

PLC GX Works2 Dasar

Kursus (e-Pembelajaran) ini dirancang bagi mereka yang memakai perangkat lunak GX Works2 untuk pertama kalinya untuk membuat program sekuens.

Pendahuluan Tujuan Kursus

Kursus ini memberikan pengetahuan dasar tentang menggunakan perangkat lunak GX Works2 untuk pemrograman, pencarian kesalahan program (debugging), dan memeriksa operasi pengontrol yang dapat diprogram (PLC). Kursus ini dimaksudkan bagi mereka yang membuat program sekuens untuk pengontrol MELSEC seri Q, MELSEC seri L, dan MELSEC seri F.

Pendahuluan Struktur Kursus

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

Bab 1 - Metode Sistem Kontrol PLC

Di sini diperkenalkan bahasa pemrograman dan perangkat lunak yang dipakai untuk pemrograman.

Bab 2 - Desain Program

Anda akan mempelajari cara merancang program berdasarkan item kontrol dan konfigurasi perangkat keras.

Bab 3 - Pemrograman

Anda akan mempelajari cara memprogram memakai perangkat lunak khusus GX Works2.

Bab 4 - Pencarian Kesalahan Program (Debugging)

Anda akan mempelajari cara menulis program sekuens ke modul CPU dan mencari kesalahannya.

Bab 5 - Tes Akhir

Nilai lulus: 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.

Pendahuluan **Perhatian Selama Penggunaan**

Petunjuk keselamatan

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

Petunjuk keselamatan dalam kursus ini

- Layar yang ditampilkan pada versi perangkat lunak yang Anda gunakan mungkin berbeda dengan yang ada di dalam kursus ini.

Bab 1 Metode Kontrol Sistem PLC

Kursus ini ditujukan bagi orang yang bekerja dengan perangkat lunak keteknikan. Di dalamnya membahas beberapa konsep dasar mengenai pengelolaan sistem MELSEC seri Q, L, dan F.

GX Works 2 (GXW2) memakai bahasa pemrograman standar Internasional termasuk bahasa Sequential Function Chart (SFC), Instruction List (IL)*1, Ladder Logic, Function Block Diagram (FBD)*2 dan Structured Text (ST).

Program dikembangkan memakai komputer pribadi yang menjalankan "perangkat lunak keteknikan," GX Works2, dan biasanya ditulis ke CPU PLC lewat kabel USB, Ethernet*3, atau Serial. Modul CPU dapat diprogram ulang sesering yang diperlukan untuk beradaptasi dengan perubahan yang diperlukan dalam kontrol yang diinginkan.

*1 Rencana GX works2 ke depan.

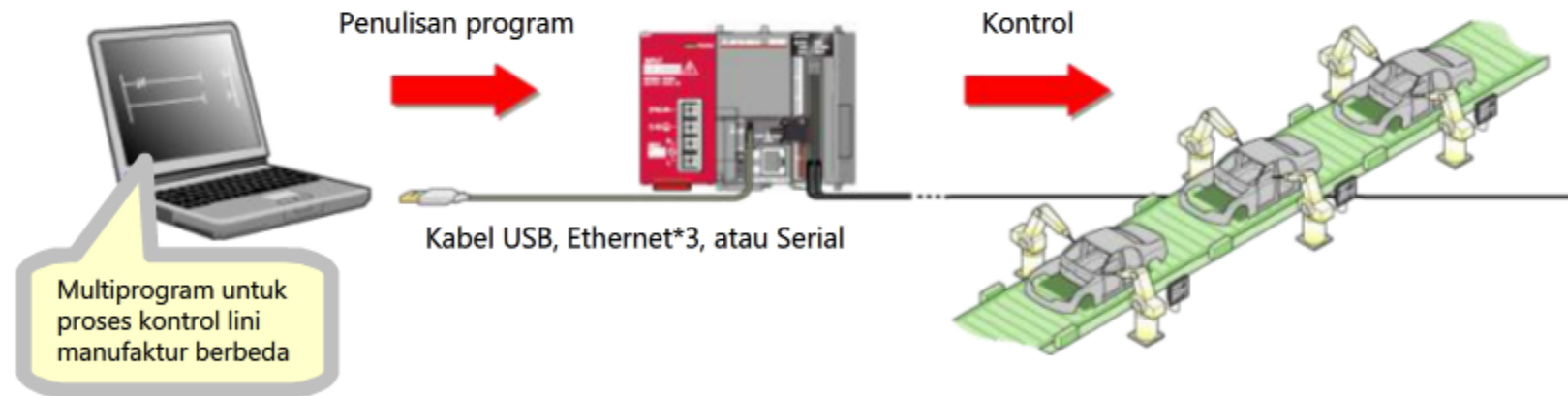
*2 Saat ini dinamai Structured Ladder di GX works2, kepatuhan IEC telah direncanakan.

*3 Ethernet adalah merek dagang dari Xerox Corp.

Komputer pribadi
(Perangkat lunak pemrograman)

Sistem PLC

Lini manufaktur mobil



Dalam kursus ini, logika ladder (salah satu bahasa pemrograman PLC terpopuler) digunakan dalam contoh program. Meskipun contoh memakai PLC Seri L, isi kursus ini juga berlaku untuk sistem Seri Q.

Metode kontrol dasar ini juga sama untuk MELSEC seri F, namun dengan perbedaan pada beberapa operasi dan fungsi.

1.1**Prosedur Konstruksi Sistem PLC**

Kursus e-pembelajaran ini mencakup langkah-langkah desain perangkat lunak (ditampilkan dalam warna hijau) yang diperlukan untuk menerapkan sistem pengontrol yang dapat diprogram.

Desain perangkat keras

(1) Desain sistem Kursus Dasar MELSEC-Q/MELSEC-L



(2) Pemilihan produk Kursus Dasar MELSEC-Q/MELSEC-L



(3) Persiapan pendahuluan Kursus Dasar MELSEC-Q/MELSEC-L



(4) Pemasangan dan pengkabelan .. Kursus Dasar MELSEC-Q/MELSEC-L



(5) Pemeriksaan pengkabelan ... Kursus Dasar MELSEC-Q/MELSEC-L

**Desain perangkat lunak**

(6) Desain program Bab 2



(7) Pemrograman Bab 3



(8) Pencarian Kesalahan Program (Debugging) Bab 4



(9) Operasi

**Cakupan
kursus ini**

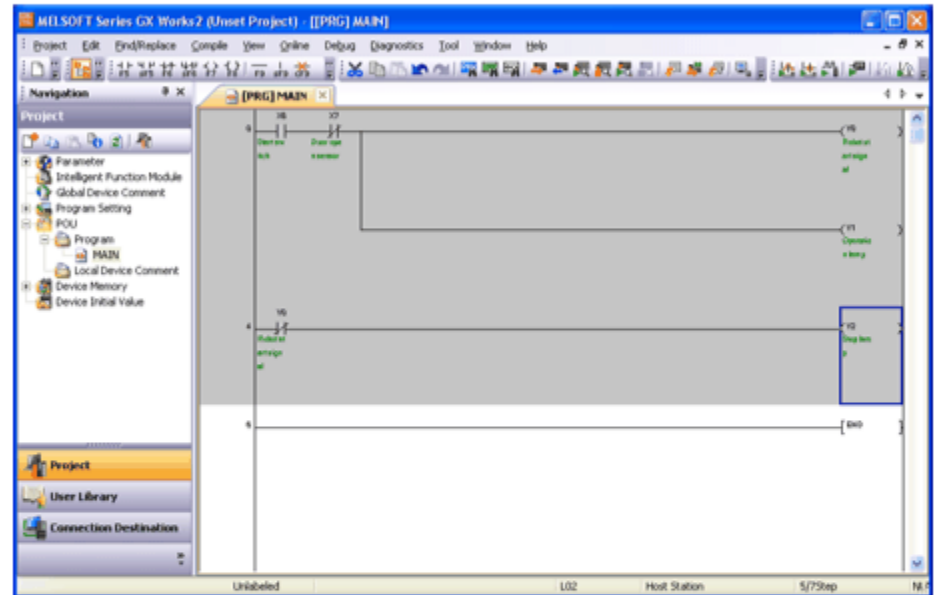
1.2

Persyaratan untuk Pemrograman

Kursus ini memfokuskan pada cara menggunakan perangkat lunak keteknikan pengontrol yang dapat diprogram GX Works2 untuk mengembangkan program sistem contoh.

Beberapa fungsi utama GX Works2 tercantum di bawah ini.

- Pengelolaan memori dan file
- Pengembangan program pengontrol yang dapat diprogram
- Pengelolaan dokumentasi program (komentar, dll)
- Pembacaan, penulisan data (terutama program) dari/ke modul CPU
- Verifikasi operasi program
 - Simulasi perangkat lunak dari perangkat keras PLC
 - Pemaksaan hidup atau mati I/O
 - Pemantauan I/O dan status memori alamat
- Pelaksanaan tugas-tugas pemeliharaan dan pemecahan masalah



1.3

Konfigurasi Layar GX Works2

Konfigurasi layar GX Works2 ditampilkan di bawah ini.

Tempatkan kursor mouse pada bingkai merah untuk menampilkan masing-masing fungsi.

The screenshot displays the MELSOFT Series GX Works2 software interface. The window title is "MELSOFT Series GX Works2 ...s\EN93632\My Documents\le_Learning\Robot_Control". The menu bar includes Project, Edit, Find/Replace, Compile, View, Online, Debug, Diagnostics, Tool, Window, and Help. The toolbar contains various icons for file operations and execution. The left sidebar shows the Project tree with folders for Parameter, Intelligent Function Module, Global Device Comment, Program Setting, POU, Program (MAIN, SUB), Local Device Comment, Device Memory, and Device Initial Value. The main workspace shows a ladder logic program with two windows: [PRG] MAIN and [PRG] SUB. The [PRG] MAIN window is highlighted with a blue border and contains a red rectangular selection box around the navigation pane and the main program area. The [PRG] SUB window is also visible, showing a similar ladder logic diagram. The status bar at the bottom indicates "Unlabeled", "L02", "Host Station", "5/7Step", and "M.S".

1.4 Prosedur Pembuatan Program Sekuens

Buatlah program Sekuens sesuai dengan prosedur berikut.

(1) Desain Program Bab 2



(2) Membuat Proyek Bagian 3.1



(3) Pemrograman Bagian 3.2



(4) Mengonversikan Program Bagian 3.4



(5) Menulis ke Modul CPU Bagian 4.2



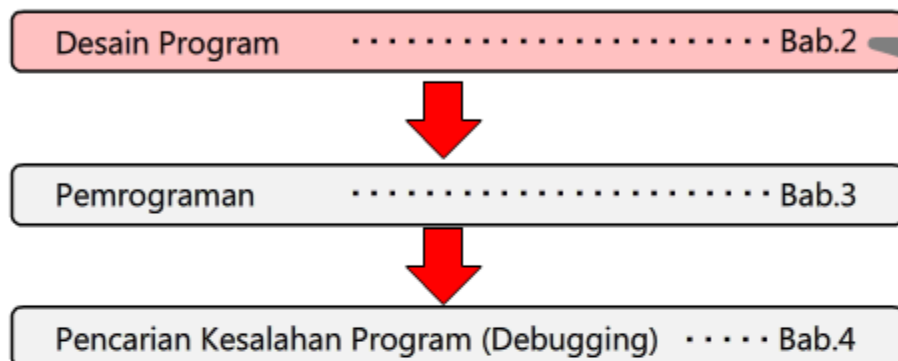
(6) Mengatur ulang dan Mengeksekusi Modul CPU ... Bagian 4.3, 4.4



(7) Pencarian Kesalahan Program (Debugging) ... Bagian 4.5

Bab 2 Membuat Data Layer

Dalam Bab 2, Anda akan mempelajari cara mendesain program, termasuk mendefinisikan isi kontrol dan mengonversikannya ke dalam sebuah program.

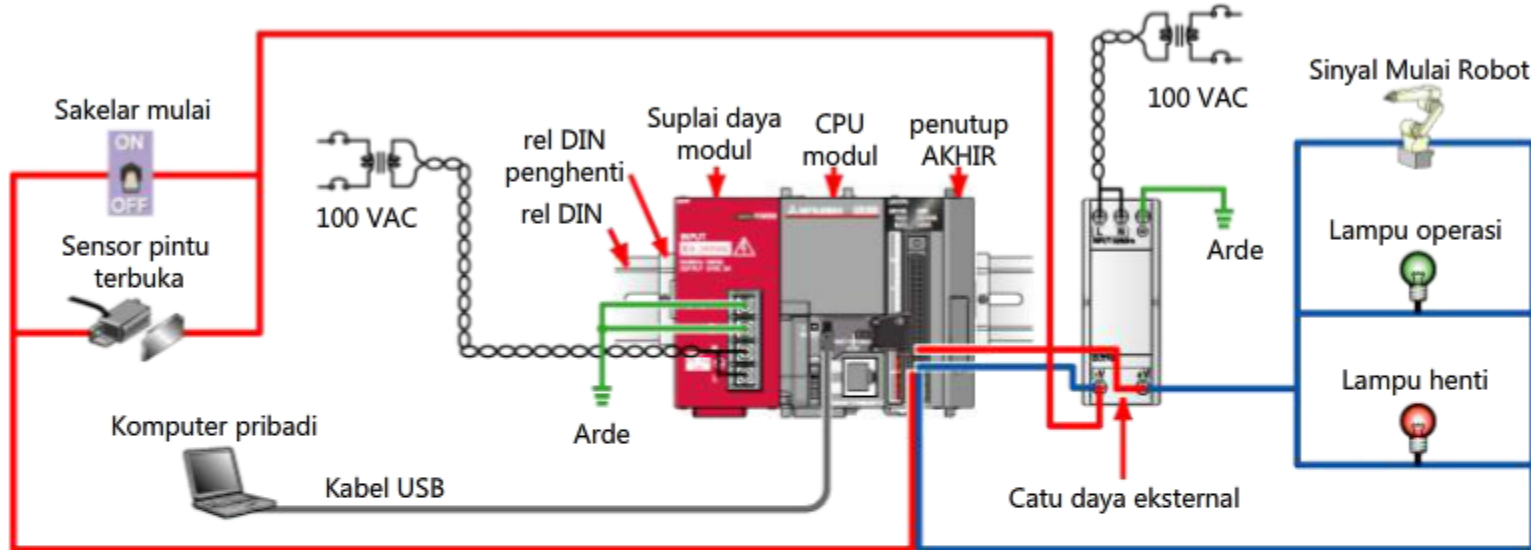


Langkah-langkah pembelajaran di Bab 2

- 2.1 Konfigurasi Perangkat Keras dari Contoh Sistem yang Digunakan untuk Belajar
- 2.2 Mendefinisikan Item Kontrol
- 2.3 Membuat Tabel Korespondensi I/O Perangkat dan Nomor Perangkat
- 2.4 Merancang Program

2.1 Konfigurasi Perangkat Keras dari Sistem Contoh yang Digunakan untuk Belajar

Dalam kursus ini, Anda akan membangun satu sistem PLC (selanjutnya disebut "sistem contoh"), yang memulai robot sesuai prosedur. Diagram konfigurasi perangkat keras dari sistem contoh ditunjukkan di bawah ini bersama daftar komponen perangkat keras.



Item	Komponen	Model	Keterangan
Sistem PLC	Modul catu daya	L61P	Menyuplai daya ke modul termasuk modul CPU dan modul I/O.
	Modul CPU	L02CPU	Mengontrol sistem PLC.
	Penutup AKHIR	L6EC	Terpasang ke sisi kanan blok sistem.
	Kabel USB	MR-J3USBCBL3M	Menghubungkan komputer pribadi, tempat GX Works2 dipasang, ke modul CPU.
	Komputer pribadi	—	Berjalan dengan GX Works2 terpasang.
Catu daya eksternal	—	—	Menyuplai daya ke perangkat I/O eksternal.
Peralatan I/O eksternal	Sakelar	—	Disetel ke ON untuk memulai kontrol.
	Sensor	—	Mendeteksi apakah pintu terbuka atau tertutup.
	Robot	—	Beroperasi sesuai dengan sinyal kontrol.
	Dua lampu	—	Menyala sesuai dengan status operasi.

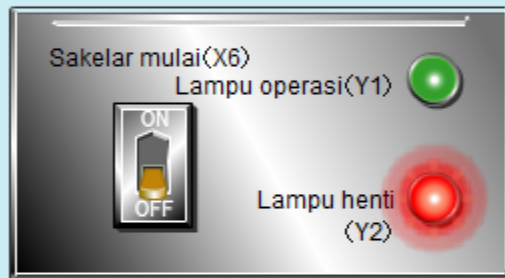
Langkah pertama dalam merancang sebuah program adalah mengidentifikasi perangkat yang akan dikontrol dan perangkat I/O yang diperlukan untuk kontrol yang diinginkan. Dalam sistem contoh ini, kontrol memulai dan menghentikan operasi pada robot dilakukan. Robot akan dihalangi dari memulai jika pintu ke pagar pengaman terbuka, dan berhenti jika pintu dibuka selama operasi.

Lihat animasi di bawah ini untuk lebih memahami cara sistem contoh akan beroperasi.

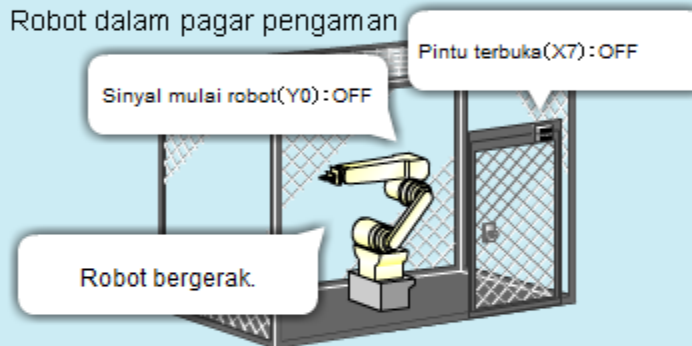
Operasi sistem contoh

🔴 Klik di dalam lingkaran merah

Panel kontrol robot



Robot dalam pagar pengaman



Ketika Anda menetapkan sakelar mulai (X6) ditetapkan ke OFF, sinyal mulai robot (Y0) mati untuk menghentikan robot beroperasi. Pada saat yang bersamaan, lampu operasi (Y1) pada panel kontrol mati, dan lampu henti (Y2) hidup.

Ulangi

◀ Sebelumnya

2.3

Membuat Tabel Keterkaitan Perangkat I/O dan Nomor Perangkat

Ada baiknya untuk membuat tabel yang menyertakan semua perangkat dan catatan I/O yang digunakan di PLC dan informasi yang terkait untuk setiap program yang dibuat. Tabel ini mengurangi kemungkinan kesalahan yang akan terjadi proses desain dan pemrograman serta berfungsi untuk meningkatkan efisiensi pemrograman. Jika tabel keterkaitan untuk sistem ini sudah ada, misalnya tabel yang dibuat oleh orang yang mengonfigurasi perangkat keras, manfaatkan tabel tersebut.

Tabel berikut ini adalah tabel keterkaitan untuk sistem contoh yang digunakan di dalam kursus ini

nama perangkat I/O	No. Perangkat	Jenis I/O	Jenis perangkat	Keterangan
Start switch (Sakelar mulai)	X6	Masukan	Bit	Sakelar ini memulai atau menghentikan robot beroperasi.
Door open sensor (Sensor pintu terbuka)	X7	Masukan	Bit	Sensor ini memeriksa apakah pintu pagar pengaman dari robot terbuka. Ketika pintu membuka, sensor menjadi aktif. Ketika pintu menutup, sensor menjadi tidak aktif.
Robot start signal (Sinyal Mulai Robot)	Y0	Keluaran	Bit	Bila sinyal ini menyala, robot mulai beroperasi.
Operation lamp (Lampu operasi)	Y1	Keluaran	Bit	Lampu ini menyala selama robot beroperasi.
Stop lamp (Lampu henti)	Y2	Keluaran	Bit	Lampu ini menyala selama robot berhenti.

* Jika data kata digunakan, nilai awal, rentang pengaturan (batas atas dan bawah), jenis data (tertanda, riil, dll), dan komentar harus disertakan di dalam tabel.

Informasi ini akan bermanfaat untuk merancang dan memodifikasi program.

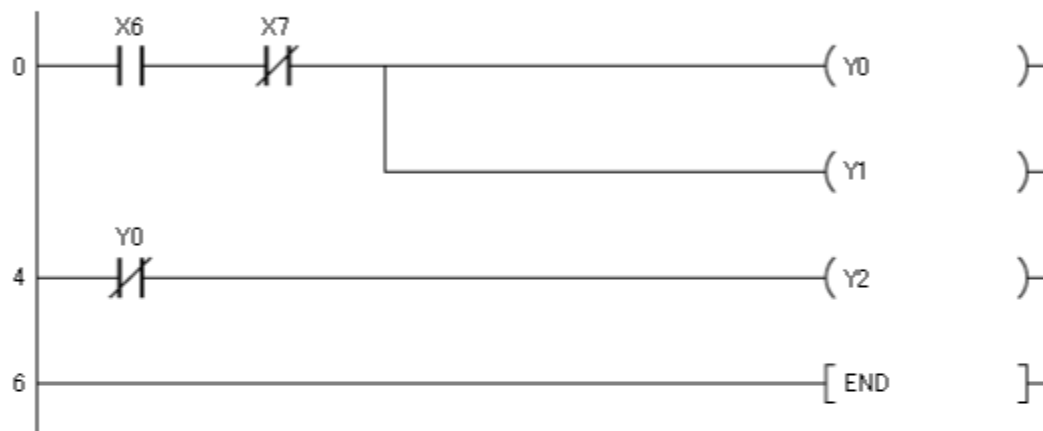
2.4

Merancang Program



Merancang program menggunakan bahasa logika Ladder berdasarkan item kontrol dan tabel keterkaitan I/O. Program ladder dan tabel keterkaitan I/O yang dirancang untuk sistem contoh ditunjukkan di bawah ini.

Program ladder

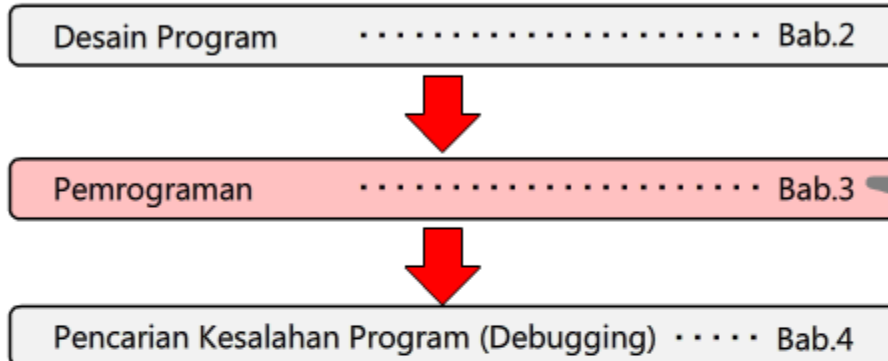


Tabel keterkaitan I/O

nama perangkat I/O	Jenis	No. Perangkat
Start switch (Sakelar mulai)	Masukan	X6
Door open sensor (Sensor pintu terbuka)	Masukan	X7
Robot start signal (Sinyal Mulai Robot)	Keluaran	Y0
Operation lamp (Lampu operasi)	Keluaran	Y1
Stop lamp (Lampu henti)	Keluaran	Y2

Bab 3 Pemrograman

Dalam Bab 3, Anda akan mempelajari cara memprogram program yang didesain memakai GX Works2.



Langkah-langkah pembelajaran di Bab 3

- 3.1 Membuat Proyek
- 3.2 Membuat Program
- 3.3 Membuat Program Mudah Dipahami
- 3.4 Mengubah Program ke Bentuk yang Dapat Dieksekusi
- 3.5 Menyimpan Proyek

Langkah pertama untuk menulis program adalah membuat proyek. Proyek merupakan sekumpulan data GX Works2 yang digunakan untuk mengelola program. Berikut adalah tabel yang berisi daftar komponen utama sebuah proyek.

Jenis data	Keterangan
Program	Kode sumber dan kode kompilasi untuk operasi sekuens CPU.
Komentar	Jenis dokumentasi yang ditampilkan di dalam program. Lihat Bagian 3.3 "Membuat Program Jadi Mudah Dipahami" untuk mengetahui detailnya.
Parameter	Berisi sebagian besar atau semua pengaturan dan konfigurasi informasi untuk sistem.
Konfigurasi transfer	Informasi rute sambungan yang diperlukan untuk menjalin komunikasi di antara sistem yang menjalankan GX Works2 dan modul CPU.

Program ladder

GX Works2 memungkinkan Anda memilih dua jenis proyek berikut. Program contoh dalam kursus ini menggunakan jenis "**simple project (proyek sederhana)**".

Jenis proyek	Keterangan
Simple project (Proyek sederhana)	Jenis proyek ini juga kompatibel dengan proyek versi sebelumnya dari Developer GX. Proyek sederhana dapat dikonversikan menjadi proyek Terstruktur sesudahnya, namun tidak sebaliknya.
Structured project (Proyek terstruktur)	Proyek ini dapat memakai bahasa pemrograman tambahan yang disebut Ladder Structured. Program juga dapat dipisahkan menjadi banyak bagian-bagian kecil dan potongan-potongan kode yang sering digunakan dapat dengan mudah dimodularkan dan dipakai lagi menggunakan perpustakaan pengguna. Dengan cara serupa, Label dapat dimodularkan agar bisa dipakai kembali dengan mudah. Hal tersebut bisa meningkatkan efisiensi dan pencarian kesalahan (debugging) pemrograman, terutama pada proyek yang berukuran sangat besar.

Label

Label adalah nama yang dibuat pengguna yang menjadi alias untuk alamat perangkat. Label bisa dipakai secara global, lokal, maupun di seluruh sistem saat diimplementasikan bersama dengan MELSOFT Navigator. Proyek sederhana bisa dibuat dengan atau tanpa kemampuan memakai label. Untuk proyek contoh, label tidak akan dipakai.

Untuk mulai membuat proyek contoh, buat pengaturan berikut. Sebelum membuat proyek, seri pengontrol yang dapat diprogram dan nama model, serta jenis proyek yang akan digunakan, harus diketahui.

Item	Keterangan
Jenis proyek	Jenis proyek menentukan fitur yang tersedia saat menulis program. Untuk contoh ini, pilih "proyek sederhana."
Gunakan label	Jika kemampuan menulis program menggunakan label diperlukan, periksa item ini. Program contoh tidak menggunakan label. Karena itu, biarkan kotak ini dicentang.
Seri PLC	Seri PLC menentukan model yang tersedia untuk dipilih dalam daftar tarik turun jenis PLC. Untuk contoh ini, pilih "LCPU."
Jenis PLC	Jenis PLC menentukan bagaimana pengompilasi mengonversi program pengguna ke dalam kode mesin. Pilih model PLC yang akan diprogram, dalam hal ini, "L02."
Bahasa pemrograman	Bahasa pemrograman menentukan jenis program dari program pertama yang dibuat otomatis (UTAMA). Program tambahan yang menggunakan bahasa berbeda bisa ditambahkan sesudahnya. Untuk contoh ini, pilih "Ladder."

Tinjau halaman berikutnya, yaitu simulasi proses membuat proyek baru.

3.1

Membuat Proyek

MELSOFT Series GX Works2

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



Navigation

Project



Project

User Library

Connection Destination

New Project

Project Type:

Simple Project

OK

Cancel

 Use Label

PLC Series:

LCPU

PLC Type:

L02

Language:

Ladder

Proyek baru kini telah dibuat.

Klik  untuk melanjutkan.

3.2

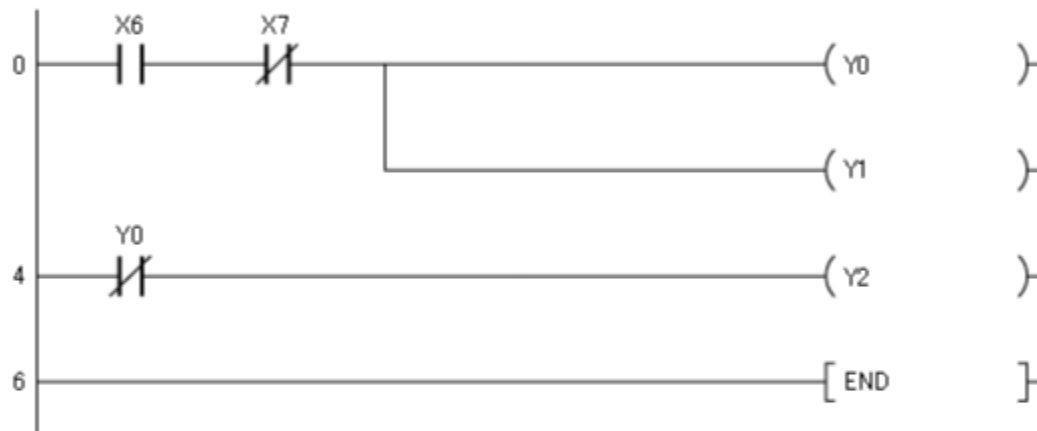
Membuat Program

Setelah membuat proyek, marilah kita membuat program.

Buatlah program berikut dan pelajari operasi dasarnya (masukan petunjuk, perubahan/hapus, salin & tempel, dan input/hapus lini aturan).

Program yang dirancang untuk sistem contoh pada Bab 2 ditampilkan di bawah ini.

Program untuk sistem contoh



Pada halaman berikutnya, coba buat program ini memakai jendela simulasi.

3.2

Membuat Program

MELSOFT Series GX Works2 (Unset Project) - [[PRG] MAIN]


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

Navigation [PRG] MAIN

Project

- Parameter
- Intelligent Function Module
- Global Device Comment
- Program Setting
- POU
 - Program
 - MAIN
 - Local Device Comment
- Device Memory
- Device Initial Value

Unlabeled L02 Host Station 0/15Step

Kini program sirkuit Ladder telah selesai.
Klik  untuk melanjutkan.

3.3

Membuat Program Mudah Dipahami

Pada kondisi saat ini, representasi visual dari program ini hanya berisi perangkat, petunjuk, garis, dan nomor-nomor langkah. Saat melihat program yang kompleks, menentukan apa yang sedang dilakukan program bisa sulit.

- Menemukan kesalahan pemrograman seperti nomor perangkat atau petunjuk yang salah menjadi sulit.
- Secara umum, analisis operasional, pencarian kesalahan program (debugging), dan ekspansi program sulit dilakukan.
- Jika pengembang program aslinya tidak lagi bisa memelihara program, orang lain yang bertugas mempelajari cara program beroperasi bisa membingungkan dan mungkin mustahil.

Penanggulangan

Di dalam program, sertakan **dokumentasi** agar siapa saja bisa memahami cara program bekerja dengan cepat.

Semua pemrogram harus membiasakan diri menambah komentar mendetail pada program mereka agar ia dan orang lain bisa lebih memahami program.



GX Works2 memungkinkan tiga jenis komentar untuk dipakai.

Untuk mendapatkan detail lebih lanjut, lihat petunjuk Proyek Sederhana GX Works2.

Jenis komentar	Cakupan komentar
Device comment (Komentar perangkat)	Masukan karakter yang akan ditampilkan di bawah perangkat yang dipilih (I/O atau alamat memori lain) hingga 32 karakter.
Statement (Pernyataan)	Masukan hingga 64 karakter per pernyataan yang akan ditambahkan pada bagian atas blok ladder (di atas nomor langkah). Setiap blok ladder dapat memiliki beberapa pernyataan.
Note (Perhatikan)	Masukan karakter yang akan ditampilkan di atas coil yang dipilih atau petunjuk aplikasi hingga 32 karakter.

Halaman berikutnya berisi simulasi proses penambahan komentar perangkat pada program contoh.

3.3 Membuat Program Mudah Dipahami

MELSOFT Series GX Works2 (Unset Project) - [[PRG] MAIN]

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

Navigation [PRG] MAIN

Project

- Parameter
- Intelligent Function Module
- Global Device Comment
- Program Setting
- POU
 - Program
 - MAIN
 - Local Device Comment
- Device Memory
- Device Initial Value

0 X6 Start switch X7 Door open sensor

Y0 Robot start signal


Y1 Operation lamp

4 Y0 Robot start signal

Y2 Stop lamp

6 [END]

Unlabeled L02 Host Station 5/75Step

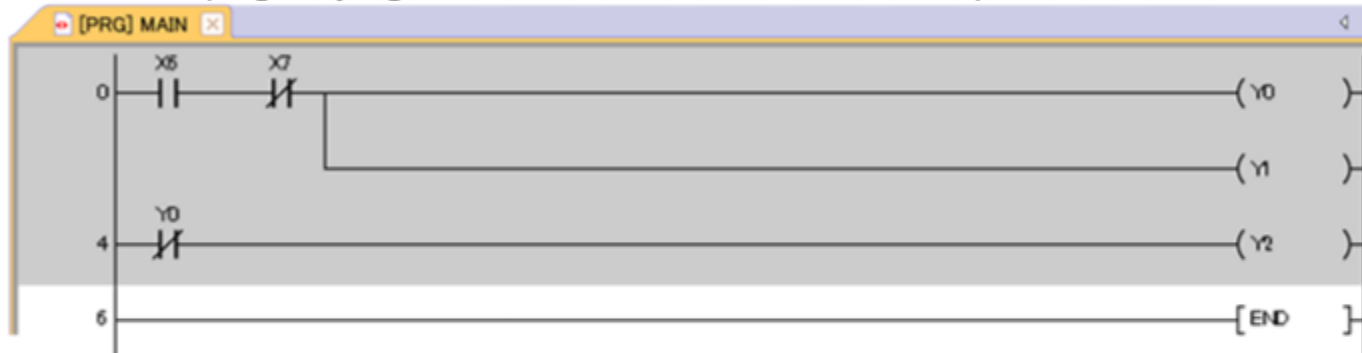
Masukan komentar perangkat telah selesai.
Klik  untuk melanjutkan.

3.4

Mengubah Program ke Bentuk yang Dapat Dieksekusi

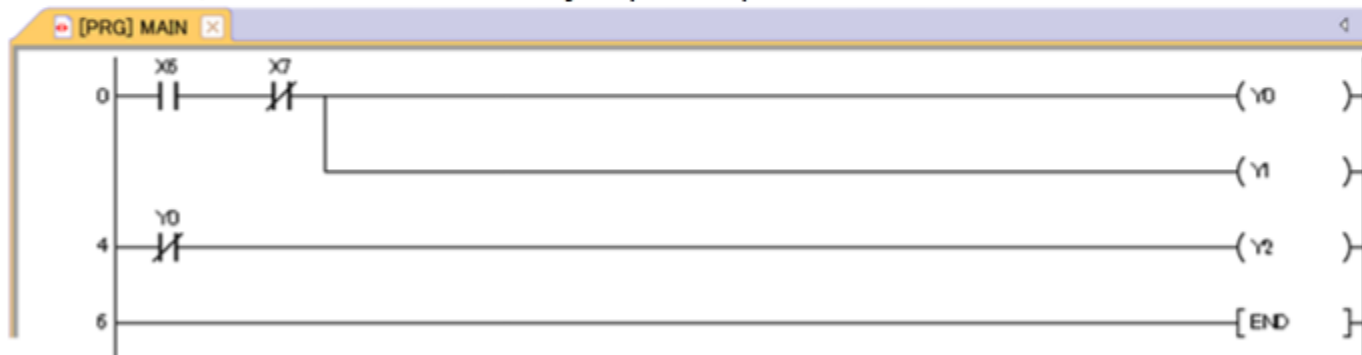
Setelah menyelesaikan program, Anda harus mengonversikannya ke dalam bentuk yang dapat dieksekusi di dalam modul CPU. Program yang belum dikonversikan tidak bisa dieksekusi atau disimpan.

Warna latar dari program yang belum dikonversikan adalah abu-abu seperti di bawah ini.



Konversi

Setelah konversi, warna latar berubah menjadi putih seperti di bawah ini.



Pada halaman berikutnya, coba konversikan program memakai jendela simulasi.

3.4

Mengubah Program ke Bentuk yang Dapat Dieksekusi

MELSOFT Series GX Works2 (Unset Project) - [[PRG] MAIN]

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

Navigation [PRG] MAIN

Project

- Parameter
- Intelligent Function Module
- Global Device Comment
- Program Setting
- POU
 - Program
 - MAIN
 - Local Device Comment
- Device Memory
- Device Initial Value

Project

User Library


Connection Destination

Unlabeled L02 Host Station 5/75Step

Diagram showing a ladder logic program with the following components:

- Step 0: Start switch (X6) and Door open sensor (X7) in series, leading to Robot start signal (Y0).
- Step 4: Robot start signal (Y0) as a normally closed contact, leading to Robot stop lamp (Y2).
- Step 6: END instruction.

Saat program telah dikonversikan. warna latar berubah dari abu-abu menjadi putih.

Program telah dikonversikan.
Klik  untuk melanjutkan.

3.5

Menyimpan Proyek

Setelah konversi program selesai, simpan proyek bersama programnya. Jika GX Works2 berhenti tanpa menyimpan proyek, program yang terkait akan dibuang, oleh karena itu Anda harus menyimpan proyek secara berkala.

Saat menyimpan proyek baru, tetapkan jenis informasi proyek berikut ini. (Ini tidak perlu untuk penyimpanan dengan menimpa)

Anda harus menyertakan informasi yang memudahkan orang lain dalam memahami isi kontrol program, nama sistem, dll.

Item	Wajib diisi	Keterangan
Simpan jalur destinasi	✓	Tetapkan folder tempat ruang kerja akan ditempatkan.
Daftar ruang kerja/proyek		Apabila satu atau beberapa ruang kerja sudah ada dalam folder yang ditetapkan pada "Simpan jalur destinasi," maka ruang kerja yang sudah ada akan dicantumkan.
Nama ruang kerja	✓	Berikan nama ruang kerja hingga 128 karakter.
Nama proyek	✓	Berikan nama proyek hingga 128 karakter.
Judul		Berikan judul proyek hingga 128 karakter. Parameter ini berguna jika Anda ingin memberi nama yang panjang yang tidak mencukupi dalam "Nama project."

Ruang kerja adalah folder untuk mengelola beberapa proyek.

Di bawah ini adalah tampilan contoh penggunaan ruang kerja. (Proyek dikelola untuk setiap jenis kendaraan di lini manufaktur mobil.)

Nama ruang kerja	Nama proyek	Judul
Lini manufaktur mobil	Lini manufaktur jenis A	Program operasi normal untuk mengontrol lini manufaktur jenis A
	Lini manufaktur jenis B	Program operasi normal untuk mengontrol lini manufaktur jenis B
	Lini manufaktur jenis C	Program operasi normal untuk mengontrol lini manufaktur jenis C

Catatan:

- Jika proyek berisi program yang belum dikonversikan disimpan, hanya program yang belum dikonversikan yang akan dibuang. Sebelum menyimpan proyek, lakukan konversi program seperti yang Anda pelajari di Bagian 3.4.
- Tentukan jalur destinasi penyimpanan, nama ruang kerja, dan nama proyek agar jumlah karakter total tidak melebihi 150. Pada halaman berikutnya, cobalah menyimpan program ini memakai jendela simulasi.

3.5

Menyimpan Proyek

The screenshot displays the MELSOFT Series GX Works2 software interface. The main window shows a ladder logic program for a robot control system. The program consists of three steps:

- Step 0: A normally open contact labeled 'Start Switch' (X6) and a normally closed contact labeled 'Door open sensor' (X7) are connected to a coil for output 'Y0 Robot start signal'.
- Step 4: A normally open contact labeled 'Y0 Robot start signal' is connected to a coil for output 'Y2 Stop lamp'.
- Step 6: An 'END' instruction is shown.

A confirmation dialog box is displayed in the bottom right corner, containing the text: "Proyek telah disimpan. Klik [Next] untuk melanjutkan." (Project has been saved. Click [Next] to continue.)

The software interface includes a menu bar (Project, Edit, Find/Replace, Compile, View, Online, Debug, Diagnostics, Tool, Window, Help), a toolbar, and a navigation pane on the left. The navigation pane shows the project structure, including 'MAIN'.

At the bottom of the window, the status bar displays: Unlabeled, L02, Host Station, 6/75Step, and NL.

Bab 4 Pencarian Kesalahan Program (Debugging)

Pada Bab 4, Anda akan mempelajari cara menulis program sekuens ke modul CPU dan mencari kesalahannya.

Desain Program Bab.2



Pemrograman Bab.3



Pencarian Kesalahan Program (Debugging) Bab.4

Langkah-langkah pembelajaran di Bab4

- 4.1 Pencarian Kesalahan Program (Debugging)
 - 4.1.1 Melakukan Debugging Program tanpa Memakai Modul CPU
 - 4.1.2 Mengubah Status Perangkat I/O
 - 4.1.3 Memantau Status Perangkat
- 4.2 Menulis Program ke Modul CPU
- 4.3 Mengaktifkan Program Tertulis
- 4.4 Menjalankan Program
- 4.5 Mencari Kesalahan (Debugging) Program
- 4.6 Memeriksa Operasi Sistem PLC
- 4.7 Mengoperasikan Sistem PLC
- 4.8 Kesimpulan

4.1 Apakah Debugging (Pencarian Kesalahan Program) Itu?

Setelah program atau segmen program ditulis, kode itu harus diuji untuk memastikan bahwa program beroperasi sesuai harapan.

Cacat perangkat lunak (ketika kode yang ditulis tidak berjalan seperti yang diinginkan) disebut "bug", dan proses pencarian penyebab perilaku yang tidak diinginkan dan memperbaikinya dikenal sebagai "debugging."

Pengujian dan debugging adalah langkah yang penting dalam membuat program.

Khususnya dalam pengontrol yang dapat diprogram karena jika bug ada, bug itu bisa menyebabkan sistem terhenti, peralatan rusak, atau insiden lainnya.

Tabel berikut ini berisi beberapa fungsi GX Works2 yang dapat membantu proses debugging.

Nama fungsi	Keterangan
Simulator	Fungsi ini digunakan untuk mensimulasikan eksekusi program sekalipun tanpa modul CPU. Fungsi ini bisa digunakan untuk debugging dalam lingkungan yang tidak menyediakan modul CPU.
Monitor	Fungsi ini memungkinkan pemantauan status eksekusi dan status setiap perangkat selama eksekusi modul CPU. Beberapa fungsi monitor tersedia bergantung pada aplikasi, seperti pemantauan ladder, pemantauan perangkat terdaftar saja, dan pemantauan semua perangkat dalam satu kelompok.
Mengubah nilai saat ini	Fungsi ini dapat mengubah status perangkat dengan paksa (bit; ON ↔ OFF, kata: nilai saat ini) selama eksekusi modul CPU. Fungsi ini berguna untuk mengubah nilai perangkat kata saat ini atau status relai internal.
pendaftaran/pembatalan masukan keluaran paksa	Fungsi ini dapat mengubah status dengan paksa (ON ↔ OFF) dari perangkat I/O terdaftar selama eksekusi modul CPU. Untuk melakukan debugging atau verifikasi operasi yang hanya menggunakan modul CPU, fungsi ini dapat dipakai sebagai pengganti sakelar.

Semua fungsi ini dijelaskan secara lebih mendetail dengan mengaitkan proses debugging di seluruh bagian akhir bab ini.

Catatan tentang debugging

Jangan lakukan tugas-tugas debugging saat pengontrol yang dapat diprogram tersambung ke perangkat I/O fisik.

Bugs pada program ini, perangkat I/O yang dipaksa, atau perubahan nilai kata bisa mengakibatkan kerusakan atau hal lain yang lebih buruk pada peralatan eksternal.

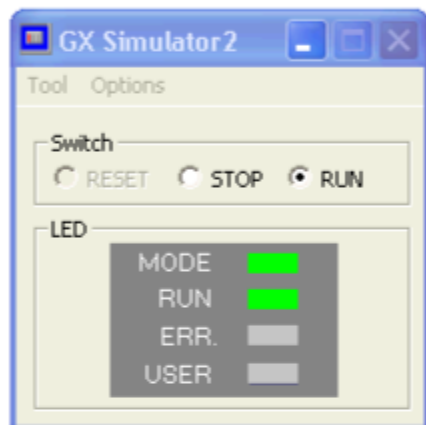
Jika sistem PLC yang terputus tidak tersedia, gunakan fungsi simulator.

4.1.1

Melakukan Debugging Program tanpa Memakai Modul CPU

Jika modul CPU tidak tersedia untuk debugging, gunakan **fungsi simulator**.

Program dapat berjalan pada modul CPU virtual yang disediakan oleh perangkat lunak tanpa harus menggunakan modul CPU sebenarnya.



 Hidup

 Mati

Item	Kondisi	Keterangan
Switch (Sakelar)	RUN	Menjalankan modul CPU virtual.
	STOP	Menghentikan modul CPU virtual.
	RESET	Mengatur ulang modul CPU virtual. (Hanya aktif dalam kondisi STOP)
LED	MODE	Menunjukkan status MODE dari CPU virtual.
	RUN	Menunjukkan status berjalan dari CPU virtual. •Hidup: Kondisi RUN •Mati: Kondisi STOP
	ERR	Menunjukkan status kesalahan dari CPU virtual. Jika ada kesalahan, LED akan menyala atau berkedip.
	USER	Menunjukkan apakah terjadi kesalahan pengguna pada CPU virtual. Menyala atau berkedip saat kesalahan terjadi.

Catatan tentang menggunakan fungsi simulator

- Debugging menggunakan fungsi simulator tidak menjamin bahwa sekuens program akan beroperasi dengan benar setelah debugging dilakukan.
- Fungsi simulator mengeksekusi masukan/keluaran data dengan modul I/O dengan menggunakan memori simulasi. Fungsi tidak mendukung beberapa petunjuk, fungsi, dan memori perangkat. Oleh karenanya, hasil operasi dengan fungsi simulator mungkin berbeda dengan hasil modul CPU sebenarnya.

Pada halaman berikutnya, cobalah menggunakan fungsi simulator dengan jendela simulasi.

4.1.1 Melakukan Debugging Program tanpa Memakai Modul CPU

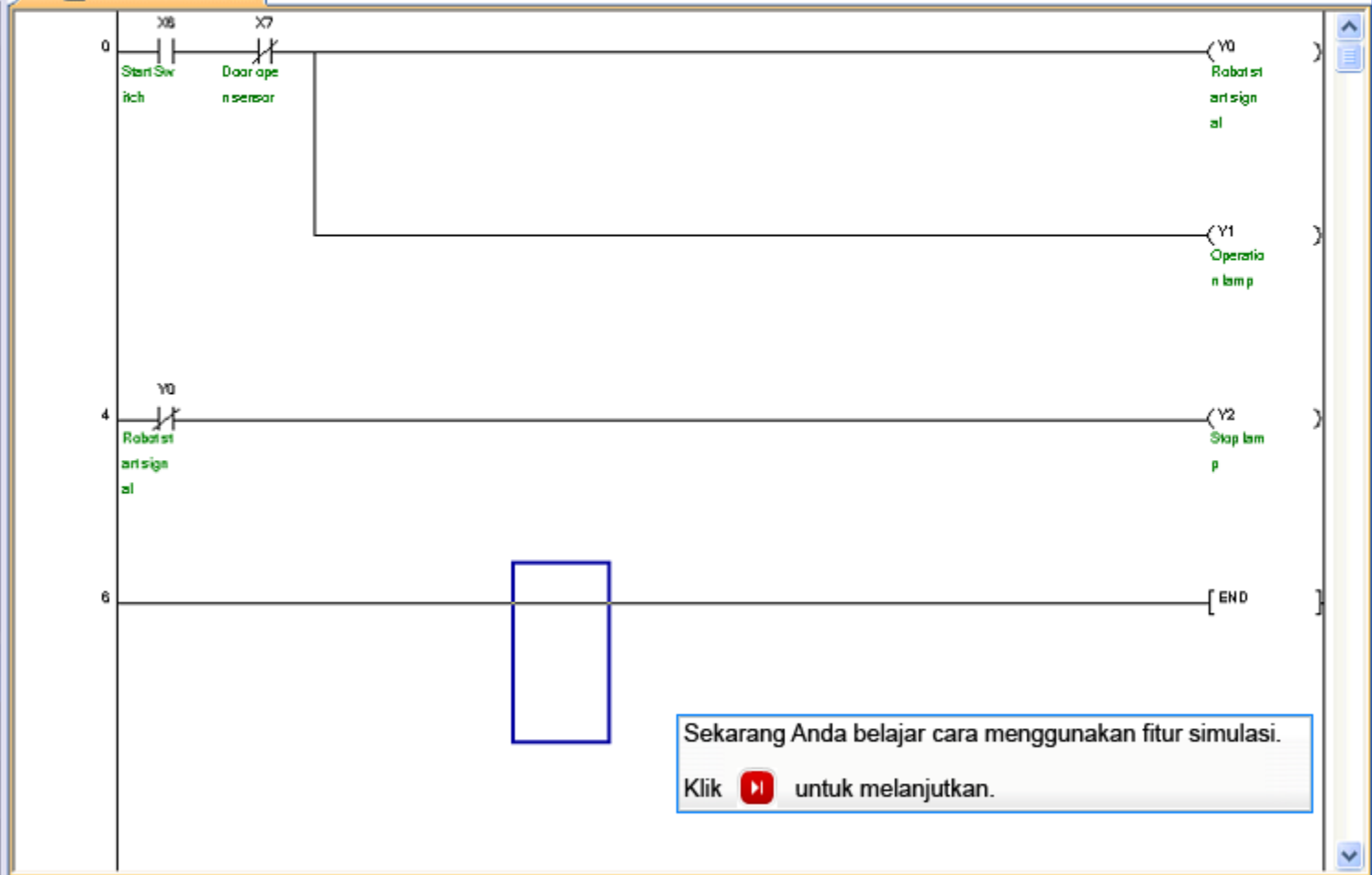
MELSOFT Series GX Works2 C:\SequenceProgram\Learning\Robot_Control - [[PRG] MAIN]


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



Navigation [PRG] MAIN

- Project
 - Parameter
 - Intelligent Function Module
 - Global Device Comment
 - Program Setting
 - POU
 - Program
 - MAIN
 - Local Device Comment
 - Device Memory
 - Device Initial Value



Sekarang Anda belajar cara menggunakan fitur simulasi.
 Klik  untuk melanjutkan.

4.1.2

Mengubah Status Perangkat I/O

Saat melakukan debugging program sekuens dengan modul CPU yang perangkat I/O-nya tidak tersambung atau memakai fungsi simulator, gunakan fungsi **Forced Input Output Registration/Cancellation** untuk mengubah keadaan ON/OFF dari perangkat I/O.

Status perangkat I/O yang terdaftar dapat diubah secara paksa menjadi ON atau OFF dengan perangkat lunak.
(MELSEC seri Q dan MELSEC seri L): Dari layar "Forced Input Output Registration/Cancellation"
(Pendaftaran/Pembatalan Input Output Paksa)

(MELSEC seri F): Dari layar "Modify Value" (Ubah Nilai)



Layar Forced Input Output Registration/Cancellation
(Pendaftaran/Pembatalan Input Output Paksa)
(MELSEC seri Q dan MELSEC seri L)



Layar Modify Value (Ubah Nilai)
(MELSEC seri F)

Untuk mengubah kondisi perangkat lain

Untuk mengubah perangkat saat ini dari perangkat kata atau kondisi ON/OFF dari relai internal, gunakan **fungsi mengubah nilai saat ini**.

Untuk memperoleh rinciannya, lihat manual.

4.1.3

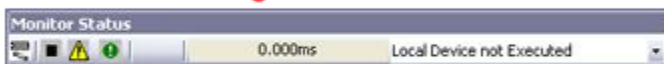
Memantau Status Perangkat

Ketika simulasi dimulai, pemantauan secara otomatis dimulai. Untuk masuk ke mode pemantauan saat tersambung ke CPU pengontrol yang dapat diprogram sebenarnya, cukup klik Online, Monitor, lalu Mulai Pemantauan. Atau pakai pintasan keyboard F3.

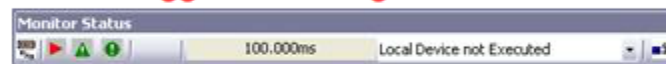
Selama mode pemantauan, nilai dan status semua perangkat yang digunakan dalam program ini dapat dilihat menimpa kode program. Ini memungkinkan pengguna melihat perubahan nilai-nilai termasuk efek dari penggunaan fungsi "pendaftaran/pembatalan input output paksa".

Selain itu, bilah **Monitor Status** diperlihatkan dan berisi informasi dasar untuk menentukan status CPU atau status virtual CPU. Lihat tabel di bawah untuk memahami informasi yang disediakan oleh bilah **Monitor Status**.

Saat tersambung ke modul CPU



Saat menggunakan fungsi simulator



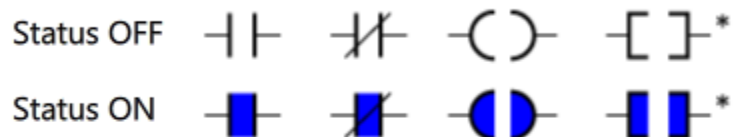
Kondisi	Ikon/indikasi	Keterangan
Status sambungan	Saat tersambung ke modul CPU	Menampilkan status sambungan dengan modul CPU atau fungsi simulator.
	Saat menggunakan fungsi simulator	
Status RUN/STOP	RUN	Menampilkan status berjalannya CPU (RUN atau STOP).
	STOP	
Status ERR.	ERR. mati	Menunjukkan status kesalahan modul CPU.
	ERR. hidup	
	ERR berkedip	
Status PENGGUNA	USER mati	Menunjukkan status kesalahan pengguna modul CPU.
	USER hidup	
	USER berkedip	
Waktu pemindaian	0.000ms	Menampilkan waktu pemindaian maksimum dari modul CPU yang sedang dipantau.
Status keberadaan/ ketiadaan petunjuk yang tidak didukung	Petunjuk yang tidak didukung ada.	Menampilkan apakah petunjuk yang tidak didukung ada saat fungsi simulator dieksekusi.
	Petunjuk yang tidak didukung tidak ada.	Mengeklik ikon akan membuka jendela Petunjuk/Perangkat yang Tidak Didukung.

4.1.3 Memantau Status Perangkat

Selama mode pemantauan, status saat ini dari semua perangkat di dalam program akan terlihat.

Tampilan status perangkat Bit (ON/OFF)

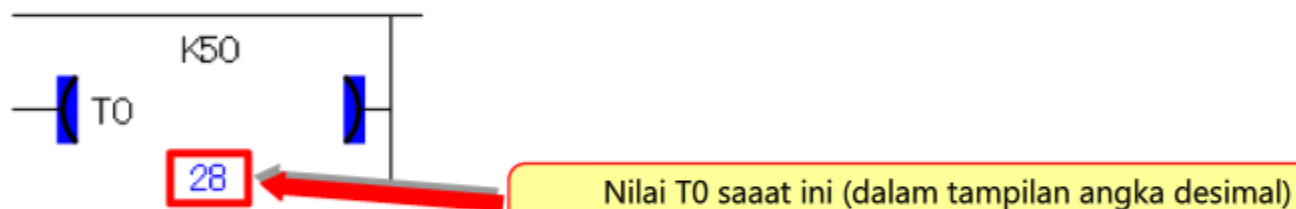
Status ON/OFF ditampilkan saat pemantauan seperti yang ditunjukkan di bawah ini.



* Jenis layar seperti ini hanya berlaku untuk SET, RST, PLS, PLF, SFT, SFTP, MC, dan petunjuk perbandingan jenis kontak. Perhatikan bahwa untuk petunjuk RST, hanya status ON/OFF akan ditampilkan.

Tampilan nilai saat ini dari perangkat kata (tampilan angka desimal/heksadesimal)

Nilai saat ini selama pemantauan ditampilkan seperti di bawah ini.



Pemantauan perangkat tertentu saja

Saat melakukan pemantauan terhadap program yang sangat besar atau kompleks, sebaiknya memantau perangkat tertentu yang diminati. Untuk mencapai hal ini, GX Works2 mencertakan jendela yang memungkinkan pengguna menambahkan perangkat yang mereka minati dengan mudah, melihat status saat ini, dan mengubah nilai-nilai mereka selama pemantauan, untuk mengetahui detailnya, lihat Panduan Operasi GX Works2 (Umum).

Watch 1					
Device/Label	Current Value	Data Type	Class	Device	Comment
X7	-	Bit		X7	Door open sensor
Y0	-	Bit		Y0	Robot start signal
Y1	-	Bit		Y1	Operation lamp
Y0	-	Bit		Y0	Robot start signal
Y2	-	Bit		Y2	Stop lamp
Y0	-	Bit		Y0	Robot start signal

4.2

Menulis Program ke Modul CPU

Sebelum melakukan debugging mEMAKAI modul CPU sebenarnya, TEMPATKAN CPU dalam **mode STOP**, pastikan sambungan ke CPU telah ditetapkan, dan tulis program dan parameter ke memori program.

Seperti yang terlihat pada tangkapan layar di bawah, fungsi utama dari **Write to PLC** memungkinkan pengguna memilih file yang diinginkan untuk ditulis, memilih lokasi mereka, dan memastikan kapasitas memori CPU. Tiga tombol di atas daftar file memungkinkan pengguna memilih file yang diinginkan untuk ditulis dengan cepat. Yang paling umum, yang digunakan dalam simulasi berikut, adalah "**Parameter+Program.**"

The screenshot shows the 'Write to PLC' dialog box with the following table of files:

Module Name/Data Name	Title	Target	Detail	Last Change	Target Memory	Size
Robot_Control						
PLC Data					Program Memory ID...	
Program(Program File)		<input checked="" type="checkbox"/>				
MA2N		<input checked="" type="checkbox"/>		2010/08/05 15:22:24		2256 Bytes
Parameter		<input checked="" type="checkbox"/>				
PLC(Network/Remote Password/Switch Setting)		<input checked="" type="checkbox"/>		2010/08/05 15:22:24		2936 Bytes
Global Device Comment		<input type="checkbox"/>				
COMMENT		<input type="checkbox"/>	Detail	2010/08/05 15:25:26		
Device Memory		<input type="checkbox"/>	Detail			
MA2N		<input type="checkbox"/>		2010/08/05 15:22:25		

The 'Writing Size' bar shows:

Writing Size	Free Volume	Use Volume
5,192Bytes	76,728	5,192Bytes

Pada halaman berikutnya, cobalah menulis modul CPU memakai jendela simulasi.

4.2

Menulis Program ke Modul CPU

MELSOFT Series GX Works2 C:\SequenceProgram\Learning\Robot_Control - [[PRG] MAIN]

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

Navigation [PRG] MAIN

Project

- Parameter
- Intelligent Function Module
- Global Device Comment
- Program Setting
- POU
 - Program
 - MAIN
 - Local Device Comment
- Device Memory
- Device Initial Value


Y0 Robot start signal

Y1 Operation lamp

Y2 Stop lamp

END

Unlabeled L02 Host Station 6/75Step

Program sekarang telah ditulis ke modul PLC.
Klik  untuk melanjutkan.

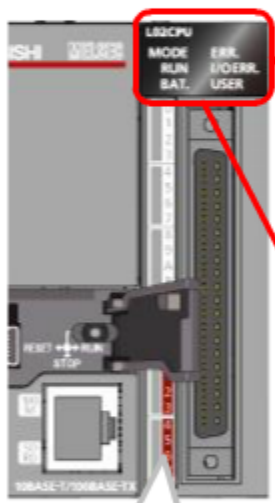
4.3 Mengaktifkan Program Tertulis

(MELSEC seri F): Operasi berikut ini tidak diperlukan.

(MELSEC seri Q dan MELSEC seri L): Operasi berikut ini diperlukan.
Setelah menulis sebuah program ke modul CPU, **atur ulang** modul CPU.
Program tertulis tidak diaktifkan kecuali modul CPU telah diatur ulang.

* Operasi ini tidak diperlukan jika fungsi simulator digunakan untuk debugging.

Atur ulang modul CPU seperti berikut ini:



RESET/STOP/
Sakelar RUN

(1) Tekan dan tahan tombol RESET/STOP/RUN pada panel depan modul CPU ke posisi RESET (selama 1 detik atau lebih).
[Pengaturan ulang sedang berlangsung]

L02CPU	
MODE ■	ERR.
RUN	I/OERR.
BAT.	USER

MODE : Nyala dalam warna hijau
RUN : Mati
ERR. : Berkedip



Tekan selama 1 detik atau lebih.

(2) Lepaskan sakelar setelah MODE LED menyala dan ERR. berkedip. Kedua LED mati.
[Pengaturan ulang selesai]

L02CPU	
MODE ■	ERR.
RUN	I/OERR.
BAT.	USER

MODE : Nyala dalam warna hijau
RUN : Mati
ERR. : Mati

(3) Sakelar kembali ke posisi STOP untuk menyelesaikan pengaturan ulang.

4.4 Menjalankan Program

MELSEC seri Q dan MELSEC seri L

Setelah pengaturan ulang selesai, jalankan program.
Tempatkan modul CPU dalam status RUN sebagai berikut untuk menjalankan program.

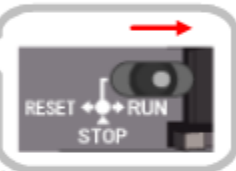
* Operasi ini tidak diperlukan jika fungsi simulator digunakan untuk debugging.



(1) Aktifkan sakelar RESET/STOP/RUN pada panel depan modul CPU ke posisi RUN.

Tampilan LED dalam status STOP

L02CPU	MODE	ERR.	MODE : Nyala dalam warna hijau
RUN	I/OERR.		RUN : Mati
BAT.	USER		



(2) Jika lampu RUN LED menyala hijau, program ini berjalan normal.

Tampilan LED dalam status RUN

L02CPU	MODE	ERR.	MODE : Nyala dalam warna hijau
RUN	I/OERR.		RUN : Nyala dalam warna hijau
BAT.	USER		

MELSEC seri F

Setelah menuliskan program ke unit utama, atur unit utama ke status RUN sebagai berikut untuk menjalankan program.
(Operasi reset tidak diperlukan.)

(1) Atur sakelar RUN/STOP pada panel depan unit utama ke posisi RUN.



Tampilan LED dalam status STOP

(2) Jika LED RUN menyala, program berjalan normal.



Tampilan LED dalam status RUN

4.5

Mencari Kesalahan (Debugging) Program

Setelah menjalankan modul CPU, pakai fimgso pendaftaran/pembatalan input output paksa untuk mengubah status masing-masing perangkat dan memantau hasilnya (output) pada ladder.

(Layar contoh MELSEC seri Q dan MELSEC seri L)

The screenshot shows the MELSOFT GX Works2 interface. A ladder logic program is visible in the background, featuring a normally open contact labeled 'Y0' and a coil labeled 'Y1'. A dialog box titled 'Forced Input Output Registration/Cancellation' is overlaid on the program. The dialog box has a 'Device' dropdown menu and buttons for 'Register FORCE ON', 'Register FORCE OFF', and 'Cancel Registration'. Below these buttons is a table with two columns of device numbers (1-16) and their corresponding ON/OFF status. The table is currently empty.

No.	Device	ON/OFF	No.	Device	ON/OFF
1	X5	ON	17		
2	X7	OFF	18		
3			19		
4			20		
5			21		
6			22		
7			23		
8			24		
9			25		
10			26		
11			27		
12			28		
13			29		
14			30		
15			31		
16			32		

Pada halaman berikutnya, cobalah melakukan debugging program ini memakai jendela simulasi.

4.5

Mencari Kesalahan (Debugging) Program



MELSOFT Series GX Works2 C:\SequenceProgram\Learning\Robot_Control - [[PRG] MAIN]

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

1.000ms Local Device not Executed

Navigation

Project

- Parameter
- Intelligent Function Module
- Global Device Comment
- Program Setting
- POU
 - Program
 - MAIN
 - Local Device Comment
- Device Memory
- Device Initial Value


Project

User Library

Connection Destination

Unlabeled L02 Host Station 6/75Step

Diagram Description: The diagram shows a ladder logic program with three rungs. Rung 0 contains two normally open contacts labeled 'Start Switch' (X6) and 'Door open sensor' (X7), connected to two outputs: 'Robot start signal' (Y0) and 'Operation lamp' (Y1). Rung 4 contains a normally open contact labeled 'Robot start signal' (Y0) connected to an output labeled 'Stop lamp' (Y2). Rung 6 contains an 'END' instruction.

Pencarian kesalahan (debugging) program telah selesai.
Klik  untuk melanjutkan.

4.6

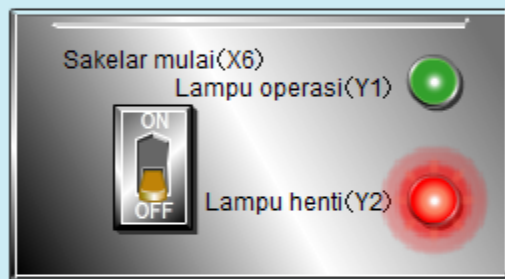
Memeriksa Operasi Sistem PLC

Setelah program debugging selesai, tulis program ke sistem PLC sebenarnya untuk memeriksa operasi dalam tahap akhir. Mengoperasikan peralatan I/O sebenarnya untuk memastikan bahwa peralatan berfungsi sesuai rencana. Bahkan saat mengoperasikan peralatan I/O, status setiap perangkat dapat diperiksa menggunakan fungsi monitor GX Works2.

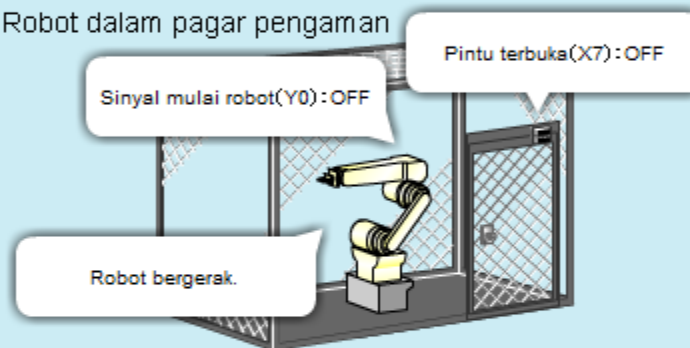
Operasi sistem contoh

 Klik di dalam lingkaran merah

Panel kontrol robot



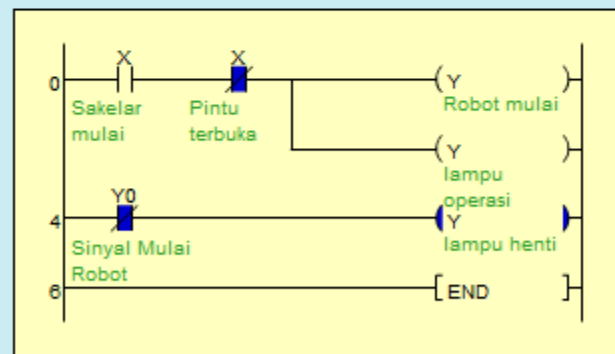
Robot dalam pagar pengaman



Ketika Anda menetapkan sakelar mulai (X6) ditetapkan ke OFF, sinyal mulai robot (Y0) mati untuk menghentikan robot beroperasi. Pada saat yang bersamaan, lampu operasi (Y1) pada panel kontrol mati, dan lampu henti (Y2) hidup.

Ulangi

 Sebelumnya



4.7

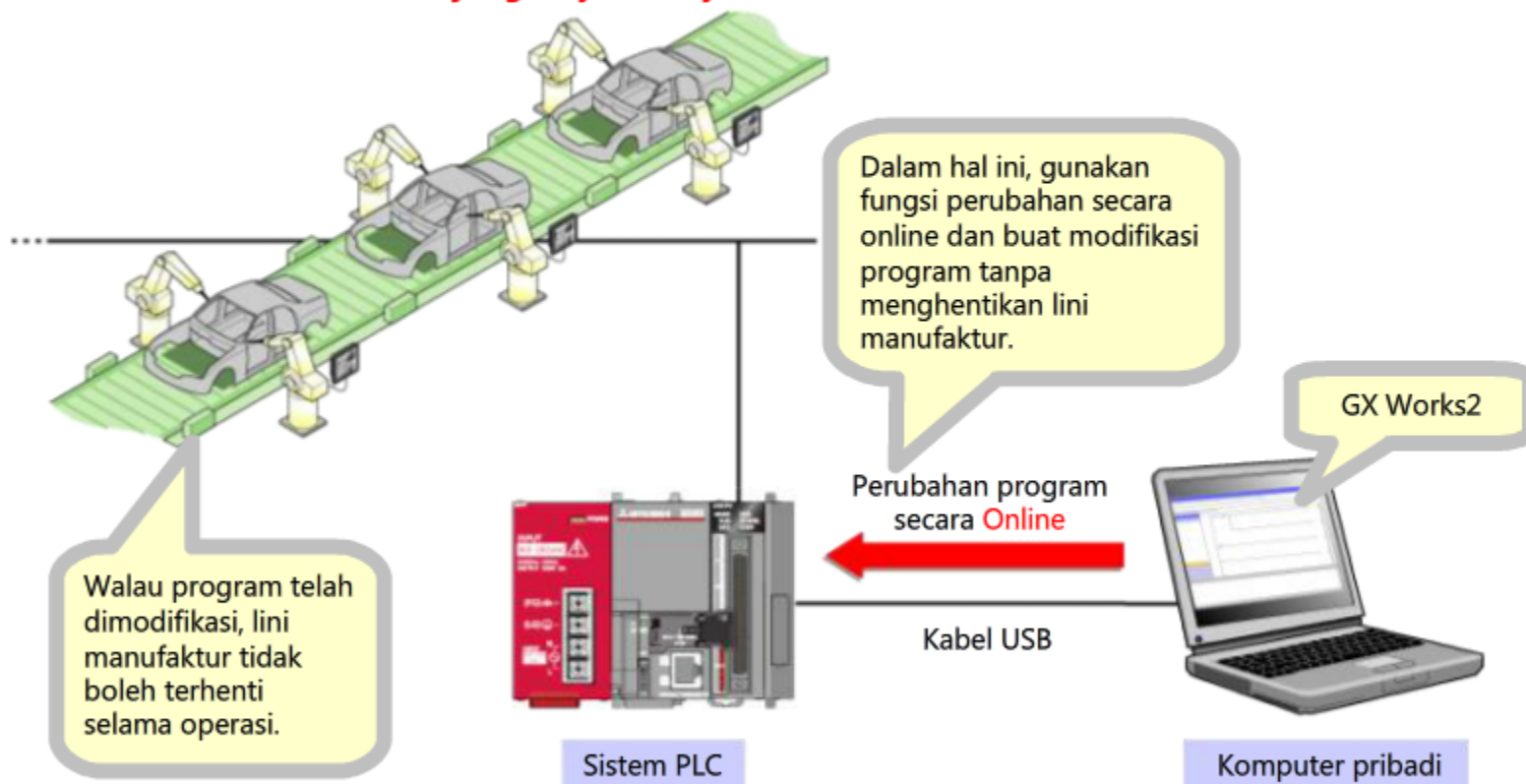
Mengoperasikan Sistem PLC

Setelah verifikasi operasi selesai, jalankan sistem PLC untuk memulai operasi.

Jika program perlu diubah dalam sistem yang berjalan

Modifikasi program seperti perbaikan bug atau perluasan sistem mungkin diperlukan setelah operasi sistem dimulai. Biasanya, sistem (modul CPU) harus dihentikan untuk menulis program yang telah dimodifikasi, namun hal ini tidak selalu mungkin dilakukan. Untuk mengatasi masalah ini, GX Works menyediakan fungsi perubahan online, yang digunakan untuk menulis program tanpa menghentikan modul CPU yang sedang berjalan.

Contoh: Lini manufaktur mobil yang berjalan 24 jam



Pada halaman berikutnya, cobalah menggunakan fungsi perubahan online menggunakan jendela simulasi.

4.7

Mengoperasikan Sistem PLC

MELSOFT Series GX Works2 C:\SequenceProgram\Learning\Robot_Control - [[PRG] MAIN]

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

1.000ms Local Device not Executed

Navigation

Project

- Parameter
- Intelligent Function Module
- Global Device Comment
- Program Setting
- POU
 - Program
 - MAIN
- Local Device Comment
- Device Memory
- Device Initial Value


Project

User Library

Connection Destination

```
graph TD
    subgraph Step_0 [0]
        X8[X8 Start Switch] --- AND --- X9[X9 Robot start signal]
        AND --- Y0[Y0 Robot start signal]
        AND --- Y1[Y1 Operation lamp]
    end
    subgraph Step_4 [4]
        Y0[Y0 Robot start signal] --- Y2[Y2 Stop lamp]
    end
    subgraph Step_6 [6]
        END[END]
    end
```

Unlabeled L02 Host Station 2/75Step N/A

Perubahan online program yang telah dimodifikasi selesai.
Klik  untuk melanjutkan.

4.8

Kesimpulan

Ini menutup penjelasan dasar dari desain perangkat lunak pengontrol yang dapat diprogram.

Di dalam kursus ini, Anda telah mempelajari tentang:

- Elemen yang diperlukan untuk memprogram sistem PLC
- Beberapa panduan dasar desain program termasuk pemakaian komentar
- Cara memakai GX Works2 untuk melakukan tugas-tugas dasar pemrograman PC
- Beberapa teknik yang dipakai untuk melakukan debugging program PLC

Setelah menyelesaikan semua pelajaran dari **PLC GX Works2 Dasar**, kini Anda siap mengikuti tes akhir. Jika Anda masih kurang memahami salah satu topik yang dibahas, gunakan kesempatan ini untuk mengulas topik tersebut. **Total terdapat 5 pertanyaan (15 pilihan) dalam Tes Akhir ini.** Anda dapat mengikuti tes akhir sesering mungkin.

Cara menilai tes

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

Hasil penilaian

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

Jawaban yang benar : 2
Jumlah total pertanyaan : 9
Persentase : 22%

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

Lanjutkan

Tinjau

Coba lagi

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

Program yang menjadi tanggung jawab Anda diambil alih oleh orang lain yang merasa mengalami kesulitan memahami item kontrol untuk program ini. Apakah tindakan penanggulangannya yang benar untuk mencegah masalah ini??

- Dengan memakai fungsi komentar dari GX Works2, berikan judul dan penjelasan yang tepat untuk program.
- Jelaskan secara lisan item kontrol kepada orang baru itu.
- Hindari mengambil alih sebuah program yang besar dan kompleks.
- Memindahkan tabel keterkaitan untuk perangkat I/O dan nomor perangkat bersama-sama dengan program ini.

[Skor](#)[Kembali](#)

Test**Final Test 2**

Selesaikan prosedur pemrograman yang benar.

Langkah 1 Desain program

Langkah 2 (Q1)

Langkah 3 (Q2)

Langkah 4 Mengonversikan program

Langkah 5 Menyimpan Proyek

Langkah 6 (Q3)

Langkah 7 (Q4)

Langkah 8 Menjalankan modul CPU (RUN)

Langkah 9 (Q5)

Langkah 10 Memeriksa Operasi Sistem PLC

Skor

Kembali

Isilah bidang yang kosong untuk melengkapi penjelasan tentang hal-hal yang harus dilakukan setelah program selesai.

Setelah program ditulis, program itu harus diuji untuk memastikan bahwa program beroperasi sesuai harapan.

A (Q1) (saat kode tertulis tidak berfungsi sesuai harapan) dipanggil

a (Q2) dan proses pencarian penyebab dan perbaikannya dipanggil

(Q3).

Proses ini merupakan langkah yang sangat penting dalam menciptakan program.

Test**Final Test 4**

Pilih aplikasi yang sesuai untuk setiap fungsi GX Works2.

Fungsi	Aplikasi
Simulasi	--Select--
Pendaftaran/pembatalan masukan keluaran paksa	--Select--
Mengubah nilai saat ini	--Select--
Monitor ladder	--Select--
Lihat	--Select--

[Skor](#)[Kembali](#)

Pilih keterangan yang tepat dari fungsi perubahan online.

- Fungsi ini otomatis menghentikan CPU, menulis program untuk CPU, lalu menjalankan CPU secara otomatis.
- Fungsi ini membandingkan program dalam modul CPU yang berjalan dengan program yang dibuka oleh GX Works2.
- Fungsi ini dapat menulis program untuk modul CPU setelah menghentikan modul CPU yang sedang berjalan dengan aman.
- Fungsi ini dapat menuliskan program untuk modul CPU yang berjalan tanpa menghentikannya.

Skor

Kembali

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

Jawaban yang benar : 0

Jumlah total pertanyaan : 5

Persentase : 0%

Lanjutkan

Tinjau

Coba lagi

Tes Anda gagal.

Anda telah menyelesaikan **PLC GX Works2 Dasar**.

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.

You can review the course as many times as you want.

Tinjau

Tutup