

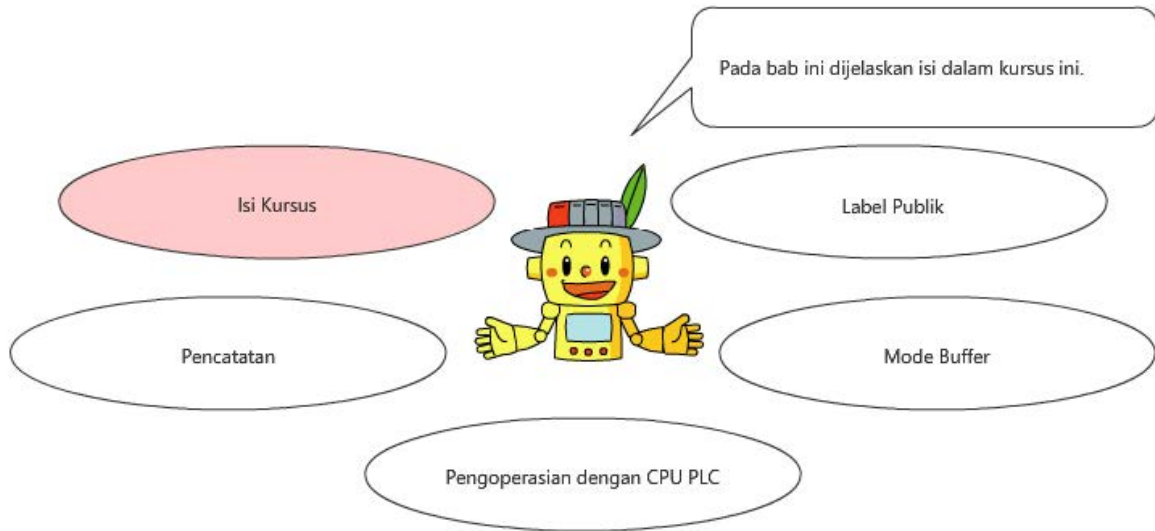
Pengontrol Sistem Servo

MELSEC iQ-R Series Motion Module Basics (Kontrol Pemosisian RD78G(H))

Kursus pelatihan ini ditujukan bagi mereka yang akan membangun sistem kendali gerak menggunakan modul Gerak Seri MELSEC iQ-R untuk pertama kalinya.

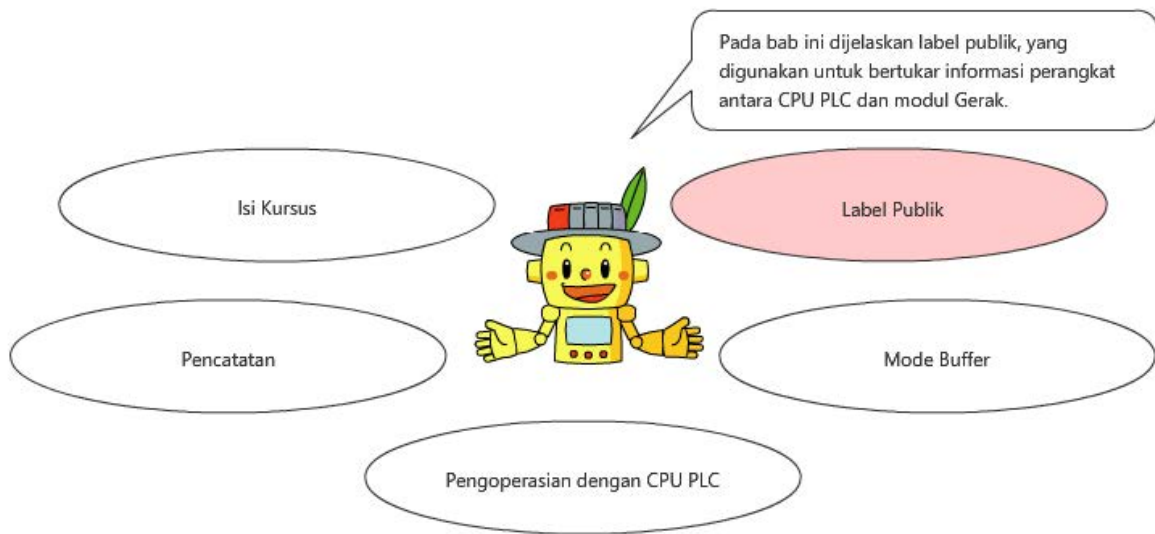
Klik tombol Berikutnya di kanan atas layar.

Kursus ini bertujuan untuk mengembangkan pengetahuan dan pemahaman tentang kendali posisi pada sistem kendali Gerak dengan menggunakan modul Gerak MELSEC iQ-R Series.



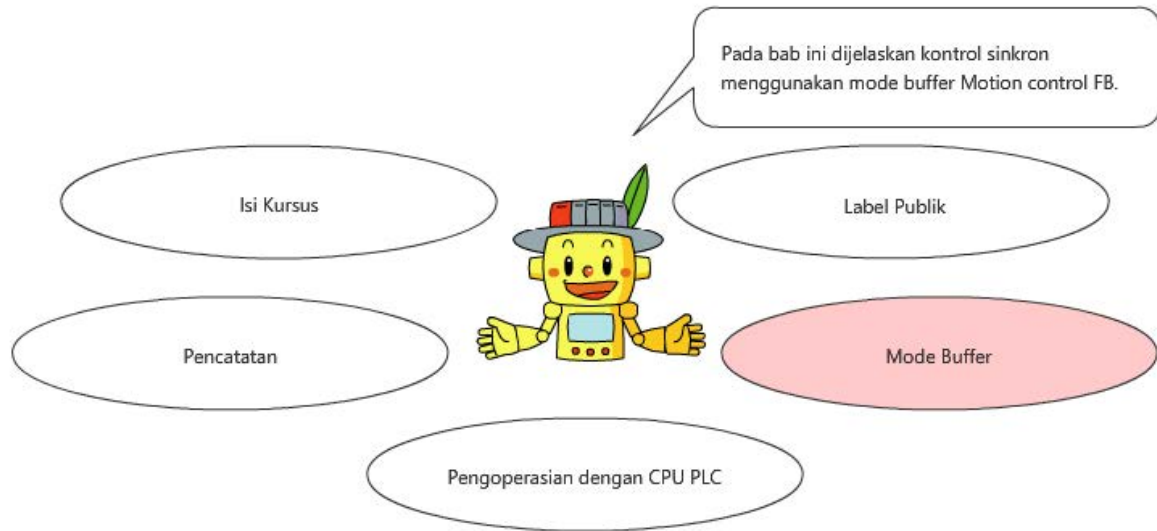
Kursus ini merupakan kelanjutan dari MELSEC iQ-R Series Motion Module Basics (Startup RD78G(H)). Pastikan Anda menyelesaikan kursus Startup sebelum mengambil kursus ini.

Kursus ini bertujuan untuk mengembangkan pengetahuan dan pemahaman tentang kendali posisi pada sistem kendali Gerak dengan menggunakan modul Gerak MELSEC iQ-R Series.



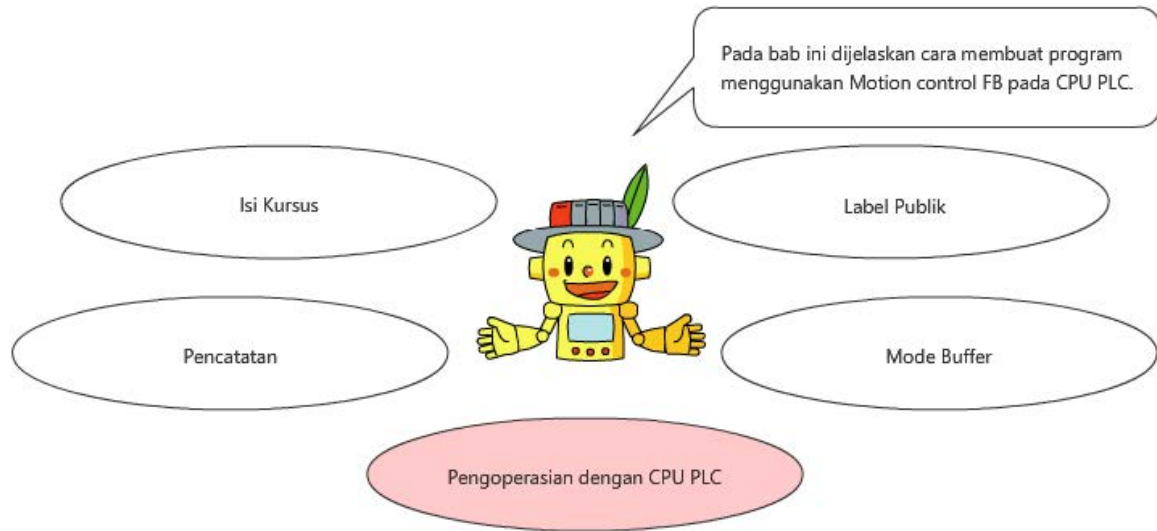
Kursus ini merupakan kelanjutan dari MELSEC iQ-R Series Motion Module Basics (Startup RD78G(H)). Pastikan Anda menyelesaikan kursus Startup sebelum mengambil kursus ini.

Kursus ini bertujuan untuk mengembangkan pengetahuan dan pemahaman tentang kendali posisi pada sistem kendali Gerak dengan menggunakan modul Gerak MELSEC iQ-R Series.



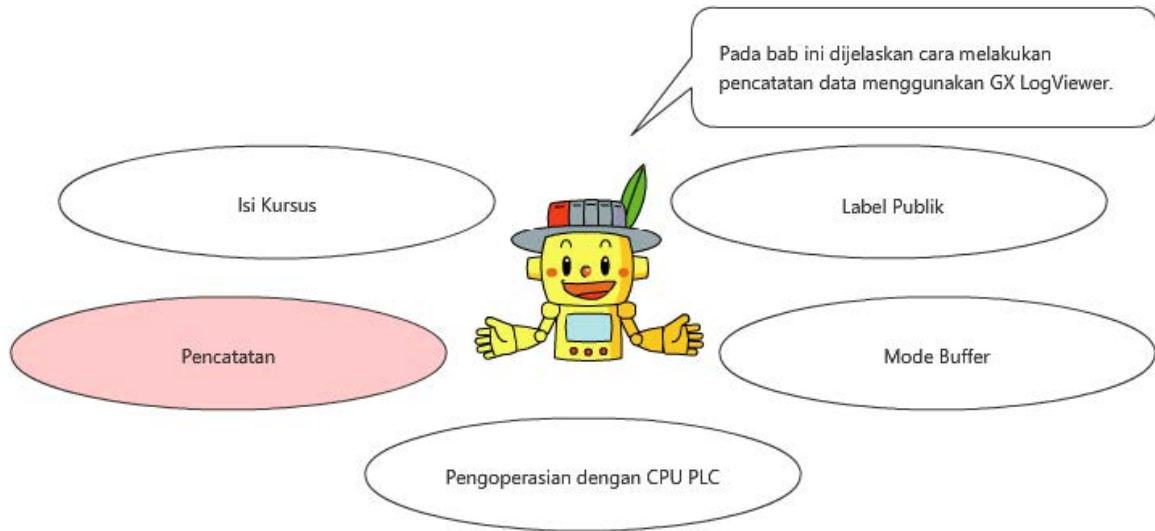
Kursus ini merupakan kelanjutan dari MELSEC iQ-R Series Motion Module Basics (Startup RD78G(H)). Pastikan Anda menyelesaikan kursus Startup sebelum mengambil kursus ini.

Kursus ini bertujuan untuk mengembangkan pengetahuan dan pemahaman tentang kendali posisi pada sistem kendali Gerak dengan menggunakan modul Gerak MELSEC iQ-R Series.



Kursus ini merupakan kelanjutan dari MELSEC iQ-R Series Motion Module Basics (Startup RD78G(H)). Pastikan Anda menyelesaikan kursus Startup sebelum mengambil kursus ini.

Kursus ini bertujuan untuk mengembangkan pengetahuan dan pemahaman tentang kendali posisi pada sistem kendali Gerak dengan menggunakan modul Gerak MELSEC iQ-R Series.



Kursus ini merupakan kelanjutan dari MELSEC iQ-R Series Motion Module Basics (Startup RD78G(H)). Pastikan Anda menyelesaikan kursus Startup sebelum mengambil kursus ini.

Materi dari kursus ini adalah.
Disarankan untuk mulai dari Bab 1.

Bab 1 - Isi Kursus

Pada bab ini dijelaskan isi dalam kursus ini.

Bab 2 - Label Publik

Pada bab ini dijelaskan label publik, yang digunakan untuk bertukar informasi perangkat antara CPU PLC dan modul Gerak.

Bab 3 - Mode Buffer

Pada bab ini dijelaskan kontrol sinkron menggunakan mode buffer Motion control FB.

Bab 4 - Pengoperasian dengan CPU PLC

Pada bab ini dijelaskan cara membuat program menggunakan Motion control FB pada CPU PLC.

Bab 5 - Pencatatan

Pada bab ini dijelaskan cara melakukan pencatatan data menggunakan GX LogViewer.

Tes Akhir

4 bagian total (7 pertanyaan) Nilai kelulusan: 60% atau lebih tinggi.

Pendahuluan Cara Menggunakan Alat e-Learning Ini

Buka halaman berikutnya		Membuka 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 Jendela seperti layar "Daftar Isi" dan kursus akan ditutup.

■Tindakan pencegahan keselamatan

Saat menggunakan produk sebenarnya untuk tujuan pembelajaran, harap baca dengan cermat "Tindakan Pencegahan" yang dijelaskan dalam panduan produk yang akan digunakan, dan perhatikan baik-baik keselamatan dan penggunaan yang benar.

■Tindakan pencegahan dalam kursus ini

Gambar layar yang ditampilkan dalam kursus mungkin berbeda dari perangkat lunak Anda yang sebenarnya tergantung pada versinya. Versi perangkat lunak berikut digunakan dalam kursus ini.

Untuk versi terbaru dari setiap perangkat lunak, periksa Situs Web Mitsubishi Electric FA.

MELSOFT GX Works3	Ver.1.066U	Motion Control Setting function	Ver.1.012N
GX LogViewer	Ver.1.106K		
MELSOFT MR Configurator2	Ver.1.110Q atau lebih baru		

Versi firmware dari CPU PLC harus pada versi 44 atau lebih baru (46 atau lebih baru untuk RD78GH).

Versi firmware modul gerak harus pada versi 10 atau lebih baru.

Untuk mengetahui cara memperbarui versi firmware, lihat Situs Web MITSUBISHI ELECTRIC FA atau panduan konfigurasi modul.

Ikun  menunjukkan panduan referensi.

Isi panduan yang dijelaskan dalam kursus ini adalah versi berikut.

Jika versi berbeda, lokasi deskripsi dan isinya mungkin sedikit berbeda.

Nama panduan	Nomor Panduan	Versi
MELSEC iQ-R Motion Module User's Manual (Startup)	IB-0300406	E
MELSEC iQ-R Motion Module User's Manual (Application)	IB-0300411	E
MELSEC iQ-R Motion Module User's Manual (Network)	IB-0300426	E
MELSEC iQ-R Programming Manual (Motion Module Instructions, Standard Functions/Function Blocks)	IB-0300431	E
MELSEC iQ-R Programming Manual (Motion Control Function Blocks)	IB-0300533	C
MELSEC iQ-R Structured Text (ST) Programming Guide Book	SH-081483	F
MELSEC iQ-R Programming Manual (CPU Module Instructions, Standard Functions/Function Blocks)	SH-081266	Z
MELSEC iQ-R CPU Module User's Manual (Application)	SH-081264	AK

Berikut ini adalah ikhtisar kursus.

Bab 1 Isi Kursus

Pada bab ini dijelaskan isi dalam kursus ini.

**Bab 2 Label Publik**

Pada bab ini dijelaskan label publik, yang digunakan untuk bertukar informasi perangkat antara CPU PLC dan modul Gerak.

**Bab 3 Mode Buffer**

Pada bab ini dijelaskan kontrol sinkron menggunakan mode buffer Motion control FB.

**Bab 4 Pengoperasian dengan CPU PLC**

Pada bab ini dijelaskan cara membuat program menggunakan Motion control FB pada CPU PLC.

**Bab 5 Pencatatan**

Pada bab ini dijelaskan cara melakukan pencatatan data menggunakan GX LogViewer untuk memeriksa pengoperasian modul Gerak.

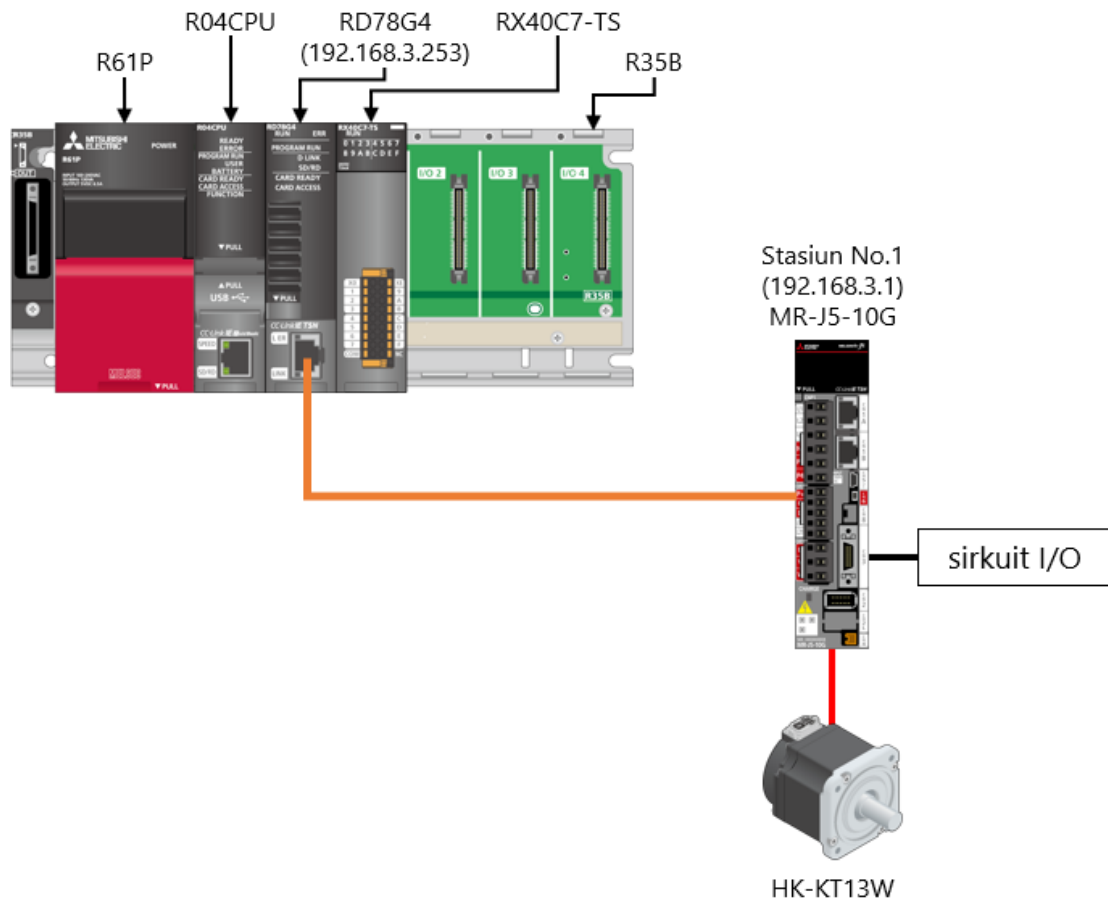
Kursus ini menggunakan mekanisme ball screw poros tunggal yang sama seperti yang digunakan dalam kursus Startup.



Konfigurasi sistem target adalah sebagai berikut.

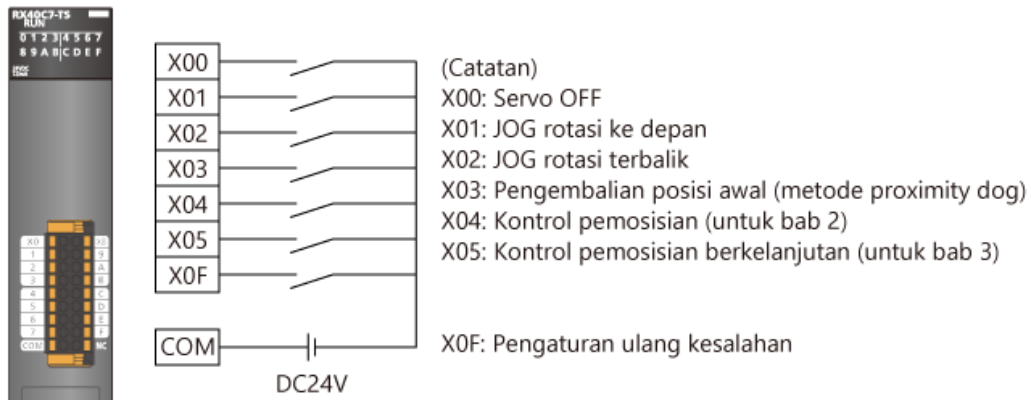
Hapus modul input jarak jauh dari sistem yang digunakan dalam kursus Startup, dan tambahkan modul input RX40C7-TS ke slot 1 unit dasar pengontrol yang dapat diprogram.

Nomor stasiun penguat servo MR-J5-10G telah diubah menjadi 1, dan alamat IP telah diubah menjadi 192.168.3.1.



Pengkabelan catu daya untuk pengontrol terprogram dan penguat servo, serta metode sambungan motor servo sama seperti yang dijelaskan dalam kursus Startup. Berikut ini menunjukkan kabel rangkaian eksternal dari modul input.

RX40C7-TS



(Catatan) Karena I/O Nomor RX40C7-TS adalah 0020H, X20 ke X25 dan X2F digunakan dalam program.

Ketika modul Gerak dikendalikan oleh modul masukan pengontrol yang dapat diprogram, seperti dalam sistem yang digunakan dalam kursus ini yang dijelaskan dalam bab 1, CPU PLC dan modul Gerak harus bertukar informasi perangkat.

Ada dua metode berikut.

1. Gunakan label publik.
2. Gunakan memori buffer modul Gerak.

Pada bab ini dijelaskan cara melakukan pencatatan data menggunakan label publik.

Unduh contoh program yang akan digunakan pada bab ini dan bab 3 dengan mengklik tautan di bawah.

[RD78GBasic2_sample1.zip \(1.34MB\)](#)

[Poin]

Saat menggunakan memori buffer, salin data yang akan ditukar ke area pengguna (U□\G11478000 to G11997999).

(Contoh program)

<PLC CPU>

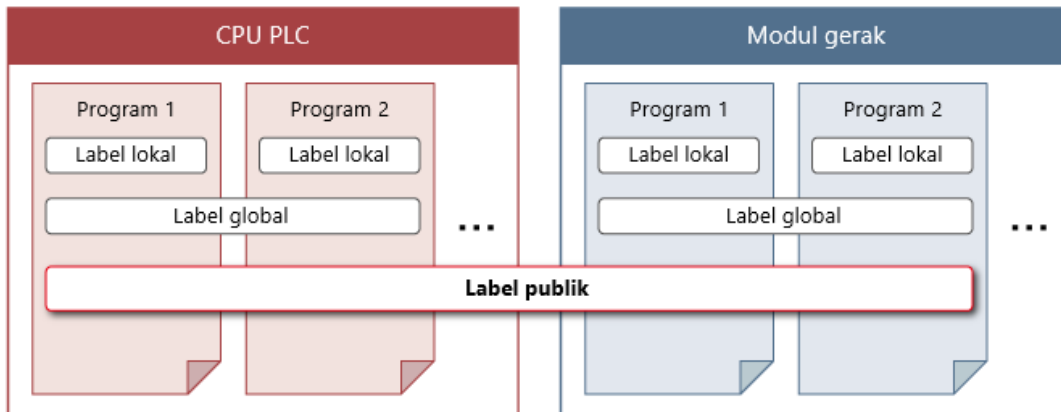
Pemosisian dimulai



<Motion module>

```
MC_MoveAbsolute_1(  
  Execute:= G11478000.0 ,  
  :  
  :  
);
```

Label publik adalah label bersama yang dapat digunakan di modul Gerak dan CPU PLC. Berikut ini adalah area yang berlaku untuk label lokal, label global, dan label publik.



(1) Cara mendaftarkan label publik

Daftarkan label publik dari label global modul Gerak.

Pastikan bahwa "Public Label" kolom terlihat di editor label global layar Motion Module Setting Function.

Atur label yang akan didaftarkan sebagai label publik ke "Enabled".

Tindakan ini akan mengaktifkan kolom "Motion Control Attribute".

Pilih apakah setiap label akan dibaca atau ditulis dari/ke CPU PLC.

	Label Name	Data Type	Class	Initial	Constant	Japanese	English(Display Target)	Chinese	Remark	Public Label	Motion Control Attribute
1	G_bSVONCMD	Bit	VAR_GLOBAL				Servo ON			Enabled	WRITE (=> Motion)
2	G_leJogVelocity	FLOAT [Double Precision]	VAR_GLOBAL				JOG Velocity			Enabled	WRITE (=> Motion)
3	G_bJogFwd	Bit	VAR_GLOBAL				JOG Forward			Enabled	WRITE (=> Motion)
4	G_bJogBwd	Bit	VAR_GLOBAL				JOG Backward			Enabled	WRITE (=> Motion)
5	G_bJogBusy	Bit	VAR_GLOBAL				JOG Busy			Enabled	READ (Motion =>)
6	G_lePosition0	FLOAT [Double Precision]	VAR_GLOBAL				Position0 Address			Disabled	-
7	G_bHomingCMD	Bit	VAR_GLOBAL				Homing Command			Enabled	WRITE (=> Motion)
8	G_bHomingDone	Bit	VAR_GLOBAL				Homing Done			Enabled	READ (Motion =>)
9	G_bHomingReq	Bit	VAR_GLOBAL				Homing Request			Enabled	READ (Motion =>)
10	G_bPosCMD	Bit	VAR_GLOBAL				Positioning Command			Enabled	WRITE (=> Motion)
11	G_bPosDone	Bit	VAR_GLOBAL				Positioning Done			Enabled	READ (Motion =>)
12	G_bPosReq	Bit	VAR_GLOBAL				Positioning Start Request			Enabled	READ (Motion =>)
13	G_bErrorReset	Bit	VAR_GLOBAL				Error Reset			Enabled	WRITE (=> Motion)
14	G_bContPosCMD	Bit	VAR_GLOBAL				Continuous Positioning Command			Enabled	WRITE (=> Motion)
15	G_bContPosReq	Bit	VAR_GLOBAL				Continuous Positioning Start Request			Enabled	WRITE (=> Motion)
16	G_bContPosDone	Bit	VAR_GLOBAL				Continuous Positioning Done			Enabled	READ (Motion =>)
17											

[Poin]

Jika kolom label publik tidak terlihat, gulir tabel ke kanan.

(2) Tipe data yang dapat didaftarkan sebagai label publik

Tabel berikut memperlihatkan tipe data yang dapat didaftarkan sebagai label publik.

Tipe variabel	Jenis	Pemilihan susunan	Pengaturan label publik	Penandaan
Label global	Tipe sederhana	Tidak	○	Pengaturan tidak dimungkinkan untuk label dan kelas berikut. ■Label • Label jenis string • Label jenis pengatur waktu • Label jenis penghitung • Label jenis penghitung panjang • Label pengatur waktu yang tersimpan • Label jenis waktu retensi yang lama • Label jenis pengatur waktu panjang ■Kelas • Kelas VAR_GLOBAL_CONSTANT
		Ya	△(Catatan 1,2)	
	Tipe data terstruktur	Tidak	△(Catatan 3)	
		Ya	△(Catatan 1,2,4,5)	
	FB (Termasuk Motion control FB)	Tidak	×	
		Ya	×	
Program	-	-	×	
Program memblokir label lokal	-	-	×	
Tipe data terstruktur	-	-	△(Catatan 3,5)	
Data terstruktur Motion control FB	-	-	△(Catatan 6,7)	

(Catatan)

1. Pengaturan label publik tidak dapat dikonfigurasi untuk setiap elemen array.
2. Ketika array tipe bit digunakan, label publik tidak dapat diatur ke "Enabled". (Dalam data terstruktur, hanya anggota terkait yang tidak dapat diatur ke "Enabled".)
3. Ketika tipe string digunakan sebagai anggota tipe data terstruktur, anggota tidak dapat diatur ke "Enabled".
4. Data terstruktur dengan maksimal empat lapisan dapat dipublikasikan.
5. Ketika array data terstruktur digunakan sebagai anggota data terstruktur, anggota tersebut tidak dapat diatur ke "Enabled".
6. Dapat digunakan dalam program PLCopen Motion control FB oleh modul CPU.
7. Ketika jenis string digunakan dalam data terstruktur Motion control FB, itu jenis data terstruktur Motion control FB itu sendiri tidak dapat diatur.

(3) Cara mendaftarkan data terstruktur sebagai label publik

Untuk mengatur anggota jenis data terstruktur yang disiapkan dalam sistem seperti data monitor axis ke label publik, daftarkan label publik berdasarkan lapisan data terstruktur seperti yang ditunjukkan di bawah ini.

Kursus ini menjelaskan cara mendaftarkan Set Posisi (SetPosition) dan Atur Kecepatan (SetVelocity), memantau data (Md) dari axis penggerak sebenarnya (Axis_Real), sebagai label publik.

[Cara mengatur AxisName.Md.SetPosition (Perintah posisi saat ini) dan AxisName.Md.SetVelocity (Perintah kecepatan saat ini) ke label publik]

1) Buka grup "Ax+Global" di label global dan atur label publik "Axis0001" ke "Enabled".

2) Buka grup "AXIS_REAL" dalam tipe data struktur, dan atur label publik "Md" (Axis Monitor Data) ke "Enabled".

AxisRef diatur ke "Diaktifkan" secara default.

3) Buka grup "AXIS_REAL_MONI" pada tipe data struktur, dan atur "SetPosition" (Atur Posisi) dan "SetVelocity" (Atur Kecepatan) ke label publik.

Atribut kontrol gerak dari tipe data terstruktur bersifat tetap.

(4) Mewakili label publik

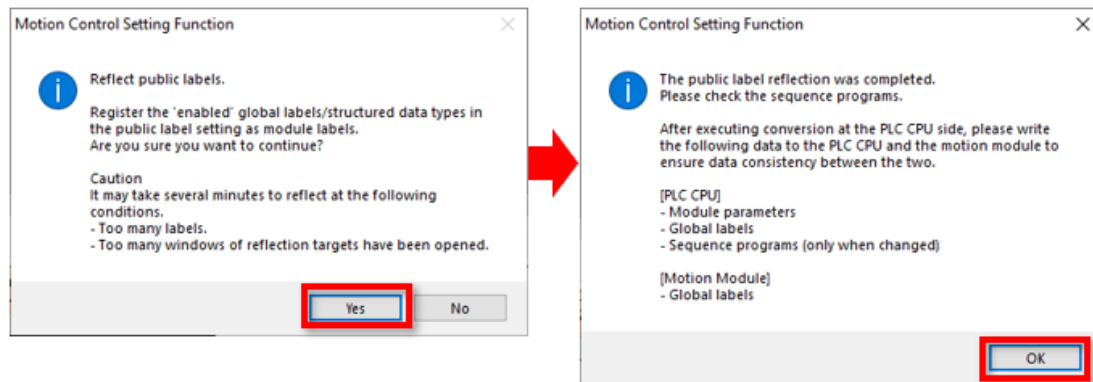
Pilih [Convert] → [Rebuild All] di menu.

Kapasitas bebas label publik ditampilkan sebagai Informasi di jendela output.

Output					
Rebuild All Error:0 Warning:0 Information					
No.	Result	Data Name	Category	Content	Error Code
1	Information	Public Label	Free Volume	99.88[%] (32728 [Word] = 32768 [Word] - (Global: 40 [Word]))	-

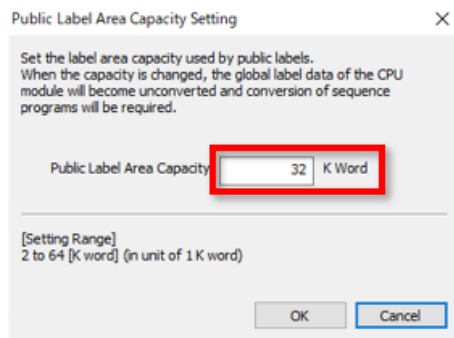
Ketika semua proses pembuatan ulang berhasil diselesaikan, pilih [Convert] → [Reflect Public Labels] di menu. Klik [Yes] di jendela pop up berikut.

Ketika pesan yang menunjukkan bahwa label publik berhasil diwakili muncul, klik [OK] tombol.



(Catatan) Kapasitas memori yang dapat digunakan untuk mendaftarkan label publik adalah 32 ribu kata secara default. Kapasitasnya bisa ditingkatkan hingga 64K kata.

Untuk mengubah kapasitas, atur ukuran memori dari [Convert] → [Public Label Capacity Setting] di menu.



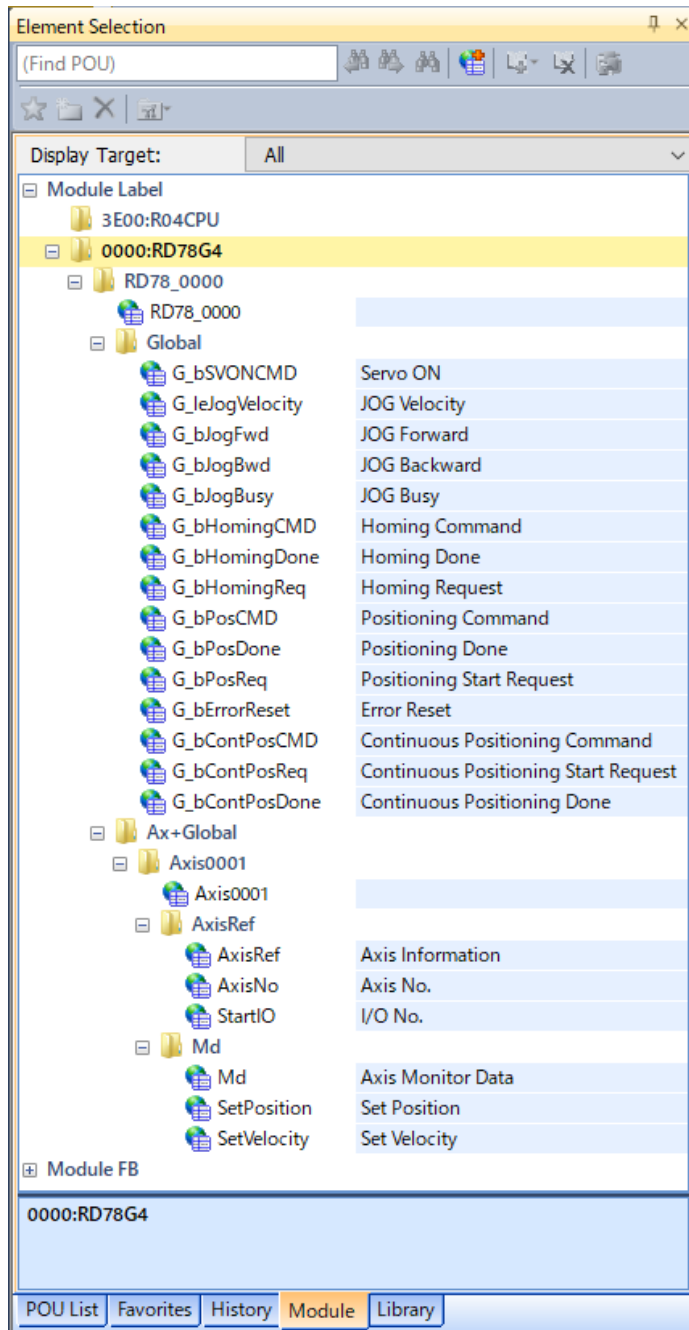
(5) Memeriksa label dari sisi CPU PLC

Label publik yang diwakili akan didaftarkan ke label modul di sisi CPU PLC.

Pilih label modul dari jendela Element Selection GX Works3, dan periksa apakah label publik telah terdaftar di [0000:RD78G4] dalam [Module Label].

Setelah mengubah pengaturan label publik, selalu jalankan "Reflect Public Labels" lagi.

Saat menggunakan label publik di CPU PLC, buat kembali semua program.

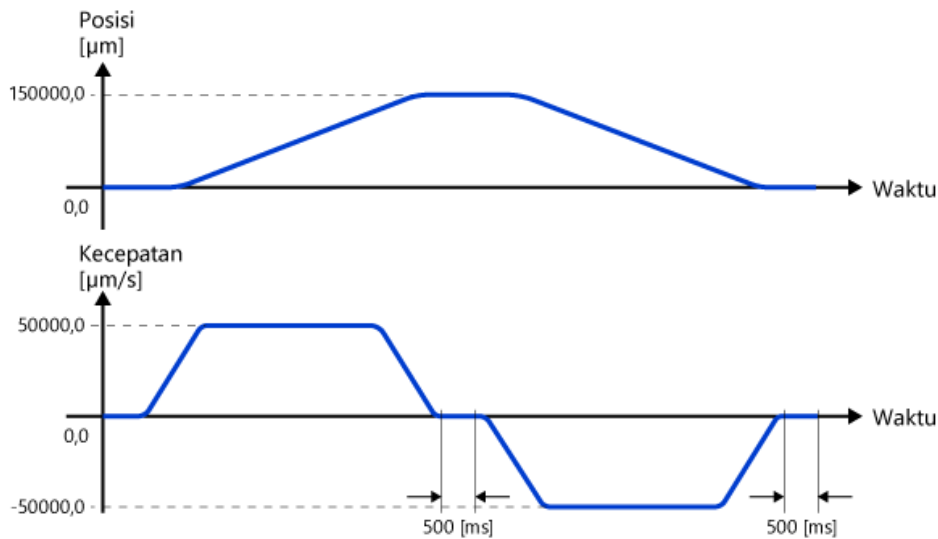


(1) Contoh operasi program

Sinyal input dari program sampel yang digunakan dalam bab ini ditetapkan sebagai berikut.

Input	Operasi
X20	Servo off (Catatan)
X21	Operasi JOG rotasi ke depan
X22	Operasi JOG rotasi terbalik
X23	Pengembalian ke posisi awal
X24	Kontrol pemosisian
X25	Kontrol pemosisian berkelanjutan (Bab 3)

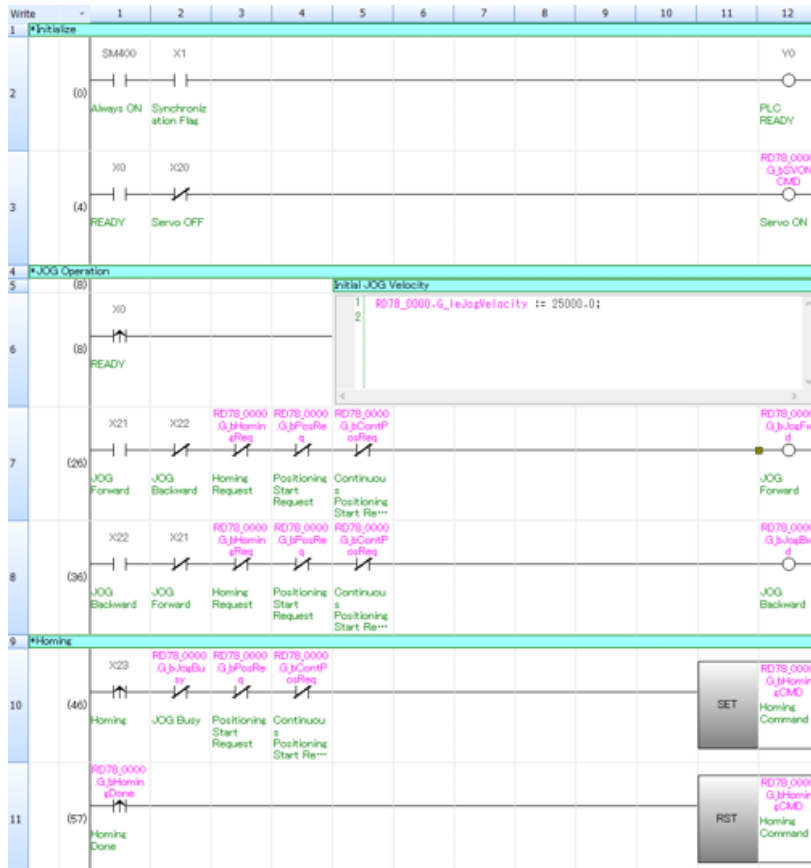
Berikut ini adalah pola operasi X24: kontrol pemosisian.



(Catatan) Contoh program ini mengeksekusi servo ON secara otomatis ketika CPU PLC diatur ke RUN. Saat daya dihidupkan dengan sinyal start ON, motor servo dapat diaktifkan.

(2) Program CPU PLC

1) MAIN (tangga, program pemindaian)



Y0 dihidupkan terlebih dahulu.

Ketika X0 dihidupkan, servo ON dijalankan.
Hidupkan X20 untuk menjalankan servo OFF.

Tetapkan nilai awal kecepatan JOG.
Program ini menggunakan ST inline.
Karena atribut kontrol gerak dari label global
"G_JogVelocity" yang menyimpan kecepatan JOG diatur
ke "WRITE (→Motion)", nilai numerik harus diatur di
CPU PLC.

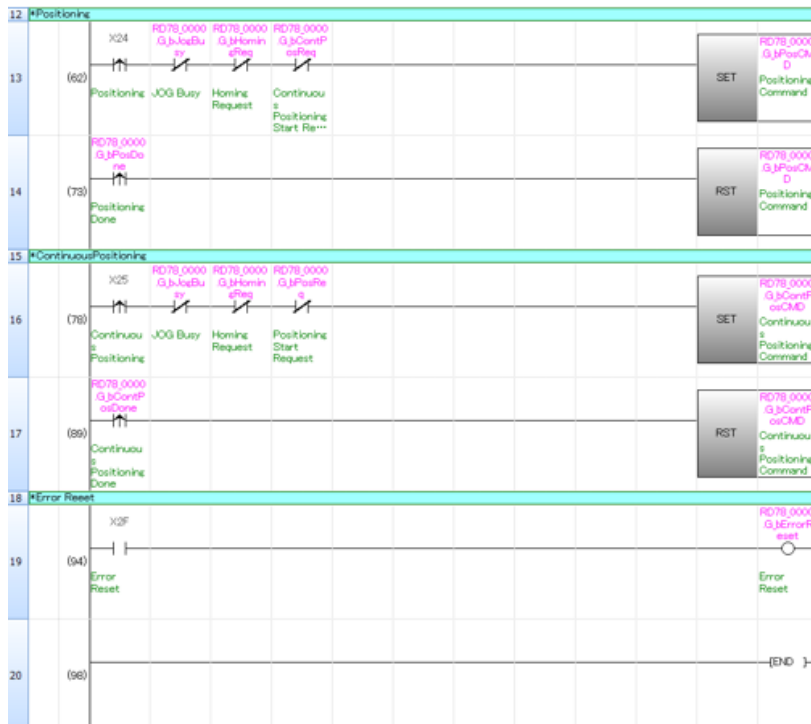
Menyalakan sinyal awal pengoperasian JOG.
Ini mencegah rotasi maju dan rotasi mundur
dimulai pada saat yang bersamaan.
Interlock diatur untuk mencegah operasi JOG
dimulai saat program lain sedang dijalankan.

Permulaan pengembalian posisi asal (X23) disimpan
dalam label publik G_bHomingCMD, dan dikirim ke
modul Gerak sebagai kondisi awal pengembalian posisi
asal.
Interlock diatur untuk mencegah pengembalian posisi
awal dimulai saat program lain sedang berjalan.
Setelah menerima bahwa modul Gerak menyalakan
sinyal penyelesaian pengembalian posisi awal,
G_bHomingCMD diatur ulang pada tepi naik sinyal
tersebut.

(2) Program CPU PLC

1) Bagian lanjutan dari MAIN (tangga, program pemindaian)

(Lanjutan dari halaman sebelumnya)



Tepi naik dari awal kontrol pemosisian (X24) disimpan di G_bPosCMD, dan dikirim ke modul Gerak sebagai kondisi awal dari kontrol pemosisian. Interlock diatur untuk mencegah pengembalian pemosisian dimulai saat program lain sedang berjalan. Setelah menerima bahwa modul Gerak menyalakan sinyal penyelesaian pengembalian posisi awal, G_bPosCMD diatur ulang pada tepi naik sinyal tersebut.

Program awal untuk pemosisian berkelanjutan dijelaskan dalam bab 3. Untuk detailnya, lihat 3.4.

Kesalahan diatur ulang ketika X2F dihidupkan.

(2) Program CPU PLC

2) MONITOR (ST, program pemindaian)

SetPosition (Atur Posisi) dan SetVelocity (Atur Kecepatan) monitor axis yang diatur sebagai label publik disimpan dalam perangkat kata D0 dan D2.

Karena SetPosition Dan SetVelocity adalah jenis bilangan nyata presisi ganda, keduanya diubah menjadi jenis kata ganda bertanda sehingga dapat dengan mudah ditangani oleh CPU PLC. (Catatan)

Meskipun perangkat kata ini tidak digunakan dalam kursus ini, perangkat tersebut digunakan untuk menampilkan data pada program urutan lain dan GOT, dan untuk tujuan lainnya.

```
1 D0:D := LREAL_TO_DINT(RD78_0000.Axis0001.Md.SetPosition);  
2 D2:D := LREAL_TO_DINT(RD78_0000.Axis0001.Md.SetVelocity);  
3
```

Tentukan jenis kata ganda yang ditandatangani dengan "D0:D".

(Catatan) Ketika tipe bilangan nyata presisi ganda dikonversi ke jenis kata ganda bertanda, jika nilai yang akan dikonversi berada di luar rentang dari -2147483648 hingga 2147483647, akan terjadi kesalahan perhitungan.

(3) Program modul Gerak

1) ServoON_JOG (tipe eksekusi normal)

```

1  //-----Servo ON-----
2  MC_Power_1(
3      Axis      := Axis0001.AxisRef,
4      Enable    := TRUE,
5      ServoON   := G_bSVONCMD,
6      Busy      => bPowerBusy
7  );
8
9  //-----JOG Operation-----
10 //Initial Value Setting
11 IF (bPowerBusy) THEN
12     leJogAcceleration := 50000.0;
13     leJogDeceleration := 50000.0;
14     leJogJerk          := 0.0;
15 -END_IF;
16
17 //JOG
18 MCv_Jog_1(
19     Axis      := Axis0001.AxisRef,
20     JogForward := G_bJogFwd,
21     JogBackward := G_bJogBwd,
22     Velocity   := G_leJogVelocity,
23     Acceleration := leJogAcceleration,
24     Deceleration := leJogDeceleration,
25     Jerk       := leJogJerk,
26     Busy       => G_bJogBusy
27 );
28

```

Menerima sinyal servo ON (G_bSVONCMD) dari CPU PLC dan mengeksekusi servo ON.

Menyimpan nilai akselerasi JOG, deselerasi JOG, dan sentakan JOG ke label ketika output Sibuk MC_Power_1 diaktifkan.

Menerima sinyal start JOG dan kecepatan JOG dari CPU PLC.

Mengembalikan keluaran Sibuk ke CPU PLC.

(Catatan) Dalam program contoh ini, sinyal I/O FB yang tidak digunakan atau yang belum diubah dari nilai awalnya akan dihilangkan.

- (3) Program modul Gerak
 2) Homing (tipe eksekusi normal)

```

1  //-----Homing Operation-----
2  //Initial Value Setting, Operation Start Request
3  IF G_bHomingCMD THEN
4      G_lePosition0 := 0.0 ;
5      G_bHomingReq  := TRUE ;
6  ELSE
7      G_bHomingReq  := FALSE ;
8  END_IF;
9
10 //Homing
11 MC_Home_1(
12     Axis          := Axis0001.AxisRef ,
13     Execute        := G_bHomingReq,
14     Position       := G_lePosition0 ,
15     Done           => bHomingDone ,
16     Busy           => G_bHomingBusy ,
17     CommandAborted => bHomingAborted ,
18     Error          => bHomingError
19 );
20
21 //Done Signal => PLC CPU
22 G_bHomingDone := bHomingDone OR bHomingAborted OR bHomingError;
23

```

Menerima sinyal perintah pengembalian posisi awal (G_bHomingCMD) dari CPU PLC.
 Menyimpan alamat posisi rumah ke label dan mengaktifkan permintaan pengembalian posisi asal (G_bHomingReq).

Mematikan G_bHomingReq ketika G_bHomingCMD dimatikan.

Mengembalikan sinyal penyelesaian eksekusi ke CPU PLC setelah pengembalian posisi awal berhasil diselesaikan (Done output ON), eksekusi terhenti (CommandAborted output ON), atau terjadi kesalahan (Error output ON).

(3) Program modul Gerak

3) Pemosisian (tipe eksekusi normal)

```

1  //-----Positioning Operation-----
2  //Initial Value Setting, Operation Start Request
3  IF G_bPosCMD THEN
4      lePosition1      := 150000.0;
5      lePosVelocity    := 50000.0;
6      lePosAcceleration := 100000.0;
7      lePosDeceleration := 100000.0;
8      lePosJerk        := 200000.0;
9      G_bPosReq := TRUE;
10 ELSE
11     G_bPosReq := FALSE;
12 END_IF;
13
14 //Positioning1
15 MC_MoveAbsolute_1(
16     Axis      := Axis0001.AxisRef ,
17     Execute   := G_bPosReq ,
18     Position  := lePosition1 ,
19     Velocity  := lePosVelocity ,
20     Acceleration := lePosAcceleration ,
21     Deceleration := lePosDeceleration ,
22     Jerk      := lePosJerk ,
23     Direction := MC_DIRECTION__mcShortestWay ,
24     BufferMode := MC_BUFFER_MODE_mcAborting ,
25     Done      => bMoveAbs1Done ,
26     CommandAborted => bMoveAbs1Aborted ,
27     Error     => bMoveAbs1Error
28 );
29 //Dwell
30 TON_1(IN:= bMoveAbs1Done ,PT:= T#500ms ,Q=> bDwell1_out );
31 //Positioning2
32 MC_MoveAbsolute_2(
33     Axis      := Axis0001.AxisRef ,
34     Execute   := bDwell1_out ,
35     Position  := G_lePosition0 ,
36     Velocity  := lePosVelocity ,
37     Acceleration := lePosAcceleration ,
38     Deceleration := lePosDeceleration ,
39     Jerk      := lePosJerk ,
40     Direction := MC_DIRECTION__mcShortestWay ,
41     BufferMode := MC_BUFFER_MODE_mcAborting ,
42     Done      => bMoveAbs2Done ,
43     CommandAborted => bMoveAbs2Aborted ,
44     Error     => bMoveAbs2Error
45 );
46 //Dwell
47 TON_2(IN:= bMoveAbs2Done ,PT:= T#500ms ,Q=> bDwell2_out );
48
49 //Error Signal, Aborted Signal
50 bPosError  := bMoveAbs1Error OR bMoveAbs2Error;
51 bPosAborted := bMoveAbs1Aborted OR bMoveAbs2Aborted;
52
53 //Done Signal => PLC CPU
54 G_bPosDone := bDwell2_out OR bPosError OR bPosAborted;

```

Menerima sinyal mulai kontrol posisi (G_bPosCMD) dari CPU PLC. Data yang diperlukan untuk pemosisian disimpan ke label, dan permintaan awal kontrol pemosisian (G_bPositioningReq) diaktifkan.

Mematikan G_bPositioningReq setelah G_bPosCMD dimatikan oleh CPU PLC.

MC_MoveAbsolute mengeksekusi pemosisian.

Memulai pengatur waktu tunda yang memicu diam ketika output Selesai dari MC_MoveAbsolute_1 diaktifkan.

Setelah waktu tunggu berlalu, MC_MoveAbsolute mengeksekusi penentuan posisi lagi.

Memulai pengatur waktu tunda yang memicu diam ketika output Selesai dari MC_MoveAbsolute_2 diaktifkan.

Mengembalikan sinyal penyelesaian eksekusi ke CPU PLC setelah waktu tunggu berlalu atau output Error atau output CommandAborted dari MC_MoveAbsolute diaktifkan.

(3) Program modul Gerak

4) ErrorReset (tipe eksekusi normal)

```
1 //Axis Error Reset
2 MC_Reset_1(
3     Axis      := Axis0001.AxisRef ,
4     Execute   := G_bErrorReset
5 );
6
7 //System Error Reset
8 MCv_MotionErrorReset_1(
9     Execute   := G_bErrorReset
10 );
11
```

Menerima sinyal mulai ulang kesalahan (G_bErrorReset) dari CPU PLC dan menjalankan mulai ulang kesalahan axis dan mulai ulang kesalahan sistem.

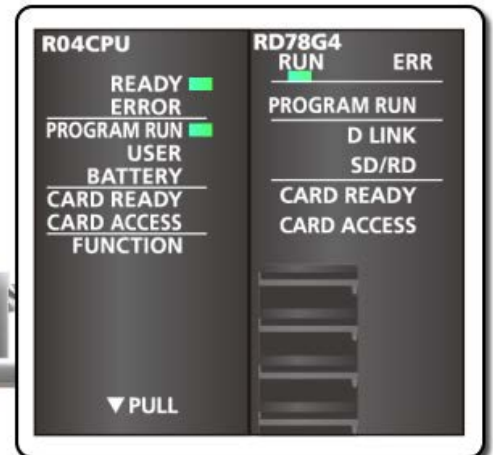
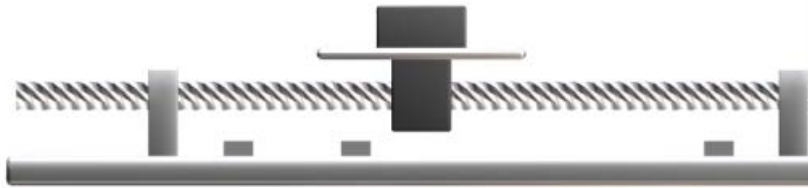
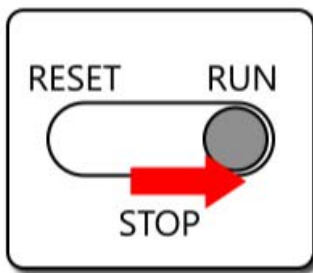
Tulis program dan parameter ke CPU PLC dan modul Gerak.

- 1) Setelah semua program di CPU PLC dibangun kembali, pilih [Online] → [Write to PLC] di bilah alat GX Works3 untuk menulis semua data ke CPU PLC.
- 2) Ketika parameter ditulis ke CPU PLC, komunikasi dengan modul Gerak diaktifkan.
Pilih [Online] → [Write to Module] di bilah alat Motion Control Setting Function untuk menulis semua data ke modul Gerak.
- 3) Reset CPU PLC untuk menyelesaikan operasi penulisan.

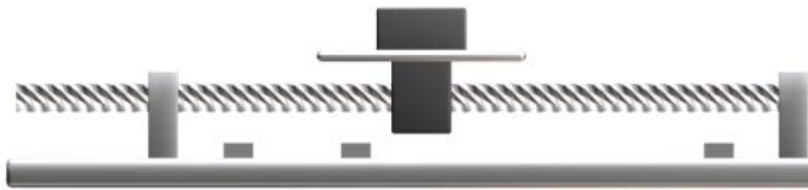
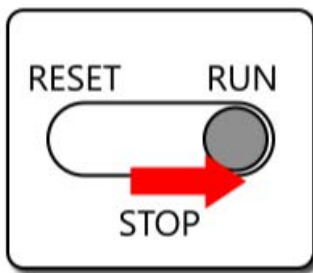
Klik tombol putar di kiri bawah jendela.



Periksa contoh operasi program.
Sebelum memulai pengoperasian, pastikan program
CPU PLC dan modul Gerak telah ditulis.



Atur tuas RUN/STOP/RESET CPU PLC ke RUN.
Lampu READY dan lampu PROGRAM RUN pada CPU PLC menyala.
Lampu RUN pada modul Gerak menyala.



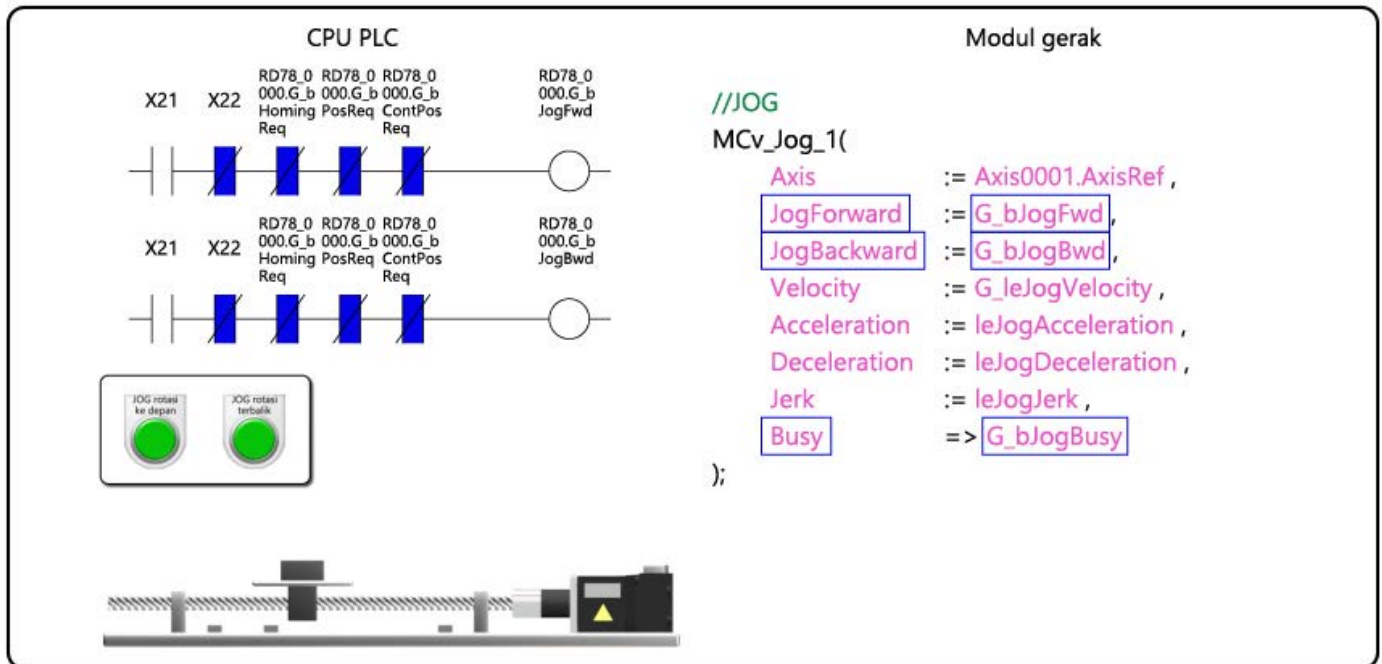
Tunggu hingga lampu PROGRAM RUN pada modul Gerak menyala.
"r.01" ditampilkan pada penguat servo. (Titik-titiknya menyala.)
Motor servo memasuki status servo ON.



Hidupkan X20 untuk menjalankan servo OFF.
"r.01" ditampilkan pada penguat servo. (Titik-titiknya berkedip.)
Hidupkan X20 untuk menjalankan servo ON lagi.



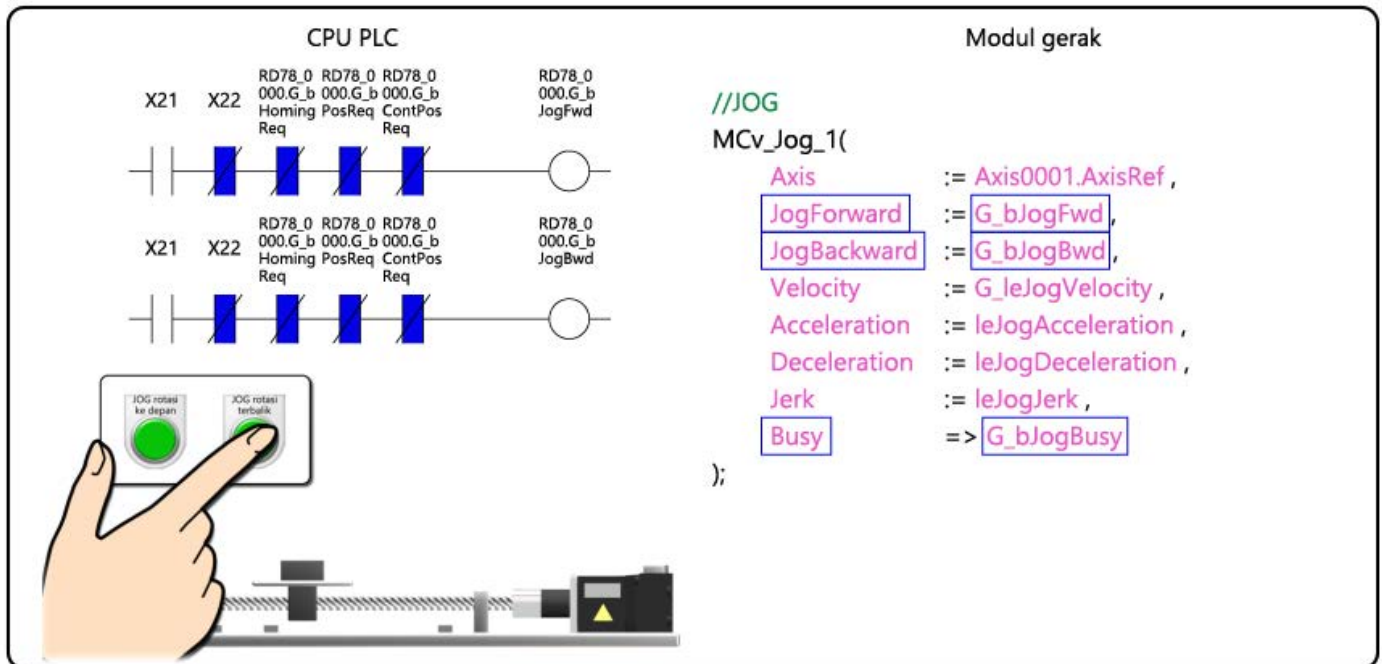
Nyalakan JOG putaran maju (X21) untuk menggerakkan axis searah kenaikan alamat, dan matikan untuk berhenti.
Nyalakan JOG putaran terbalik (X22) untuk menggerakkan axis ke arah penurunan alamat, dan matikan untuk berhenti.



Periksa monitor program.

Saat X21 dihidupkan, "RD78_0000.G_bJogFwd" menyala dan "G_bJogFwd" di sisi modul Gerak menyala.

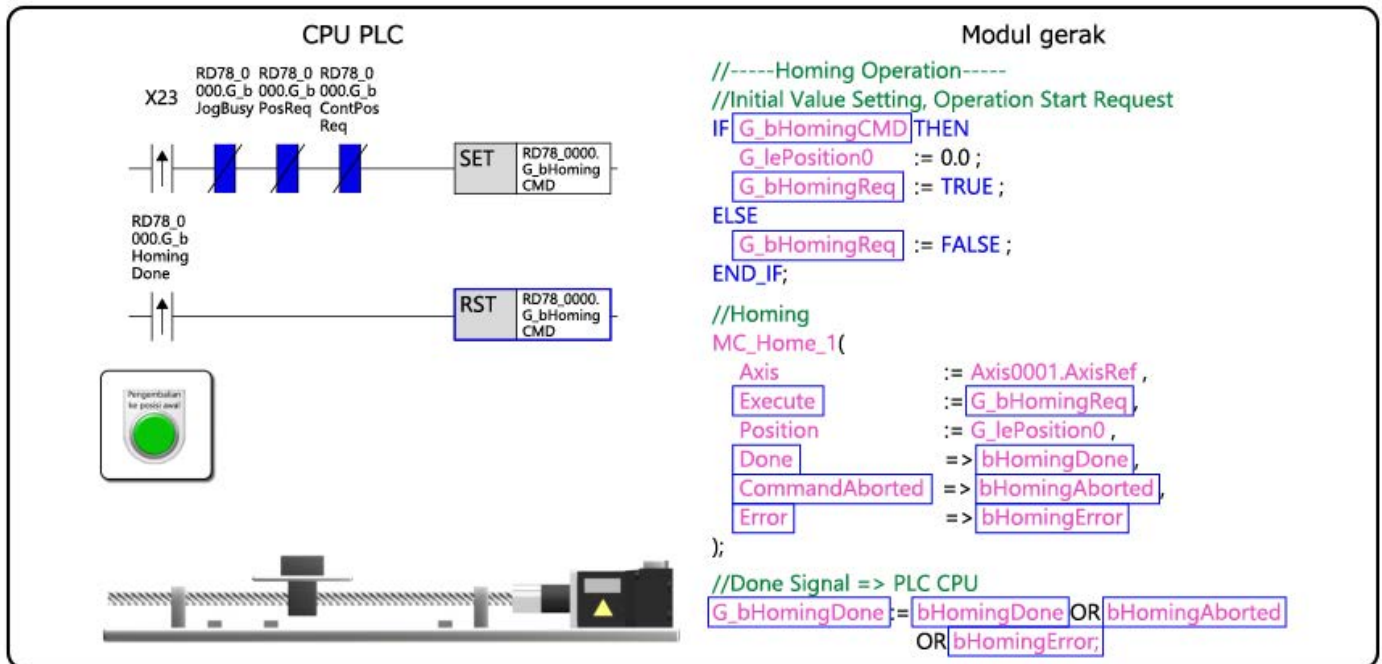
Ketika input JogForward dari MCv_Jog_1 menyala, JOG rotasi maju dimulai.



Saat X22 dihidupkan, "RD78_0000.G_bJogBwd" menyala dan "G_bJogFwd" di sisi modul Gerak menyala. Ketika input JogBackward dari MCv_Jog_1 menyala, JOG rotasi terbalik dimulai.

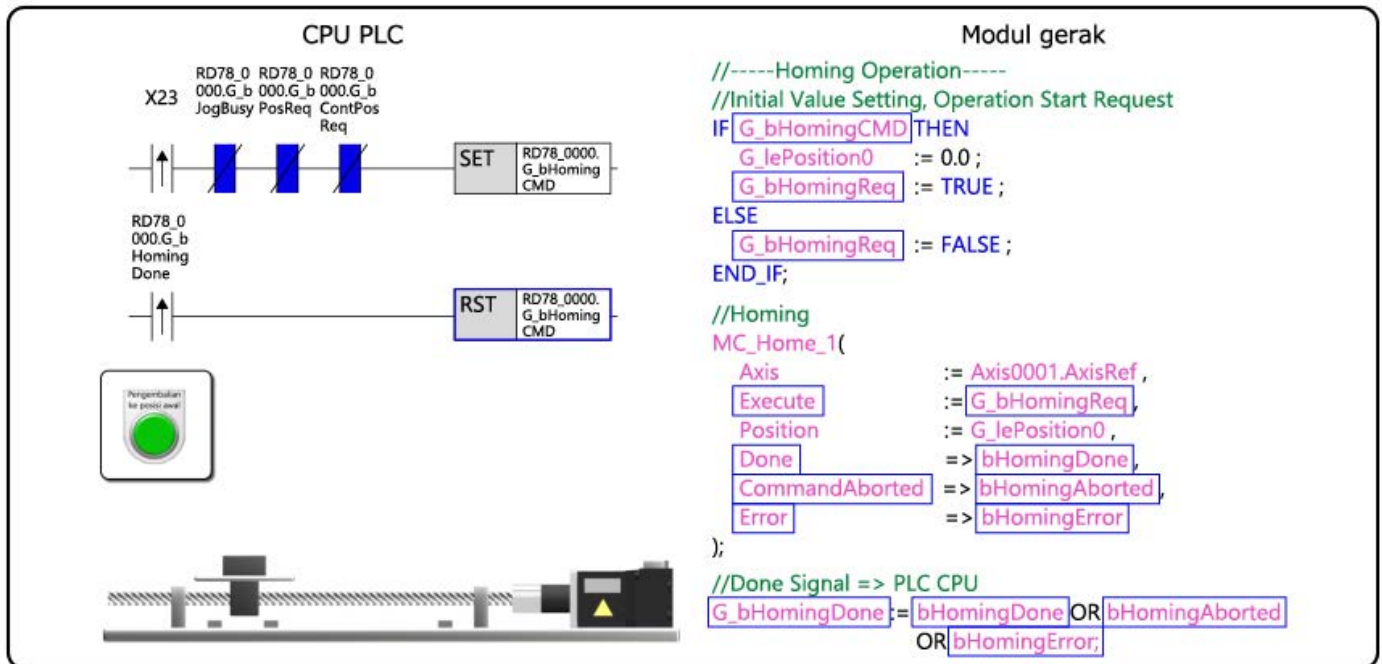


Aktifkan pengembalian posisi awal (X23) untuk memulai pengembalian posisi awal.
Jalankan pengembalian posisi asal dengan metode proximity dog
(33 dikurangi dari Pr.PT45)
Axis nya berhenti sedikit lebih jauh di luar dog,
dan menetapkan titik tersebut sebagai posisi awal.



Periksa monitor program.

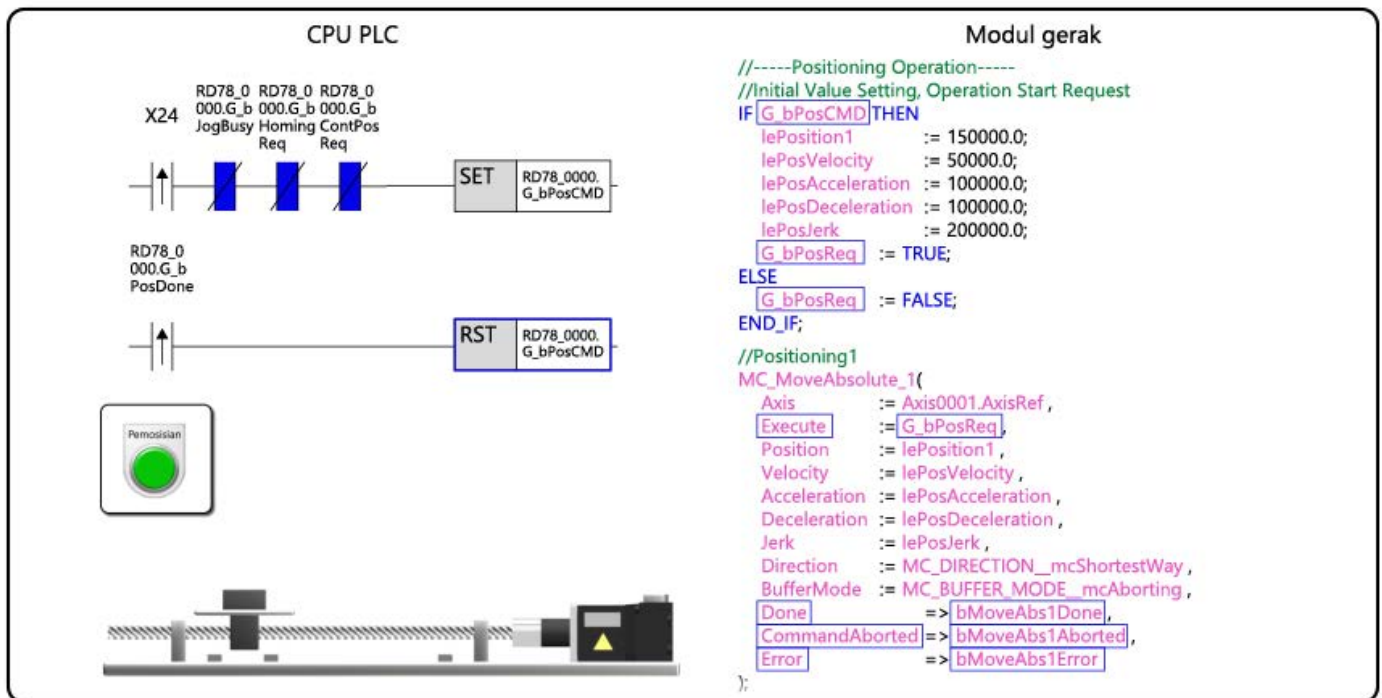
Ketika X23 dihidupkan, "RD78_0000.G_bHomingCMD" dijalankan. "G_bHomingCMD" di sisi modul Gerak menyala, dan "G_bHomingReq", yang merupakan perintah eksekusi MC_Home_1, menyala.



Ketika pengembalian posisi awal selesai, output Selesai dan "G_bHomingDone" menyala.
 "G_bHomingCMD" di sisi CPU PLC diatur ulang ke keadaan awal.

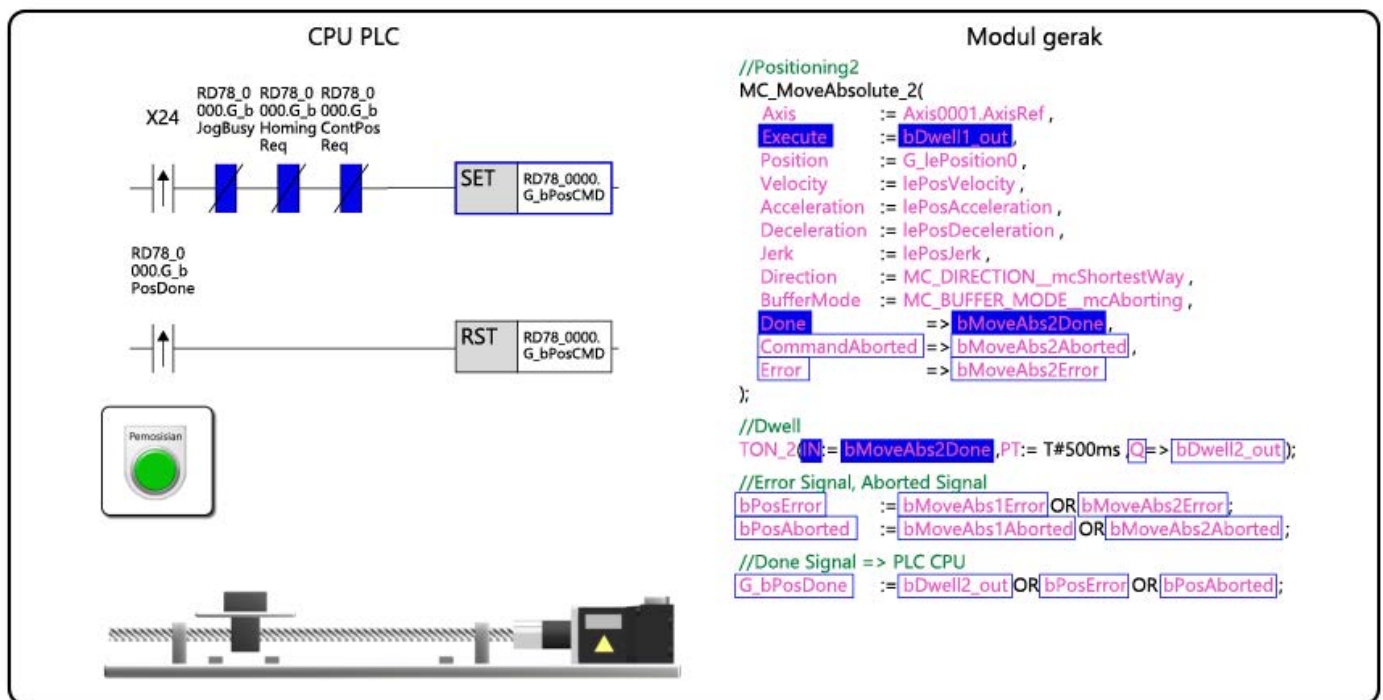


Mengaktifkan pemosisian dimulai (X24) akan memulai gerakan bolak-balik.
Axis bergerak maju 150 mm dan berhenti selama 0,5 detik, serta mundur 150 mm dan berhenti selama 0,5 detik.

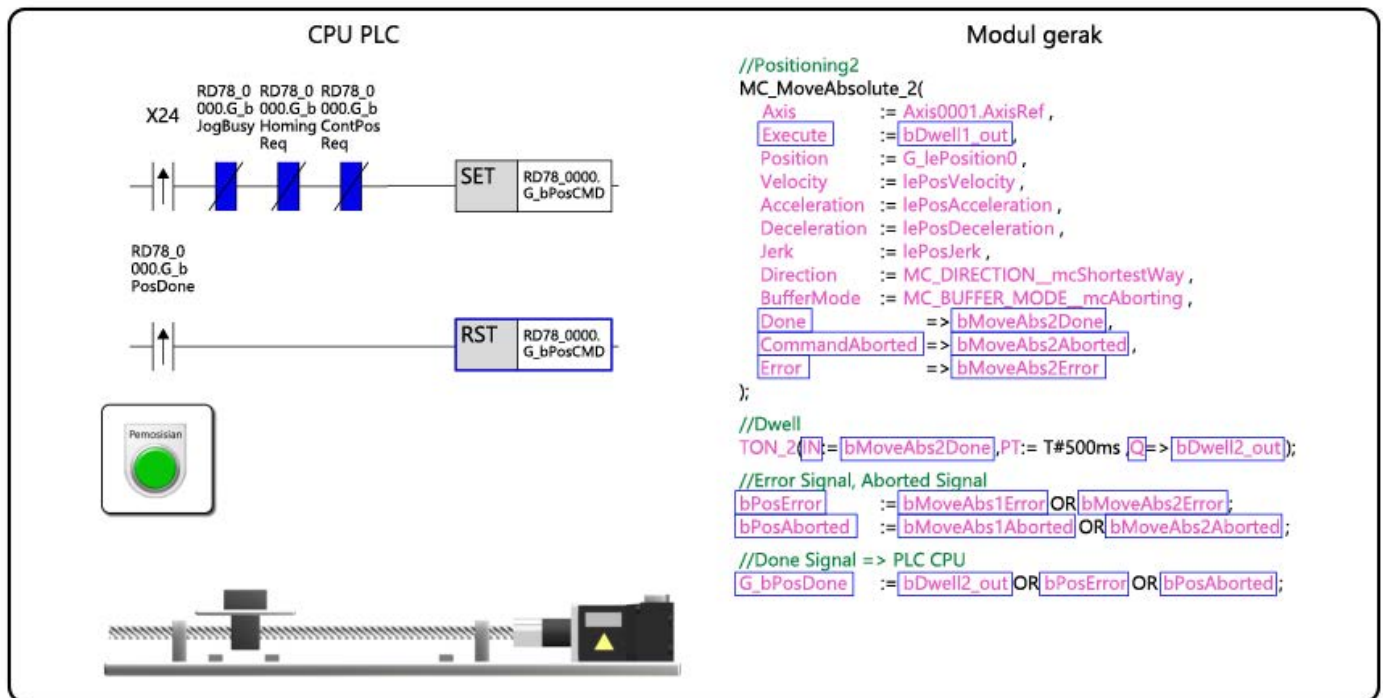


Periksa monitor program.

Ketika X24 dihidupkan, "RD78_0000.G_bPosCMD" dijalankan. "G_bPosCMD" di sisi modul Gerak menyala, dan "G_bPosReq", yang merupakan perintah eksekusi MC_MoveAbsolute_1, menyala.



Setelah gerakan bolak-balik selesai dan waktu tunggu berlalu,
 "G_bPosDone" menyala.
 "G_bPosCMD" di sisi CPU PLC diatur ulang ke keadaan awal.



Tindakan ini akan menyelesaikan pemeriksaan operasi.
Buka halaman berikutnya.

Dalam bab ini, Anda telah mempelajari:

- Apa itu Label Publik?
- Pengaturan Label Publik
- Contoh program
- Program Penulisan
- Pemeriksaan Operasi

Poin penting

Apa itu Label Publik?	• Label publik adalah label bersama yang dapat digunakan di modul Gerak dan CPU PLC.
Pengaturan Label Publik	• Daftarkan label publik dari label global modul Gerak. • Pilih apakah setiap label akan dibaca atau ditulis dari/ke CPU PLC. • Untuk mengatur anggota jenis data terstruktur yang disiapkan dalam sistem ke label publik, daftarkan label publik berdasarkan jenis lapisan data terstruktur. • Setelah mengatur label publik di modul Gerak, bangun kembali semua program dan cerminkan label publik. • Label publik didaftarkan ke label modul di sisi CPU PLC.
Contoh program	• Pada bab ini dijelaskan contoh program berikut: program tangga CPU PLC yang menggunakan label publik untuk menukar sinyal awal pemosisian dan sinyal penyelesaian pemosisian.
Program Penulisan	• Tulis data ke CPU PLC terlebih dahulu, lalu modul Gerak.
Pemeriksaan Operasi	• Anda telah memeriksa pengoperasian program contoh di video.

Mode buffer menjalankan operasi secara terus menerus dengan memulai beberapa FB operasi dari Motion control FB. Pengaturan ini dapat diatur melalui input BufferMode dari Motion control FB. Dapat memulai hingga dua FB secara bersamaan untuk setiap axis dan grup sumbu.

(Contoh) MC_MoveAbsolute

```
MC_MoveAbsolute_1(
  Axis      := Axis0001.AxisRef ,
  Execute   := G_bPositioningReq ,
  ContinuousUpdate := FALSE ,
  Position  := lePosition1 ,
  Velocity  := lePosVelocity ,
  Acceleration := lePosAcceleration ,
  Deceleration := lePosDeceleration ,
  Jerk      := lePosJerk ,
  Direction := MC_DIRECTION_mcShortestWay ,
  BufferMode := MC_BUFFER_MODE_mcAborting ,
  Options   := 0 , //mcAccDec
  Done      => bMoveAbs1Done ,
  CommandAborted => bMoveAbs1Aborted ,
  Error     => bMoveAbs1Error
);
```

0 atau MC_BUFFER_MODE_mcAborting

*** FB yang sedang dieksekusi akan terhenti dan FB berikutnya segera dijalankan.

1 atau MC_BUFFER_MODE_mcBuffered

*** Setelah pengoperasian FB yang dijalankan selesai, FB berikutnya dijalankan.

2 atau MC_BUFFER_MODE_mcBlendingLow

*** Kecepatan target yang lebih rendah untuk FB yang sedang dieksekusi dan FB yang akan di-buffer ditetapkan sebagai kecepatan switching.

3 atau MC_BUFFER_MODE_mcBlendingPrevious

*** Kecepatan target FB yang dijalankan ditetapkan sebagai kecepatan switching.

4 atau MC_BUFFER_MODE_mcBlendingNext

*** Kecepatan target FB yang akan di-buffer ditetapkan sebagai kecepatan switching.

5 atau MC_BUFFER_MODE_mcBlendingHigh

*** Kecepatan target yang lebih tinggi untuk FB yang sedang dieksekusi dan FB yang akan di-buffer ditetapkan sebagai kecepatan switching.

[Poin]

Untuk input Direction dan BufferMode, tentukan angka atau enumerator ENUM dimulai dengan MC_BUFFER_MODE dan MC_DIRECTION.

Untuk detail tentang enumerator ENUM, lihat panduan berikut.



MELSEC iQ-R Programming Manual (Motion Control Function Blocks)

2 VARIABLES AND MOTION CONTROL FB

2.2 List of Enumerators

Diagram berikut menunjukkan operasi saat BufferMode diatur ke 0:mcAborting. FB yang sedang dieksekusi akan terhenti dan FB berikutnya segera dijalankan.

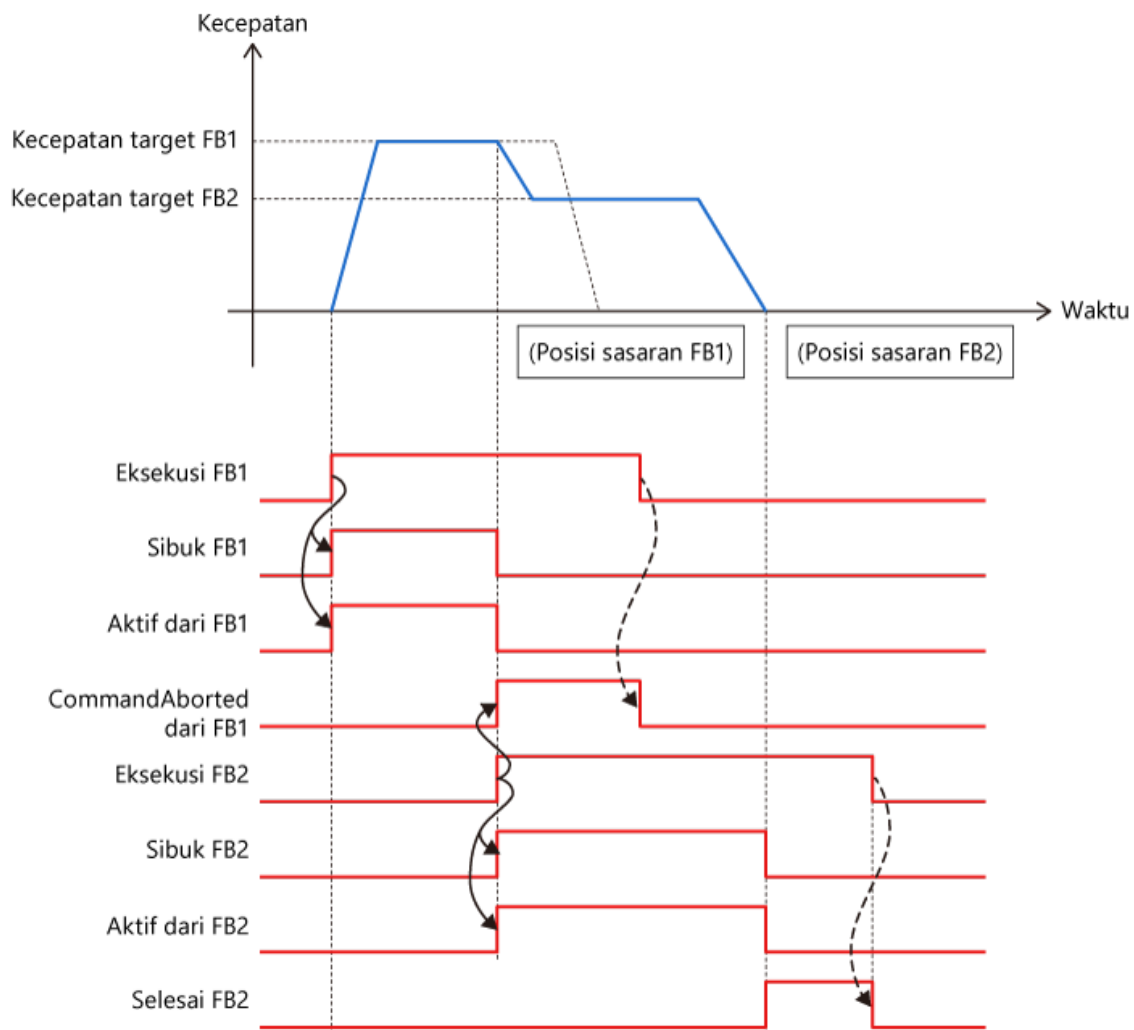
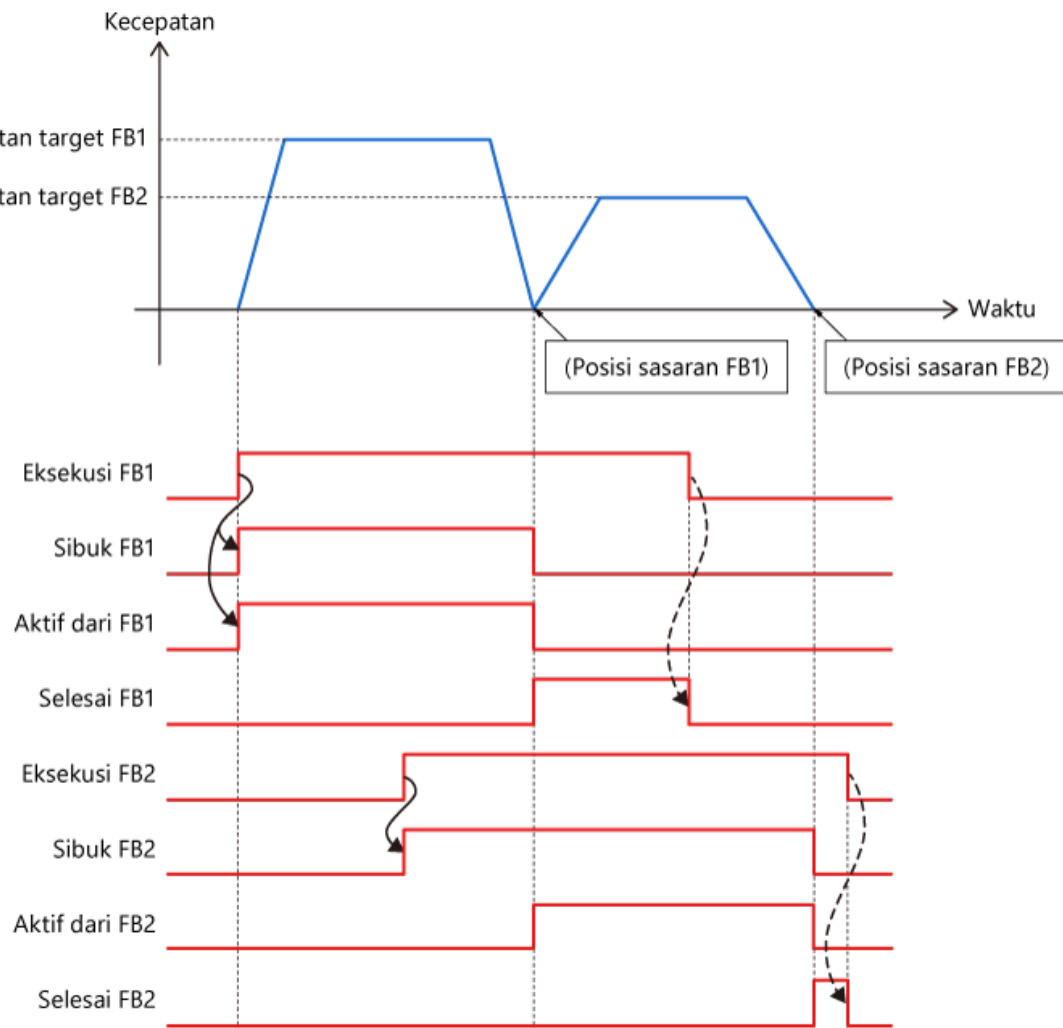


Diagram berikut menunjukkan operasi saat BufferMode diatur ke 1:mcBuffered. Setelah pengoperasian FB yang dijalankan selesai, FB berikutnya dijalankan.



Saat BufferMode diatur ke mcBlending***, FB berikutnya akan terus dieksekusi setelah posisi target FB yang sedang dieksekusi tercapai.

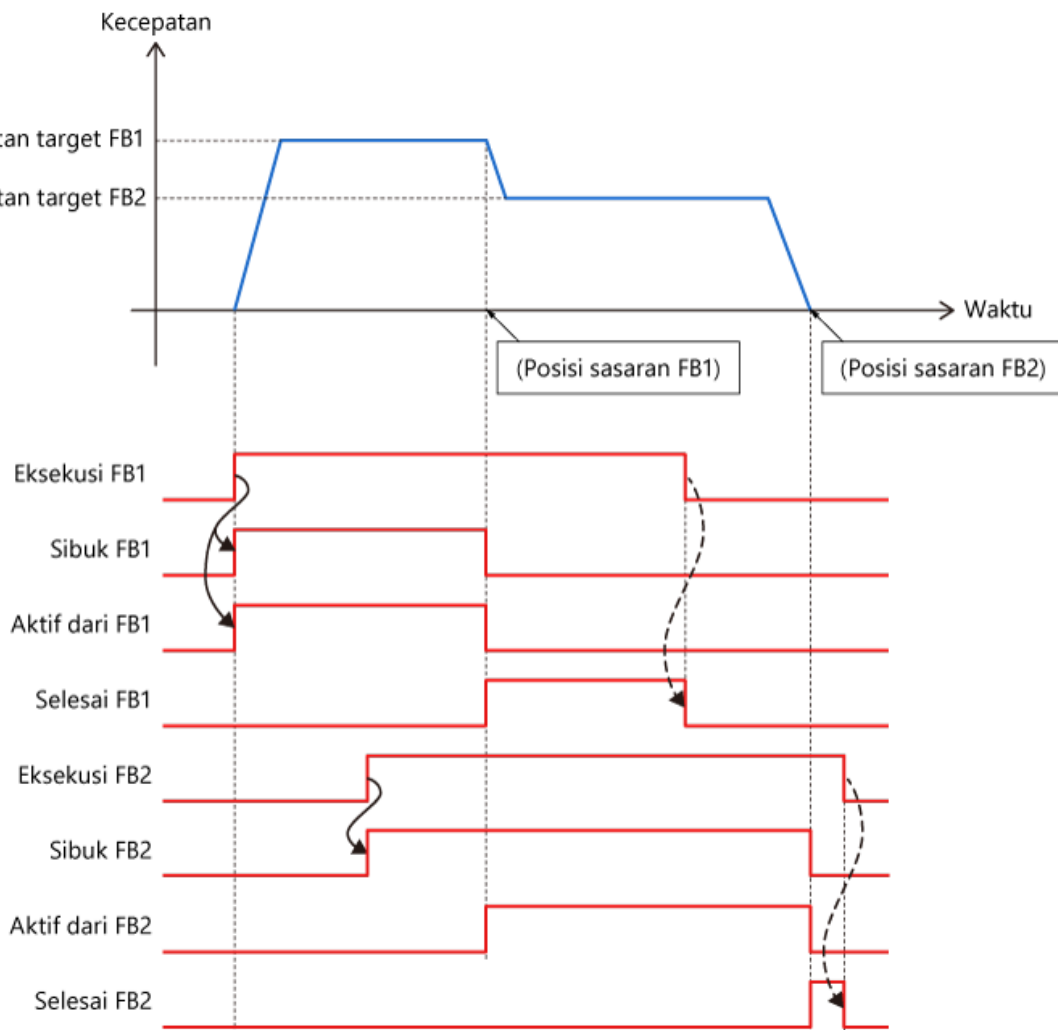
Pada uraian berikut, FB yang akan dieksekusi terlebih dahulu adalah FB1, dan FB yang akan di-buffer adalah FB2.

(1) BlendingPrevious

Diagram berikut menunjukkan operasi saat BufferMode diatur ke 3: mcBlendingPrevious.

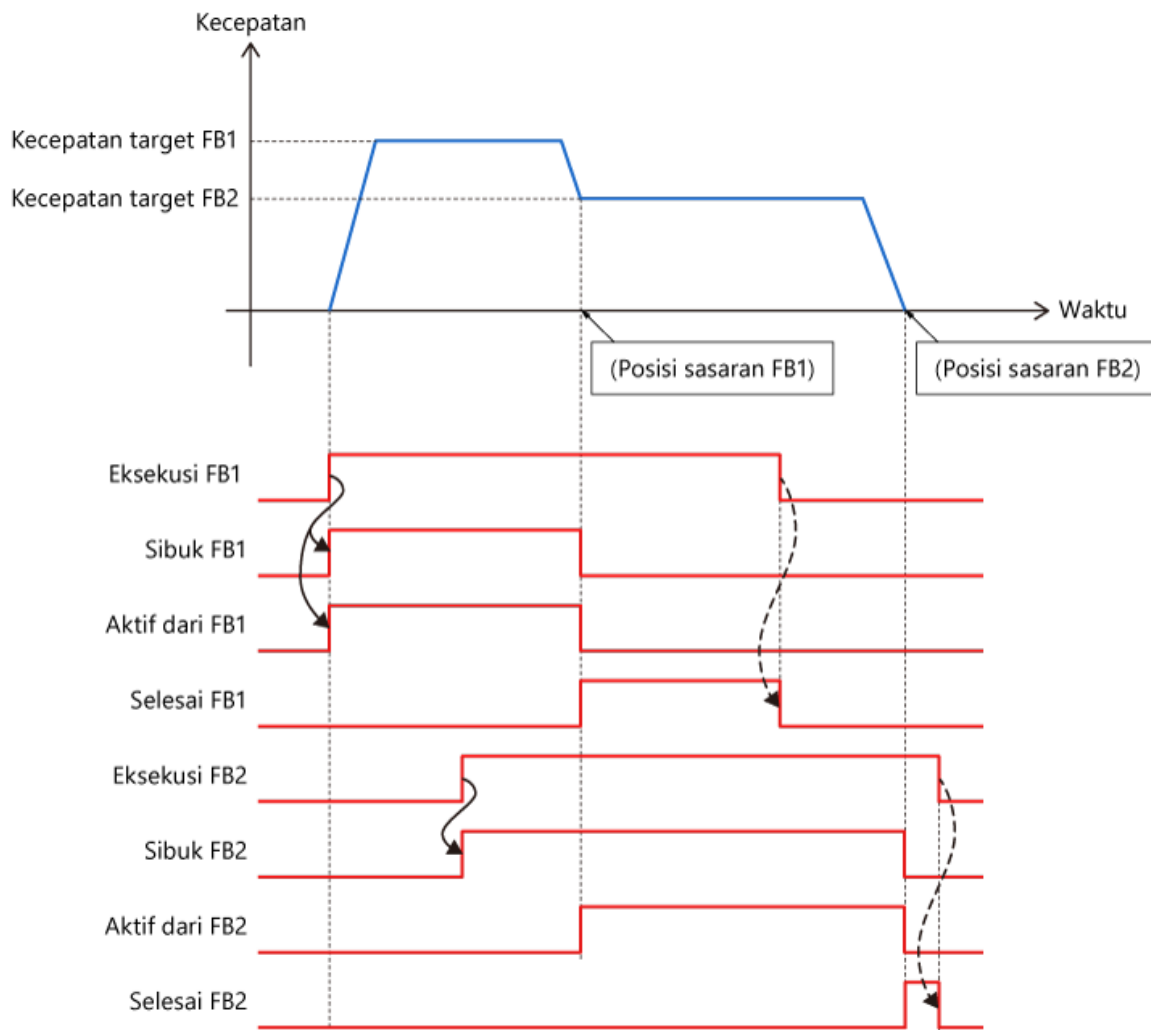
Operasi dilakukan pada kecepatan target FB1 hingga posisi target FB1.

Ketika operasi dialihkan ke FB2, kecepatan akan diubah ke kecepatan target FB2, dan berpindah ke posisi target FB2.



(2) BlendingNext

Diagram berikut menunjukkan operasi saat BufferMode diatur ke 4: mcBlendingNext. Kecepatan berubah ke kecepatan target FB2 saat operasi mencapai posisi target FB1.



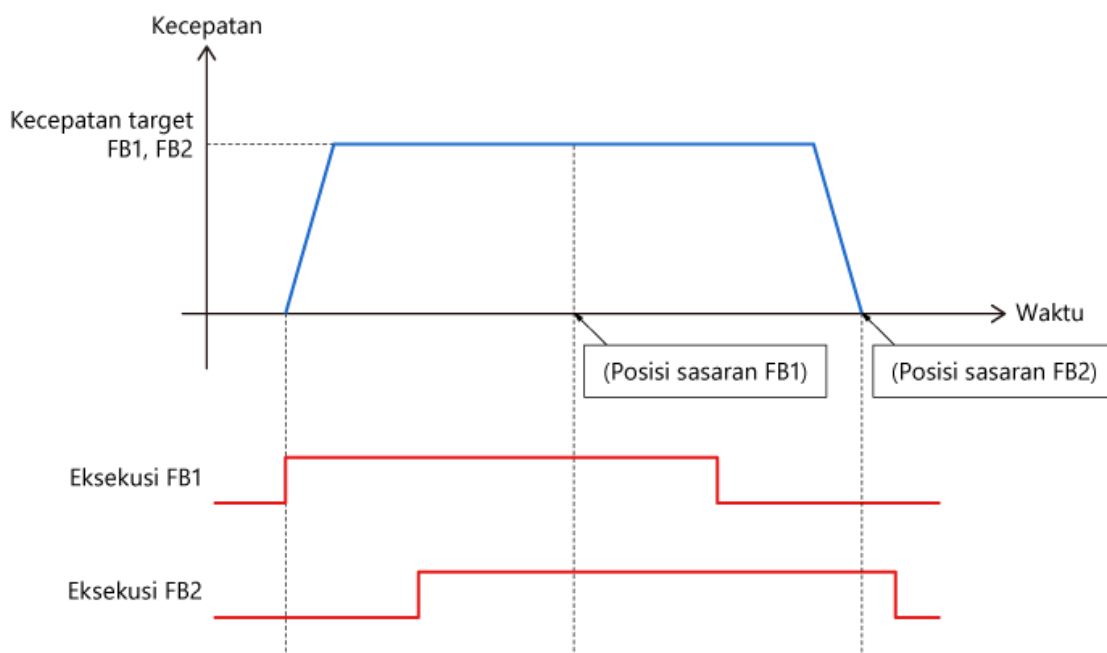
(3) BlendingLow, BlendingHigh

Operasi saat BufferMode diatur ke 2: mcBlendingLow atau 5: mcBlendingHigh bervariasi tergantung pada kecepatan target mana antara FB1 dan FB2 yang lebih besar.

Menetapkan nilai	Kecepatan target FB1 > Kecepatan target FB2	Kecepatan target FB1 < Kecepatan target MFB2
2: mcBlendingLow	Operasi yang sama seperti BlendingPrevious	Operasi yang sama seperti BlendingNext
5: mcBlendingHigh	Operasi yang sama seperti BlendingNext	Operasi yang sama seperti BlendingPrevious

[Poin]

Diagram berikut menunjukkan bentuk gelombang kecepatan untuk BlendingPrevious, BlendingNext, BlendingHigh, dan BlendingLow ketika kecepatan target FB1 dan FB2 sama.



(1) Contoh operasi program

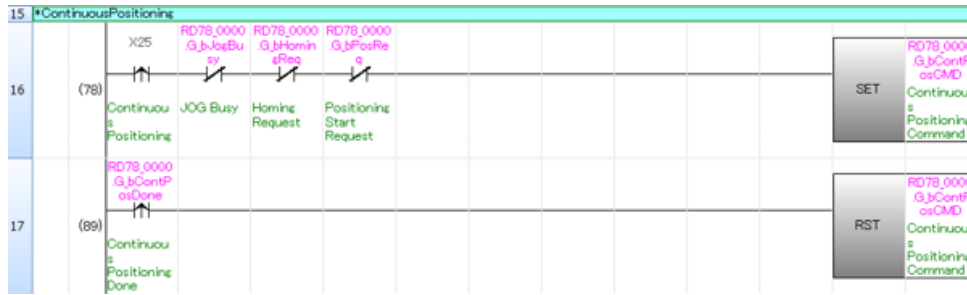
Bab ini menggunakan contoh program yang digunakan pada Bab 2.

Periksa perbedaan operasi mode buffer dalam program yang dimulai dengan X25.

Item	FB1 (MC_MoveAbsolute)	FB2 (MC_MoveAbsolute)
Alamat pemosisian	75000.0 [μm]	150000.0 [μm]
Kecepatan	50000.0 [$\mu\text{m/s}$]	25000.0 [$\mu\text{m/s}$]
Akselerasi, deselerasi	100000.0 [$\mu\text{m/s}^2$]	50000.0 [$\mu\text{m/s}^2$]
Sentakan	200000.0 [$\mu\text{m/s}^3$]	100000.0 [$\mu\text{m/s}^3$]

(2) Program CPU PLC

MAIN (tangga, program pemindaian)

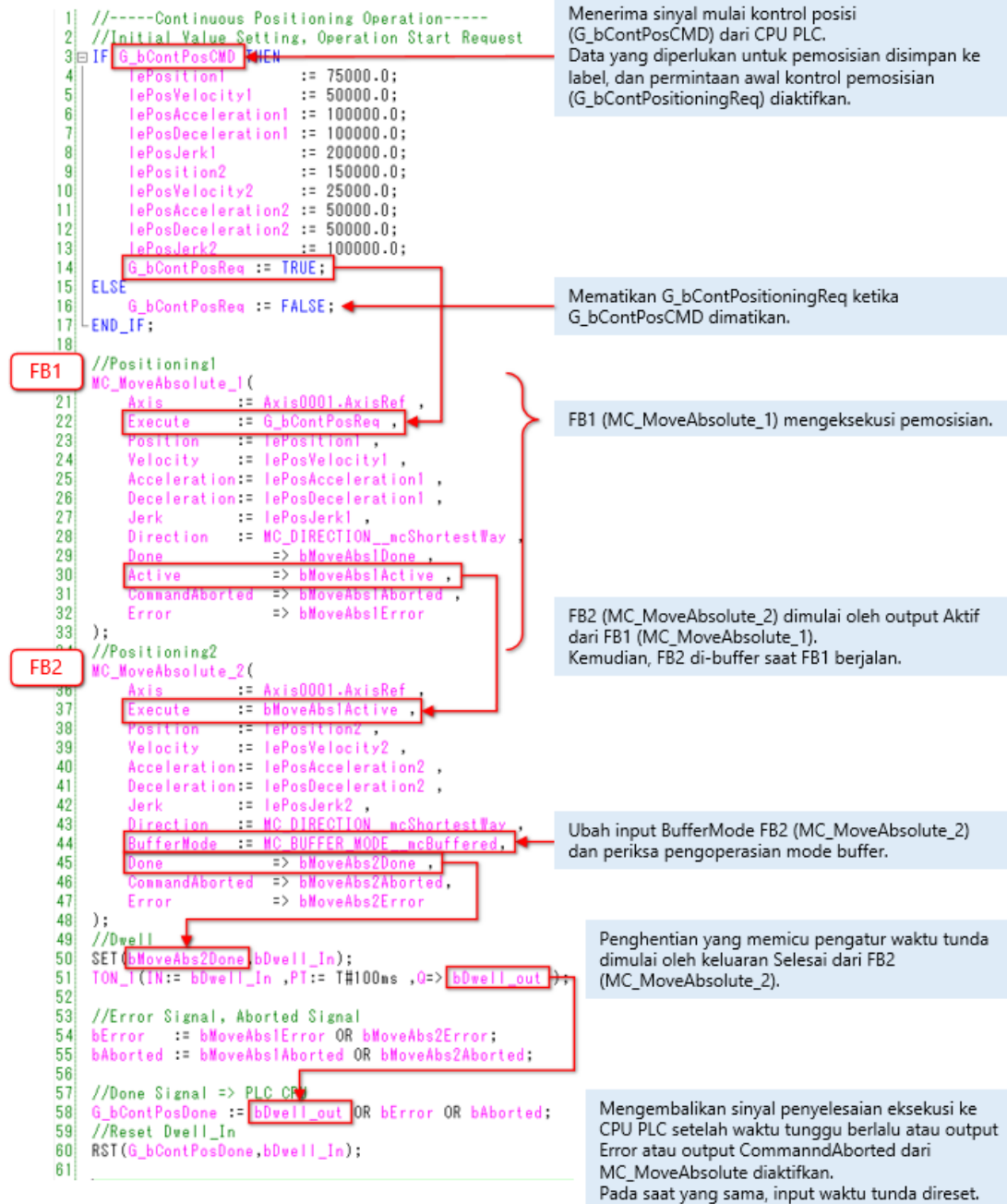


Tepi naik dari awal kontrol pemosisian (X25) disimpan di G_bContPosCMD, dan dikirim ke modul Gerak sebagai kondisi awal dari kontrol pemosisian.

Interlock diatur untuk mencegah pengembalian pemosisian dimulai saat program lain sedang berjalan. Setelah menerima bahwa modul Gerak menyalakan sinyal penyelesaian pengembalian posisi awal, G_bContPosCMD diatur ulang pada tepi naik sinyal tersebut.

(3) Program modul gerak

Pemosisian Berkelanjutan (tipe eksekusi normal)



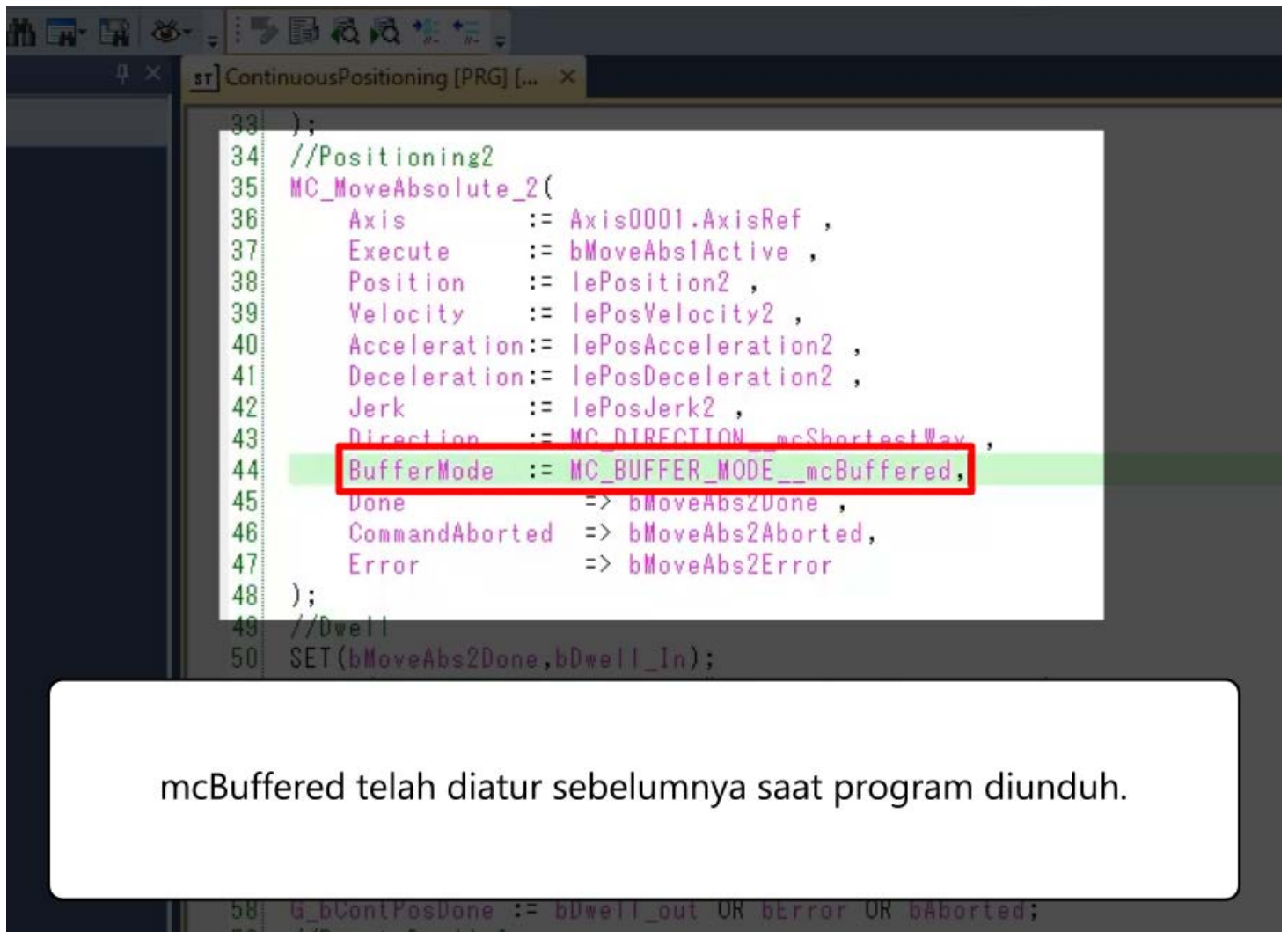
Klik tombol putar di kiri bawah jendela.

```
33 );  
34 //Positioning2  
35 MC_MoveAbsolute_2(  
36     Axis      := Axis0001.AxisRef ,  
37     Execute   := bMoveAbs1Active ,  
38     Position  := lePosition2 ,  
39     Velocity  := lePosVelocity2 ,  
40     Acceleration:= lePosAcceleration2 ,  
41     Deceleration:= lePosDeceleration2 ,  
42     Jerk      := lePosJerk2 ,  
43     Direction  := MC_DIRECTION__mcShortestWay ,  
44     BufferMode := MC_BUFFER_MODE__mcBuffered ,  
45     Done       => bMoveAbs2Done ,  
46     CommandAborted => bMoveAbs2Aborted ,  
47     Error      => bMoveAbs2Error  
48 );  
49 //Dwell  
50 SET(bMoveAbs2Done,bDwell_In);
```

Periksa operasi mode buffer.

Buka "LinearInterpolation" di program contoh dan ubah input BufferMode MC_MoveAbsolute_2 untuk memeriksa pengoperasian mode buffer.

```
58 G_bContPosDone := bDwell_out OR bError OR bAborted;  
59 //Dwell out
```



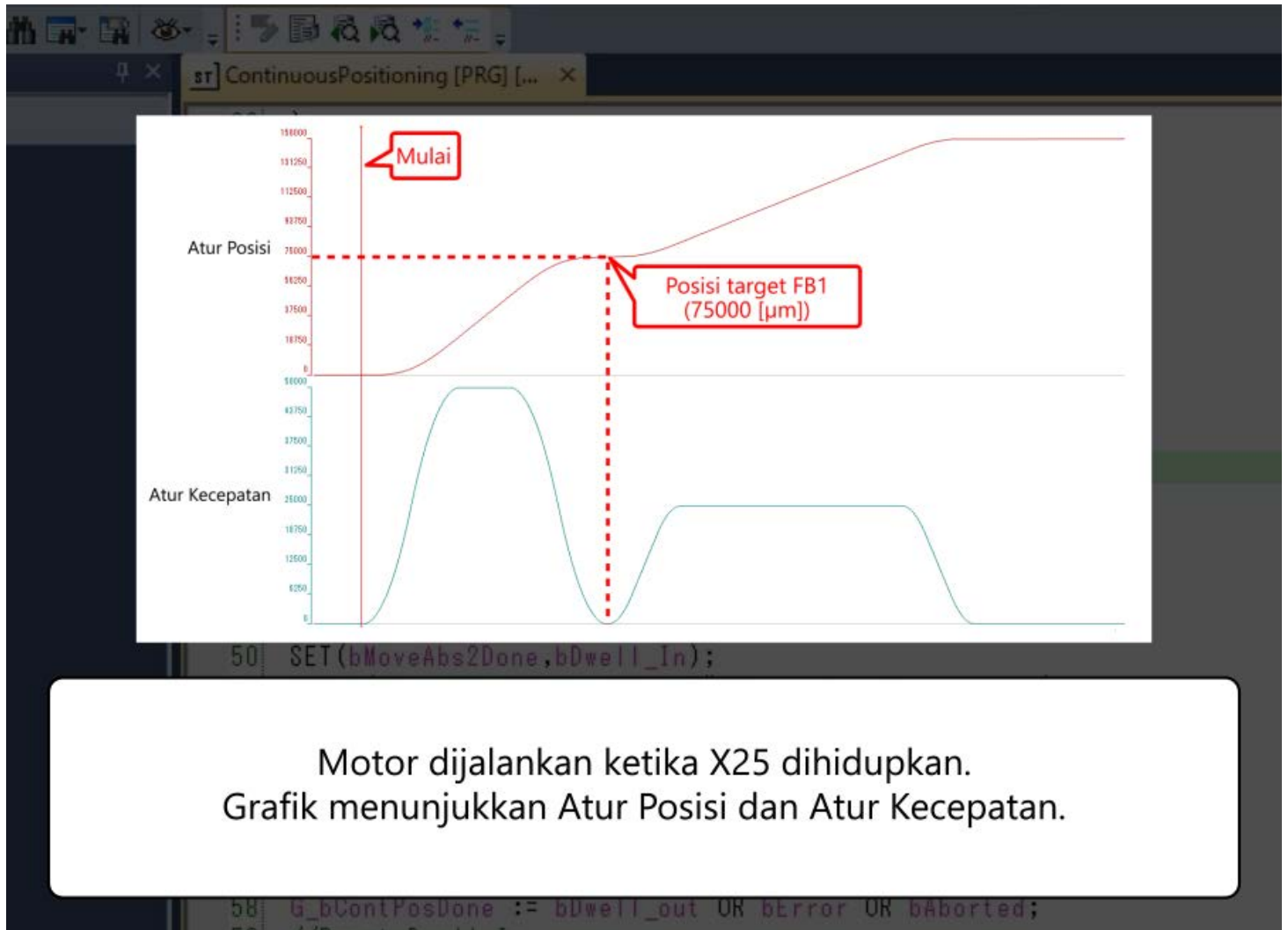
```
33 );
34 //Positioning2
35 MC_MoveAbsolute_2(
36     Axis      := Axis0001.AxisRef ,
37     Execute   := bMoveAbs1Active ,
38     Position  := lePosition2 ,
39     Velocity  := lePosVelocity2 ,
40     Acceleration:= lePosAcceleration2 ,
41     Deceleration:= lePosDeceleration2 ,
42     Jerk      := lePosJerk2 ,
43     Direction  := MC_DIRECTION__mcShortestWay ,
44     BufferMode := MC_BUFFER_MODE__mcBuffered ,
45     Done       => bMoveAbs2Done ,
46     CommandAborted => bMoveAbs2Aborted ,
47     Error      => bMoveAbs2Error
48 );
49 //Dwell
50 SET(bMoveAbs2Done,bDwell_In);
```

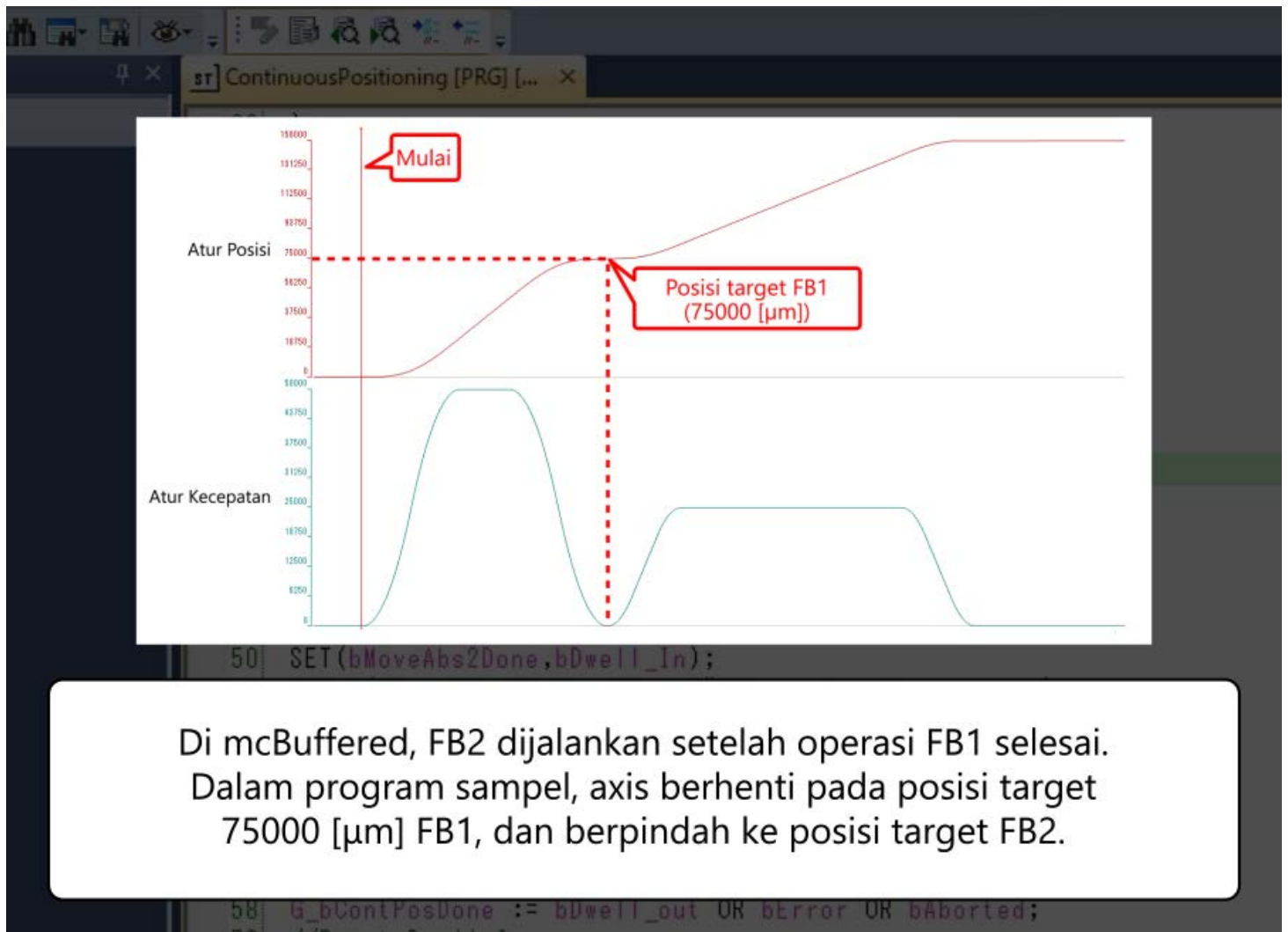
mcBuffered telah diatur sebelumnya saat program diunduh.

```
33 );  
34 //Positioning2  
35 MC_MoveAbsolute_2(  
36     Axis      := Axis0001.AxisRef ,  
37     Execute   := bMoveAbs1Active ,  
38     Position  := lePosition2 ,  
39     Velocity   := lePosVelocity2 ,  
40     Acceleration := lePosAcceleration2 ,  
41     Deceleration := lePosDeceleration2 ,  
42     Jerk      := lePosJerk2 ,  
43     Direction  := MC_DIRECTION__mcShortestWay ,  
44     BufferMode := MC_BUFFER_MODE__mcBuffered ,  
45     Done       => bMoveAbs2Done ,  
46     CommandAborted => bMoveAbs2Aborted ,  
47     Error      => bMoveAbs2Error  
48 );  
49 //Dwell  
50 SET(bMoveAbs2Done,bDwell_In);
```

Pertama, periksa pengoperasian mcBuffered.
Periksa apakah input BufferMode dari MC_MoveAbsolute_2 diatur ke "MC_BUFFER_MODE__mcBuffered", dan tulis program ke modul Gerak.

```
58 G_bContPosDone := bDwell_out OR bError OR bAborted;
```

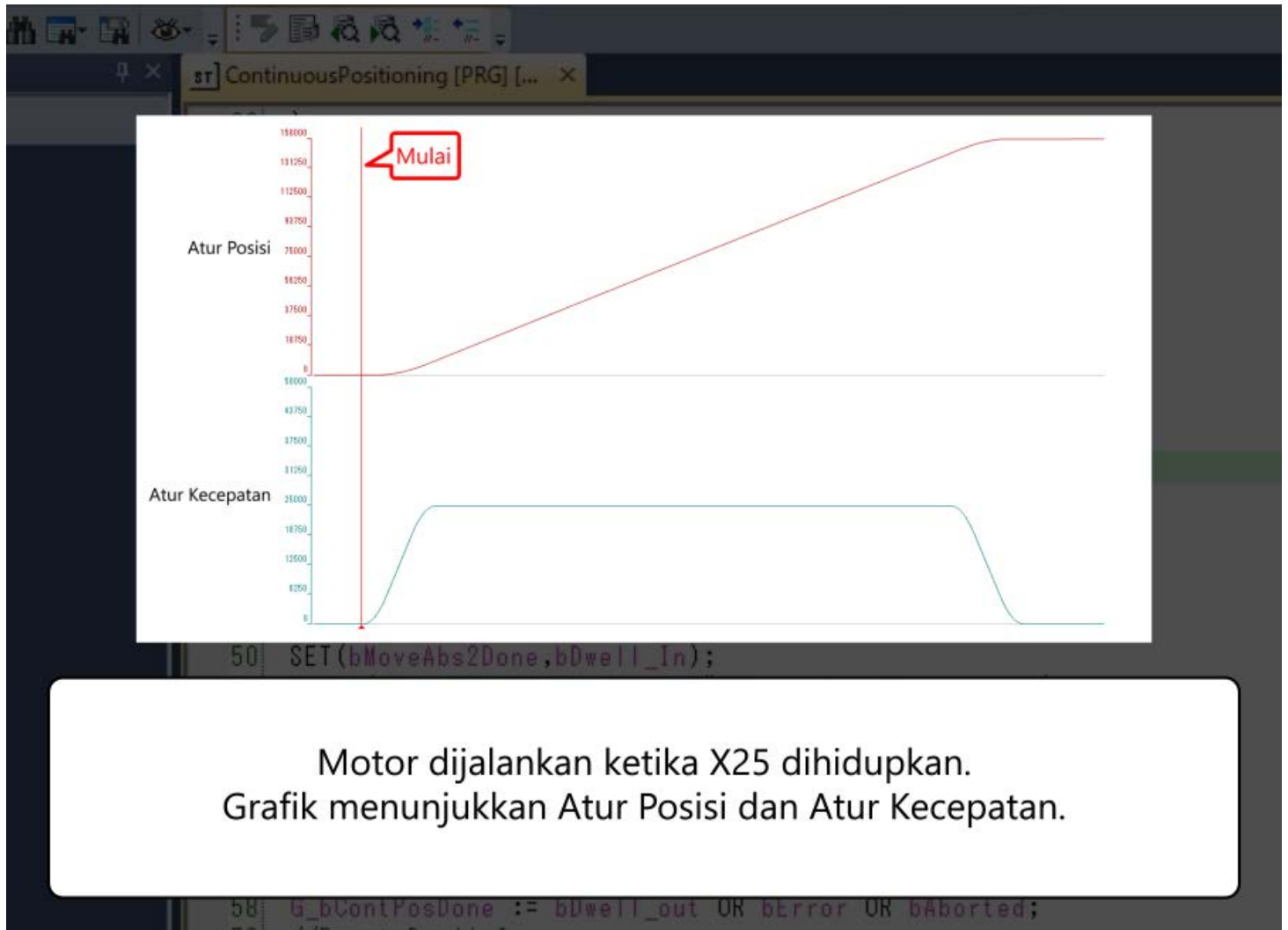




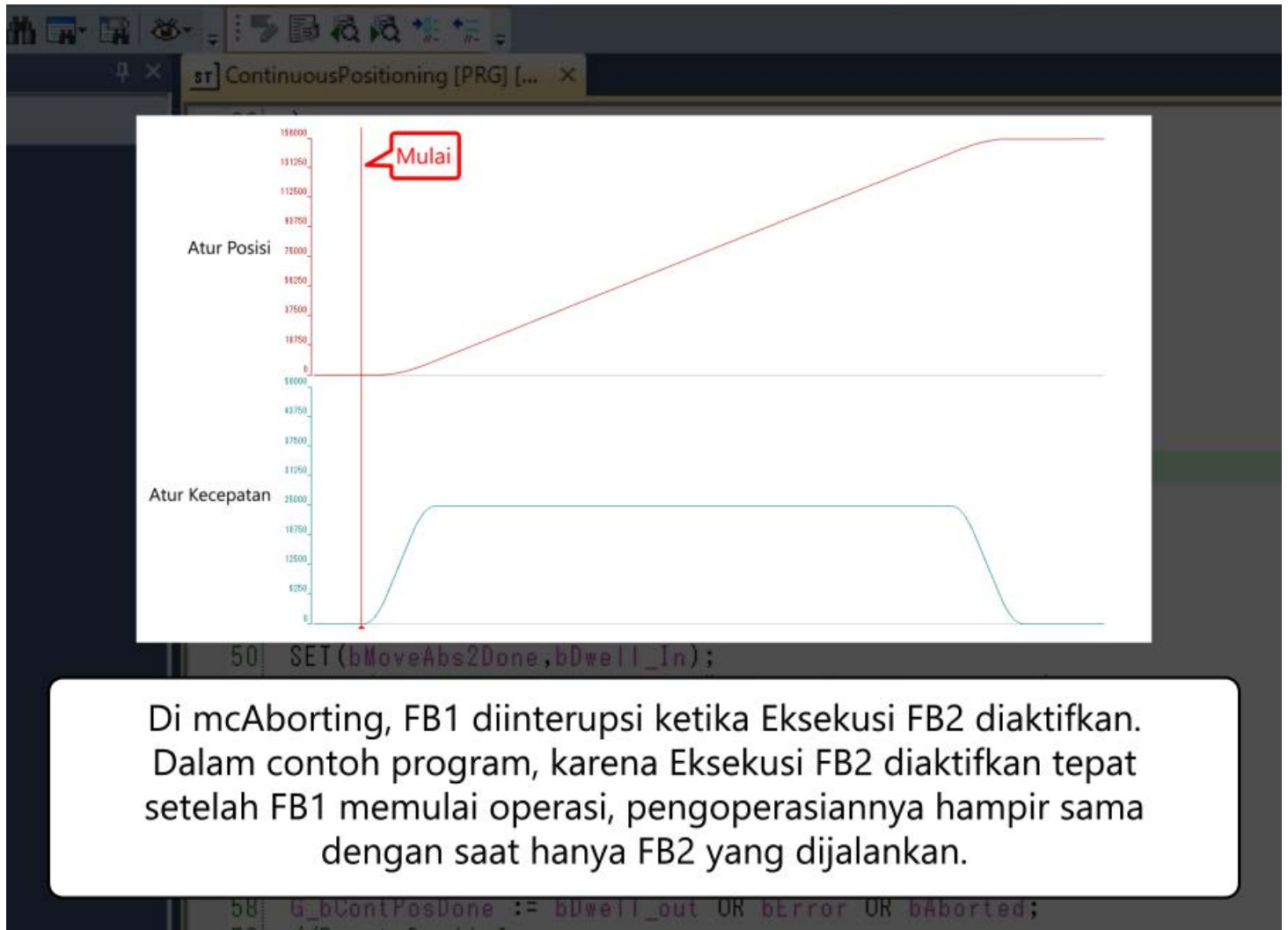
Di mcBuffered, FB2 dijalankan setelah operasi FB1 selesai. Dalam program sampel, axis berhenti pada posisi target 75000 μm FB1, dan berpindah ke posisi target FB2.

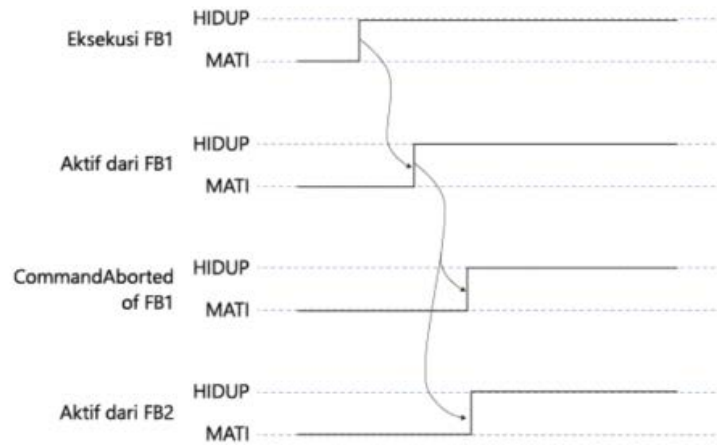
```
33 );  
34 //Positioning2  
35 MC_MoveAbsolute_2(  
36     Axis      := Axis0001.AxisRef ,  
37     Execute   := bMoveAbs1Active ,  
38     Position  := lePosition2 ,  
39     Velocity  := lePosVelocity2 ,  
40     Acceleration:= lePosAcceleration2 ,  
41     Deceleration:= lePosDeceleration2 ,  
42     Jerk      := lePosJerk2 ,  
43     Direction  := MC_DIRECTION__mcShortestWay ,  
44     BufferMode := MC_BUFFER_MODE__mcAborting ,  
45     Done       => bMoveAbs2Done ,  
46     CommandAborted => bMoveAbs2Aborted ,  
47     Error      => bMoveAbs2Error  
48 );  
49 //Dwell  
50 SET(bMoveAbs2Done,bDwell_In);  
  
58 G_bContPosDone := bDwell_out OR bError OR bAborted;  
59 //Dwell out
```

Selanjutnya, periksa operasi mcAborting.
Ubah input BufferMode dari MC_MoveAbsolute_2 menjadi
"MC_BUFFER_MODE__mcAborting", buat ulang semua program,
dan tulis ke modul Gerak.



Motor dijalankan ketika X25 dihidupkan.
Grafik menunjukkan Atur Posisi dan Atur Kecepatan.





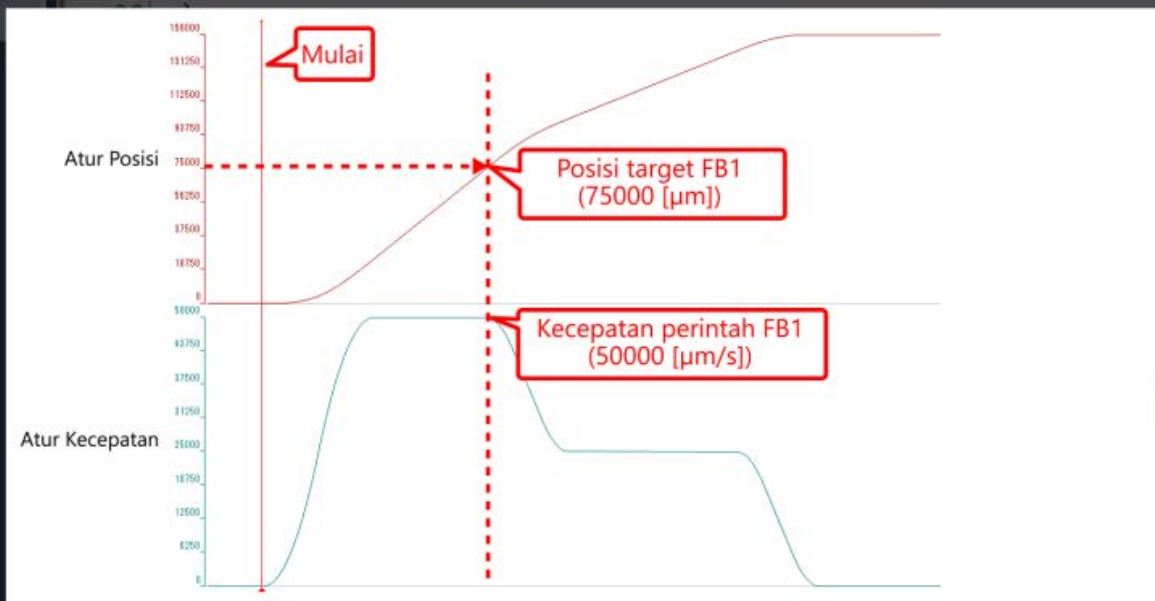
Periksa sinyal I/O FB1 dan FB2 pada saat dimulai.
Output CommandAborted dari FB1 dihidupkan,
menunjukkan bahwa FB1 terganggu.

```
50 SET(bMoveAbs2Done,bDwell_In);  
58 G_bContPosDone := bDwell_out OR bError OR bAborted;  
59 //Dwell out
```

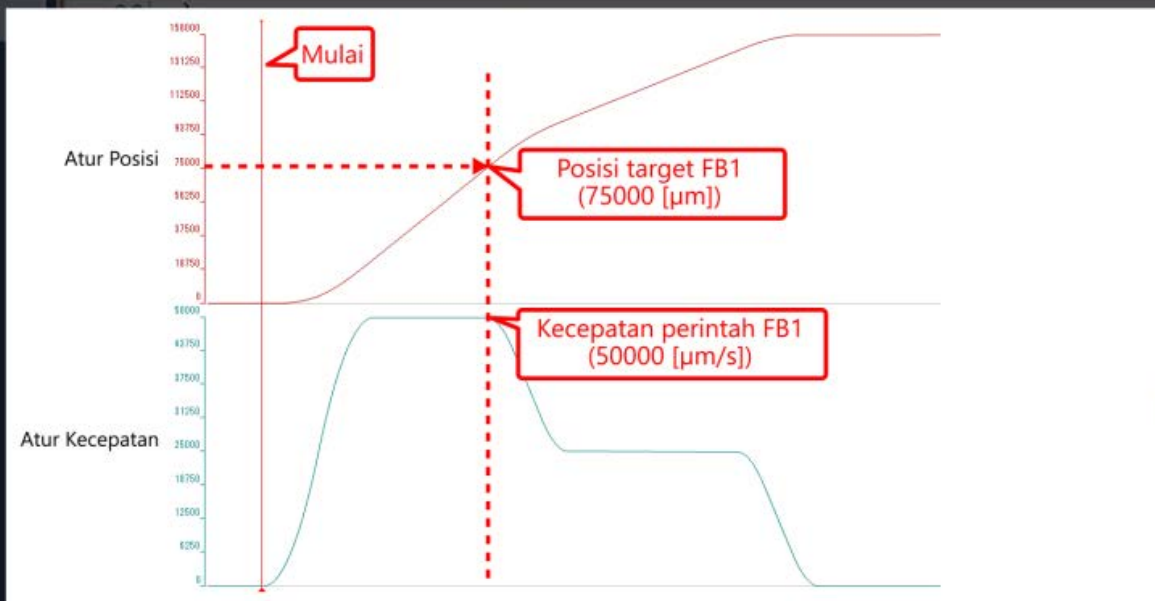
```
33 );  
34 //Positioning2  
35 MC_MoveAbsolute_2(  
36     Axis      := Axis0001.AxisRef ,  
37     Execute    := bMoveAbs1Active ,  
38     Position   := lePosition2 ,  
39     Velocity   := lePosVelocity2 ,  
40     Acceleration := lePosAcceleration2 ,  
41     Deceleration := lePosDeceleration2 ,  
42     Jerk       := lePosJerk2 ,  
43     Direction  := MC_DIRECTION_mcShortestWay ,  
44     BufferMode := MC_BUFFER_MODE_mcBlendingPrevious ,  
45     Done       => bMoveAbs2Done ,  
46     CommandAborted => bMoveAbs2Aborted ,  
47     Error      => bMoveAbs2Error  
48 );  
49 //Dwell  
50 SET(bMoveAbs2Done, bDwell_In);
```

Selanjutnya, periksa operasi mcBlendingPrevious. Ubah input BufferMode dari MC_MoveAbsolute_2 menjadi "MC_BUFFER_MODE_mcBlendingPrevious", buat ulang semua program, dan tulis ke modul Gerak.

```
58 G_bContPosDone := bDwell_out OR bError OR bAborted;
```



Motor dijalankan ketika X25 dihidupkan.
Grafik menunjukkan Atur Posisi dan Atur Kecepatan.



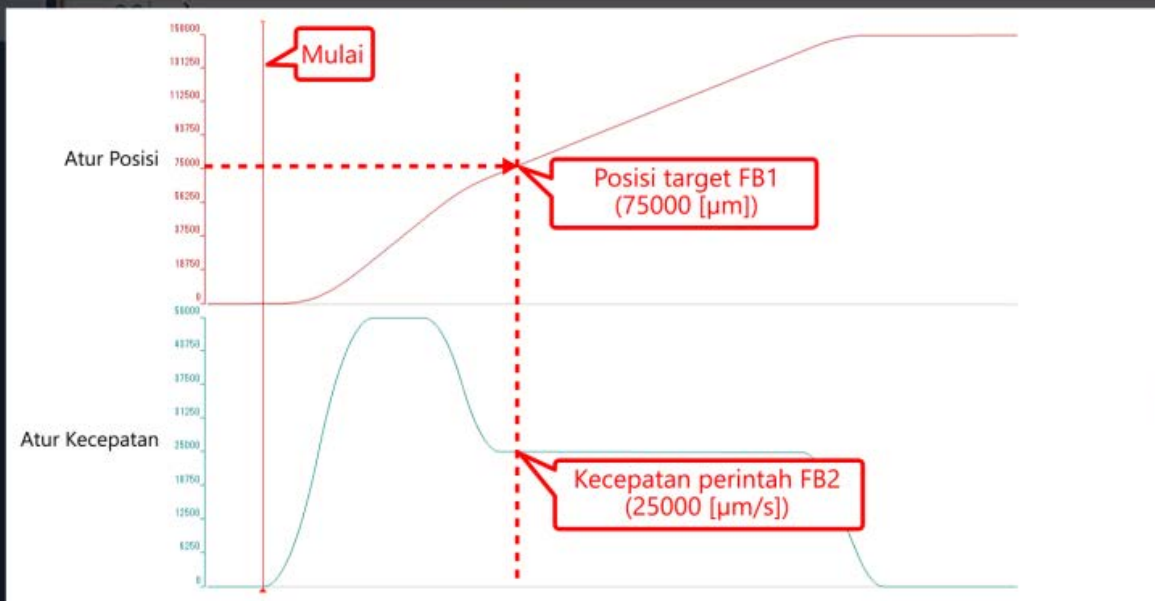
Di mcBlendingPrevious, kecepatan perintah FB1 diterapkan ketika posisi target FB1 tercapai. Kemudian, kecepatan perintah FB2 akan diterapkan, dan axis bergerak ke posisi target FB2.

58 G_bContPosDone := bDwell_out OR bError OR bAborted;
59 //Dwell out

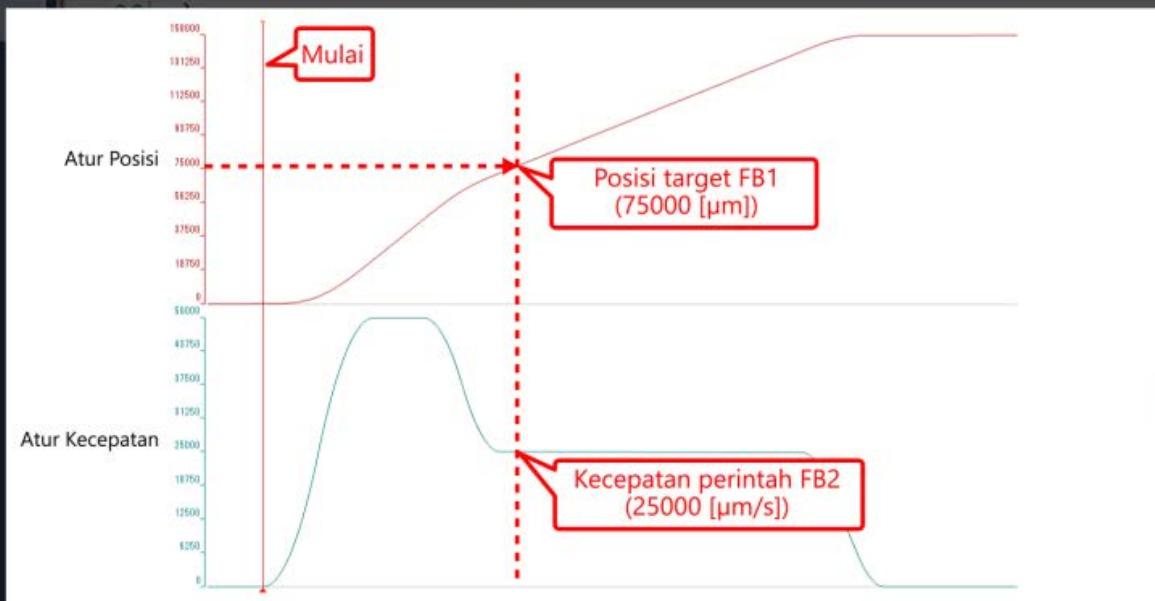
```
33 );  
34 //Positioning2  
35 MC_MoveAbsolute_2(  
36     Axis      := Axis0001.AxisRef ,  
37     Execute   := bMoveAbs1Active ,  
38     Position  := lePosition2 ,  
39     Velocity   := lePosVelocity2 ,  
40     Acceleration:= lePosAcceleration2 ,  
41     Deceleration:= lePosDeceleration2 ,  
42     Jerk      := lePosJerk2 ,  
43     Direction  := MC_DIRECTION__mcShortestWay ,  
44     BufferMode := MC_BUFFER_MODE__mcBlendingNext ,  
45     Done       => bMoveAbs2Done ,  
46     CommandAborted => bMoveAbs2Aborted ,  
47     Error      => bMoveAbs2Error  
48 );  
49 //Dwell  
50 SET(bMoveAbs2Done,bDwell_In);
```

Selanjutnya, periksa operasi mcBlendingNext.
Ubah input BufferMode dari MC_MoveAbsolute_2 menjadi
"MC_BUFFER_MODE__mcBlendingNext", buat ulang semua program,
dan tulis ke modul Gerak.

```
58 G_bContPosDone := bDwell_out OR bError OR bAborted;  
59 //Dwell out
```



Motor dijalankan ketika X25 dihidupkan.
Grafik menunjukkan Atur Posisi dan Atur Kecepatan.

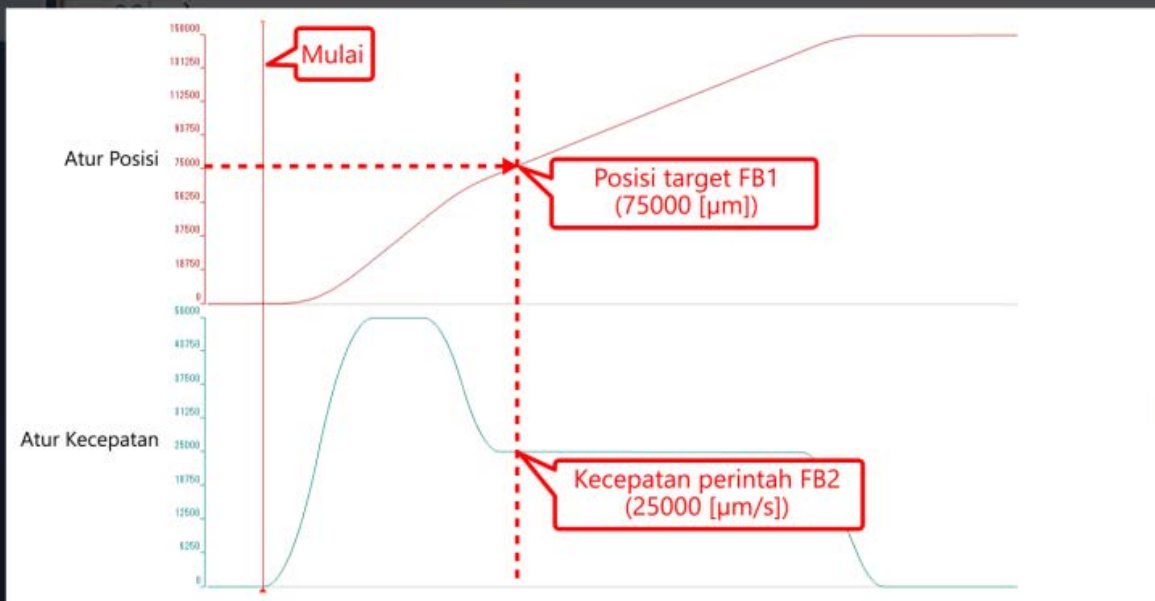


Di mcBlendingNext, kecepatan perintah FB1 diterapkan ketika posisi target FB2 tercapai. Kemudian axis berpindah ke posisi target FB2.

```
33 );  
34 //Positioning2  
35 MC_MoveAbsolute_2(  
36     Axis      := Axis0001.AxisRef ,  
37     Execute   := bMoveAbs1Active ,  
38     Position  := lePosition2 ,  
39     Velocity   := lePosVelocity2 ,  
40     Acceleration := lePosAcceleration2 ,  
41     Deceleration := lePosDeceleration2 ,  
42     Jerk      := lePosJerk2 ,  
43     Direction  := MC_DIRECTION__mcShortestWay ,  
44     BufferMode := MC_BUFFER_MODE__mcBlendingLow ,  
45     Done       => bMoveAbs2Done ,  
46     CommandAborted => bMoveAbs2Aborted ,  
47     Error      => bMoveAbs2Error  
48 );  
49 //Dwell  
50 SET(bMoveAbs2Done,bDwell_In);
```

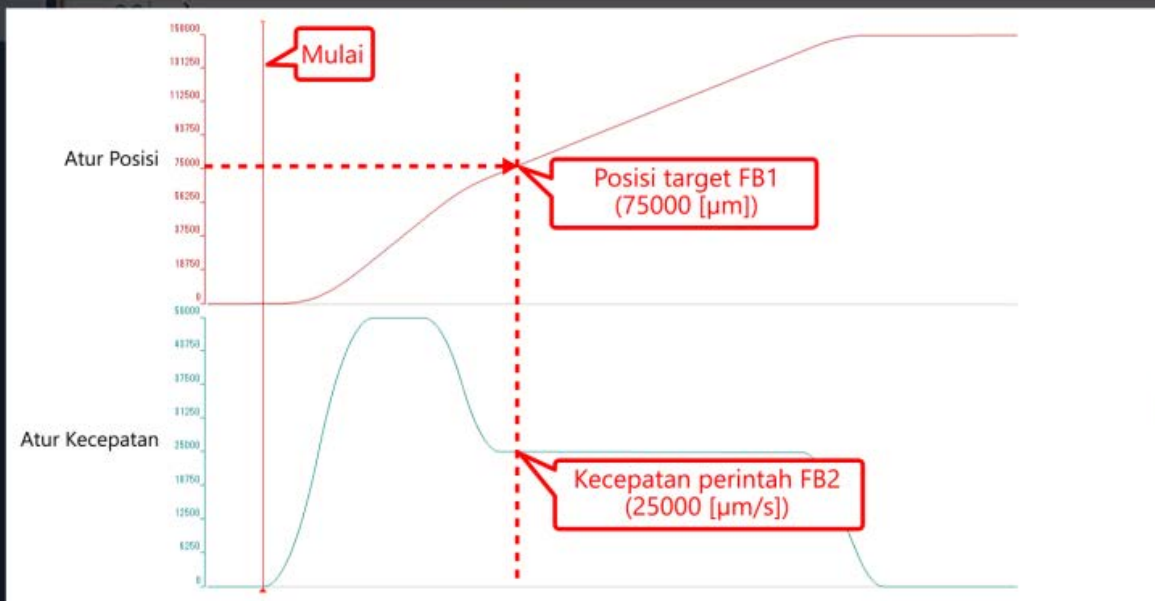
Selanjutnya, periksa operasi mcBlendingLow.
Ubah input BufferMode dari MC_MoveAbsolute_2 menjadi
"MC_BUFFER_MODE__mcBlendingLow", buat ulang semua program,
dan tulis ke modul Gerak.

```
58 G_bContPosDone := bDwell_out OR bError OR bAborted;  
59 //Dwell out
```



Motor dijalankan ketika X25 dihidupkan.
Grafik menunjukkan Atur Posisi dan Atur Kecepatan.

```
58 G_bContPosDone := bDwell_out OR bError OR bAborted;  
59 //Dwell out
```



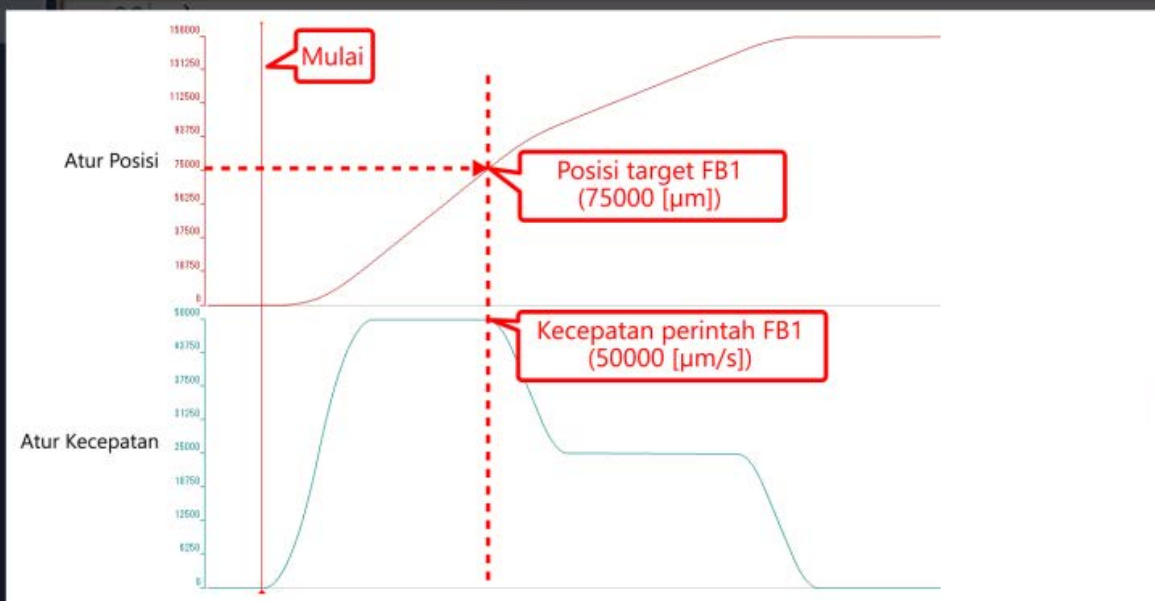
Di mcBlendingLow, kecepatan perintah lebih lambat diterapkan ketika posisi target FB1 tercapai.

Pada contoh kursus ini, karena kecepatan perintah FB2 lebih lambat, maka bentuk gelombang kecepatannya sama dengan BlendingNext.

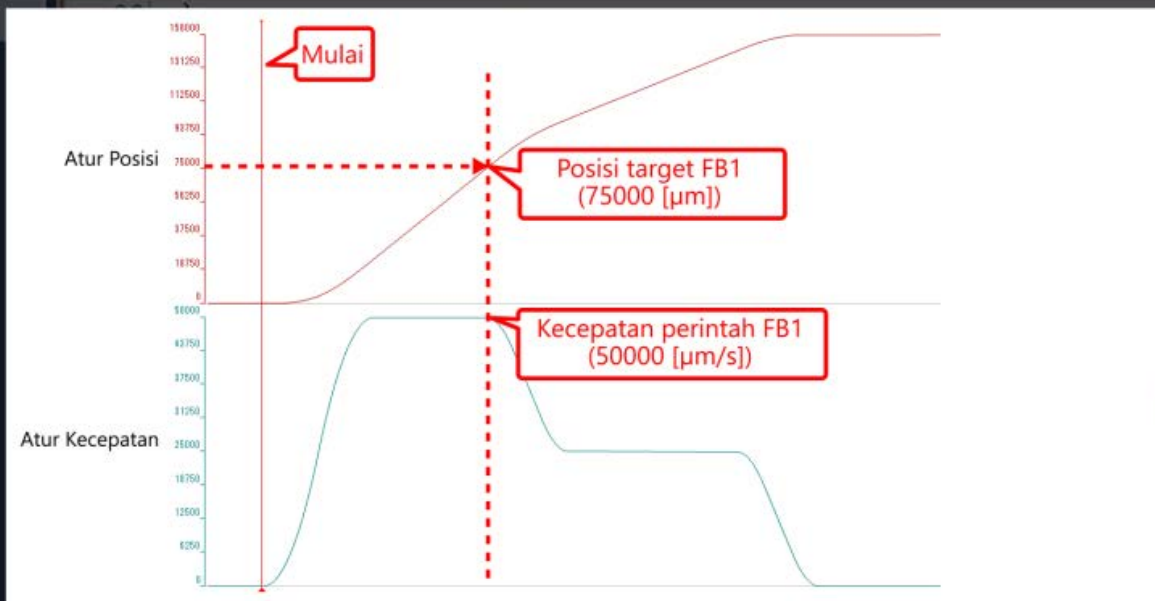
```
33 );  
34 //Positioning2  
35 MC_MoveAbsolute_2(  
36     Axis      := Axis0001.AxisRef ,  
37     Execute   := bMoveAbs1Active ,  
38     Position  := lePosition2 ,  
39     Velocity   := lePosVelocity2 ,  
40     Acceleration := lePosAcceleration2 ,  
41     Deceleration := lePosDeceleration2 ,  
42     Jerk      := lePosJerk2 ,  
43     Direction  := MC_DIRECTION__mcShortestWay ,  
44     BufferMode := MC_BUFFER_MODE__mcBlendingHigh ,  
45     Done       => bMoveAbs2Done ,  
46     CommandAborted => bMoveAbs2Aborted ,  
47     Error      => bMoveAbs2Error  
48 );  
49 //Dwell  
50 SET(bMoveAbs2Done,bDwell_In);
```

Selanjutnya, periksa operasi mcBlendingHigh.
Ubah input BufferMode dari MC_MoveAbsolute_2 menjadi
"MC_BUFFER_MODE__mcBlendingHigh", buat ulang semua program,
dan tulis ke modul Gerak.

```
58 G_bContPosDone := bDwell_out OR bError OR bAborted;  
59 //Dwell out
```

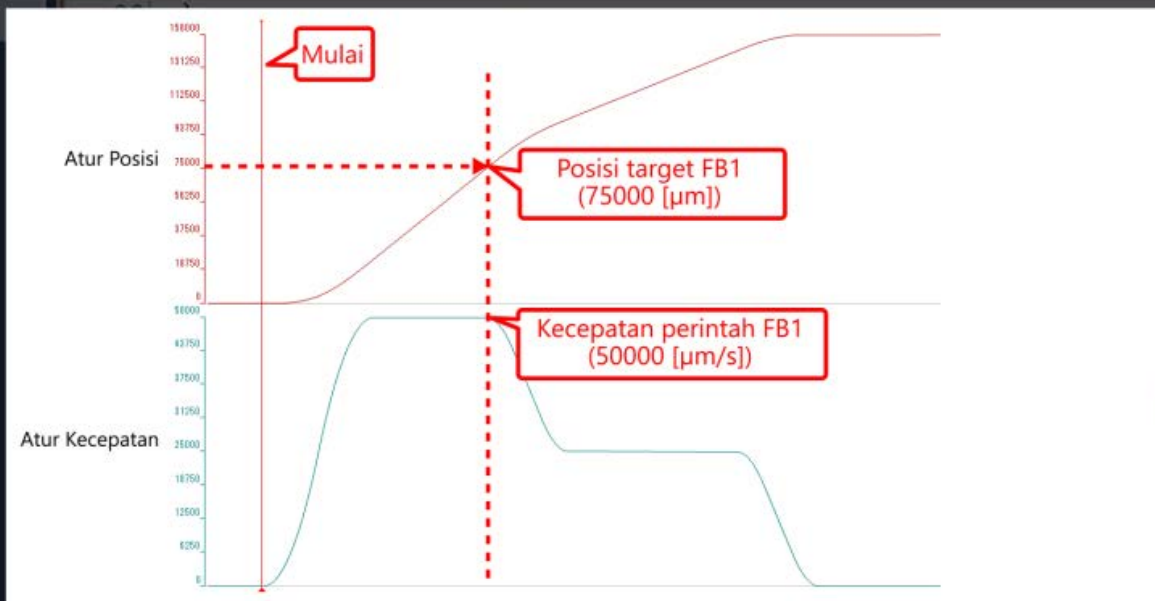


Motor dijalankan ketika X25 dihidupkan.
Grafik menunjukkan Atur Posisi dan Atur Kecepatan.



Di mcBlendingHigh, kecepatan perintah lebih cepat diterapkan ketika posisi target FB1 tercapai.

Pada contoh kursus ini, karena kecepatan perintah FB1 lebih lambat, maka bentuk gelombang kecepataannya sama dengan BlendingPrevious.



Tindakan ini akan menyelesaikan pemeriksaan operasi BufferMode.
Buka halaman berikutnya.

Dalam bab ini, Anda telah mempelajari:

- Pembatalan
- Di-buffer
- Pencampuran
- Contoh program
- Pemeriksaan Operasi

Poin penting

Pembatalan	• Ketika FB berjenis operasi sedang berjalan dan FB berjenis operasi berikutnya dijalankan, Pembatalan akan menginterupsi FB yang sedang dijalankan dan mengeksekusi FB berikutnya.
Di-buffer	• Ketika FB tipe operasi berjalan dan FB tipe operasi berikutnya dijalankan, Buffer akan menunggu hingga FB yang dieksekusi selesai dan mengeksekusi FB berikutnya.
Pencampuran	• Ketika FB berjenis operasi sedang berjalan dan FB berjenis operasi berikutnya dijalankan, Pencampuran akan menjalankan FB berikutnya tanpa menghentikan operasi FB yang sedang dijalankan. • Dalam Pencampuran, ada empat metode peralihan kecepatan: BlendingLow, BlendingHigh, BlendingPrevious, dan BlendingNext.
Contoh program	• Pilih mode buffer dengan input BufferMode dari operasi FB.
Pemeriksaan Operasi	• Anda telah memeriksa perbedaan operasi dari setiap mode buffer di video.

Unduh contoh program yang akan digunakan pada bab ini dengan mengklik tautan di bawah. Isi program tersebut sama dengan contoh program yang dijelaskan pada bab 2 dan bab 3. Hanya metode pemrogramannya saja yang berbeda.

[RD78GBasic2_sample2.zip \(1.39 MB\)](#)

4.1**Mendaftarkan Library FB Modul Gerak**

(1) Unduh library FB

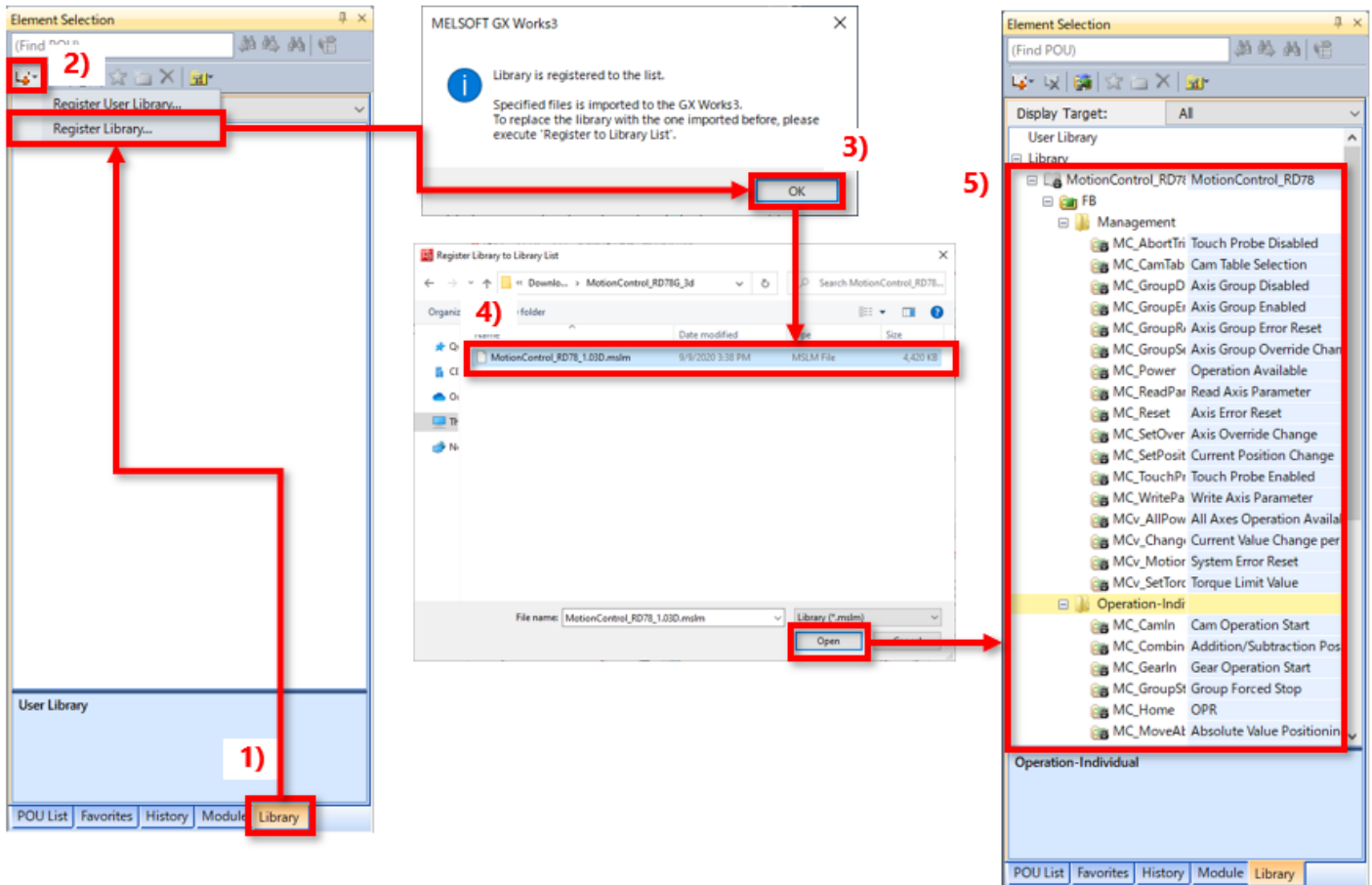
Kontrol gerak FB dapat digunakan dalam program CPU PLC dengan mendaftar library FB untuk modul Gerak ke GX Works3. Unduh library FB dari tautan di bawah, dan ekstrak file ZIP ke tujuan yang diinginkan.

[MotionControl_RD78G_3d.zip\(4.29 MB\)](#)

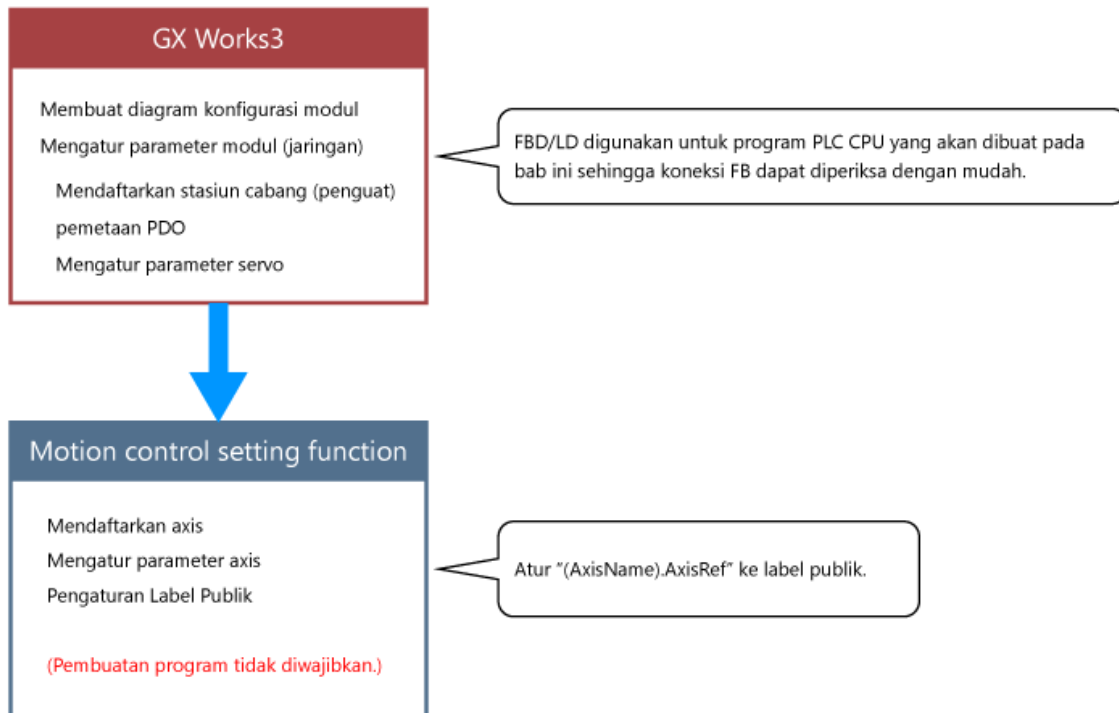
(Catatan) Library FB versi terbaru dapat diunduh dari Situs Web Global MITSUBISHI ELECTRIC FA.

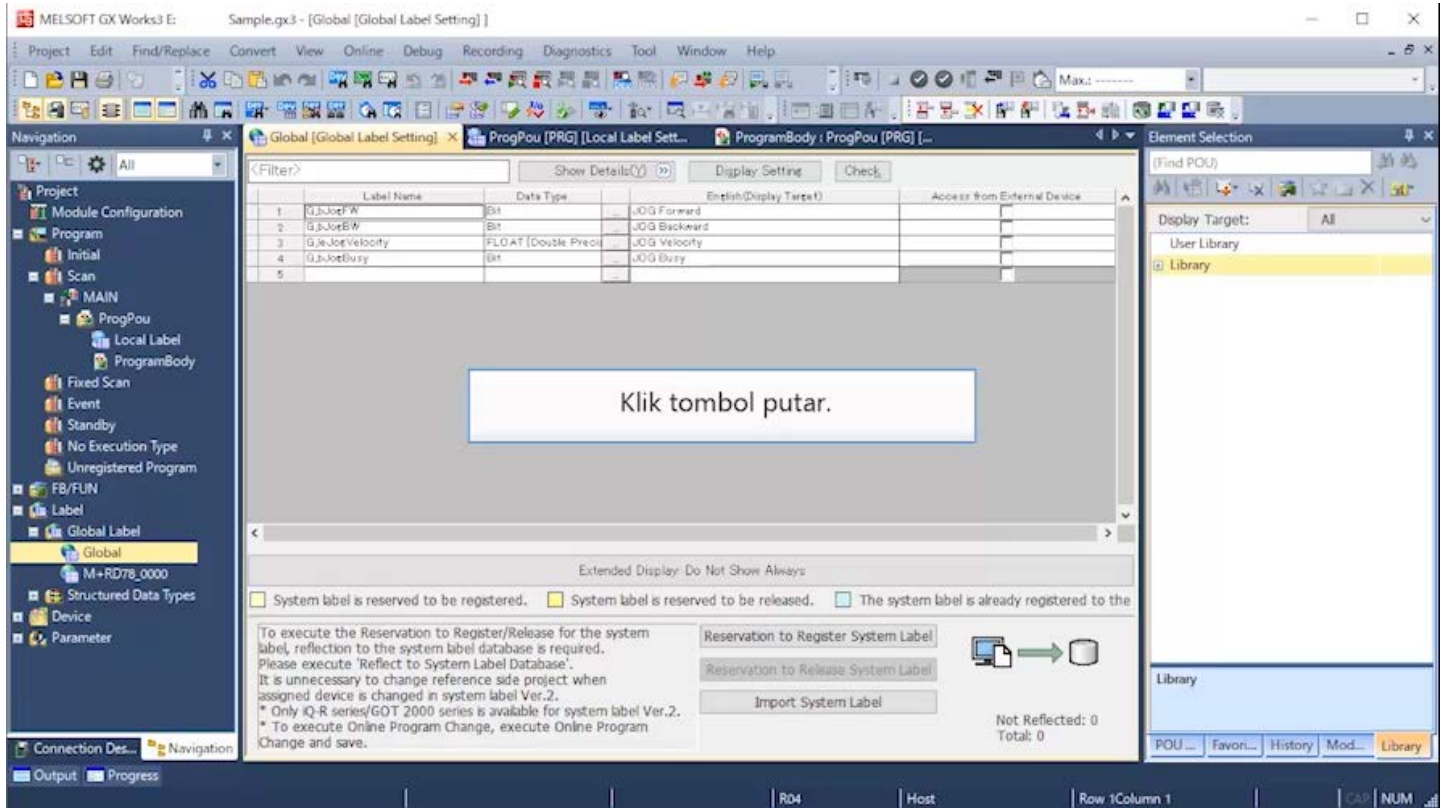
(2) Mendaftarkan library FB

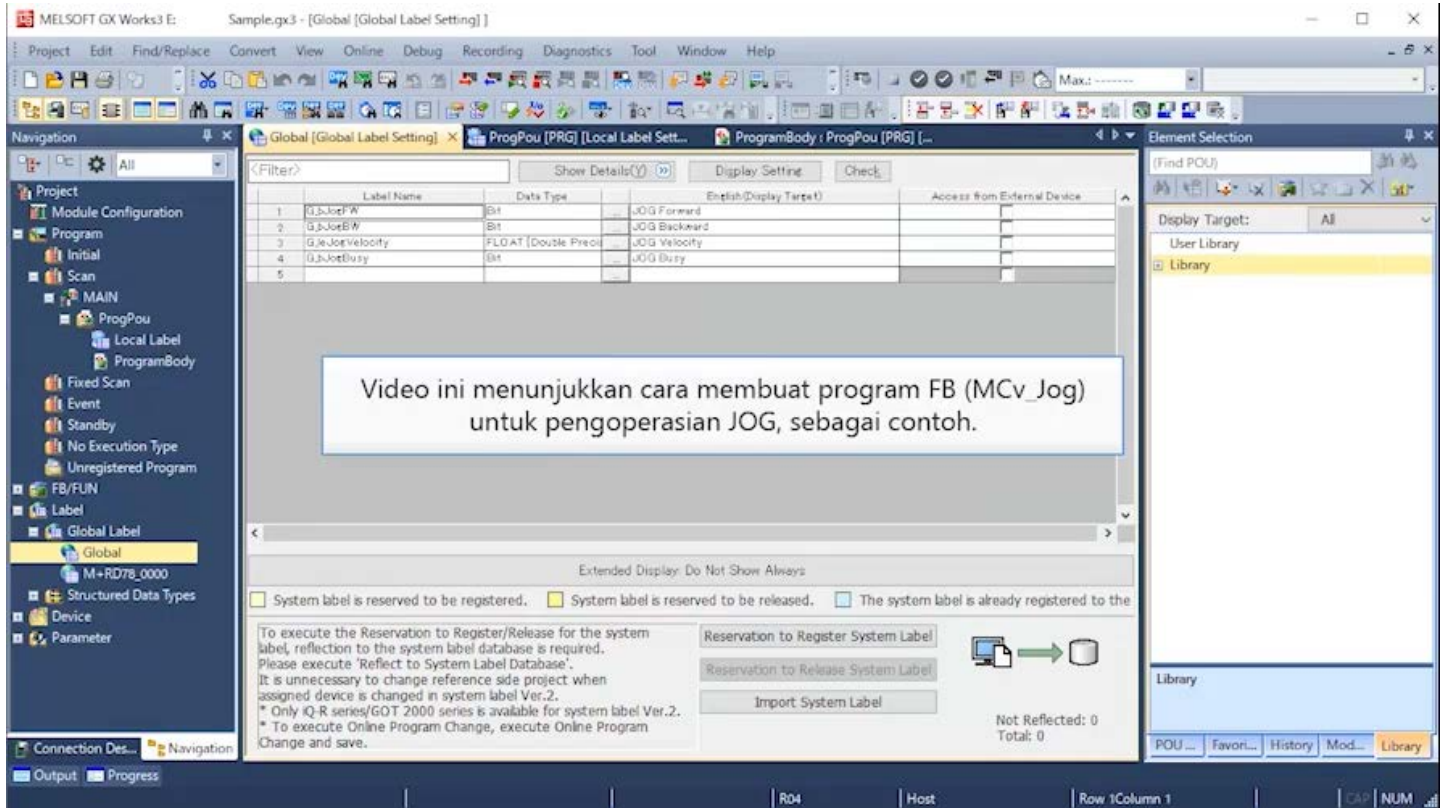
- 1) Buka proyek apa pun di GX Works3, dan buka tab Library di jendela Element Selection.
- 2) Klik tombol [Register to Library List] di bagian atas, dan pilih [Register Library].
- 3) Ketika pesan "Library is registered to the list" ditampilkan, klik [OK].
- 4) Pilih file library FB "MotionControl_RD78_****.mslm", dan klik [Open].
(**** menunjukkan versi.)
- 5) Motion control FB didaftarkan ke library di jendela Elemen Selection.



Prosedur untuk membuat proyek sama seperti yang dijelaskan pada bagian sebelumnya.







Daftarkan perintah JOG, kecepatan JOG, dan Jog sibuk ke label global.
(Kecepatan JOG didaftarkan ke label global dengan asumsi bahwa kecepatan tersebut dapat diatur dari perangkat eksternal seperti GOT.)

Label Name	Data Type	English/Display Target	Access from External Device
G.JogFW	Bit	JOG Forward	
G.JogBW	Bit	JOG Backward	
G.JogVelocity	Float (Double Precision)	JOG Velocity	
G.JogBusy	Bit	JOG Busy	

Extended Display: Do Not Show Always

☐ System label is reserved to be registered. ☐ System label is reserved to be released. ☐ The system label is already registered to the

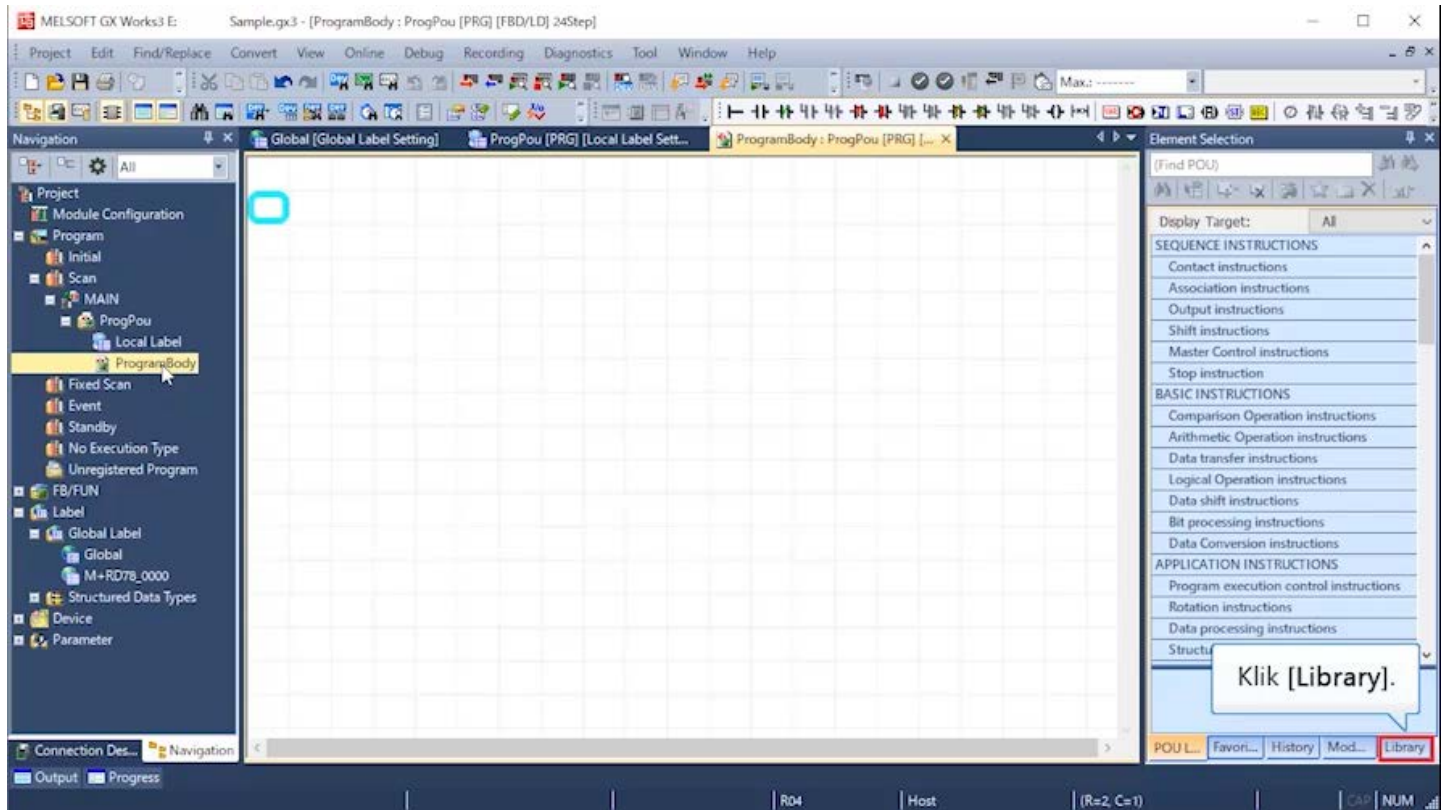
To execute the Reservation to Register/Release for the system label, reflection to the system label database is required. Please execute 'Reflect to System Label Database'. It is unnecessary to change reference side project when assigned device is changed in system label Ver.2.
* Only Q-R series/GOT 2000 series is available for system label Ver.2.
* To execute Online Program Change, execute Online Program Change and save.

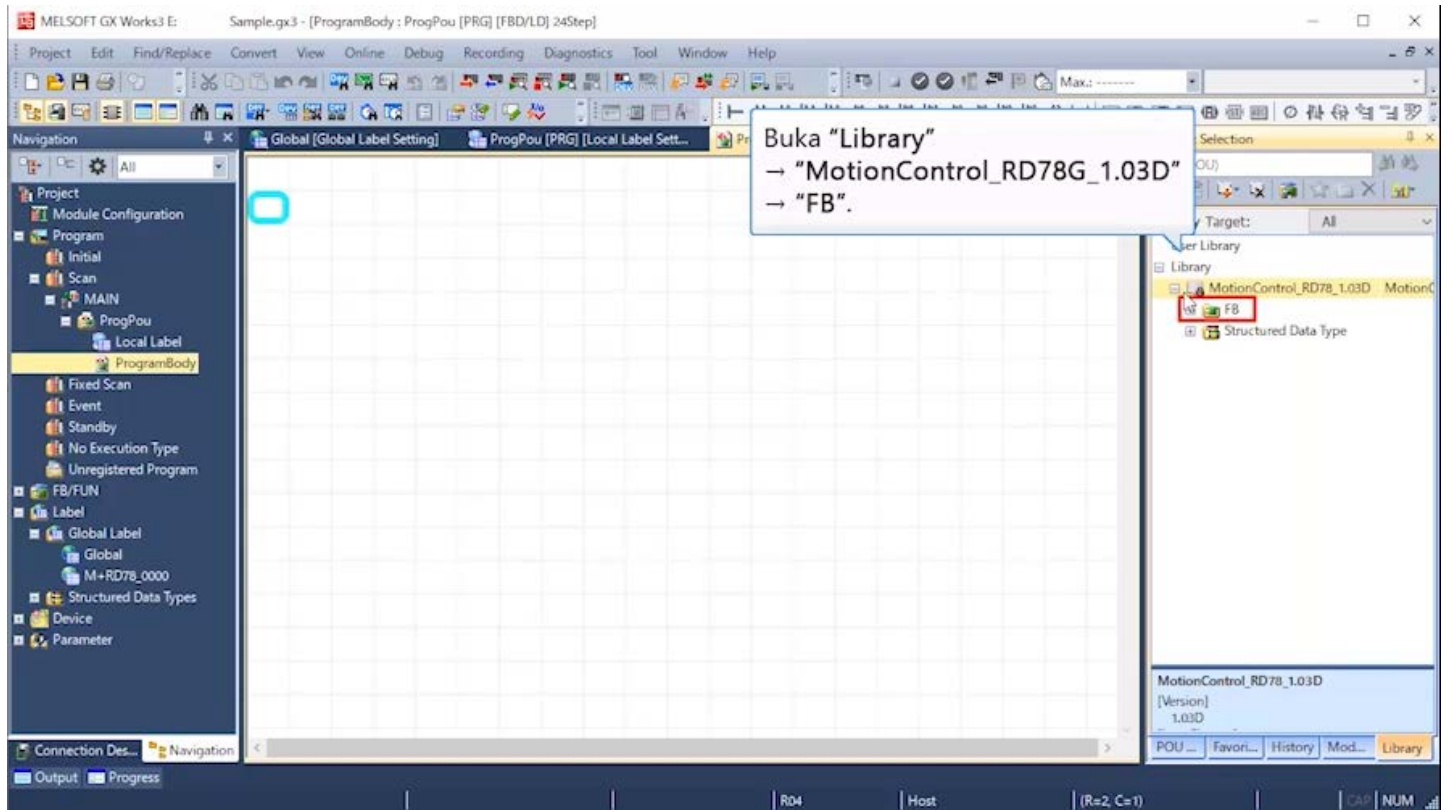
Reservation to Register System Label
Reservation to Release System Label
Import System Label

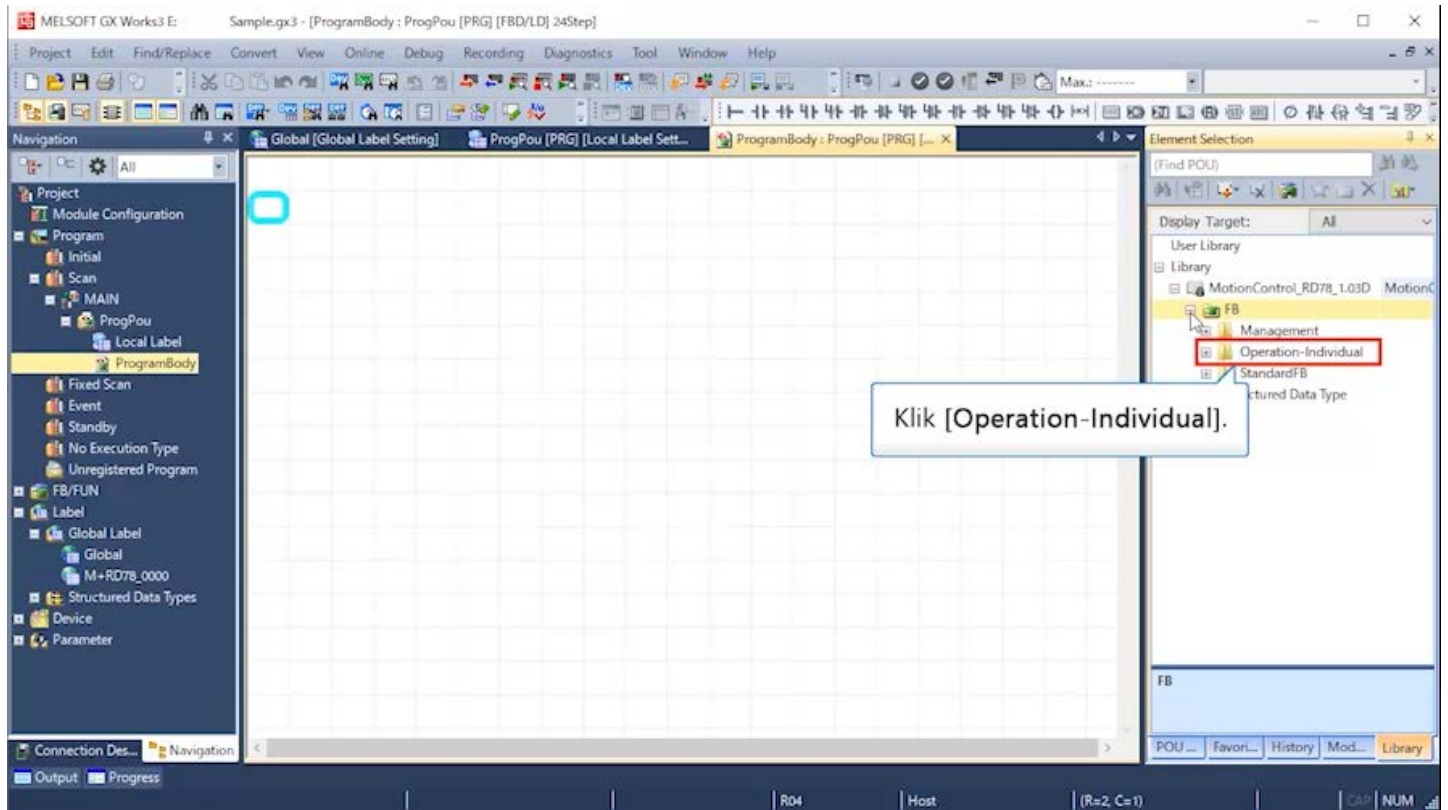
Not Reflected: 0
Total: 0

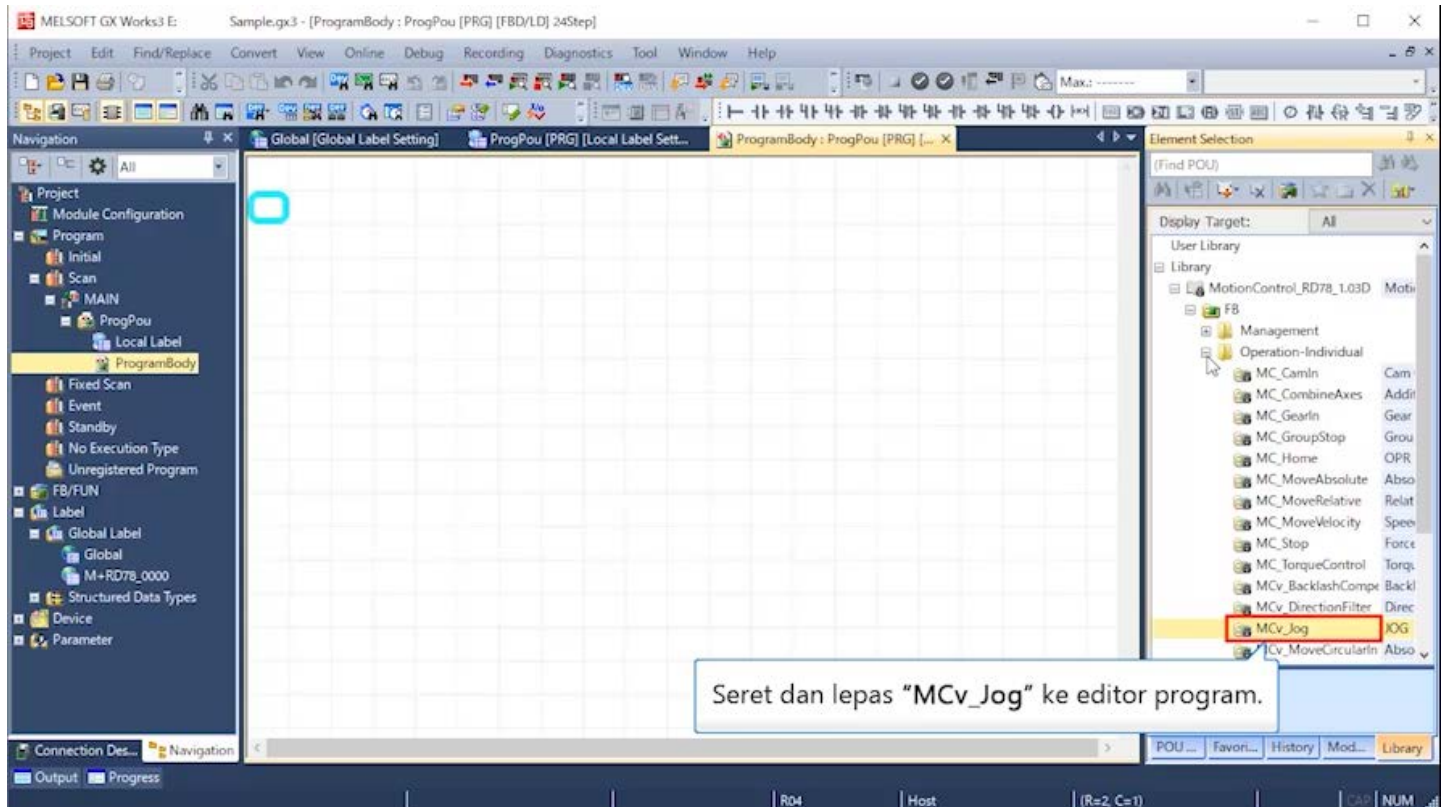
Daftarkan akselerasi JOG, deselerasi JOG, sentakan JOG dan lain-lain yang hanya digunakan dalam program ini ke label lokal program "ServoON_Jog".

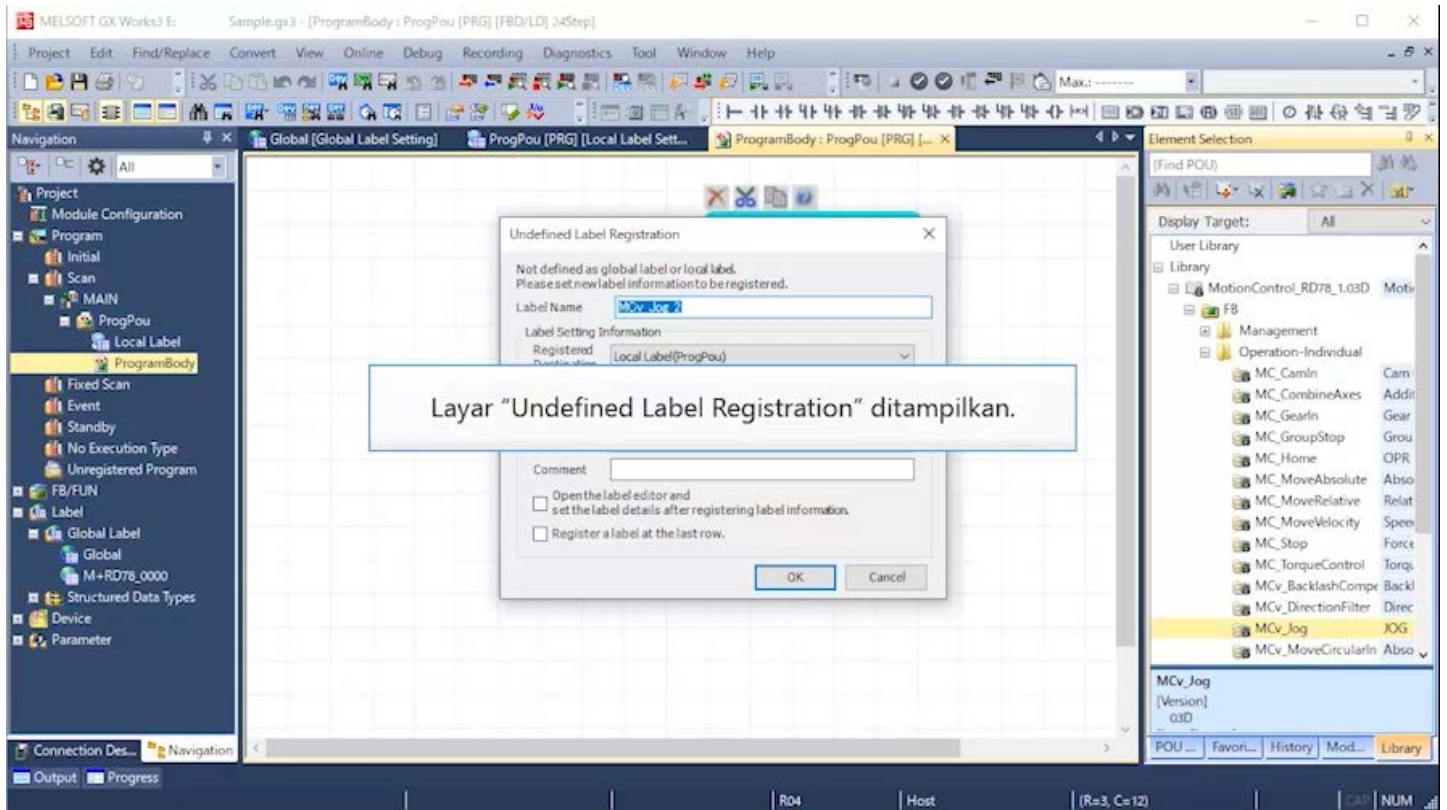
	Label Name	Data Type	English (Display Target)
1	%JogAcc	FLOAT (Double Precision)	JOG Acceleration
2	%JogDec	FLOAT (Double Precision)	JOG Deceleration
3	%JogJerk	FLOAT (Double Precision)	JOG Jerk
4			

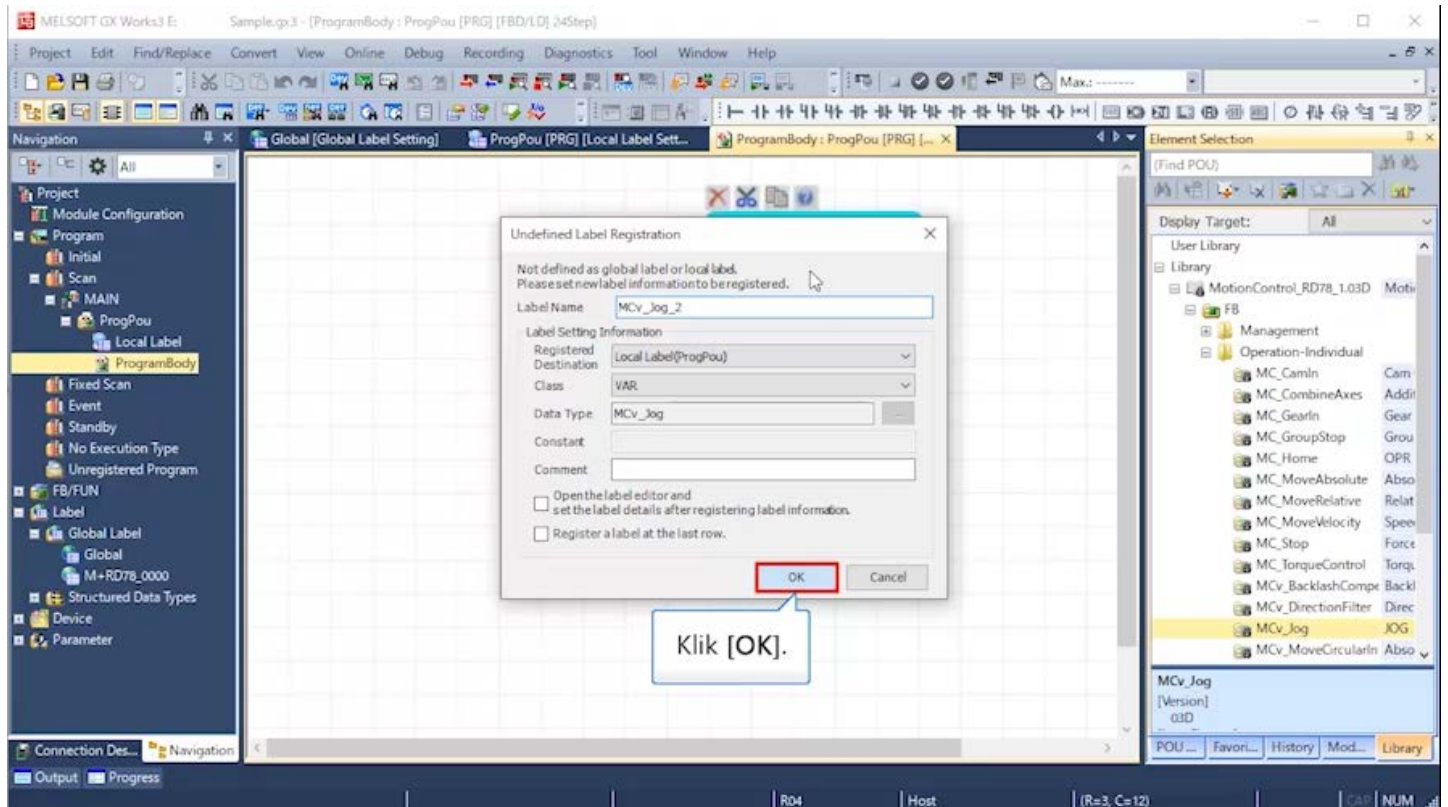




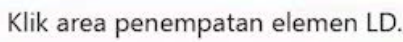


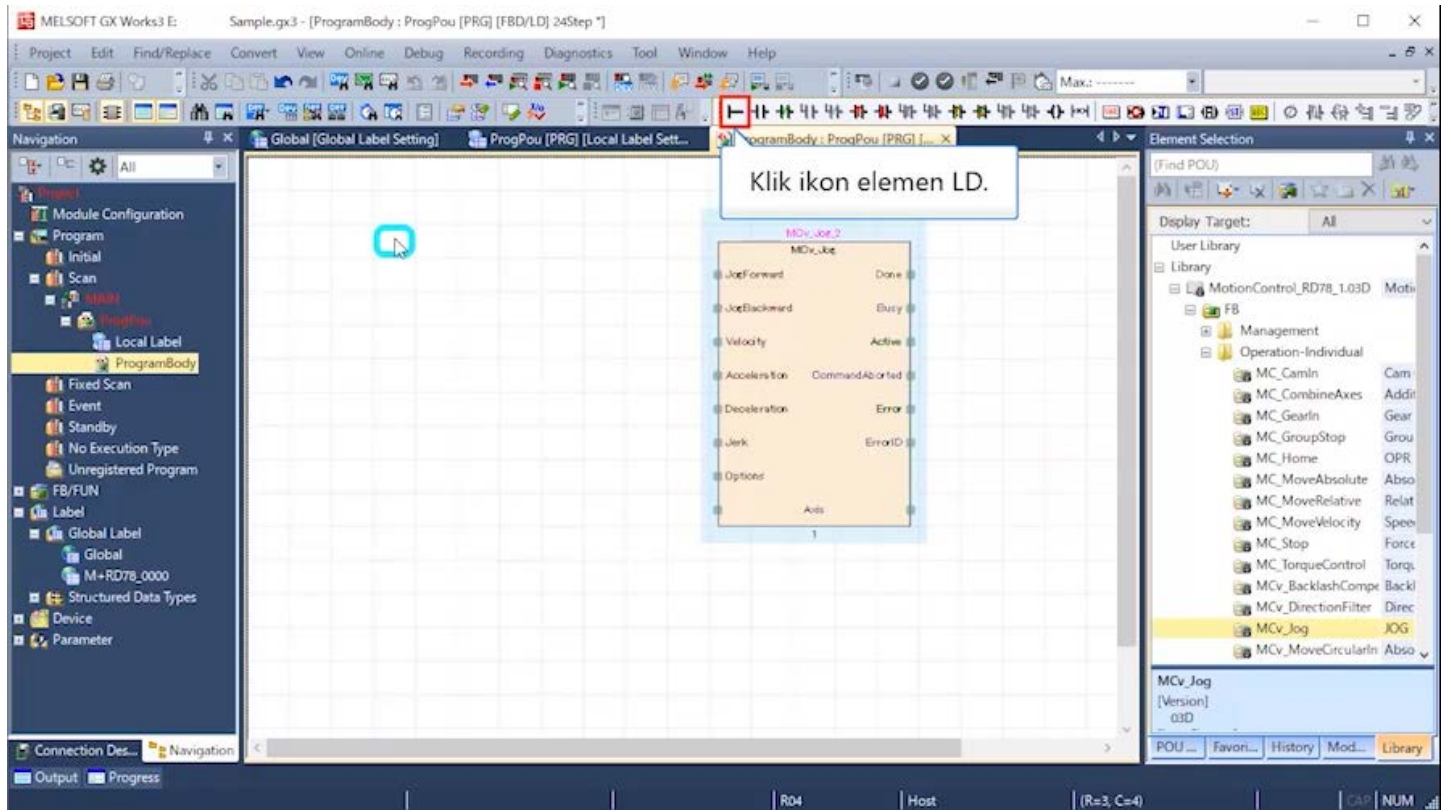


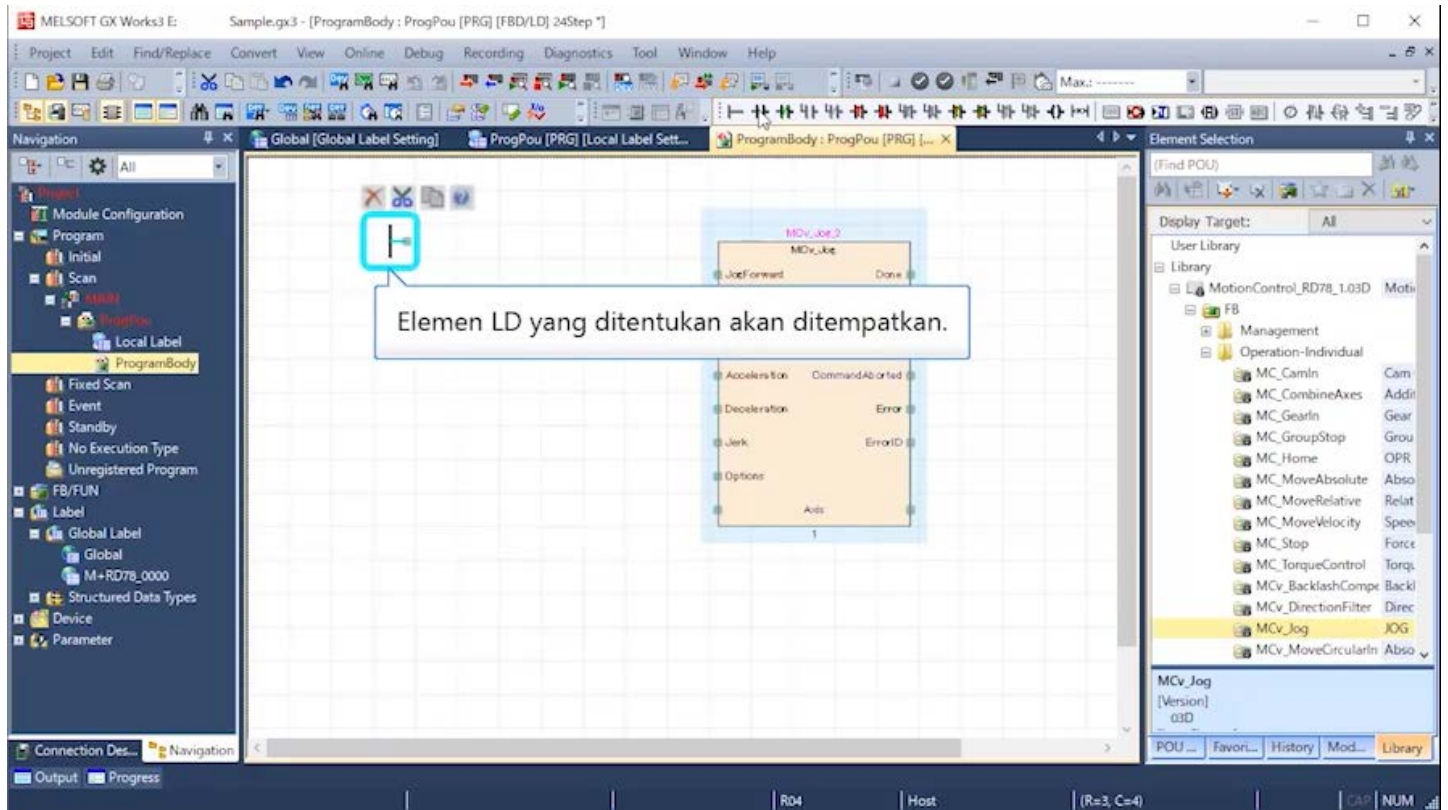


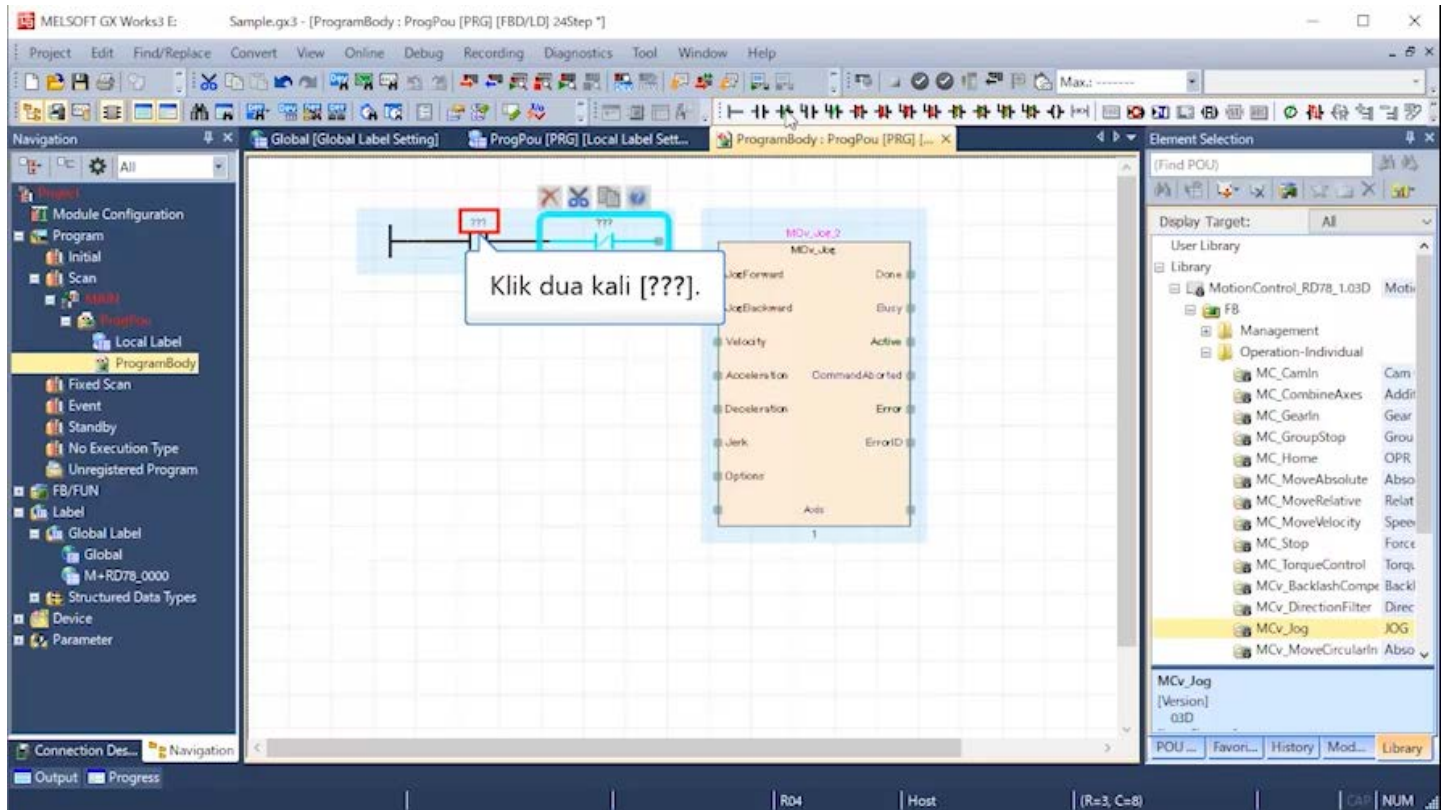


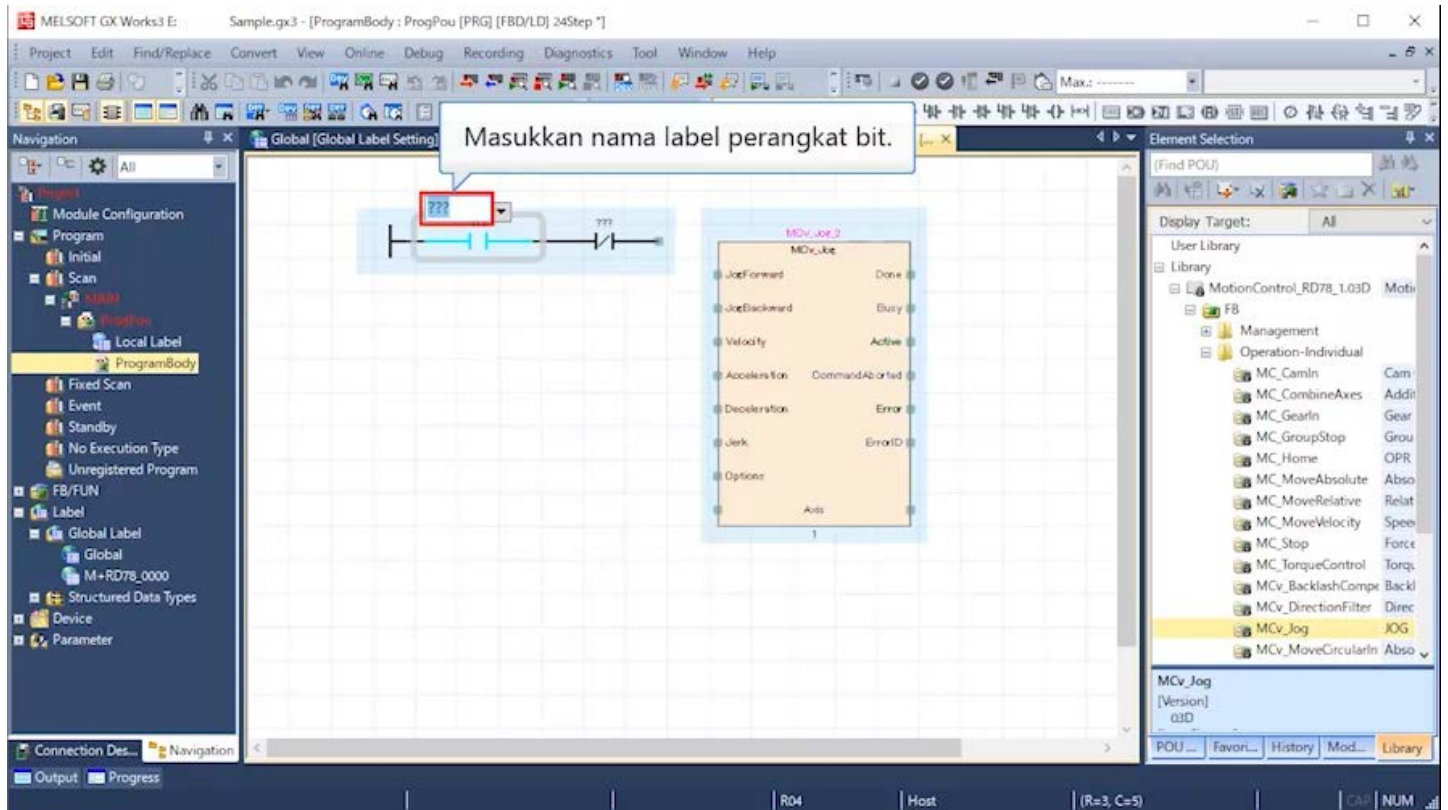
Cara Menggunakan Motion Control FB









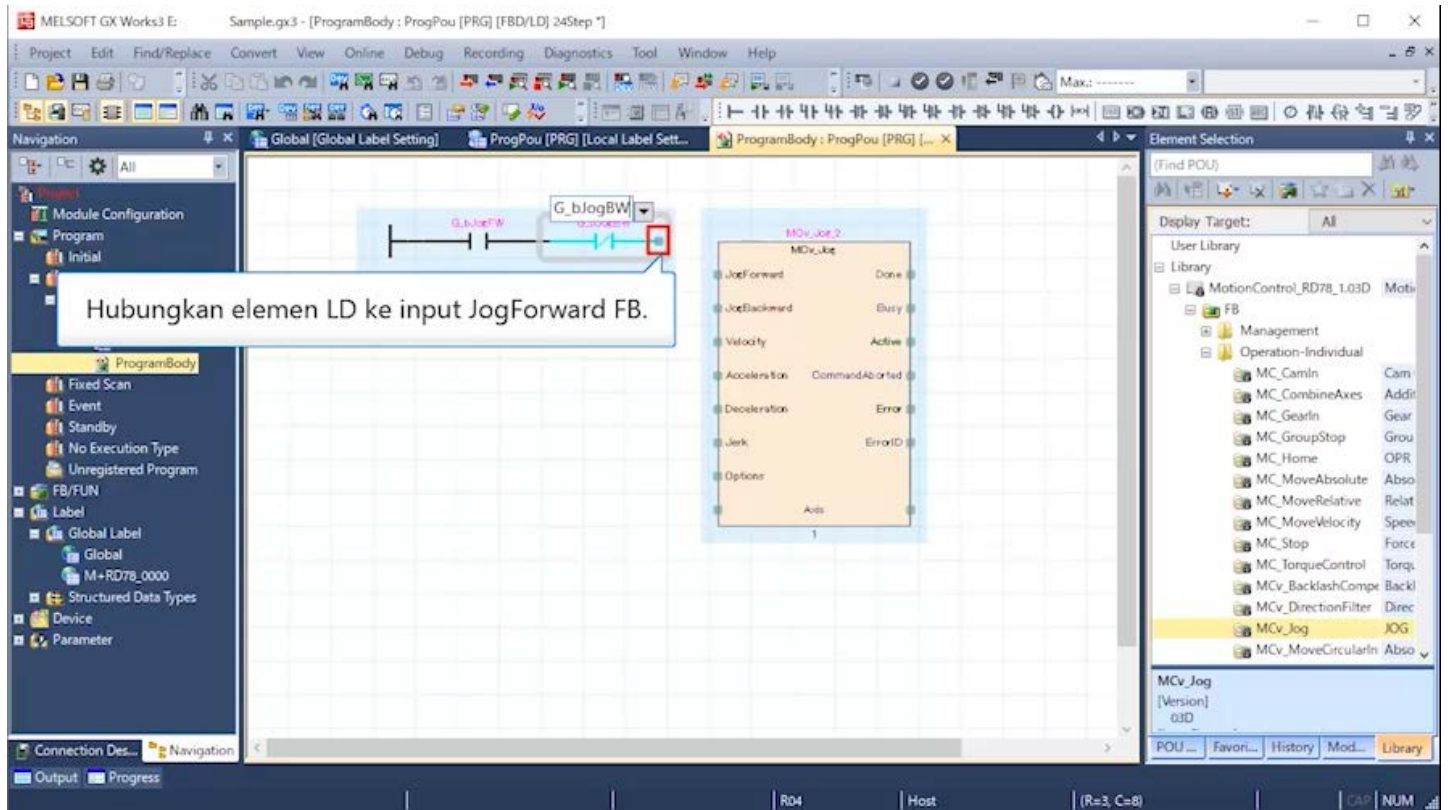


The screenshot displays the MELSOFT GX Works3 interface. The main workspace shows a ladder logic diagram with a network containing a normally open contact labeled 'G_'. A tooltip points to this contact, displaying a list of global labels (G_) and their corresponding data types and functions:

Label	Data Type	Function
G_bJogBusy	BOOL	JOG Busy
G_bJogBW	BOOL	JOG Backward
G_bJogFW	BOOL	JOG Forward
G_leJogNotify	REAL	JOG Velocity

The right-hand pane shows the 'Element Selection' window with the 'Library' tab active. The 'MotionControl_RD78_1.03D' library is expanded, showing various motion control functions. The 'MCv_Jog' function is highlighted, showing its parameters: 'MDv_Jog' (Done), 'Busy', 'Active', and 'Options' (Axis).

A text box overlay on the diagram states: "Label yang sudah terdaftar pada label global ditampilkan sebagai kandidat." (Labels already registered in the global label are displayed as candidates.)



MELSOFT GX Works3 E: Sample.gx3 - [ProgramBody : ProgPou [PRG] [FBD/LD] 24Step *]

Project Edit Find/Replace Convert View Online Debug Recording Diagnostics Tool Window Help

Navigation

Global [Glob]

Project

- Module Configuration
- Program
 - Initial
 - Scan
 - Input
 - Output
 - Local Label
 - ProgramBody
 - Fixed Scan
 - Event
 - Standby
 - No Execution Type
 - Unregistered Program
 - FB/FUN
 - Label
 - Global Label
 - Global
 - M+RD78_0000
 - Structured Data Types
 - Device
 - Parameter

Connection Des... Navigation

Output Progress

Hasilnya akan terlihat seperti ini ketika elemen LD terhubung ke input JogForward dan input JogBackward.

Element Selection

(Find POU)

Display Target: All

User Library

Library

- MotionControl_RD78_1.03D
 - Management
 - Operation-Individual
 - MC_CamIn Cam
 - MC_CombineAxes Addi
 - MC_GearIn Gear
 - MC_GroupStop Grou
 - MC_Home OPR
 - MC_MoveAbsolute Abso
 - MC_MoveRelative Relat
 - MC_MoveVelocity Spee
 - MC_Stop Force
 - MC_TorqueControl Torqs
 - MCv_BacklashCompe Backl
 - MCv_DirectionFilter Direc
 - MCv_Jog JOG
 - MCv_MoveCircularIn Abso

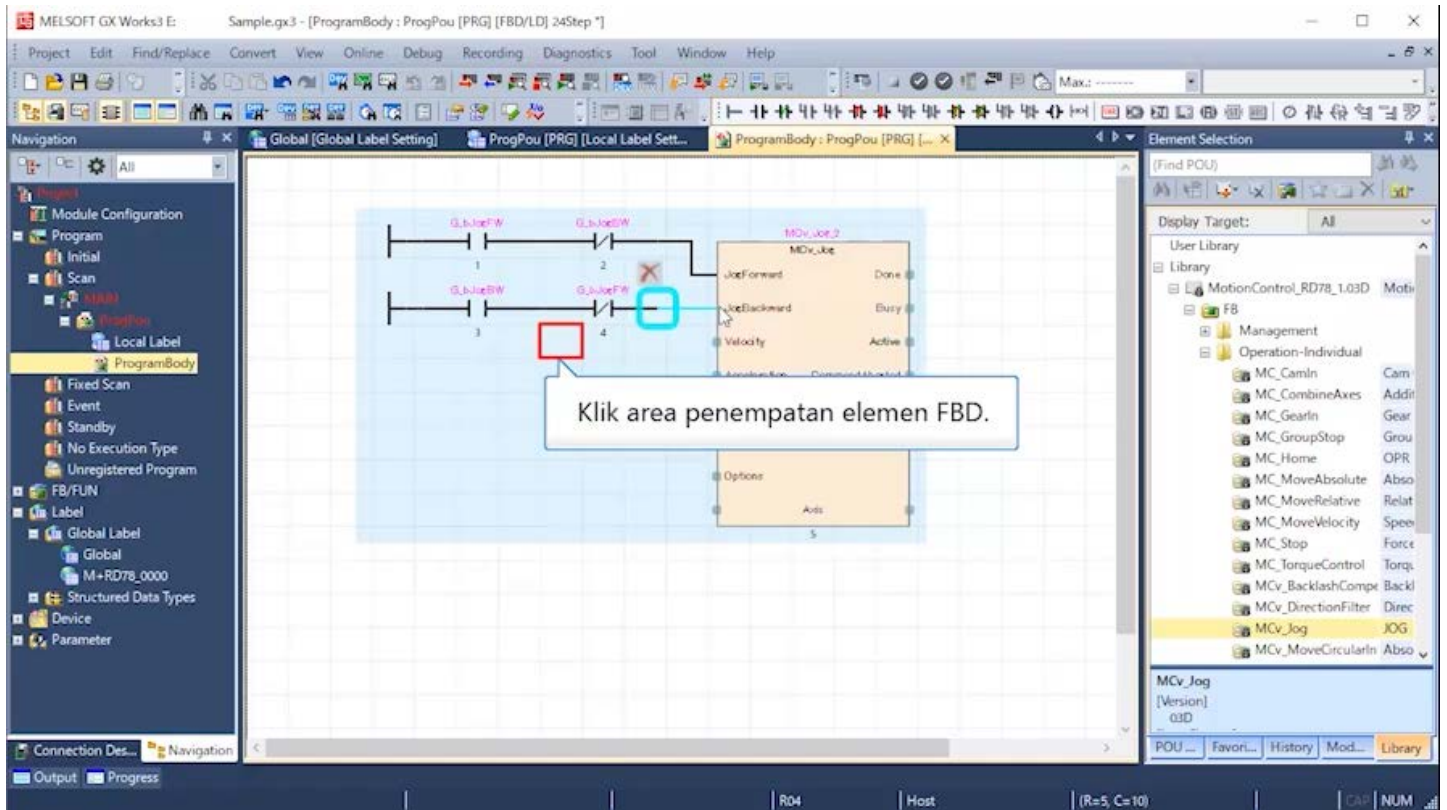
MCv_Jog

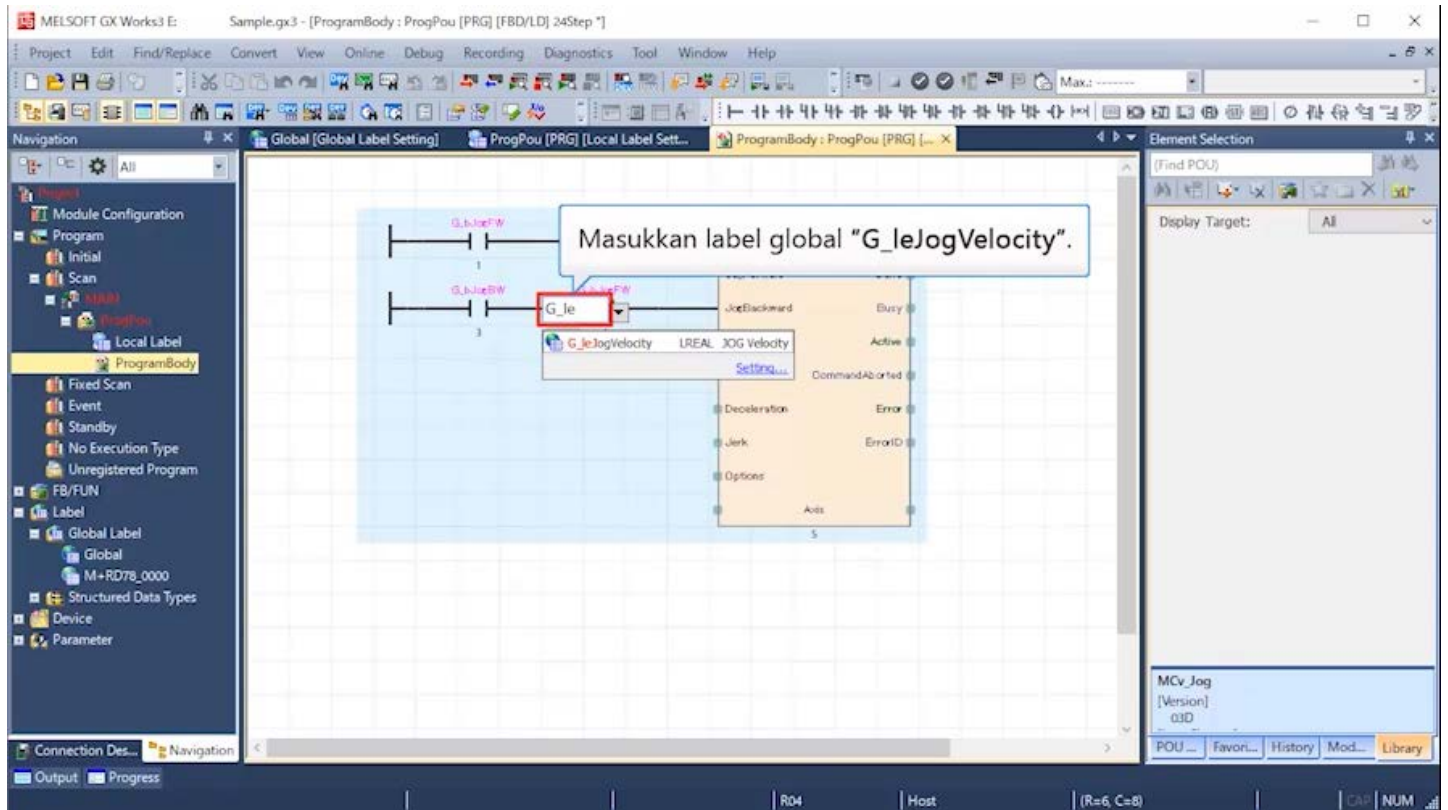
[Version]

03D

POU... Favori... History Mod... Library

R04 Host (R=5, C=10) CAP NUM



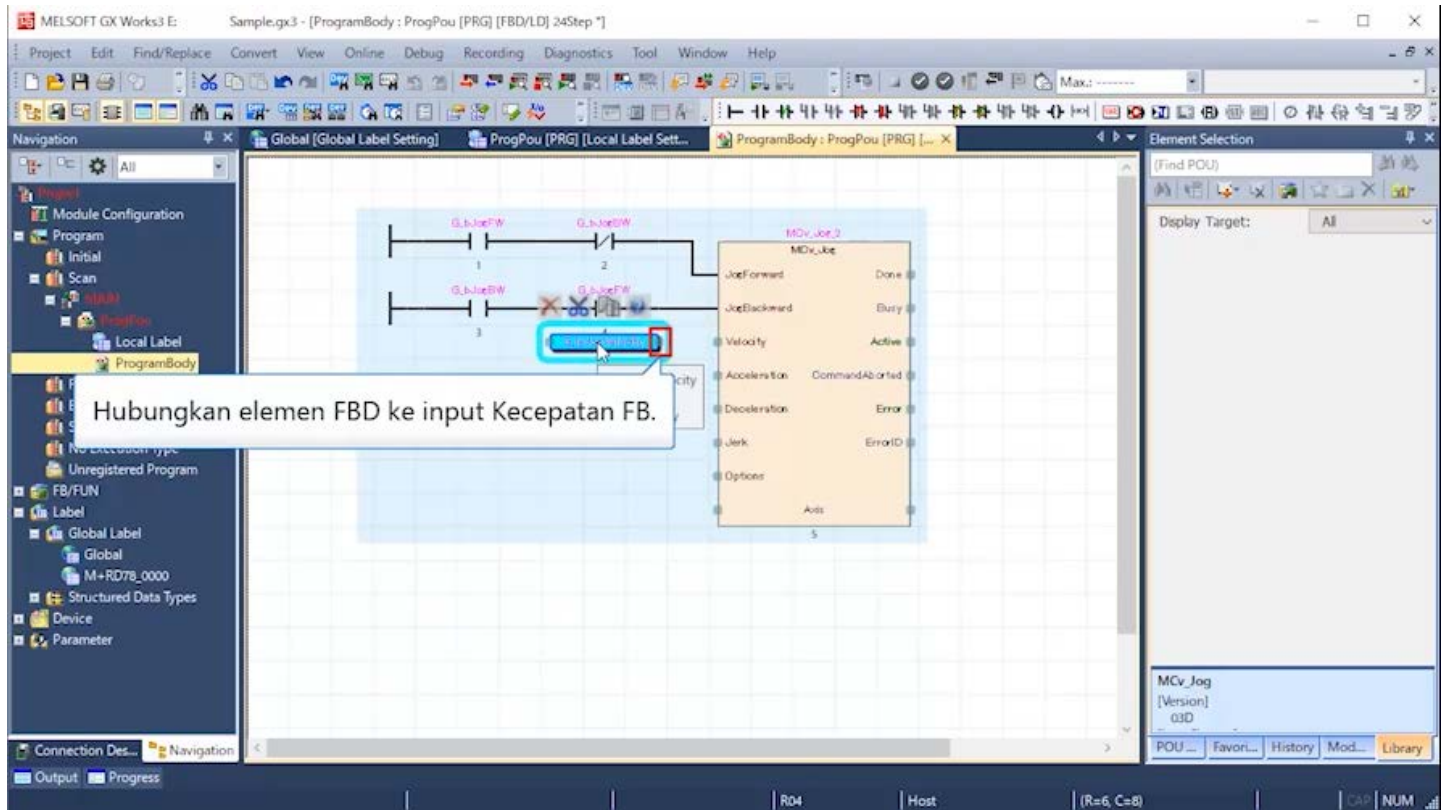


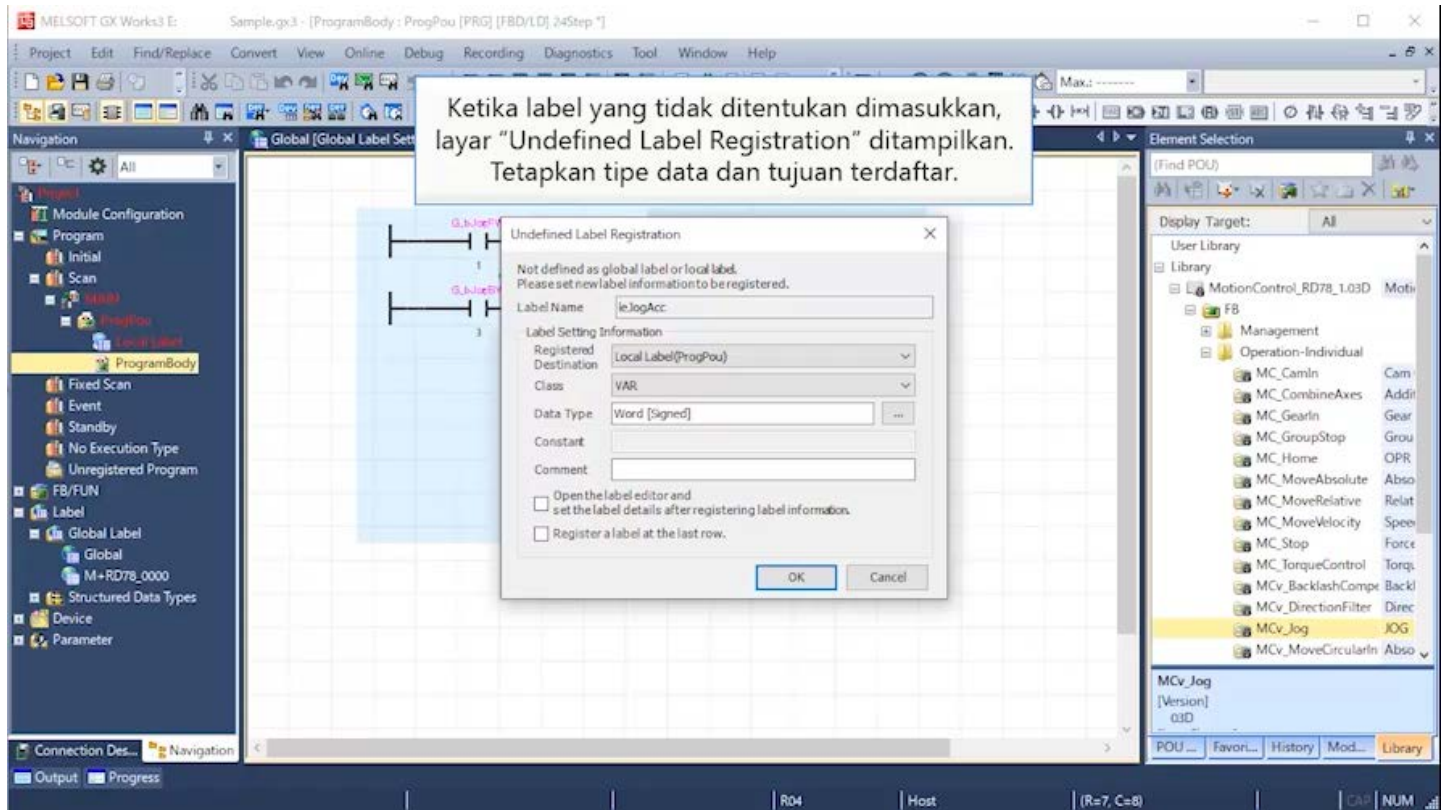
The screenshot displays the MELSOFT GX Works3 software interface. The main workspace shows a ladder logic diagram for a motion control function block. The diagram includes a callout box with the following text:

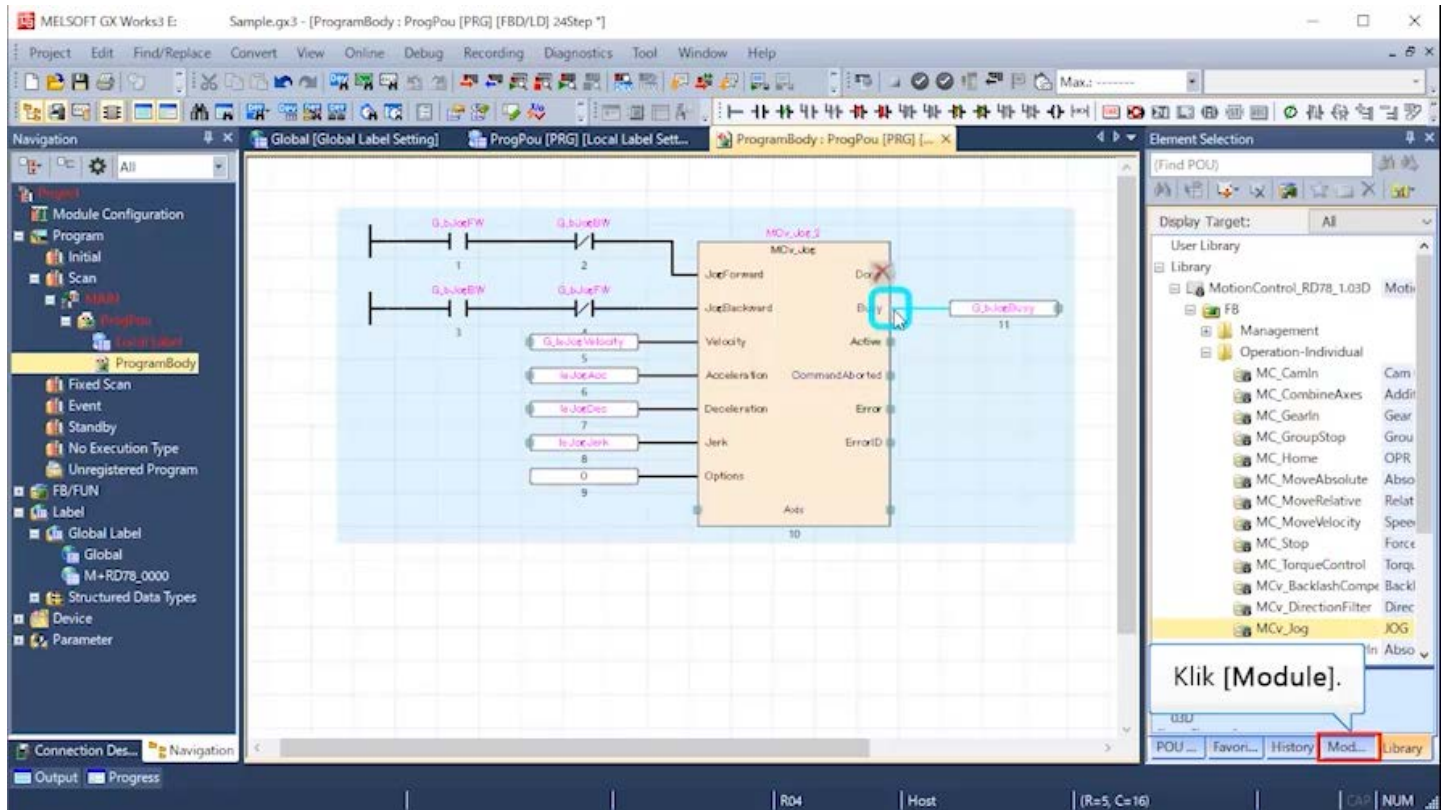
Label yang sudah terdaftar pada label global ditampilkan sebagai kandidat.

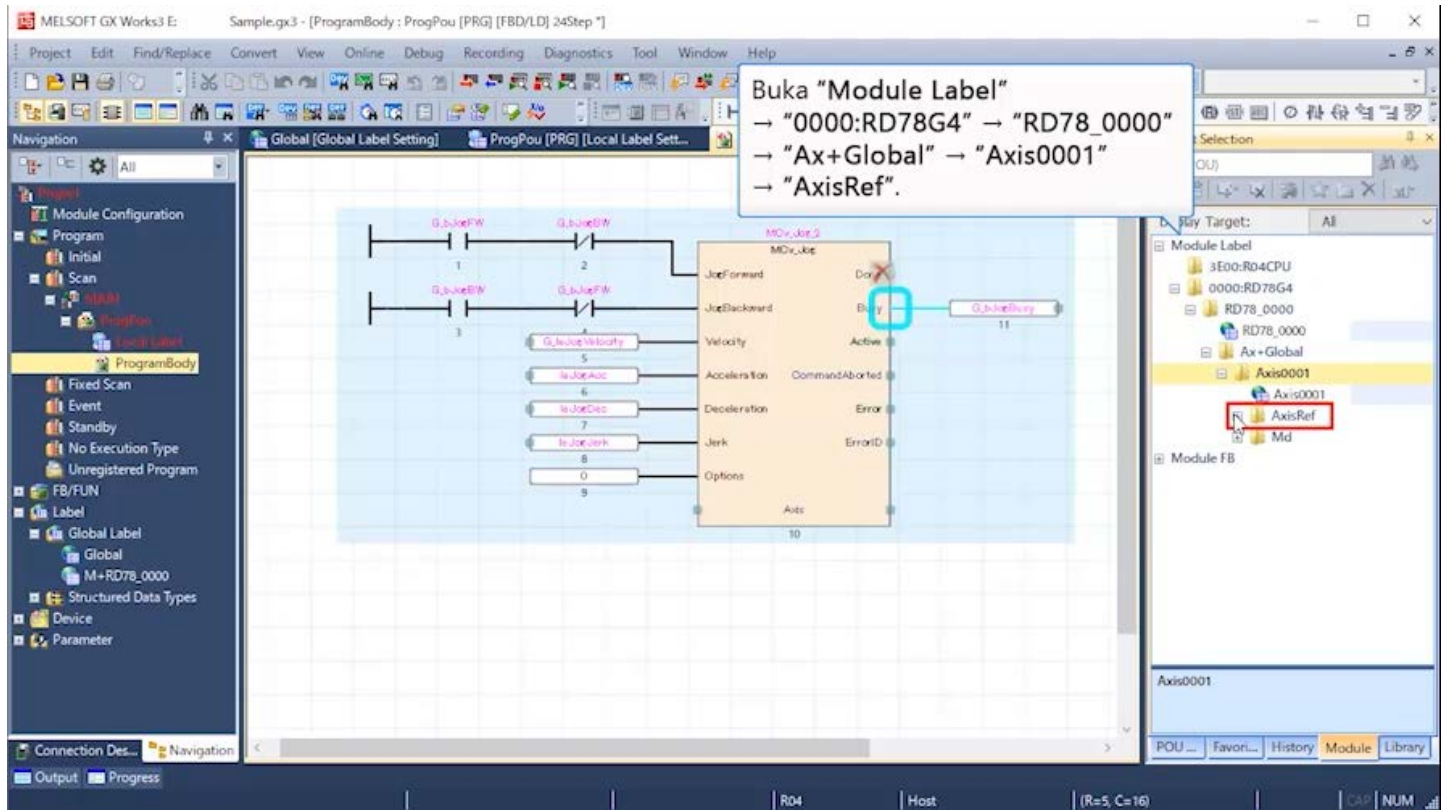
The diagram shows a function block named `MDV Jog` with inputs `JogForward` and `JogBackward`. The output is `MDV Jog`. The function block is connected to a global label `G_JogVelocity` and a local label `REAL JOG Velocity`. The function block is also connected to a global label `G_JogVelocity` and a local label `REAL JOG Velocity`.

The interface includes a navigation pane on the left with a tree view showing the project structure. The right pane shows the 'Element Selection' dialog with a search bar and a list of elements. The bottom status bar shows the current position (R04) and host connection status.









Hubungkan data terstruktur tipe AxisRef yang terdaftar sebagai label publik ke input Axis.
Seret dan lepas "AxisRef" (informasi axis) ke editor program.

The screenshot shows the MELSOFT GX Works3 interface. The main workspace displays a ladder logic program with a function block 'MC_MoveJelt'. The inputs for this block are:

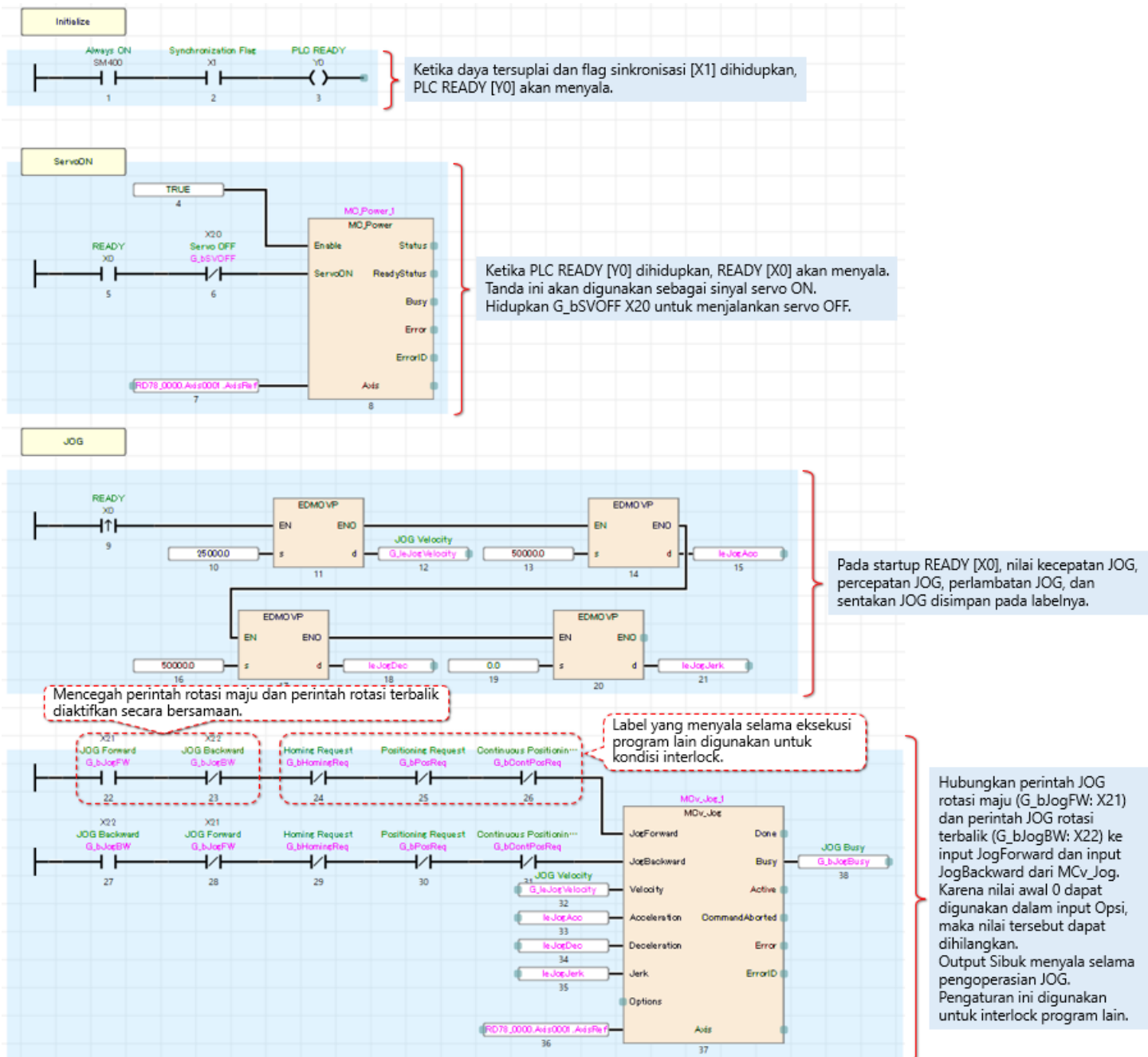
- 1: G-JerkFW
- 2: G-JerkBW
- 3: G-JerkFW
- 4: G-JerkBW
- 5: G-JerkVelocity
- 6: G-JerkAcc
- 7: G-JerkDec
- 8: G-JerkJerk

 The 'Axis' input is highlighted with a blue circle, and a callout box provides instructions to link 'AxisRef' data to it. The right-hand pane shows the 'Element Selection' window with 'AxisRef' selected under the 'Module Label' section. The bottom status bar indicates 'R04 | Host | (R=5, C=16) | CAP | NUM'.

The screenshot displays the MELSOFT GX Works3 interface. The main workspace shows a ladder logic diagram for a motion control function block. The diagram includes several inputs and outputs connected to the function block, which is labeled 'MDV_Job_0'. The inputs are: 'JobForward' (labeled 1), 'JobBackward' (labeled 2), 'Velocity' (labeled 3), 'Acceleration' (labeled 4), 'Deceleration' (labeled 5), 'Jerk' (labeled 6), and 'Options' (labeled 7). The outputs are: 'Done' (labeled 8), 'Busy' (labeled 9), 'Active' (labeled 10), 'CommandAborted' (labeled 11), 'Error' (labeled 12), and 'ErrorID' (labeled 13). The function block is also labeled 'Axis' at the bottom. The left sidebar shows the project structure, including 'Module Configuration', 'Program', 'Initial', 'Scan', 'ProgramBody', 'Fixed Scan', 'Event', 'Standby', 'No Execution Type', 'Unregistered Program', 'FB/FUN', 'Label', 'Global Label', 'Global', 'M+RD78_0000', 'Structured Data Types', 'Device', and 'Parameter'. The right sidebar shows the 'Element Selection' panel with a search bar and a list of elements, including 'Module Label', '3E00-R04CPU', '0000-RD78G4', 'RD78_0000', 'Ax+Global', 'Axis0001', 'Axis0001', 'AxisRef', 'AxisNo', 'StartIO', and 'Md'. A text box at the bottom right of the workspace contains the following text: 'Tindakan ini akan melengkapi masukan untuk pemrograman. Klik > untuk melanjutkan ke halaman berikutnya.'

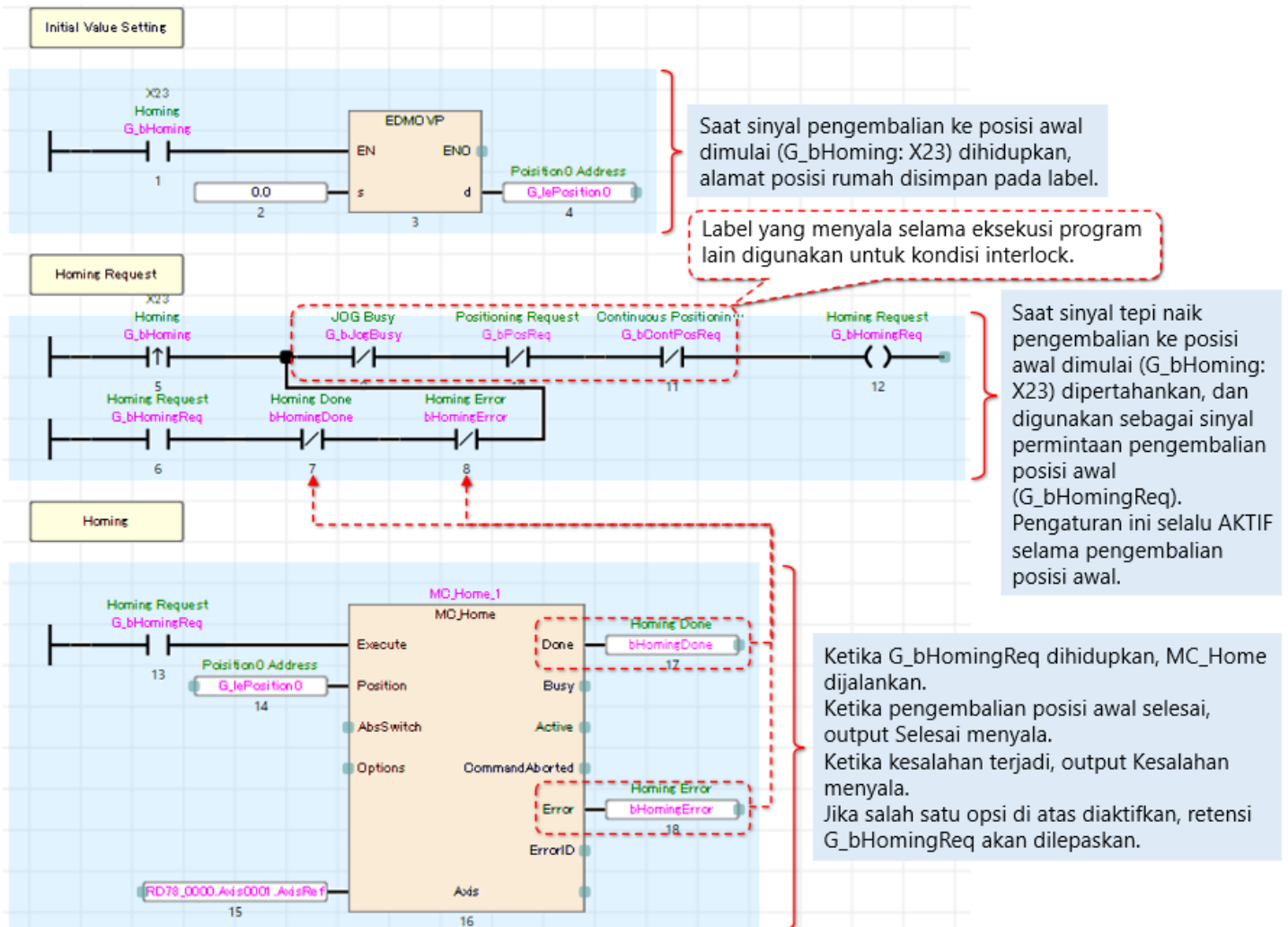
(1) ServoON_Jog

Program ini mengeksekusi operasi PLC ready ON, servo ON, dan JOG.



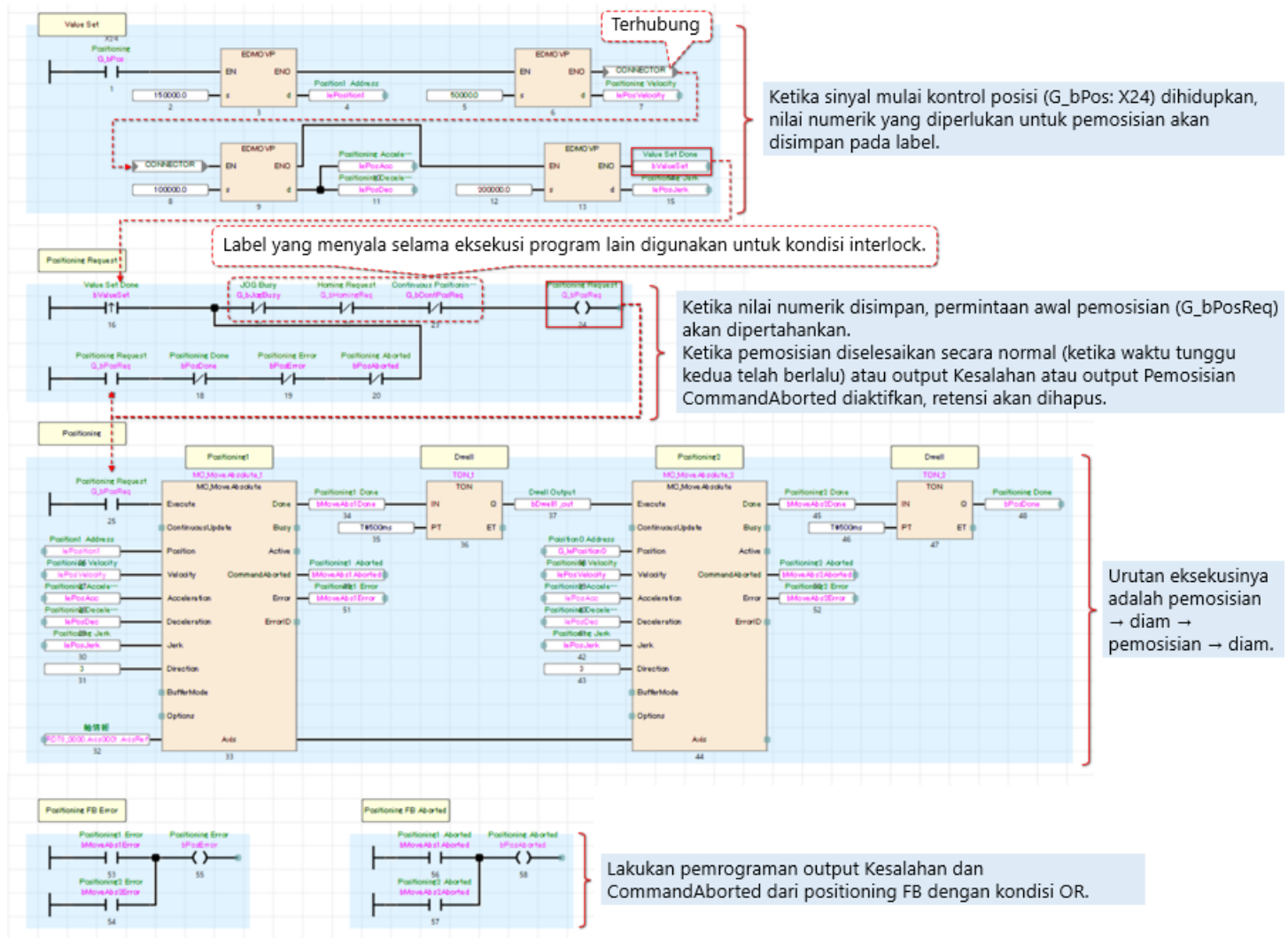
(2) Homing

Program ini menjalankan pengembalian ke posisi awal



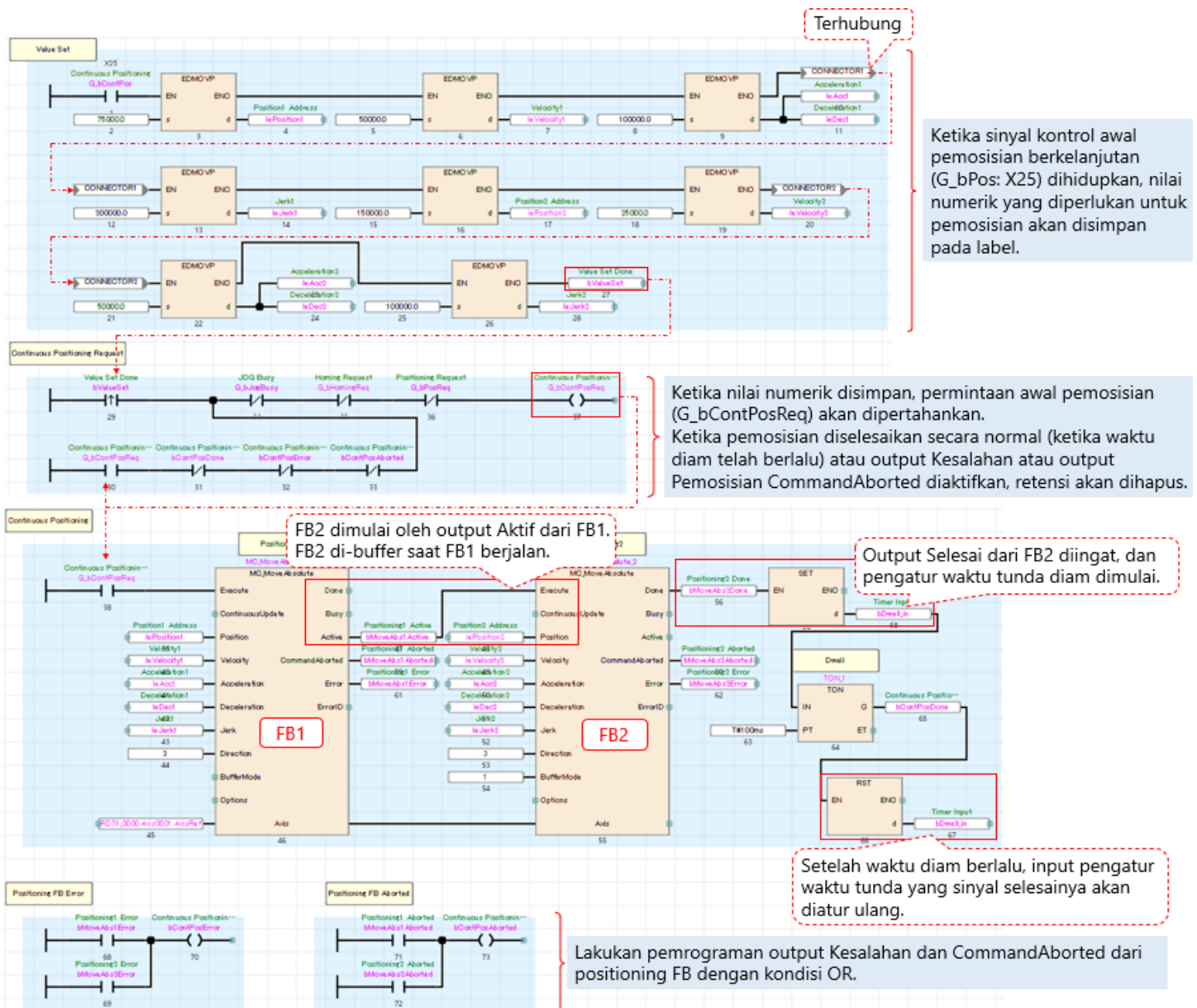
(3) Pemosisian

Program ini menjalankan operasi penentuan posisi axis tunggal.



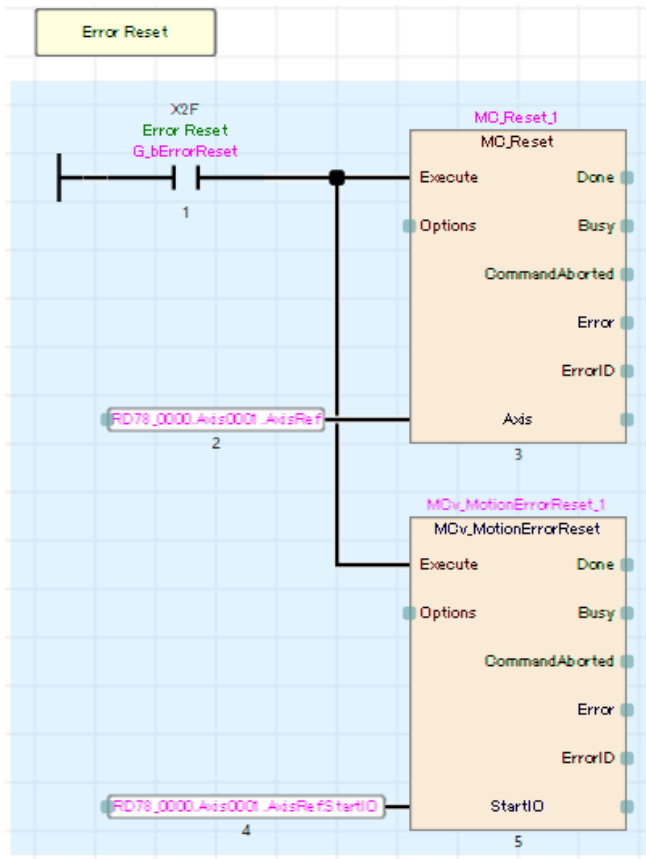
(4) ContinuousPositioning

Program ini menjalankan pemosisian berkelanjutan dengan menggunakan mode buffer.



(5) ErrorReset

Program ini menjalankan mulai ulang kesalahan.



Ketika sinyal reset kesalahan (G_bErrorReset: X2F) dihidupkan, pemulaian ulang kesalahan axis (MC_Reset) dan pemulaian ulang kesalahan sistem (MCv_MotionErrorReset) akan dijalankan.

(6) Pemantauan

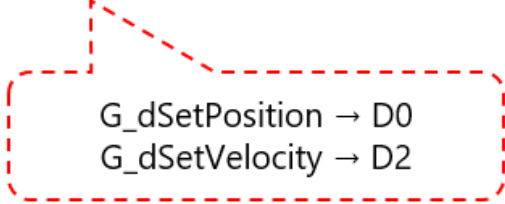
Program ini menyimpan SetPosition (Atur Posisi) dan SetVelocity (Atur Kecepatan) label global monitor axis yang ditetapkan ke D0 dan D2 CPU PLC.

Karena SetPosition Dan SetVelocity adalah jenis bilangan nyata presisi ganda, keduanya diubah menjadi jenis kata ganda bertanda sehingga dapat dengan mudah ditangani oleh CPU PLC. (Catatan)

Perangkat kata ini tidak digunakan dalam subjek.

Perangkat tersebut digunakan untuk ditampilkan pada program urutan lain dan GOT, dan untuk tujuan lainnya. Program ini dijelaskan dengan ST.

```
1 G_dSetPosition := LREAL_TO_DINT(RD78_0000.Axis0001.Md.SetPosition);  
2 G_dSetVelocity := LREAL_TO_DINT(RD78_0000.Axis0001.Md.SetVelocity);  
3
```



G_dSetPosition → D0
G_dSetVelocity → D2

(Catatan) Ketika tipe bilangan nyata presisi ganda dikonversi ke jenis kata ganda bertanda, jika nilai yang akan dikonversi berada di luar rentang dari -2147483648 hingga 2147483647, akan terjadi kesalahan perhitungan.

Tulis program dan parameter ke CPU PLC dan modul Gerak.

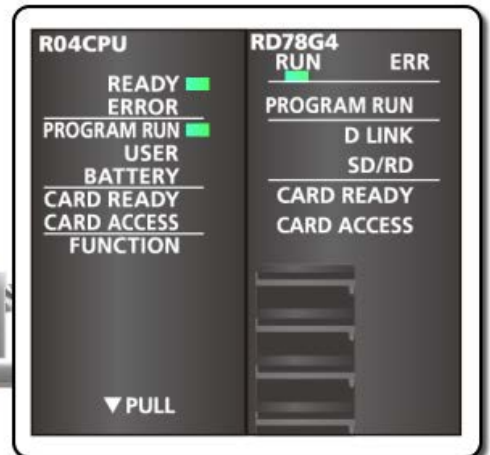
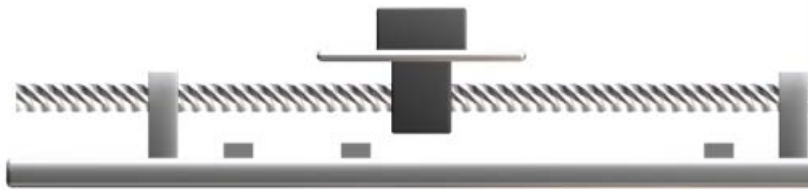
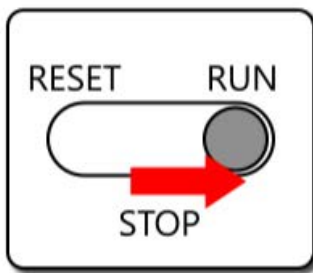
Program ini hanya ditulis ke modul CPU. Parameter axis dan pengaturan label publik harus ditulis ke sisi modul Gerak.

- 1) Setelah semua program di CPU PLC dibangun kembali, pilih [Online] → [Write to PLC] di bilah alat GX Works3 untuk menulis semua data ke CPU PLC.
- 2) Ketika parameter ditulis ke CPU PLC, komunikasi dengan modul Gerak diaktifkan. Pilih [Online] → [Write to Module] di bilah alat Motion Control Setting Function untuk menulis semua data ke modul Gerak.
- 3) Reset CPU PLC untuk menyelesaikan operasi penulisan.

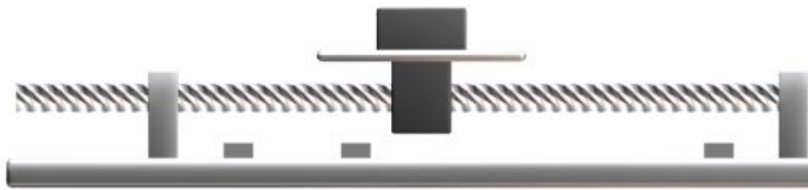
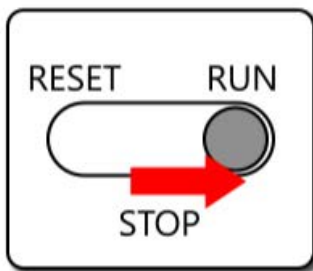
Klik tombol putar di kiri bawah jendela.



Periksa contoh operasi program.
Sebelum memulai pengoperasian, pastikan program dan parameter telah ditulis ke CPU PLC dan modul Gerak.



Atur tuas RUN/STOP/RESET CPU PLC ke RUN.
Lampu READY dan lampu PROGRAM RUN pada kontroler yang dapat diprogram menyala.
Lampu RUN pada modul Gerak menyala.



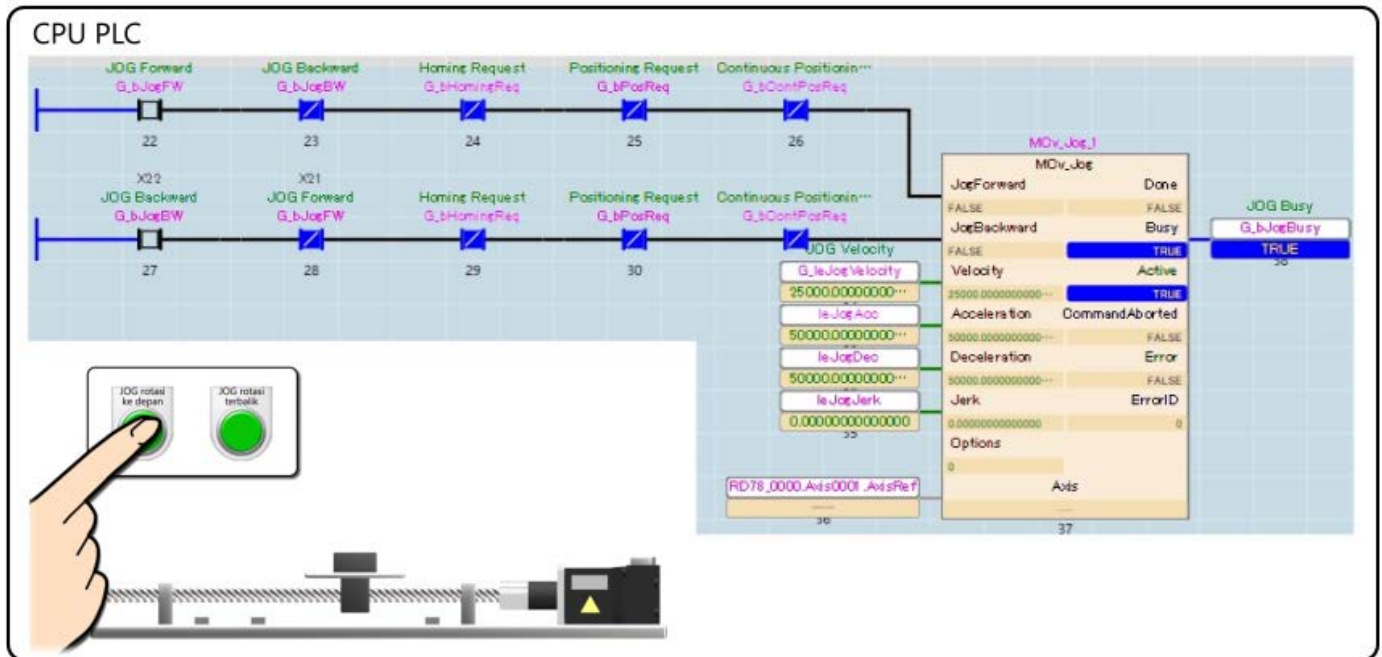
Tunggu hingga lampu PROGRAM RUN pada modul Gerak menyala.
 "r.01" ditampilkan pada penguat servo. (Titik-titiknya menyala.)
 Motor servo memasuki status servo ON.



Hidupkan X20 untuk menjalankan servo OFF.
"r.01" ditampilkan pada penguat servo. (Titik-titiknya berkedip.)
Hidupkan X20 untuk menjalankan servo ON lagi.



Hidupkan JOG putaran maju (X21) untuk menggerakkan arah kenaikan alamat, dan matikan untuk berhenti.
Nyalakan JOG putaran terbalik (X22) untuk menggerakkan arah penurunan alamat, dan matikan untuk berhenti.

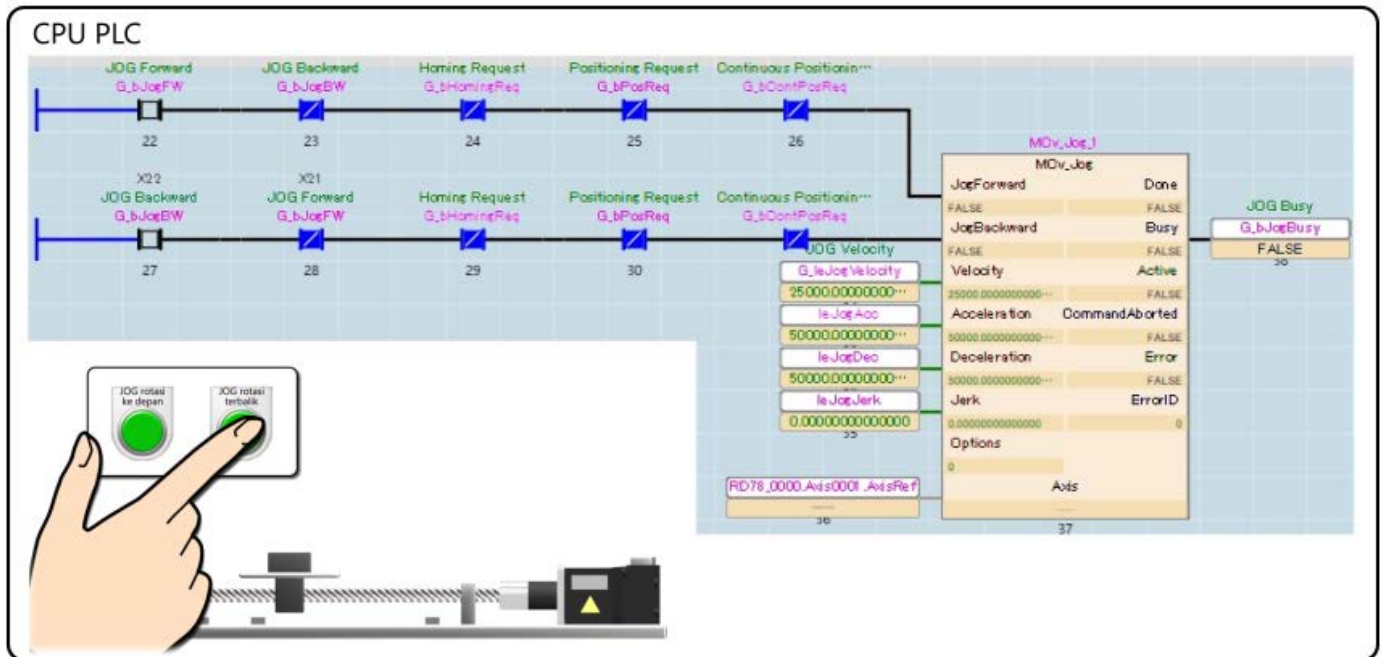


Periksa monitor program.

Saat X21 dihidupkan, input JogForward dari MCv_Jog_1 akan menyala.

Operasi JOG rotasi normal dilakukan.

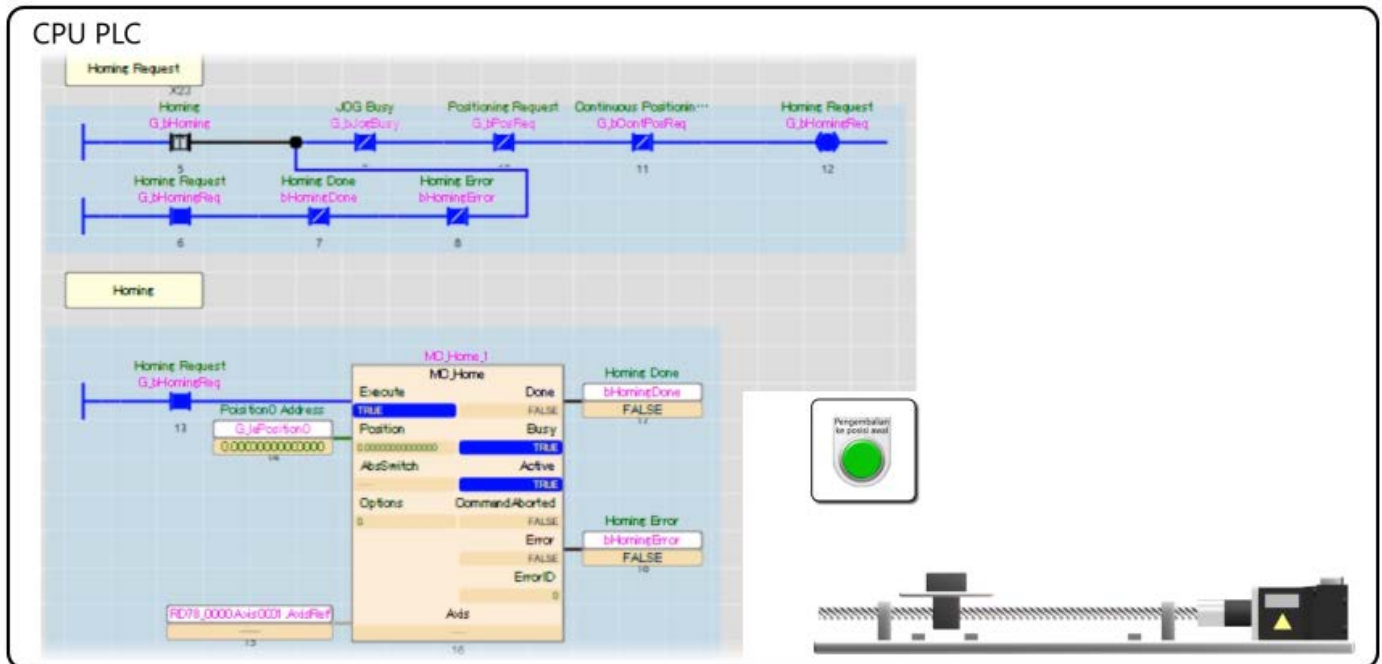
Output Sibuk dan "G_bJogBusy" menyala selama pengoperasian.



Saat X22 dihidupkan, input JogBackward dari MCv_Jog_1 akan menyala.
 Operasi JOG rotasi terbalik dilakukan.
 Output Sibuk dan "G_bJogBusy" menyala selama pengoperasian.

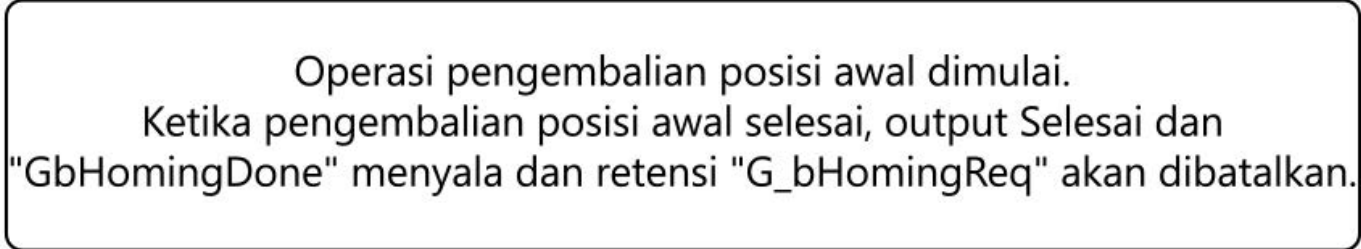


Aktifkan pengembalian posisi awal (X23) untuk memulai pengembalian posisi awal.
Jalankan pengembalian posisi asal dengan metode proximity dog
(33 dikurangi dari Pr.PT45)
Axis nya berhenti sedikit lebih jauh di luar dog,
dan menetapkan titik tersebut sebagai posisi awal.



Periksa monitor program.

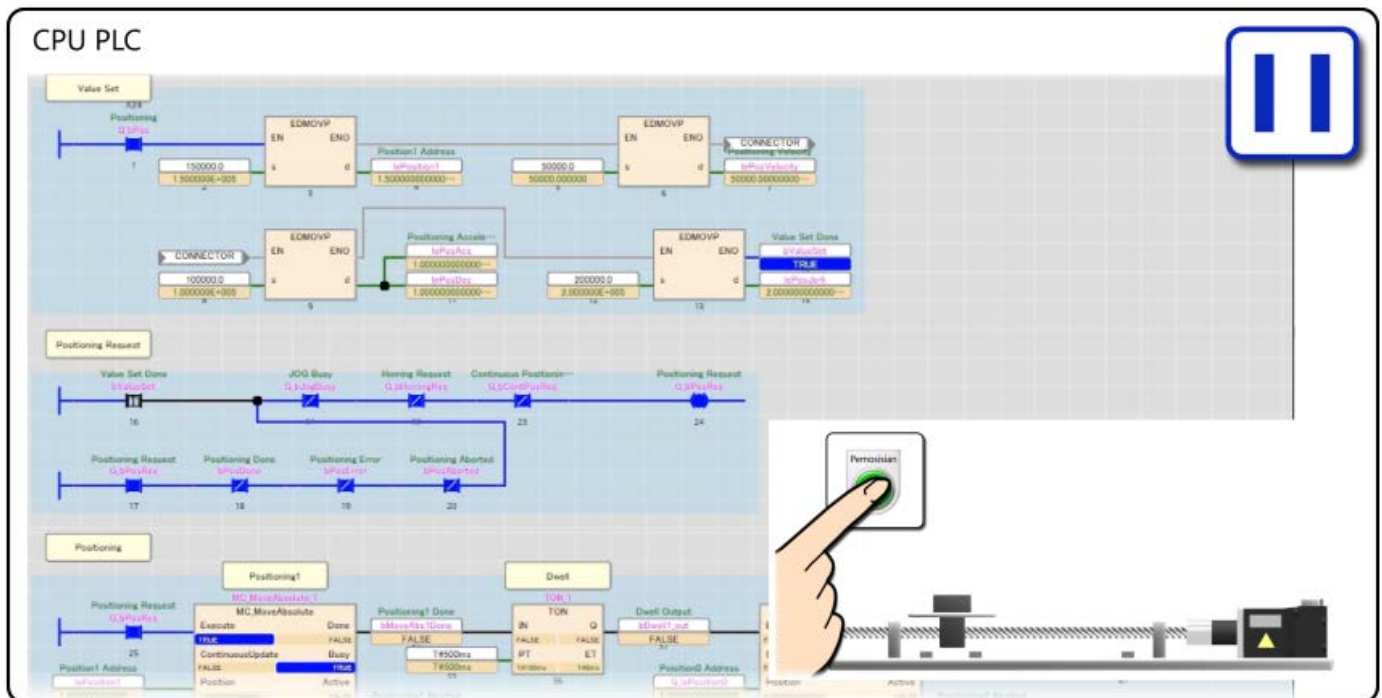
Ketika X23 dihidupkan, alamat posisi awal akan disimpan pada label. "G_bHomingReq", yang merupakan perintah eksekusi MC_Home_1, diaktifkan dan dipertahankan.



"GbHomingDone" menyala dan retensi "G_bHomingReq" akan dibatalkan.



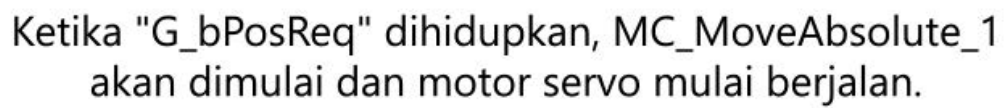
Mengaktifkan pemosisian dimulai (X24) akan memulai gerakan bolak-balik.
Axis bergerak maju 150 mm dan berhenti selama 0,5 detik, serta mundur 150 mm dan berhenti selama 0,5 detik.



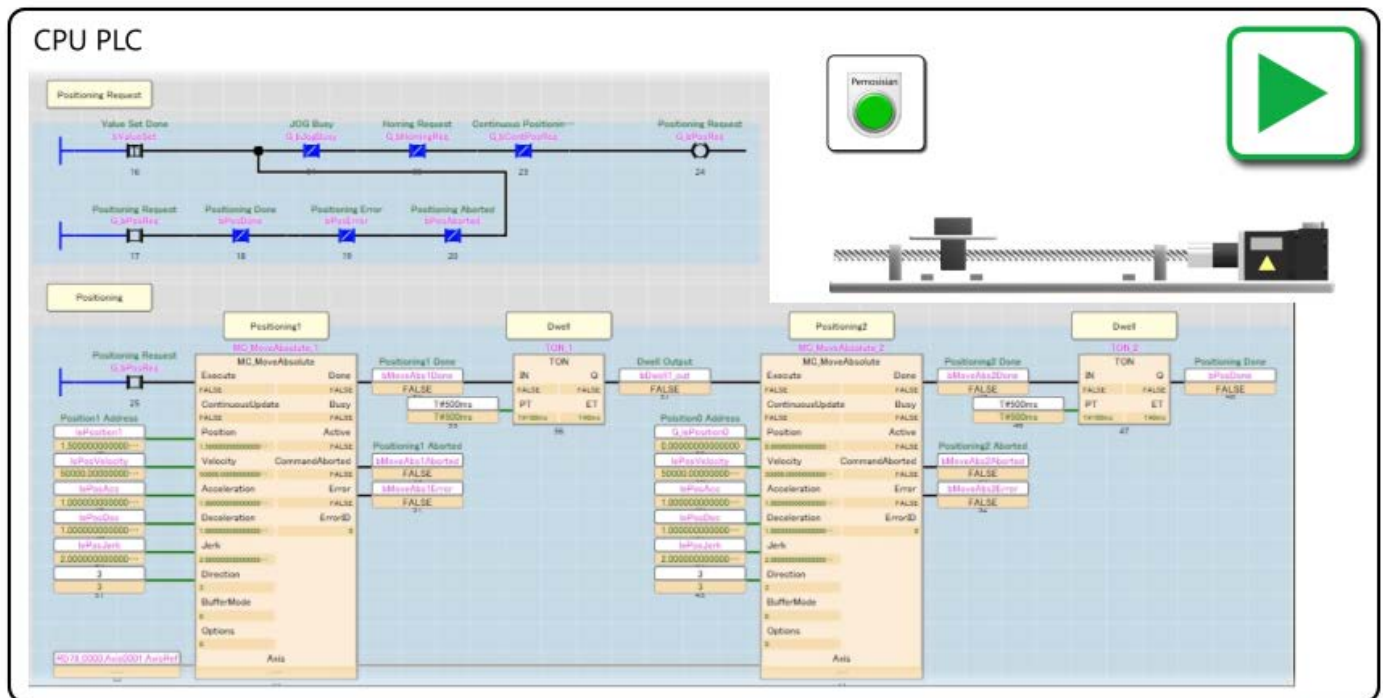
Periksa monitor program.

Saat X24 diaktifkan, data untuk pemosisian disimpan ke setiap label, dan "bValueSet" diaktifkan.

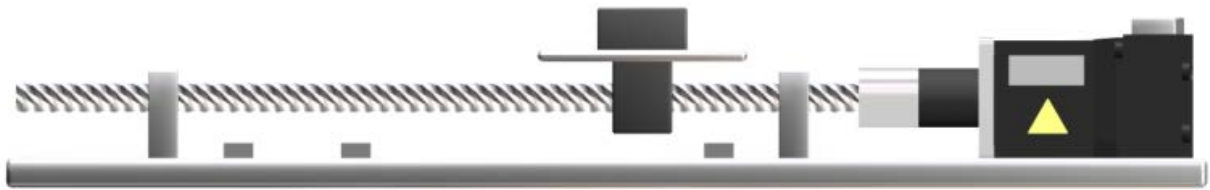
"G_bPosReq", yang merupakan perintah eksekusi `MC_MoveAbsolute_1`, diaktifkan dan dipertahankan pada tepi naik "bValueSet".



Ketika "G_bPosReq" dihidupkan, MC_MoveAbsolute_1 akan dimulai dan motor servo mulai berjalan.



Ketika pemosisian oleh MC_MoveAbsolute_1 selesai, TON_1, yang merupakan tempat tinggal, akan beroperasi. Ketika 500 ms sudah terlewati, MC_MoveAbsolute_2 akan dimulai dan motor servo mulai berjalan.

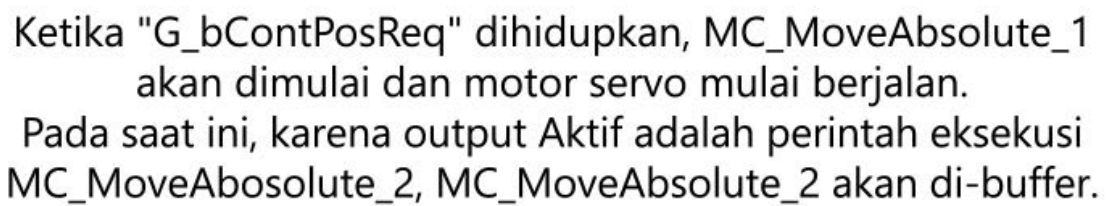


Aktifkan pemosisian dimulai berkelanjutan (X25) untuk memulai pengoperasian mode buffer (mc_Buffered).



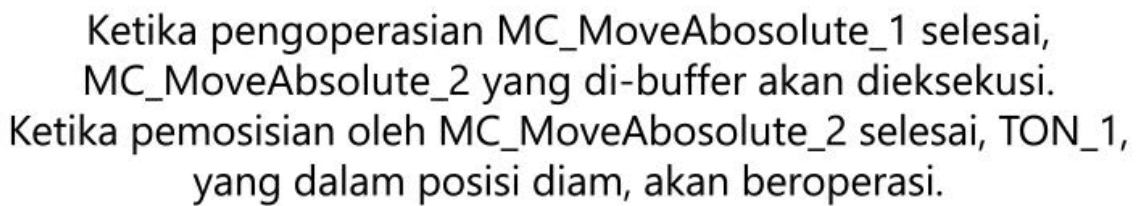
Saat X25 diaktifkan, data untuk pemosisian disimpan ke setiap label, dan "bValueSet" diaktifkan.

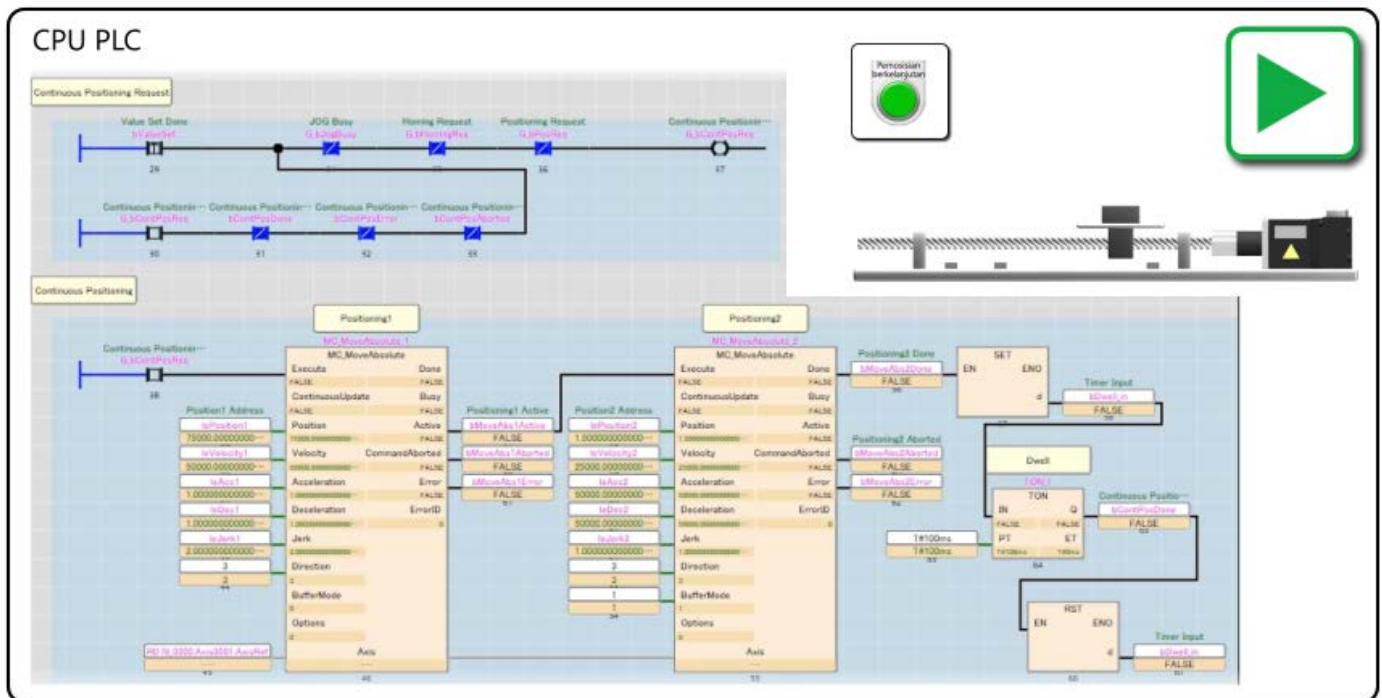
"G_bContPosReq", yang merupakan perintah eksekusi MC_MoAbsolute_1, diaktifkan dan dipertahankan pada tepi naik "bValueSet".



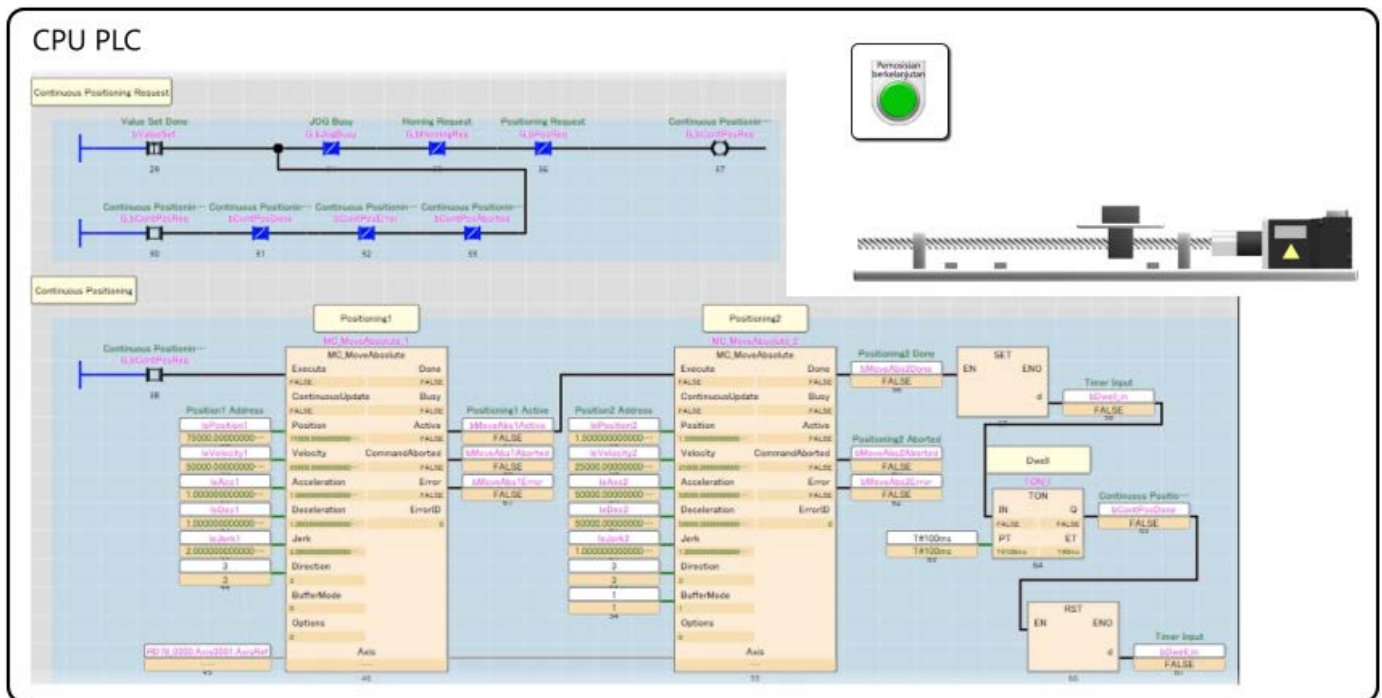
Ketika "G_bContPosReq" dihidupkan, MC_MoveAbsolute_1 akan dimulai dan motor servo mulai berjalan.

Pada saat ini, karena output Aktif adalah perintah eksekusi MC_MoveAbosolute_2, MC_MoveAbsolute_2 akan di-buffer.





Ketika 100 ms berlalu, retensi "G_bContPosReq" akan dihapus, dan diatur ulang ke keadaan awal.



Tindakan ini akan menyelesaikan pemeriksaan operasi.
Buka halaman berikutnya.

Dalam bab ini, Anda telah mempelajari:

- Mendaftarkan Library FB Modul Gerak
- Membuat Proyek
- Cara Menggunakan Motion Control FB
- Contoh Deskripsi Program
- Contoh Pemeriksaan Operasi Program

Poin penting

Mendaftarkan Library FB Modul Gerak	• Library FB harus didaftarkan ke GX Works3 untuk menggunakan Motion control FB dalam CPU PLC.
Membuat Proyek	• Konfigurasi parameter axis dan pengaturan lainnya seperti saat memprogram ke modul Gerak.
Cara Menggunakan Motion Control FB	• Gerakan yang mengendalikan FB dapat ditempatkan ke editor program dengan menyeret dan melepaskannya dari tab Library di jendela Element Selection GX Works3. • Hubungkan kontak dan label ke input/output FB.
Contoh Deskripsi Program	• Anda telah membuat program yang mirip dengan contoh program di bab 2 dan bab 3 hanya dengan menggunakan CPU PLC.
Contoh Pemeriksaan Operasi Program	• Anda telah memeriksa pengoperasian program contoh di video.

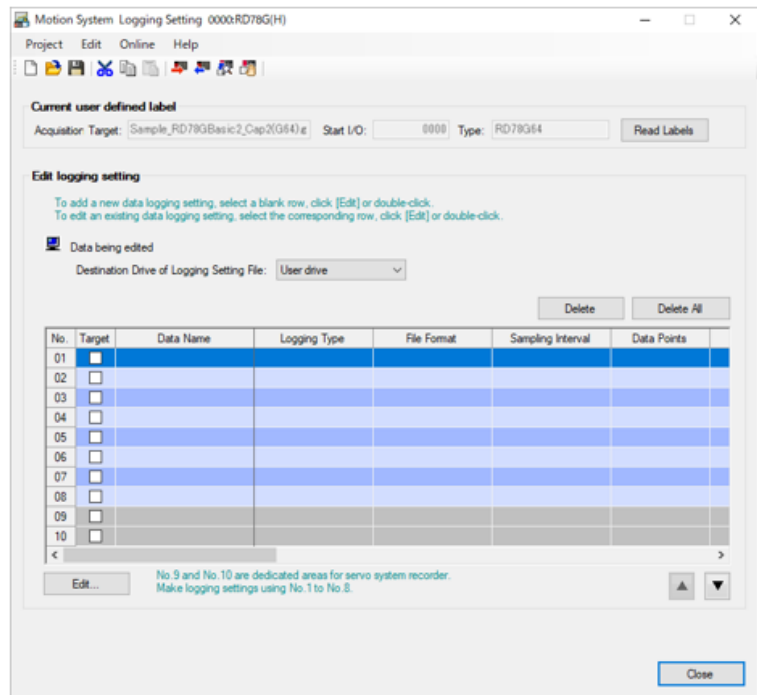
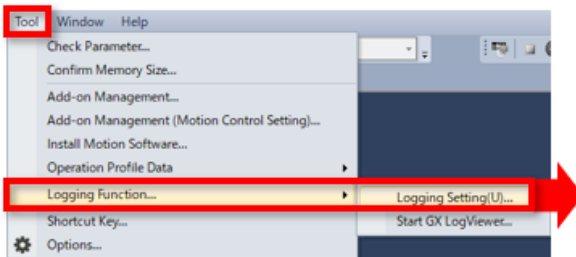
Pada bab ini dijelaskan cara mencatat data modul Motion dan menampilkannya dalam grafik. Dalam kursus ini, program awal pemosisian dari program contoh di bab 2 dan bab 3 akan dicatat sebagai contoh.

(Catatan) Program di bab 4 tidak dapat dicatat dengan prosedur yang dijelaskan dalam bab ini.

"CPU Module Logging Configuration Tool" harus digunakan.

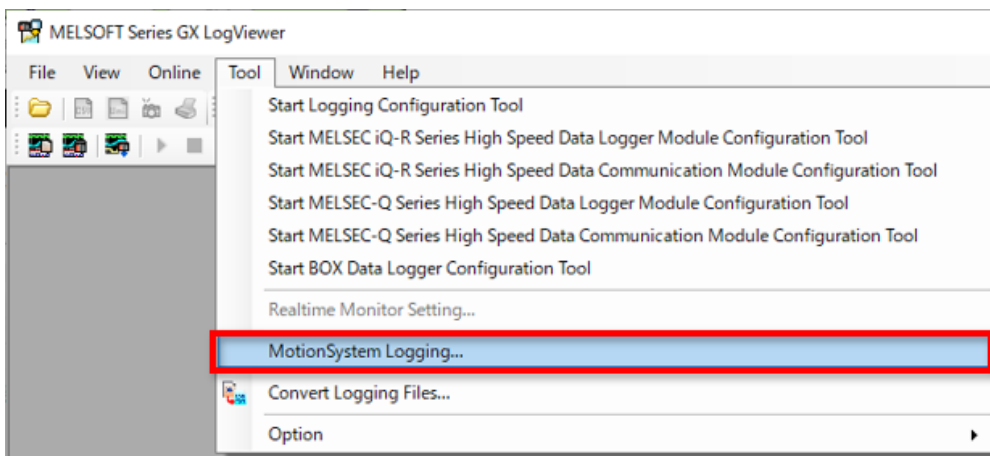
5.1 Memulai Alat Konfigurasi Pencatatan

Pilih [Tool] → [Logging Function] → [Logging Setting] dari bilah alat layar Motion Control Setting Function. Alat pengaturan pencatatan sistem gerak akan dimulai.



[Poin]

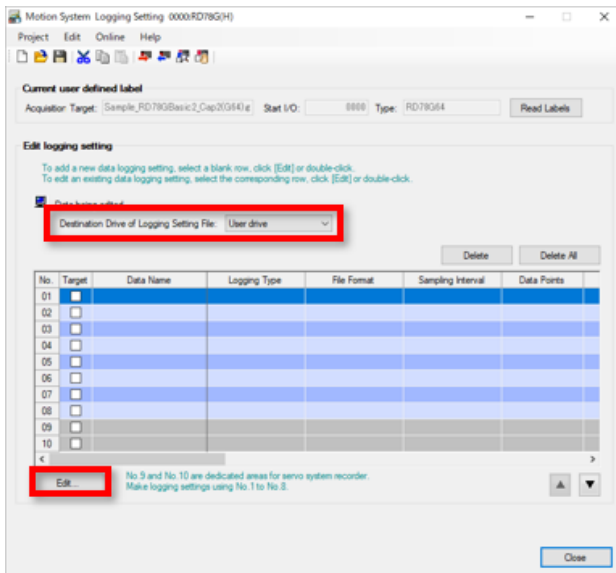
Alat pengaturan pencatatan sistem gerak dapat dimulai dari [Tool] → [MotionSystem Logging] dalam GX LogViewer.



- (1) Di kolom edit pengaturan pencatatan pada alat pengaturan pencatatan sistem gerak, tetapkan destinasi penyimpanan pencatatan data.

Kemudian, klik tombol [Edit].

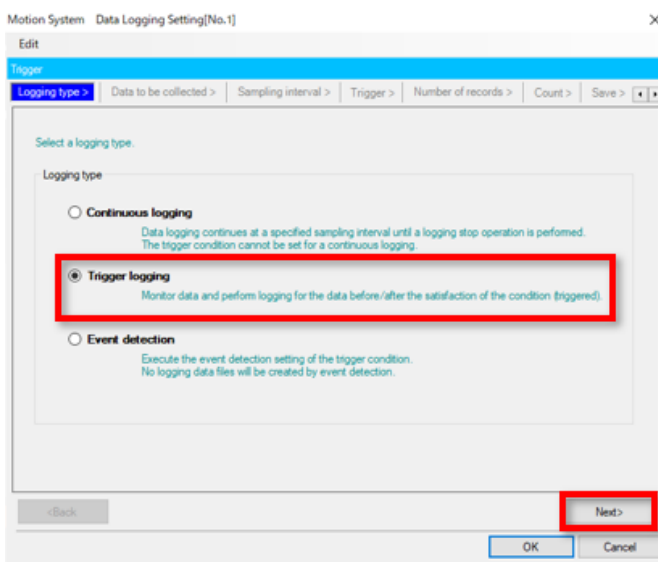
Layar Data Logging Setting ditampilkan.



- (2) Pilih [Logging type] dari pencatatan berkelanjutan, pemicu pencatatan, dan deteksi peristiwa.

Dalam kursus ini, akan dijelaskan detail pemicu pencatatan.

Pilih pemicu pencatatan, dan klik tombol [Next].



(3) Label data yang akan dicatat terdaftar di [Data to be collected].

1) Label global

Masukkan nama label global ke kolom nama data.
Di kolom jenis data, pilih jenis data label.

2) Label lokal

Masukkan nama data dalam format "nama program/nama label lokal".
Di kolom jenis data, pilih jenis data label.

3) Jenis data terstruktur

Klik tombol [Label Input Assistant] dan pilih anggota jenis data terstruktur dari daftar.
Pilih dari daftar dan klik tombol [Register] untuk mewakili data yang akan dikumpulkan.

No.	Display	Type	Data Name	Data Type	Size [Word]	Axis Name
0001	☒	VAR	G_bPosReq	Bit	1	
0002	☒	VAR	Positioning/bMoveAbs1Done	Bit	1	
0003	☒	VAR	Positioning/bDwell1_out	Bit	1	
0004	☒	VAR	Positioning/bMoveAbs2Done	Bit	1	
0005	☒	VAR	G_bPosDone	Bit	1	
0006	☒	VAR	Axis0001.Md.SetPosition	Float (double precision)	4	Axis0001
0007	☒	VAR	Axis0001.Md.SetVelocity	Float (double precision)	4	Axis0001

Dalam kursus ini, data berikut dicatat sebagai contoh.

Nama data
G_bPosReq
Positioning/bMoveAbs1Done
Positioning/bDwell1_out
Positioning/bMoveAbs2Done
G_bPosDone
Axis0001.Md.SetPosition
Axis0001.Md.SetVelocity

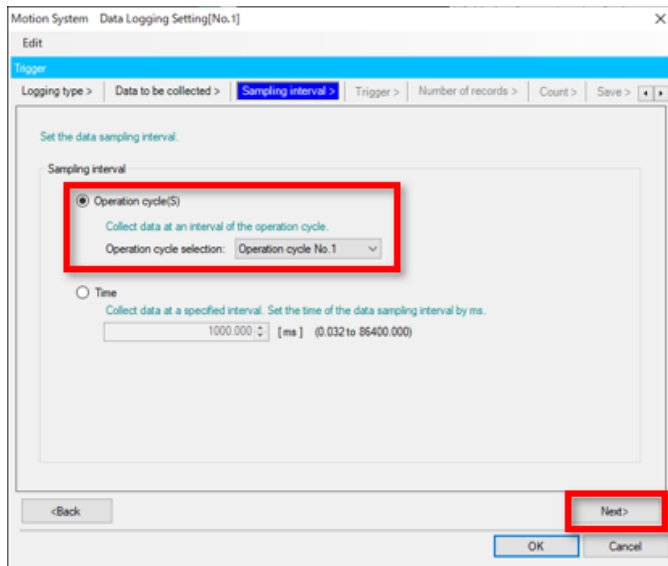
Klik tombol [Next] ketika pendaftaran selesai.

Name	Data Name (Label)
Md Object Data_VelActualValue	* Md Io_VelActualValue
Md Jerk Limit Value	* Md JerkLimit
Md Control Cycle	* Md OperationCycle
Md Operation Setting at Overrun	* Md OverrunOperation
Md Current Position Restoration Status	* Md PosRestoration_Status
Md Execution Profile ID No.	* Md ProfileID
Md Set Acceleration	* Md SetAcceleration
Md Set Position	* Md SetPosition
Md Set Velocity	* Md SetVelocity
Md Slave Emulating	* Md SlaveEmulate_Enable
Md Start Permission at Homing Uncompleted	* Md StartableAtUnhomed
Md Deceleration at Stop	* Md StopMode_Deceleration
Md Stop Signal.Signal Detection Method	* Md StopSignal.Detection

(4) Atur interval pengambilan sampel di [Sampling Interval].

Dalam kursus ini, gunakan siklus operasi No. 1 untuk pengambilan sampel.

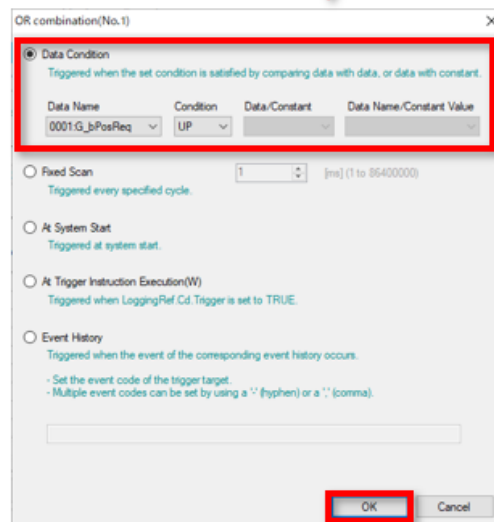
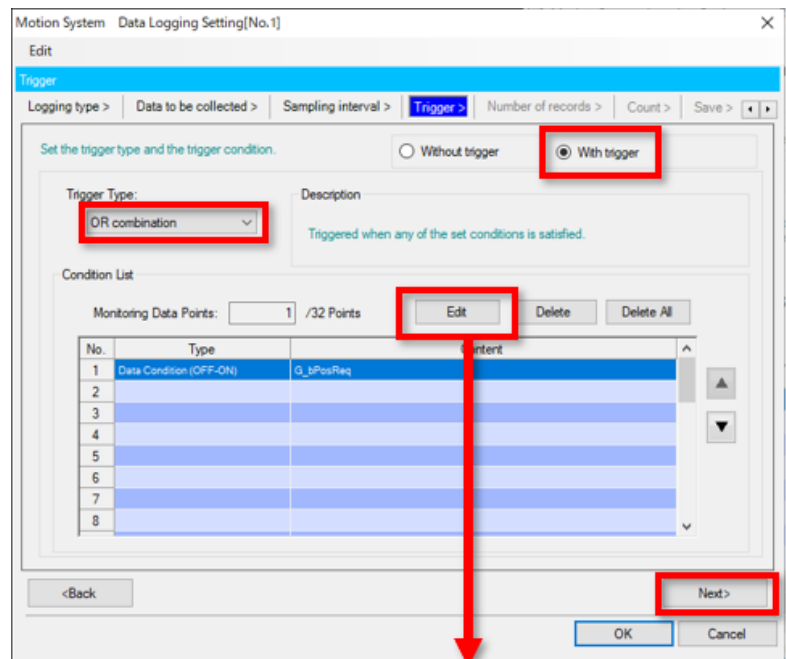
Setelah memilih interval pengambilan sampel, klik tombol [Next].



(5) Kondisi untuk memulai pencatatan diatur di [Trigger].

Dalam kursus ini, bit startup yang merupakan sinyal start positioning digunakan sebagai pemicu.

- 1) Pilih [With trigger].
- 2) Pilih "OR combination" untuk jenis pemicu.
- 3) Pilih No. 1 dari daftar kondisi, dan klik tombol edit. Subjendela akan muncul.
- 4) Pilih "Data Condition", dan pilih "0001:G_bPosReq" untuk nama datanya. Pilih "UP" untuk kondisinya. Jika pemilihan sudah selesai, klik tombol [OK].
- 5) Setelah Anda kembali ke layar asli, klik tombol [Next].



(6) Jumlah titik pengambilan sampel ditetapkan pada [Number of records].

Dalam kursus ini, Jumlah rekaman (sebelum pemicu) diatur ke "500", dan Jumlah rekaman (setelah pemicu) menjadi "19500". Jika pengaturan sudah selesai, klik tombol [Next].

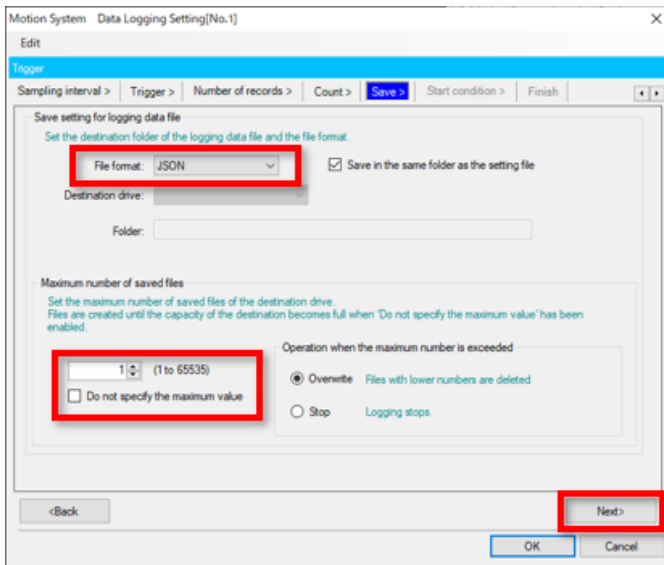
The screenshot shows the 'Data Logging Setting' dialog box with the 'Number of records' tab selected. The dialog has a title bar 'Motion System Data Logging Setting[No.1]' and a close button. Below the title bar is an 'Edit' button and a 'Trigger' tab. The main area contains three input fields: 'No. of records (before trigger):' with a value of 500, 'No. of records (after trigger):' with a value of 19500, and 'Total No. of records:' with a value of 20000. Each field has a '[Record]' button and a range in parentheses. At the bottom, there are '<Back', 'Next>', 'OK', and 'Cancel' buttons. The 'Next>' button is highlighted with a red box.

(7) Jumlah pencatatan diatur di [Count]. Dalam kursus ini, hitungannya diatur ke 1.

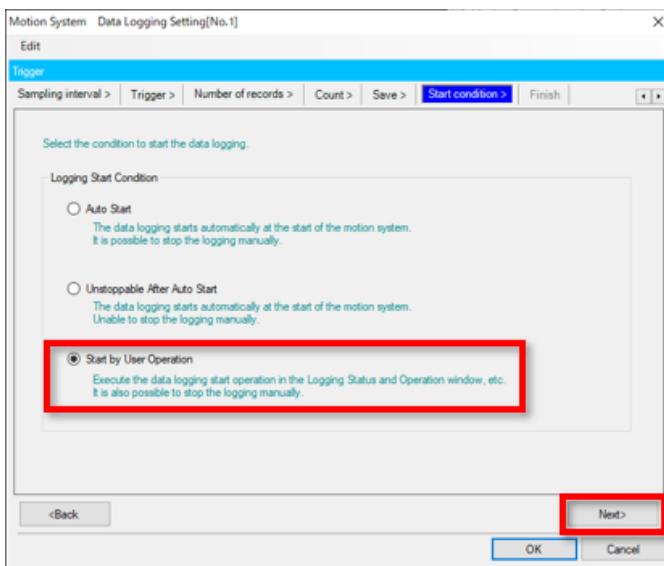
Jika pengaturan sudah selesai, klik tombol [Next].

The screenshot shows the 'Data Logging Setting' dialog box with the 'Count' tab selected. The dialog has a title bar 'Motion System Data Logging Setting[No.1]' and a close button. Below the title bar is an 'Edit' button and a 'Trigger' tab. The main area contains two radio button options: 'Specified Count' (selected) and 'Specified Number of Saved Files'. The 'Specified Count' option has a value of 1 and a range of (1 to 32767). Below the radio buttons, there is a description: 'Execute trigger logging repeatedly for the specified count. The operation will be 'Overwrite' when the maximum number of saved files is exceeded.' At the bottom, there are '<Back', 'Next>', 'OK', and 'Cancel' buttons. The 'Next>' button is highlighted with a red box.

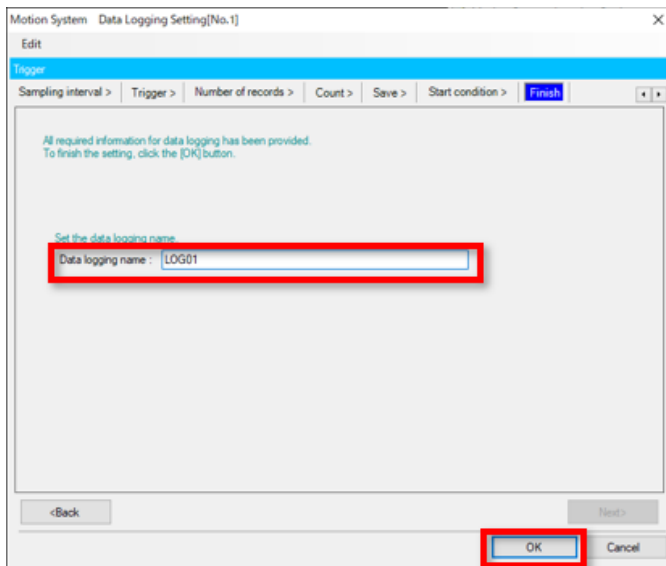
- (8) Format file dan jumlah file yang disimpan dari data pencatatan diatur di [Save].
Dalam kursus ini, nilai default (format: JSON, jumlah file yang disimpan: 1) sudah diatur.
Jika pengaturan sudah selesai, klik tombol [Next].



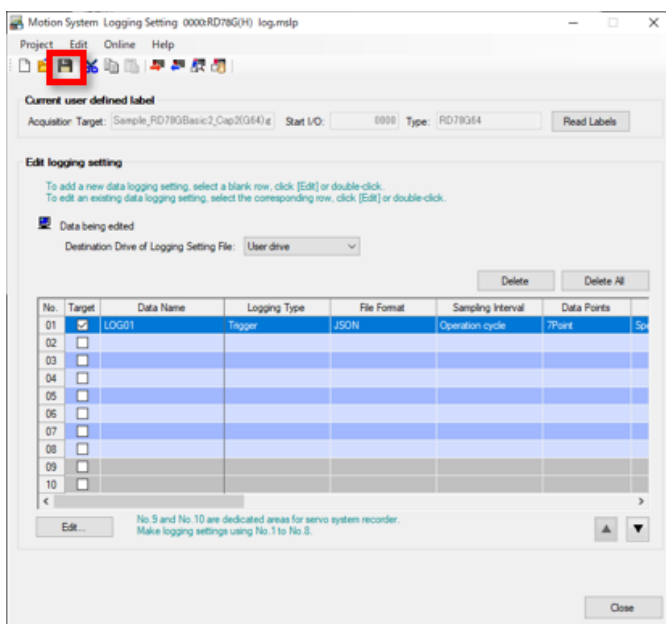
- (9) Kondisi untuk memulai pencatatan diatur di [Start condition].
Dalam kursus ini, "Start by User Operation" sudah diatur.
Jika pengaturan sudah selesai, klik tombol [Next].



- (10) Nama pencatatan data diatur di [Finish].
 Dalam kursus ini, nilai default ditetapkan ke (LOG01).
 Jika pengaturan sudah selesai, klik tombol [Next].



- (11) Kembali ke alat pengaturan pencatatan sistem gerak.
 Pengaturan yang telah dikonfigurasi dapat disimpan.
 Klik ikon simpan dan simpan ke tujuan pilihan Anda.

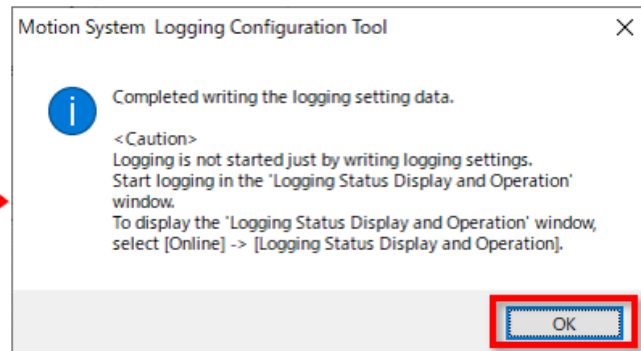
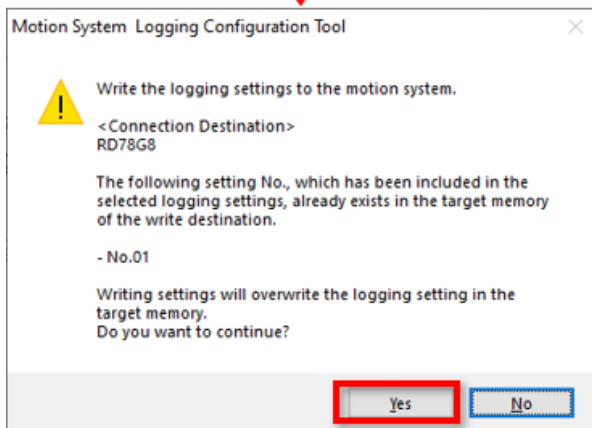
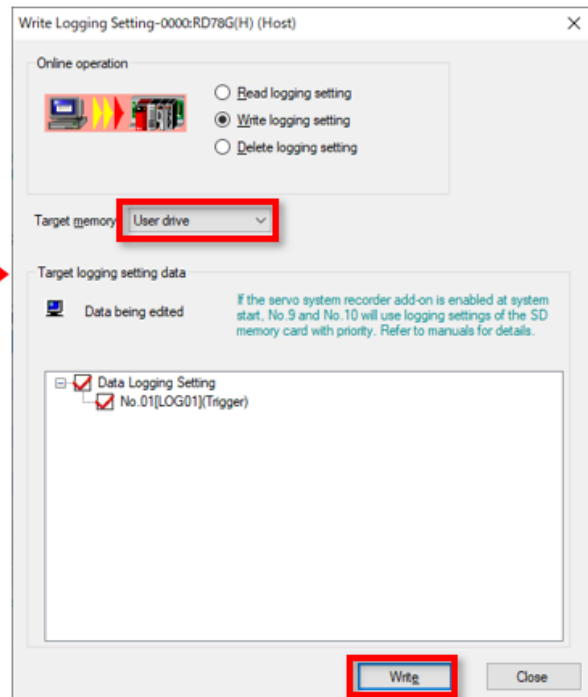
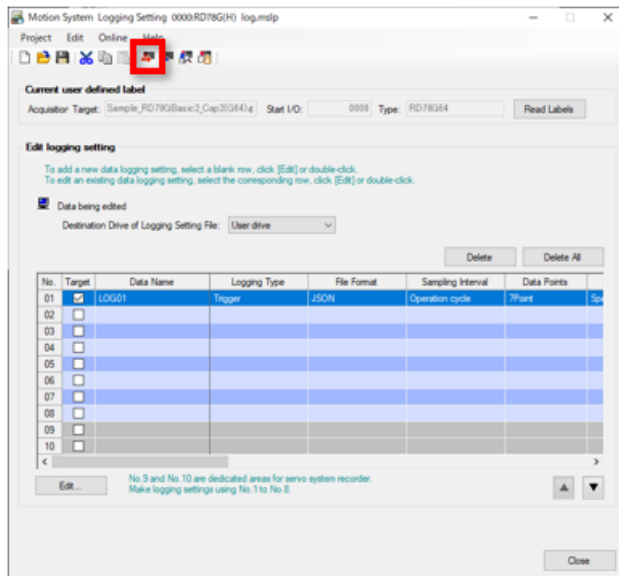


Informasi pengaturan pencatatan yang ditulis.

Klik ikon pengaturan pencatatan, pilih memori target, dan klik tombol [Write].

Jendela konfirmasi akan muncul.

Klik tombol [Yes] dan lanjutkan. Jika penulisan sudah selesai, klik tombol [OK] dan tutup layar.



Ketika "Start by User Operation" diatur di 5.2 (9), klik ikon [Logging Status and Operation] untuk menampilkan layar [Logging Status and Operation] dan untuk memulai pencatatan.

Ketika nama pengaturan data pencatatan yang akan dijalankan dipilih, tombol [Start] akan diklik, LoggingStatus switches berubah ke "Waiting for trigger".

Ketika program dijalankan dalam keadaan ini dan kondisi pemicu (ketika X24 AKTIF dalam contoh kursus ini) terpenuhi, statusnya beralih ke "Triggerred".

Ketika pencatatan selesai, statusnya akan beralih ke "CollectionCompleted" dari "Saving".

The image shows two screenshots of the Motion System software interface and a flowchart illustrating the logging status transitions.

Left Screenshot: Motion System Logging Setting

Current user defined label: Acquisition Target: Sample_RD78G8asic2_Cap20(G44) Start I/O: 0000 Type: RD78G84 Read Labels

Edit logging setting

To add a new data logging setting, select a blank row, click [Edit] or double-click.
To edit an existing data logging setting, select the corresponding row, click [Edit] or double-click.

Data being edited

Destination Drive of Logging Setting File: User drive

No.	Target	Data Name	Logging Type	File Format	Sampling Interval	Data Points
01	☒	LOG01	Trigger	JSON	Operation cycle	77urst
02	☐					
03	☐					
04	☐					
05	☐					
06	☐					
07	☐					
08	☐					
09	☐					
10	☐					

No. 9 and No. 10 are dedicated areas for servo system recorder.
Make logging settings using No. 1 to No. 8.

Right Screenshot: Motion System Logging Status and Operation

Monitor status: Monitoring Stop

User Drive Free Volume: 47852 MB RAM Drive Free Volume: 47852 MB SD Memory Card Free Volume: 47852 MB

Logging status and operation

Display the executing logging status or execute a logging start/stop operation.
To execute logging, check the operation target No., and select the reading destination of logging settings, then click [Start].
In a row where the logging is stopped, the setting of the selected logging setting reading destination is displayed, while in a row where the logging is executing, the setting value at the logging start is displayed.
Clicking [Update] will read the logging setting of the row where the logging is stopped.

Motion System Data

Select All Select None Update

No.	Target	Data Name	Logging Type	File Format	Sampling Interval	Stopped
01	☒	User drive	LOG01	Trigger		☒
02	☐	User drive				☐
03	☐	User drive				☐
04	☐	User drive				☐
05	☐	User drive				☐
06	☐	User drive				☐
07	☐	User drive				☐
08	☐	User drive				☐
09	☐	User drive				☐
10	☐	User drive				☐

Logging operation

Select target data for logging operation and click an appropriate button.

Start Stop

Flowchart:

```

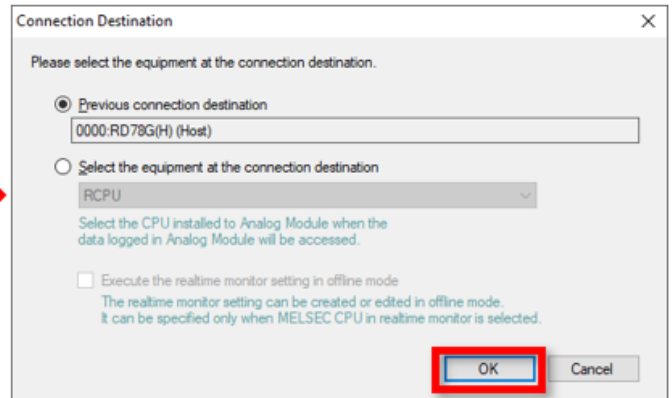
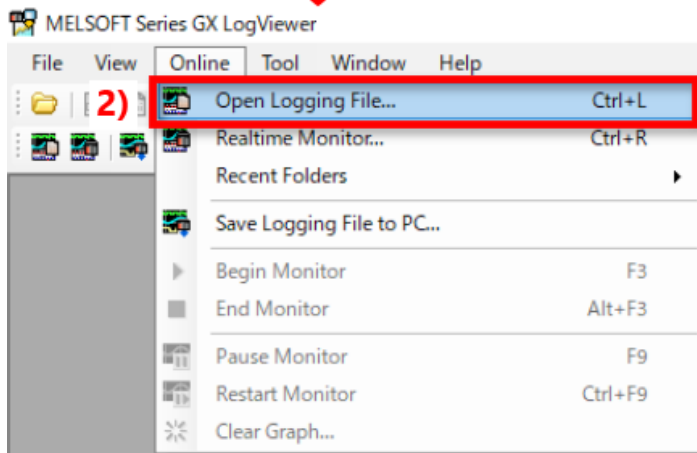
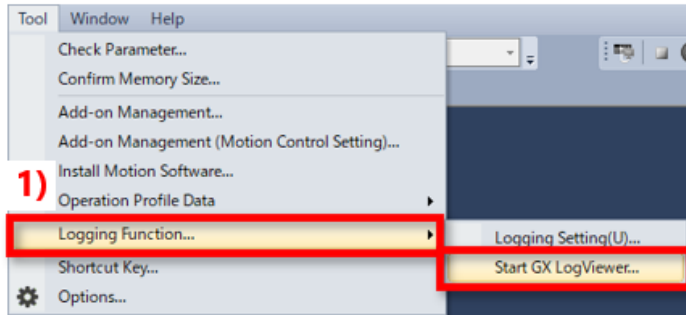
graph TD
    A[Logging Status Waiting for trigger] --> B[Logging Status Triggered]
    B --> C[Logging Status Saving]
    C --> D[Logging Status CollectionCompleted]
  
```

GX LogViewer digunakan untuk membaca data pencatatan.

Pilih [Tool] → [Logging Function] → [Start GX LogViewer] dari bilah alat layar Motion Control Setting Function.

Saat GX LogViewer dimulai, pilih [Online] → [Open Logging File].

Pilih "0000:RD78G(H) (Host)" di layar Connection Destination. (Catatan)

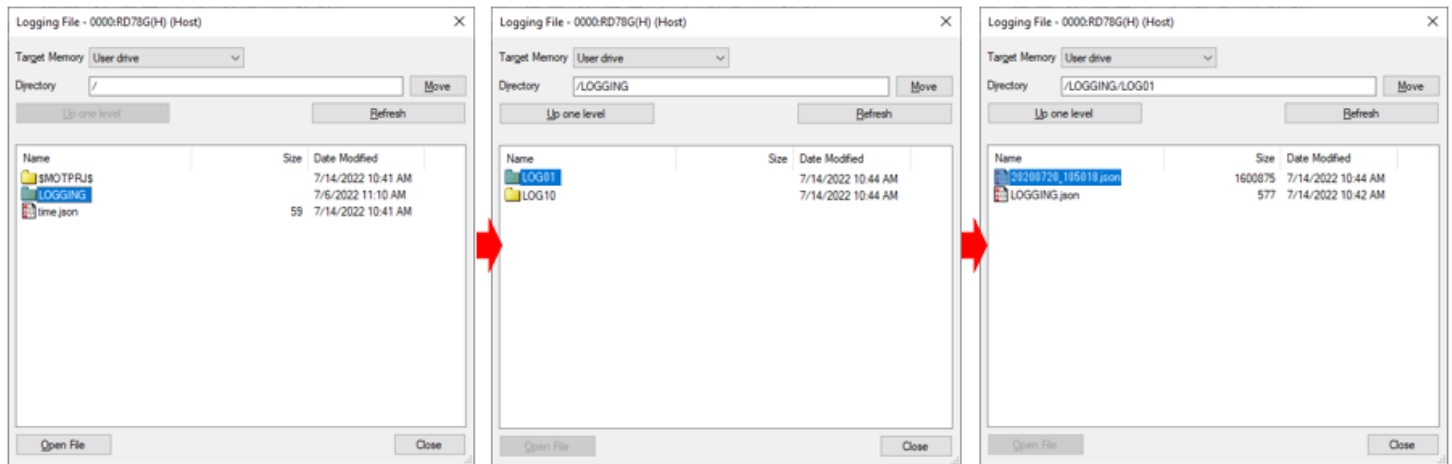


(Catatan) Jika GX LogViewer sudah dimulai, dan komunikasi dengan modul Gerak sudah diatur, layar ini tidak ditampilkan.

Pilih file pencatatan yang akan dibaca.

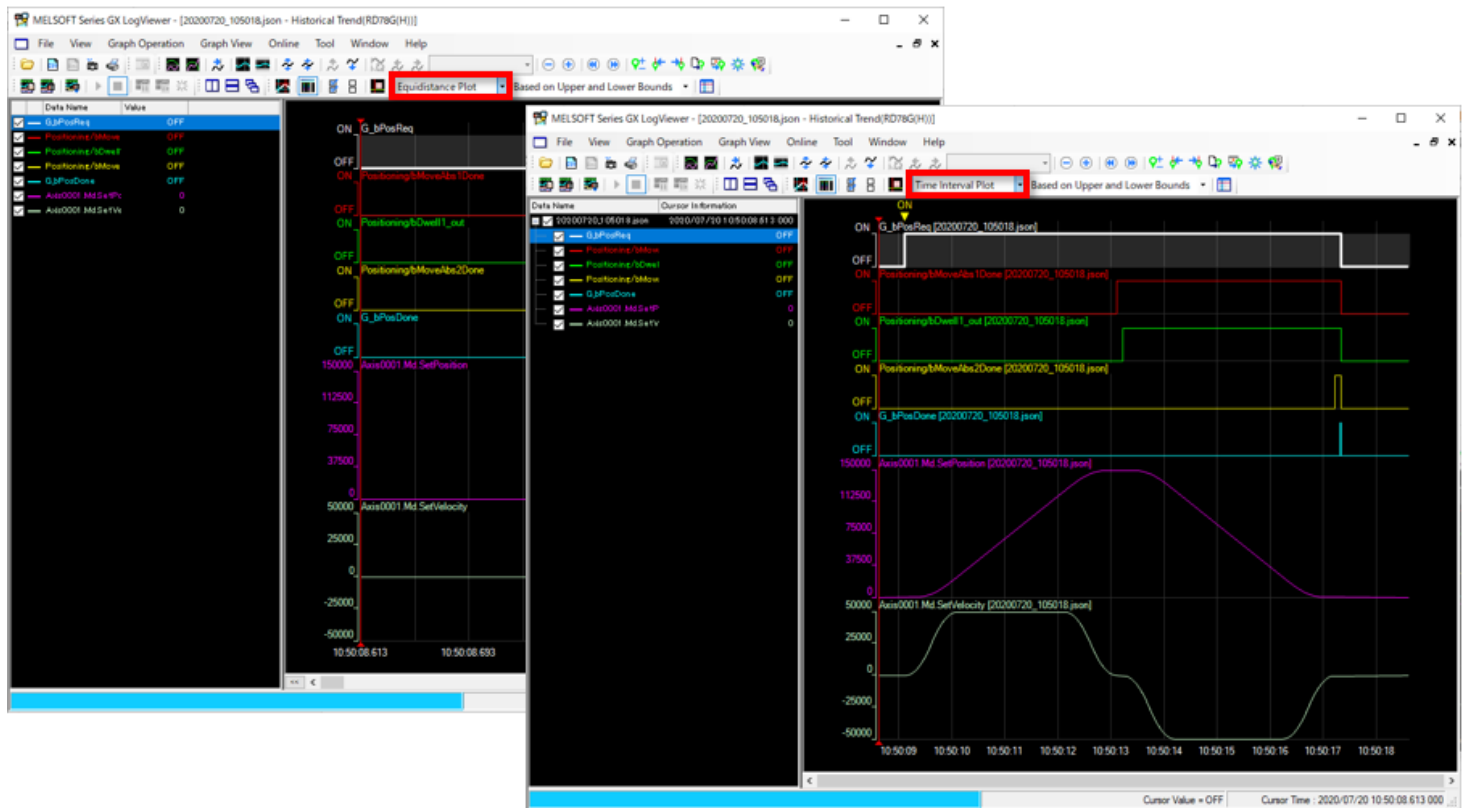
Dalam contoh bab ini, pengguna memasukkan "LOGGING" → "LOG1" → "(Logged date and time).json" dipilih.

Pilih nama file dan klik tombol [Open File].



Data bentuk gelombang yang tercatat di GX LogViewer akan ditampilkan.

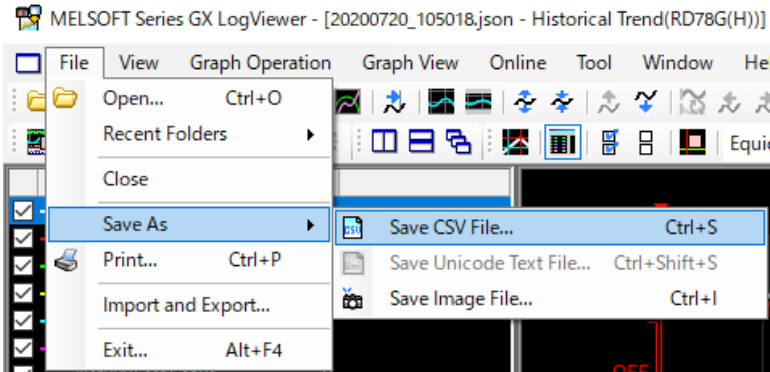
Ketika format plot diubah dari "Equidistance Plot" ke "Time Interval Plot", seluruh bentuk gelombang yang dicatat dapat ditampilkan.



Data bentuk gelombang yang dicatat dapat disimpan sebagai file csv atau file json.
(Jika data dicatat dalam format CSV, maka dapat disimpan sebagai file CSV.)

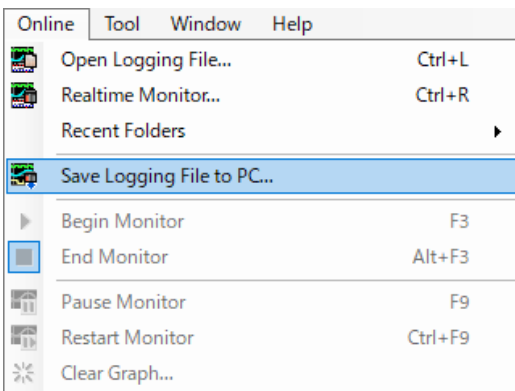
1) Jika menyimpan sebagai file csv

Pilih [File] → [Save As] → [Save CSV File] dari bilah alat GX LogViewer.



2) Jika menyimpan sebagai file json

Pilih [Online] → [Save Logging File to PC] dari bilah alat GX LogViewer.



Dalam bab ini, Anda telah mempelajari:

- Memulai Alat Konfigurasi Pencatatan
- Mengatur Data yang Akan Dicatat
- Menulis Pengaturan Pencatatan
- Memulai Pencatatan
- Membaca Data Pencatatan
- Menyimpan Data Pencatatan

Poin penting

Memulai Alat Konfigurasi Pencatatan	• Mulai alat pengaturan pencatatan sistem gerak dari motion control setting function.
Mengatur Data yang Akan Dicatat	• Atur data yang akan dicatat, kondisi pemicu, dan lain-lain dengan mengikuti prosedur yang ditampilkan pada alat pengaturan pencatatan sistem gerak.
Menulis Pengaturan Pencatatan	• Tulis data pengaturan pencatatan ke modul Gerak sebelum pencatatan.
Memulai Pencatatan	• Ketika kondisi awal pencatatan diatur ke "Start by User Operation", klik tombol mulai di layar "Logging Status and Operation" untuk mulai mencatat.
Membaca Data Pencatatan	• GX LogViewer digunakan untuk membaca data pencatatan.
Menyimpan Data Pencatatan	• Data bentuk gelombang yang dicatat dapat disimpan sebagai file csv atau file json.

Kini setelah Anda menyelesaikan semua pelajaran **MELSEC iQ-R Series Motion Module Basics (Kontrol Pemosisian RD78G(H))**, Anda siap untuk mengikuti tes akhir. Jika Anda masih belum memahami topik yang sudah dibahas, silakan gunakan kesempatan ini untuk meninjau topik tersebut.

Total terdapat 4 pertanyaan (7 pilihan) dalam Tes Akhir ini.

Anda dapat mengikuti tes akhir sesering mungkin.

Hasil penilaian

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

		1	2	3	4	5	6	7	8	9	10	11	12	
Coba lagi	Tes 1	✓	✓	✓	✗									Jumlah total pertanyaan: 28
	Tes 2	✓	✓	✓	✓									Jawaban yang benar: 23
	Tes 3	✓												Persentase: 82 %
	Tes 4	✓	✓											
	Tes 5	✓	✓											
Coba lagi	Tes 6	✓	✗	✗	✗									
	Tes 7	✓	✓	✓	✓									
	Tes 8	✓	✓	✓	✓	✓								
	Tes 9	✓												
Coba lagi	Tes 10	✗												

Untuk berhasil lulus tes, diperlukan jawaban yang benar sebanyak **60%**.

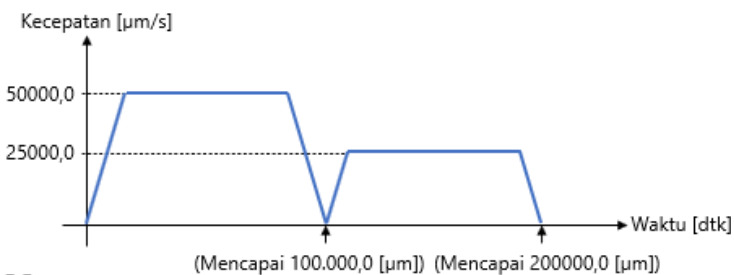
Pilih deskripsi label publik yang benar. (Anda dapat memilih beberapa jawaban.)

- ☐ Label publik adalah label bersama yang dapat digunakan di modul Gerak dan CPU PLC.
- ☐ Label publik didaftarkan dari label global CPU PLC.
- ☐ Ketika label global diatur ke label publik, pilih apakah label dibaca atau ditulis dari/ke CPU PLC.

FB1 dijalankan terlebih dahulu, lalu FB2 dijalankan setelahnya.

Ketika posisi target dan kecepatan target FB1 dan FB2 seperti yang ditunjukkan pada tabel berikut, pilih mode buffer yang dilakukan selanjutnya.

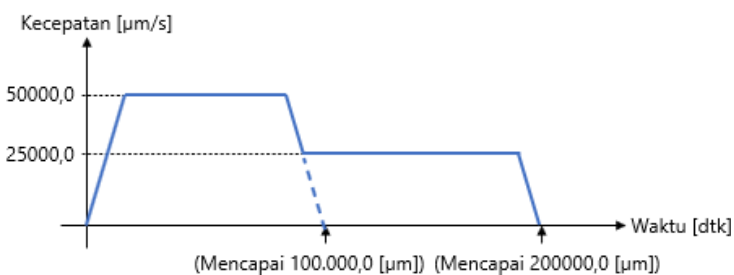
	Posisi target	Kecepatan target
FB1	100000,0 [μm]	50000,0 [μm/s]
FB2	200000,0 [μm]	25000,0 [μm/s]

Q1**Q1**

Pilih jawaban yang sesuai.

**Q2**

Pilih jawaban yang sesuai.

**Q2**

Q1: 1: mcAborting
2: mcBuffered
3: mcBlendingNext
4: mcBlendingPrevious

Q2: 1: mcBlendingNext dan mcBlendingHigh
2: mcBlendingPrevious dan mcBlendingHigh
3: mcBlendingNext dan mcBlendingLow
4: mcBlendingPrevious dan mcBlendingLow

Pilih kalimat untuk pemrograman dengan CPU PLC yang benar dari pilihan berikut. (Anda dapat memilih beberapa jawaban.)



Library FB harus didaftarkan ke GX Works3 agar bisa menggunakan Motion control FB untuk modul Gerak dalam CPU PLC.



Tempatkan motion control FB di editor program dari pohon proyek GX Works3.



Tidak ada parameter yang harus diatur untuk modul Gerak.

Pilih jawaban yang sesuai untuk mengisi bagian yang kosong.

- Mulai (Q1) untuk mengatur data yang akan dicatat.
- Tulis data pencatatan ke (Q2) untuk melakukan pencatatan.
- (Q3) digunakan untuk membaca data pencatatan dan memeriksa bentuk gelombang.

Q1

Pilih jawaban yang sesuai.

**Q2**

Pilih jawaban yang sesuai.

**Q3**

Pilih jawaban yang sesuai.



Q1: 1: CPU module logging configuration tool
2: Alat pengaturan pencatatan sistem gerak

Q2: 1: CPU
2: Modul Gerak
3: Penguat servo

Q3: 1: MR Configurator2
2: GX LogViewer

Pilih deskripsi label publik yang benar. (Anda dapat memilih beberapa jawaban.)

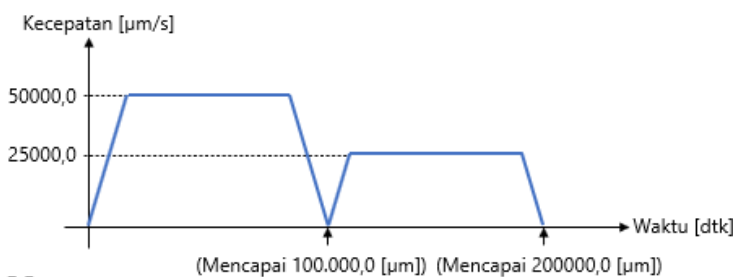
- ☒ Label publik adalah label bersama yang dapat digunakan di modul Gerak dan CPU PLC.
- ☐ Label publik didaftarkan dari label global CPU PLC.
- ☒ Ketika label global diatur ke label publik, pilih apakah label dibaca atau ditulis dari/ke CPU PLC.

FB1 dijalankan terlebih dahulu, lalu FB2 dijalankan setelahnya.

Ketika posisi target dan kecepatan target FB1 dan FB2 seperti yang ditunjukkan pada tabel berikut, pilih mode buffer yang dilakukan selanjutnya.

	Posisi target	Kecepatan target
FB1	100000,0 [μm]	50000,0 [μm/s]
FB2	200000,0 [μm]	25000,0 [μm/s]

Q1



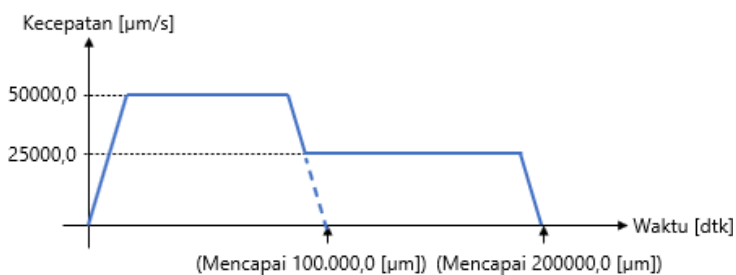
Q1

2: mcBuffered

Q2

3: mcBlendingNext dan mcBlendingLow

Q2



Q1: 1: mcAborting
2: mcBuffered
3: mcBlendingNext
4: mcBlendingPrevious

Q2: 1: mcBlendingNext dan mcBlendingHigh
2: mcBlendingPrevious dan mcBlendingHigh
3: mcBlendingNext dan mcBlendingLow
4: mcBlendingPrevious dan mcBlendingLow

Pilih kalimat untuk pemrograman dengan CPU PLC yang benar dari pilihan berikut. (Anda dapat memilih beberapa jawaban.)



Library FB harus didaftarkan ke GX Works3 agar bisa menggunakan Motion control FB untuk modul Gerak dalam CPU PLC.



Tempatkan motion control FB di editor program dari pohon proyek GX Works3.



Tidak ada parameter yang harus diatur untuk modul Gerak.

Pilih jawaban yang sesuai untuk mengisi bagian yang kosong.

- Mulai (Q1) untuk mengatur data yang akan dicatat.
- Tulis data pencatatan ke (Q2) untuk melakukan pencatatan.
- (Q3) digunakan untuk membaca data pencatatan dan memeriksa bentuk gelombang.

Q1

2: Alat pengaturan pencatatan sistem gerak

**Q2**

2: Modul Gerak

**Q3**

2: GX LogViewer



Q1: 1: CPU module logging configuration tool
2: Alat pengaturan pencatatan sistem gerak

Q2: 1: CPU
2: Modul Gerak
3: Penguat servo

Q3: 1: MR Configurator2
2: GX LogViewer

Anda telah menyelesaikan Tes Akhir.
Hasil Anda adalah sebagai berikut.

		1	2	3	4	5	6	7	8	9	10
	Tes Akhir 1	✓									
	Tes Akhir 2	✓	✓								
	Tes Akhir 3	✓									
	Tes Akhir 4	✓	✓	✓							

Jumlah total pertanyaan: 7
Jawaban yang benar: 7
Persentase: 100 %

Hapus

Anda telah menyelesaikan Kursus "MELSEC iQ-R Series Motion Module Basics (Kontrol Pemosisian RD78G(H))" .

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 kapanpun Anda mau.

Tinjau

Tutup