unit
C: Nomor slot

Contoh pengaturan

- Tetapkan 5 slot ke main base unit (Q33B) yang memiliki 3 slot fisik (agar tersisa 2 slot kosong).
- Tetapkan 8 slot ke extension base unit (Q65B) yang memiliki 5 slot fisik (agar dapat menampung 3 slot lagi).

Base Mode
 Auto
 Detail

8 Slot Default
 12 Slot Default
 Select module name

Mode pengaturan base unit

Dalam bab ini, Anda telah mempelajari:

- Konsep nomor I/O
- Nomor I/O untuk unitudukan ekstensi
- Pemeriksaan penetapan nomor I/O pada monitor sistem
- Cara menggunakan fungsi penetapan I/O

Hal-hal penting

Konsep nomor I/O dan cara menggunakan fungsi penetapan I/O

- Nomor I/O setiap modul input/output ditetapkan secara berurutan dalam satuan 16 point (0 hingga F), dari slot yang paling dekat dengan modul CPU.
- Jika ada slot kosong di tengah base unit, nomor I/O juga ditetapkan ke slot kosong tersebut.
- Nomor I/O modul pada extension base unit ditetapkan secara otomatis, setelah nomor I/O terakhir pada main base unit.
- Untuk fungsi penetapan I/O, nomor I/O dapat ditetapkan terlepas dari urutan fisik slot pada base unit.

Tes**Tes Akhir**

Setelah menyelesaikan semua pelajaran dari Kursus **PLC Aplikasi Pemrograman**, kini Anda siap mengikuti tes akhir. Jika Anda masih kurang memahami salah satu topik yang dibahas, gunakan kesempatan ini untuk mengulas topik tersebut. **Total terdapat 6 pertanyaan (29 pilihan) dalam Tes Akhir ini.** Anda dapat mengikuti tes akhir sesering mungkin.

Cara menilai tes

Setelah memilih jawaban, pastikan untuk 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 : 2
Jumlah total pertanyaan : 9
Persentase : 22%

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

Lanjutkan

Tinjau

Coba lagi

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

Kalimat berikut ini mendeskripsikan pengatur waktu retentif. Pilih kata yang tepat di setiap tempat yang kosong untuk melengkapi kalimat.

Bila terpenuhi, kumparan berubah dan pengatur waktu retentif mulai mengukur waktu.

Nilai pengatur waktu retentif dipertahankan sekalipun kondisi berubah menjadi kondisi input selama pengukuran.

Saat kumparan kembali, pengatur waktu me-restart pengukuran dari nilai yang dipertahankan.

Setelah nilai pengukuran mencapai nilai pengaturan, timeout (lewat waktu) terjadi dan menyala.

Sekalipun kumparan mati setelah lewat waktu, nilai pengukuran tidak dihapus dan kontak tetap ON.

Instruksi digunakan untuk menghapus nilai pengukuran dan mematikan kontak.

Tes

Tes Akhir 2



Lengkapi program sekuens yang menjalankan operasi dengan pengatur waktu retentif berikut:

Detail operasi:

- 1) Pengatur waktu retentif (ST0) mengukur berapa lama sinyal input X0 atau X1 terus menyala.
- 2) Ketika waktu nyala X0 atau X1 mencapai 30 detik, kumparan (Y70) aktif untuk menghidupkan lampu timeout (lewat waktu).
- 3) Ketika X2 menyala, kontak pengatur waktu retentif (ST0) mati dan nilai pengukuran (nilai saat ini) di-reset.

Q1

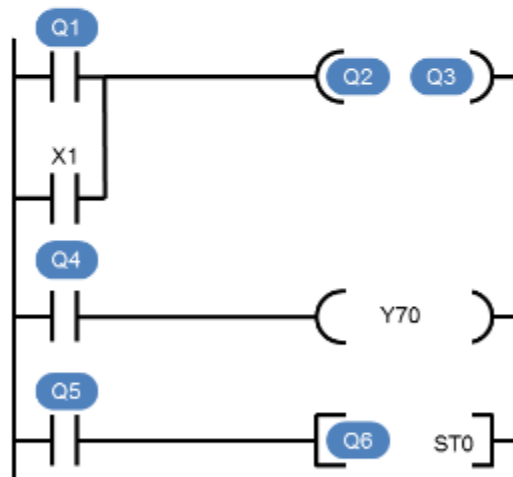
Q2

Q3

Q4

Q5

Q6

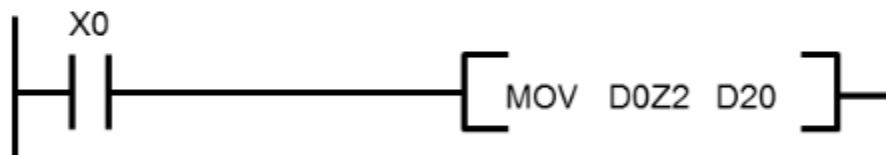


Jawab

Kembali

Di bawah ini adalah program yang menggunakan register indeks "Z2". Pilih nilai yang akan disimpan di register data (D20) ketika X0 diaktifkan pada setiap kondisi:

- 1) Ketika nilai yang disimpan di Z2 adalah 0, disimpan di D20.
- 2) Ketika nilai yang disimpan di Z2 adalah 1, disimpan di D20.
- 3) Ketika nilai yang disimpan di Z2 adalah 2, disimpan di D20.
- 4) Ketika nilai yang disimpan di Z2 adalah 3, disimpan di D20.



Nilai yang disimpan di register data

D0	100
D1	200
D2	400
D3	500

Kalimat berikut ini mendeskripsikan register file pada QCPU. Pilih kata yang tepat di setiap tempat yang kosong untuk melengkapi kalimat.

1) Register file adalah perangkat kata yang digunakan untuk memperluas ekstensi register data (D) dan dinyatakan dengan simbol perangkat .

2) Tidak seperti register data, data yang disimpan di register file tidak akan sekalipun daya dimatikan atau modul CPU di-reset.

3) Biasanya, register file disimpan sebagai file dalam kartu memori (RAM) atau dalam di modul CPU.

4) Untuk menggunakan register file, Anda harus membuat pengaturan yang diperlukan di tab pada jendela PLC Parameter (Parameter PLC).

Di antara nilai numerik yang digunakan pada pengontrol terprogram, nilai numerik tanpa desimal disebut bilangan bulat, dan yang mengandung desimal disebut bilangan asli.

Pilih kata yang tepat di setiap tempat yang kosong untuk melengkapi teks tentang bilangan asli berikut ini.

- 1) Satu bilangan asli menggunakan perangkat kata dan menempati ruang memori bit.
- 2) Nilai numerik yang menangani bilangan asli disebut . Misalnya, nilai numerik 2.035 dinyatakan sebagai pada program sekuens.
- 3) Perintah yang menangani bilangan asli diawali dengan .
- 4) Perintah aritmetika yang menangani bilangan asli memuat bilangan bulat dan bilangan asli sekaligus untuk satu operasi.

Tes**Skor Tes**

Anda telah menyelesaikan Tes Akhir. Hasil Anda adalah sebagai berikut.
Untuk mengakhiri Tes Akhir, lanjutkan ke halaman berikutnya.

Jawaban benar: **6**

Total pertanyaan: **6**

Persentase: **100%**

Selamat. Anda lulus tes ini.

Anda telah menyelesaikan kursus **PLC Aplikasi Pemrograman**.

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