

**Author : Muhammad Mardiansyah**

# **LAPORAN SIMULASI SCADA - SISTEM KONVEYOR**

---

## **1. Pendahuluan**

Laporan ini dibuat sebagai dokumentasi pelaksanaan simulasi sistem SCADA berbasis Modbus TCP menggunakan aplikasi Factory I/O dan OpenPLC. Simulasi ini bertujuan untuk memodelkan proses otomasi sederhana berupa sistem konveyor, serta menguji komunikasi antar perangkat secara virtual sebagai bagian dari pembelajaran sistem kontrol industri dan dasar Operational Technology (OT).

---

## **2. Rancangan Sistem**

Pada proyek ini, sistem otomasi yang disimulasikan terdiri dari:

- Satu unit konveyor dengan panjang 4 meter.
- Sebuah tombol Start (Push Button).
- Sebuah tombol Stop.
- Variabel Motor untuk menjalankan dan menghentikan konveyor.

Desain tersebut diimplementasikan pada platform Factory I/O dengan konfigurasi sebagai Modbus TCP Server, yang berperan sebagai antarmuka fisik (I/O) dari sistem

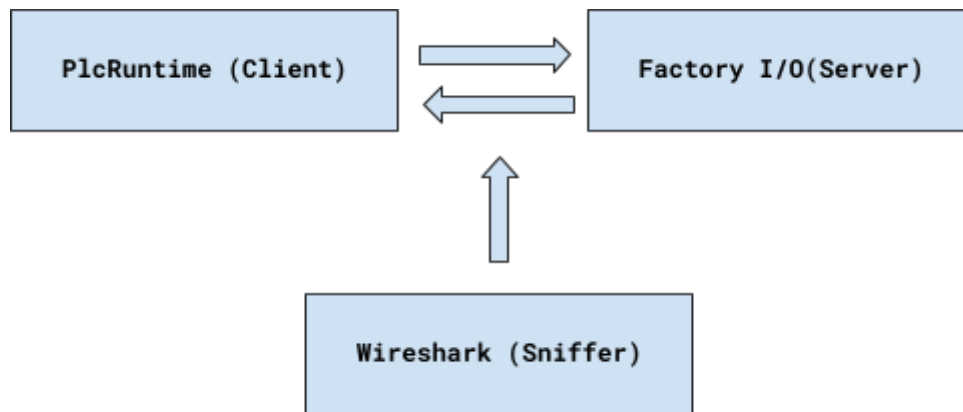
---

## **3. Implementasi Logika**

Logika pengendalian sistem dikembangkan menggunakan bahasa pemrograman Ladder Diagram (LD) melalui software OpenPLC Editor. Dalam logika tersebut, tombol Start digunakan untuk mengaktifkan motor penggerak konveyor, sedangkan tombol Stop digunakan untuk menghentikannya. Motor hanya akan aktif apabila tombol Start ditekan dan kondisi Stop tidak aktif.

Setelah logika dirancang dan diuji secara lokal, file program disimpan dalam format .st (Structured Text), lalu diunggah ke OpenPLC Runtime untuk dijalankan secara langsung. OpenPLC Runtime dikonfigurasi sebagai Modbus TCP Client (slave) dengan alamat IP 192.168.155.55, yang memastikan koneksi antara logika kontrol dan antarmuka konveyor di Factory I/O.

## 4. Topologi Simulasi



Seluruh simulasi dilakukan di dalam satu komputer (localhost: 127.0.0.1)

---

## 5. Langkah-Langkah Simulasi

### A. Desain Sistem di Factory I/O

- Buka *Factory I/O*.
- Pilih mode **Modbus TCP Server** pada pengaturan koneksi.
- Rancang sistem conveyor dengan spesifikasi:
  - **1 conveyor** sepanjang 4 meter.
  - **1 tombol Start** (input digital).
  - **1 tombol Stop** (input digital).
  - **1 variabel Motor** (output digital) untuk menggerakkan conveyor.

### B. Buat Logika Ladder (LD) di OpenPLC Editor

- Buka *OpenPLC Editor* dan buat program Ladder Diagram.
- Logika dasar:
  - Jika tombol Start ditekan, nyalakan motor.
  - Jika tombol Stop ditekan, matikan motor.
- Setelah selesai, **kompilasi program menjadi file** (Structured Text).

### c. Unggah Logika ke OpenPLC Runtime

- Jalankan *OpenPLC Runtime* di perangkat.
- Unggah file .st ke runtime melalui web interface.
- Konfigurasi:
  - **PLC sebagai Modbus TCP Client (Master).**
  - **Modbus TCP Slave IP = 192.168.155.55** (IP Factory I/O).
  - Sesuaikan *polling interval* dan port (default Modbus: 502).

### D. Uji Koneksi dan Jalankan Simulasi

- Pastikan OpenPLC berhasil terkoneksi ke Factory I/O.
- Jalankan simulasi.
- Tekan tombol Start dan Stop secara bergantian untuk mengamati apakah conveyor bekerja sesuai logika.

#### E. Monitoring dan Analisis dengan Wireshark

- Buka *Wireshark* dan tangkap traffic jaringan.
  - Filter protokol: modbus atau gunakan port 502.
  - Amati:
    - Komunikasi *read/write register* dalam bentuk **plaintext**.
    - Tidak adanya enkripsi dalam komunikasi
    - Jika menggunakan antarmuka monitoring berbasis HTTP, **cookie dan session token** akan terlihat jelas (vulnerable terhadap sniffing).
- 

## 6. Hasil Observasi

Setelah semua konfigurasi selesai, koneksi antar komponen divalidasi dan dinyatakan berhasil. Sistem simulasi dapat dijalankan dengan baik, di mana tombol Start dan Stop dapat mengendalikan motor conveyor secara real-time sesuai dengan logika LD yang dibuat. Setelah itu pantau hasil logika di monitoring openPLC Runtime untuk memantau apakah sistem dalam kondisi start atau stop.

Untuk memverifikasi komunikasi data dan kestabilan jaringan, dilakukan pemantauan lalu lintas jaringan menggunakan Wireshark. Hasilnya menunjukkan bahwa komunikasi Modbus TCP antara Factory I/O dan OpenPLC berjalan stabil dan sesuai dengan ekspektasi.

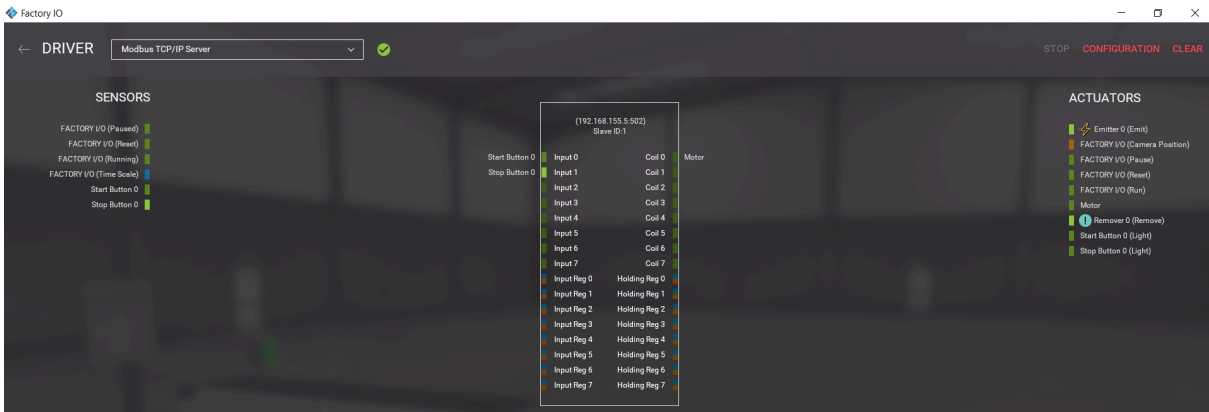
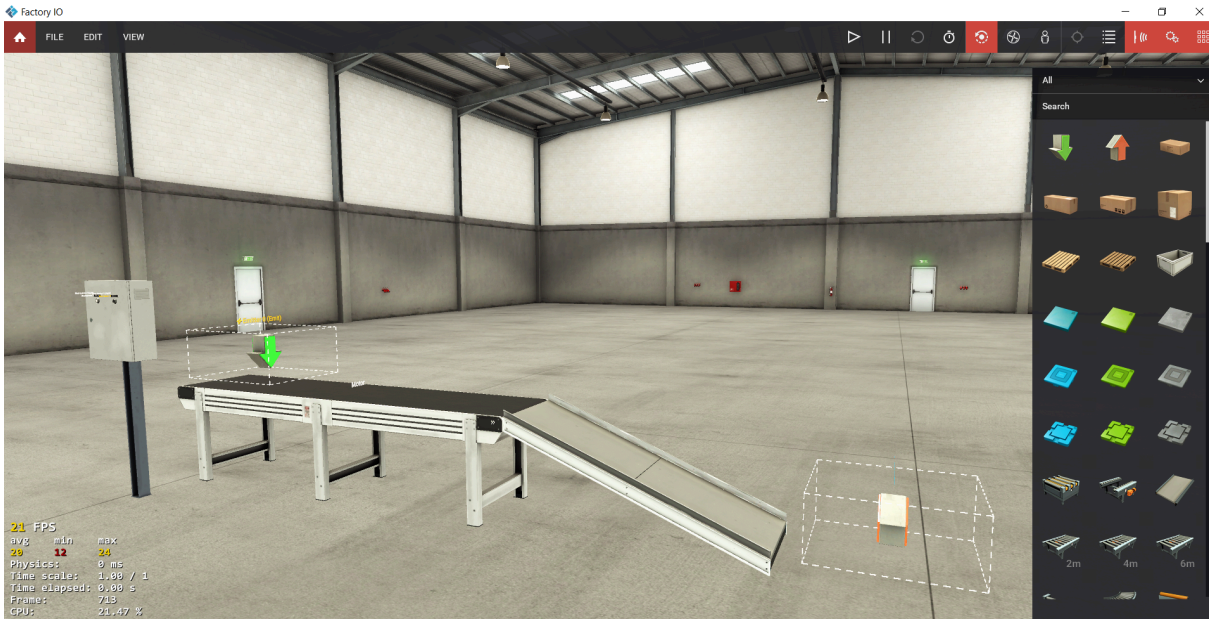
---

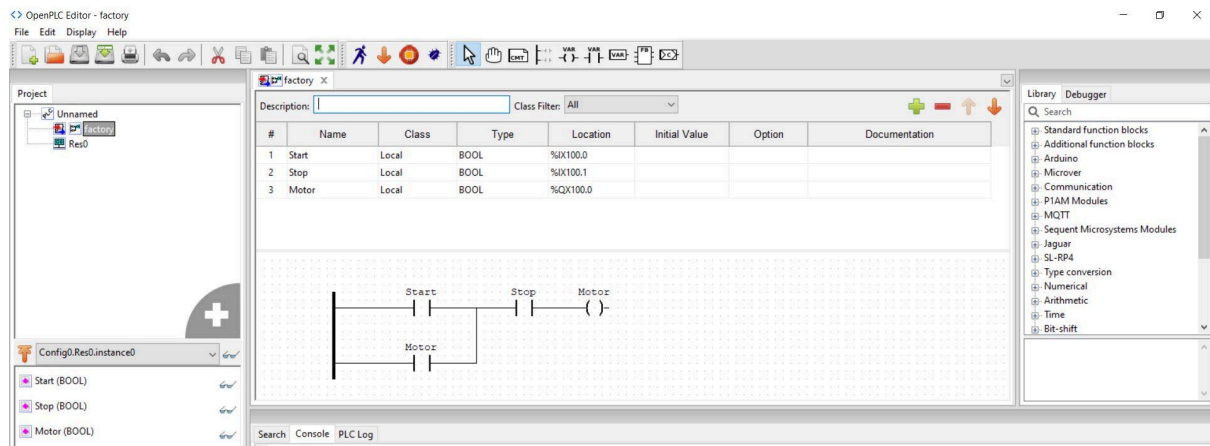
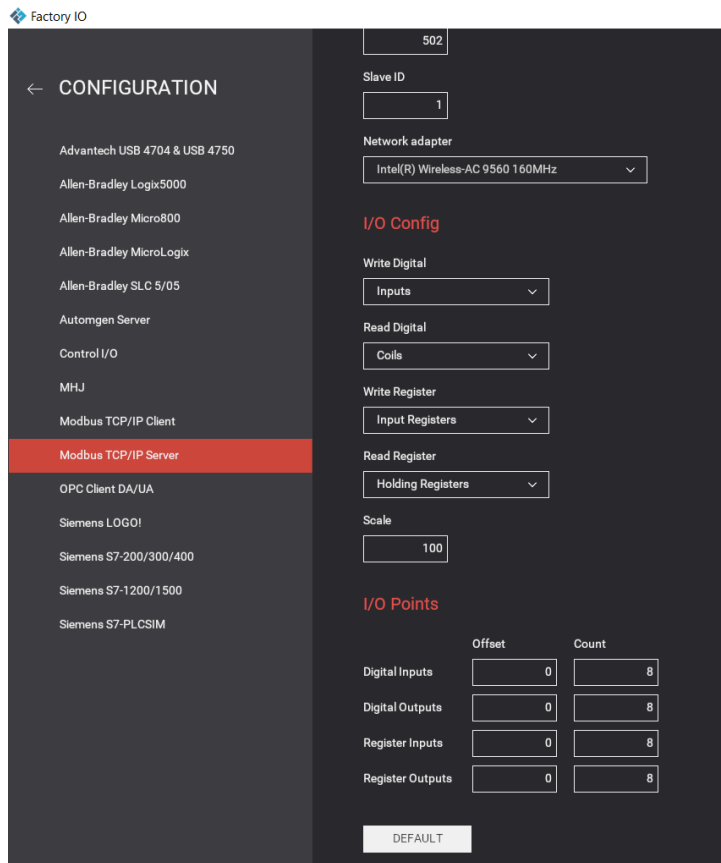
## 7. Kesimpulan

Simulasi sistem SCADA berbasis conveyor berhasil dilakukan menggunakan Factory I/O dan OpenPLC. Sistem mampu merepresentasikan proses otomasi dasar dengan logika kontrol yang sesuai, serta komunikasi jaringan yang valid dan aktif. Keberhasilan konektivitas dan fungsionalitas simulasi membuktikan bahwa sistem dapat digunakan sebagai sarana pembelajaran dasar kontrol industri dan komunikasi OT berbasis Modbus TCP.

---

## 8. Dokumentasi Tambahan





## Programs

Here you can upload a new program to OpenPLC or revert back to a previous uploaded program shown on the table.

Program Name	File	Date Uploaded
factory	638641.st	Jul 06, 2025 - 09:36AM
Snap7_Map	4968.st	Mar 18, 2025 - 02:22PM
Blank Program	blank_program.st	May 25, 2018 - 01:02AM

[List all programs](#)

## Upload Program

Slave Devices

List of Slave devices attached to OpenPLC.

Attention: Slave devices are attached to address 100 onward (i.e. %IX100.0, %IW100, %QX100.0, and %QW100)




Device Name	Device Type	DI	DO	AI	AO
factory	TCP	%IX100.0 to %IX100.7	%QX100.0 to %QX100.7	%IW100 to %IW107	%QW100 to %QW107

Add new device

Monitoring

Refresh Rate (ms): 500

Update

Point Name	Type	Location	Write	Value
Start	BOOL	%IX100.0		 FALSE
Stop	BOOL	%IX100.1		 TRUE
Motor	BOOL	%QX100.0	<div>truefalse</div>	 FALSE

factory.pcapng

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-F>

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	:::1	:::1	TCP	76	61467 → 8080 [SYN] Seq=0 Win=65535 Len=0 MSS=65475 WS=256 SACK_PERM
2	0.000238	:::1	:::1	TCP	64	8080 → 61467 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
3	0.021791	192.168.155.5	192.168.155.5	Modbus...	56	Query: Trans: 12905; Unit: 1, Func: 2: Read Discrete Inputs
4	0.022060	192.168.155.5	192.168.155.5	TCP	44	502 → 60471 [ACK] Seq=1 Ack=13 Win=10156 Len=0
5	0.024047	192.168.155.5	192.168.155.5	Modbus...	54	Response: Trans: 12905; Unit: 1, Func: 2: Read Discrete Inputs
6	0.024263	192.168.155.5	192.168.155.5	TCP	44	60471 → 502 [ACK] Seq=13 Ack=11 Win=10001 Len=0
7	0.026153	192.168.155.5	192.168.155.5	Modbus...	58	Query: Trans: 12906; Unit: 1, Func: 15: Write Multiple Coils
8	0.026231	192.168.155.5	192.168.155.5	TCP	44	502 → 60471 [ACK] Seq=11 Ack=27 Win=10156 Len=0
9	0.026743	192.168.155.5	192.168.155.5	Modbus...	56	Response: Trans: 12906; Unit: 1, Func: 15: Write Multiple Coils
10	0.026815	192.168.155.5	192.168.155.5	TCP	44	60471 → 502 [ACK] Seq=27 Ack=23 Win=10001 Len=0
11	0.027112	192.168.155.5	192.168.155.5	Modbus...	56	Query: Trans: 12907; Unit: 1, Func: 4: Read Input Registers
12	0.027147	192.168.155.5	192.168.155.5	TCP	44	502 → 60471 [ACK] Seq=23 Ack=39 Win=10156 Len=0
13	0.027772	192.168.155.5	192.168.155.5	Modbus...	69	Response: Trans: 12907; Unit: 1, Func: 4: Read Input Registers
14	0.027839	192.168.155.5	192.168.155.5	TCP	44	60471 → 502 [ACK] Seq=39 Ack=48 Win=10001 Len=0
15	0.028425	192.168.155.5	192.168.155.5	Modbus...	73	Query: Trans: 12908; Unit: 1, Func: 16: Write Multiple Registers
16	0.028476	192.168.155.5	192.168.155.5	TCP	44	502 → 60471 [ACK] Seq=48 Ack=68 Win=10156 Len=0
17	0.029983	192.168.155.5	192.168.155.5	Modbus...	56	Response: Trans: 12908; Unit: 1, Func: 16: Write Multiple Registers
18	0.030059	192.168.155.5	192.168.155.5	TCP	44	60471 → 502 [ACK] Seq=68 Ack=60 Win=10001 Len=0
19	0.071921	127.0.0.1	127.0.0.1	TCP	56	61468 → 8080 [SYN] Seq=0 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM
20	0.072112	127.0.0.1	127.0.0.1	TCP	56	8080 → 61468 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM
21	0.072263	127.0.0.1	127.0.0.1	TCP	44	61468 → 8080 [ACK] Seq=1 Ack=1 Win=2619648 Len=0
22	0.076208	127.0.0.1	127.0.0.1	HTTP	862	GET /monitor-update?wb_port=502 HTTP/1.1
23	0.076295	127.0.0.1	127.0.0.1	TCP	44	8080 → 61468 [ACK] Seq=1 Ack=819 Win=2619648 Len=0
24	0.147830	192.168.155.5	192.168.155.5	Modbus...	56	Query: Trans: 12909; Unit: 1, Func: 2: Read Discrete Inputs
25	0.147978	192.168.155.5	192.168.155.5	TCP	44	502 → 60471 [ACK] Seq=60 Ack=80 Win=10156 Len=0
26	0.148600	192.168.155.5	192.168.155.5	Modbus...	54	Response: Trans: 12909; Unit: 1, Func: 2: Read Discrete Inputs
27	0.148690	192.168.155.5	192.168.155.5	TCP	44	60471 → 502 [ACK] Seq=80 Ack=70 Win=10001 Len=0
28	0.149306	192.168.155.5	192.168.155.5	Modbus...	58	Query: Trans: 12910; Unit: 1, Func: 15: Write Multiple Coils
29	0.149366	192.168.155.5	192.168.155.5	TCP	44	502 → 60471 [ACK] Seq=70 Ack=94 Win=10156 Len=0
30	0.151306	192.168.155.5	192.168.155.5	Modbus...	56	Response: Trans: 12910; Unit: 1, Func: 15: Write Multiple Coils

> Frame 1: 76 bytes on wire (608 bits), 76 bytes captured (608 bits) on interface \Device\NPF\_{...} id 0

> Null/Loopback

> Internet Protocol Version 6, Src: ::1, Dst: ::1

> Transmission Control Protocol, Src Port: 61467, Dst Port: 8080, Seq: 0, Len: 0

0000 18 00 00 00 00 00 04 05 00 20 06 80 00 00 00 00 .....  
0010 00 00 00 00 00 00 00 00 00 00 01 00 00 00 .....  
0020 00 00 00 00 00 00 00 00 00 00 01 f0 1b 1f 90 .....  
0030 e3 98 59 62 00 00 00 00 02 ff 28 58 00 00 ..Yb.....(X-  
0040 02 04 ff c3 01 03 03 08 01 01 04 02 .....

factory.pcapng

Packets: 14899

Profile: Default

1632

06/07/2025

21°C Berawan

