



TUGAS AKHIR (AE43250)

RANCANG BANGUN SISTEM WEB-SCADA PADA *MODULAR PRODUCTION SYSTEM* MENGGUNAKAN MULTI KONTROLLER

Muhammad Reza Alfiano
NRP. 0922040088

Dosen Pembimbing
Lilik Subiyanto, S.T., M.T.
Adianto, S.T., M.T.

PROGRAM STUDI TEKNIK OTOMASI
JURUSAN TEKNIK KELISTRIKAN KAPAL
POLITEKNIK PERKAPALAN NEGERI SURABAYA
SURABAYA
2026

(Halaman ini sengaja dikosongkan)



PPNS POLITEKNIK
PERKAPALAN
NEGERI SURABAYA

TUGAS AKHIR (AE43250)

RANCANG BANGUN SISTEM WEB-SCADA PADA MODULAR PRODUCTION SYSTEM MENGGUNAKAN MULTI KONTROLLER

Muhammad Reza Alfiano
NRP. 0922040088

Dosen Pembimbing
Lilik Subiyanto, S.T., M.T.
Adianto, S.T., M.T.

PROGRAM STUDI TEKNIK OTOMASI
JURUSAN TEKNIK KELISTRIKAN KAPAL
POLITEKNIK PERKAPALAN NEGERI SURABAYA
SURABAYA
2026

(Halaman ini sengaja dikosongkan)

LEMBAR PENGESAHAN

TUGAS AKHIR

RANCANG BANGUN SISTEM WEB-SCADA PADA MODULAR PRODUCTION SYSTEM MENGGUNAKAN MULTI KONTROLLER

Disusun Oleh:

Muhammad Reza Alfiano
0922040088

Diajukan Untuk Memenuhi Salah Satu Syarat Kelulusan
Program Studi D4 Teknik Otomasi
Jurusan Teknik Kelistrikan Kapal
POLITEKNIK PERKAPALAN NEGERI SURABAYA

Disetujui oleh Tim penguji Tugas Akhir Tanggal Ujian : 21 Juli 2026
Periode Wisuda : Oktober 2026

Menyetujui,

Dosen Penguji

NIDN

Tanda Tangan

1. Lilik Subiyanto, S.T., M.T.

(0030016903)

(.....)

2. Ryan Yudha Adhitya, S.ST., M.T.

(0016069101)

(.....)

3. Adianto, S.T., M.T.

(0002077704)

(.....)

4. Ludia Ni'matuzzahroh, S.Stat., M.Stat.

(0613059601)

(.....)

Dosen Pembimbing

NIDN

Tanda Tangan

1. Lilik Subiyanto, S.T., M.T.

(0030016903)

(.....)

2. Adianto, S.T., M.T.

(0002077704)

(.....)

Menyetujui

Ketua Jurusan,

Mengetahui,

Koordinator Program Studi



Isa Rachman, S.T., M.T.

NIP. 198008162008121001

Agus Khumaidi, S.ST., M.T.

NIP. 199308172020121004

(Halaman ini sengaja dikosongkan)



PERNYATAAN BEBAS PLAGIAT

No. : F.WD I. 021
Date : 3 Nopember 2015
Rev. : 01
Page : 1 dari 1

Yang bertandatangan dibawah ini :

Nama : Muhammad Reza Alfiano

NRP. : 0922040088

Jurusan/Prodi : Teknik Kelistrikan Kapal/Teknik Otomasi

Dengan ini menyatakan dengan sesungguhnya bahwa :

Tugas Akhir yang akan saya kerjakan dengan judul :

Rancang Bangun Sistem WEB-SCADA Pada *Modular Production System* Menggunakan Multi Kontroller

Adalah benar karya saya sendiri dan bukan plagiat dari karya orang lain.

Apabila dikemudian hari terbukti terdapat plagiat dalam karya ilmiah tersebut, maka saya bersedia menerima **sanksi** sesuai ketentuan peraturan yang berlaku.

Demikian surat pernyataan ini saya buat dengan penuh tanggung jawab.

Surabaya, 27 Agustus 2026

Yang membuat pernyataan,



(Muhammad Reza Alfiano)
NRP. 0922040088

(Halaman ini sengaja dikosongkan)

KATA PENGANTAR



Dengan mengucapkan syukur Alhamdulillah, segala puji penulis panjatkan ke hadirat Tuhan Yang Maha Esa atas limpahan rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul **“Rancang Bangun Sistem WEB-SCADA Pada Modular Production System Menggunakan Multi Kontroller”** dengan baik dan lancar. Sholawat serta salam senantiasa tercurah kepada junjungan Nabi Muhammad SAW, yang telah membimbing umat manusia dari zaman kegelapan menuju era yang terang benderang penuh ilmu pengetahuan

Tugas Akhir ini disusun sebagai salah satu syarat untuk menyelesaikan studi pada Program Sarjana Terapan D-IV Teknik Otomasi di Politeknik Perkapalan Negeri Surabaya serta untuk memperoleh gelar Sarjana Terapan Teknik Otomasi. Penulis menyadari sepenuhnya bahwa tersusunnya laporan ini tidak lepas dari dukungan, bimbingan, serta doa dari berbagai pihak yang telah memberikan kontribusi baik secara langsung maupun tidak langsung. Oleh karena itu, penulis ingin menyampaikan banyak rasa terima kasih dan penghargaan yang sebesar-besarnya khususnya kepada :

1. Allah SWT, karena berkat limpahan nikmat sehat, nikmat ilmu, nikmat waktu, serta limpahan rahmat dan hidayah-Nya sehingga Penulis dapat menyelesaikan Tugas Akhir ini tepat waktu.
2. Orang tua penulis, yang selalu memberikan doa, kasih sayang, dukungan, perhatian, serta motivasi kepada penulis selama proses penyusunan Tugas Akhir ini. Terima kasih atas segala pengorbanan, kerja keras, dan kepercayaan yang telah diberikan. Bapak dan Ibu senantiasa memberikan semangat kepada penulis dalam setiap proses yang dijalani, serta mengingatkan penulis untuk tetap berusaha dan menyelesaikan Tugas Akhir ini dengan sebaik-baiknya. Dukungan yang tidak pernah putus dari Bapak dan Ibu menjadi salah satu alasan penulis dapat menyelesaikan Tugas Akhir ini hingga tahap akhir.
3. Nenek penulis yang senantiasa mendoakan, memberikan kasih sayang, dukungan, dan semangat kepada penulis selama proses penyusunan tugas akhir ini. Terima kasih atas perhatian serta masakan yang selalu menjadi penyemangat bagi penulis.

4. Bapak Rachmad Tri Soelistijono, S.T., M.T. yaitu selaku Direktur Politeknik Perkapalan Negeri Surabaya.
5. Bapak Isa Rachman, ST., MT. selaku Ketua Jurusan Teknik Kelistrikan Kapal Politeknik Perkapalan Negeri Surabaya.
6. Bapak Agus Khumaidi, S.ST., M.T. selaku Koordinator Program Studi Teknik Otomasi Politeknik Perkapalan Negeri Surabaya.
7. Bapak Lilik Subiyanto, S.T., M.T. selaku Dosen Pembimbing I yang dengan penuh kesabaran telah memberikan motivasi, arahan, serta bimbingan kepada penulis selama proses penyusunan Tugas Akhir ini. Berbagai masukan dan ilmu yang diberikan menjadi bekal yang sangat berarti sehingga penulis dapat menyelesaikan Tugas Akhir ini.
8. Bapak Adianto, S.T., M.T. selaku Dosen Pembimbing II yang senantiasa mendampingi penulis dari awal hingga akhir proses penyusunan Tugas Akhir ini. Dengan penuh perhatian, semangat, dan ketulusan, beliau selalu memberikan kepercayaan, arahan, serta bimbingan yang sangat berarti sehingga penulis dapat menyelesaikan Tugas Akhir ini.
9. Seluruh jajaran SMKN 1 Driyorejo, khususnya Bapak Asrul, Ibu Luluk, serta teman-teman Elektro Skandry yang telah memberikan bantuan, dukungan, serta memfasilitasi penulis dalam pelaksanaan dan penyelesaian Tugas Akhir ini hingga dapat berjalan dengan baik.

RANCANG BANGUN SISTEM WEB-SCADA PADA *MODULAR PRODUCTION SYSTEM* MENGGUNAKAN MULTI KONTROLLER

Muhammad Reza Alfiano

ABSTRAK

Perkembangan industri 4.0 menuntut sistem otomasi yang mampu mengendalikan proses produksi sekaligus menyediakan informasi operasional terintegrasi untuk mendukung pengambilan keputusan berbasis data. Penelitian ini merancang Web-SCADA berbasis *multi-controller* yang mengintegrasikan PLC Omron CP2E-N30 DRA melalui FINS UDP dan PLC Siemens S7-1200 melalui S7 TCP/IP pada *Modular Production System*, khususnya *distribution station* dan *sorting station*. *Gateway* multi-protokol berbasis Node.js digunakan sebagai penghubung antar-PLC serta pengirim data menuju Web-SCADA lokal dan berbasis *cloud*. Data operasional ditampilkan dalam bentuk *monitoring* proses, status operasi, alarm, *data logger*, serta perhitungan *Overall Equipment Effectiveness* yang meliputi *Availability*, *Performance*, dan *Quality*. Hasil pengujian menunjukkan rata-rata *delay read* FINS UDP sebesar 1,7–2,0 ms dan S7 TCP/IP sebesar 12–13 ms, sedangkan rata-rata *latency gateway* ke *cloud* sebesar 315,2 ms. Setelah kinerja komunikasi diuji, pengujian dilanjutkan pada aspek produksi sebagai dasar perhitungan OEE. Pengujian *cycle time* menghasilkan rata-rata 0,31 menit per produk sebagai parameter *Performance*. Sistem menghitung OEE otomatis sebesar 43,92%, 70,27%, dan 85,25% pada tiga skenario, serta menghasilkan *Performance* aktual *plant* sebesar 92,47% dengan *Availability* 100%, *Quality* 100%, dan OEE 92,47%. Hasil sistem sesuai dengan perhitungan manual dengan *error* 0%. Dengan demikian, sistem mampu melakukan supervisi, akuisisi data, integrasi multi-protokol, pemantauan jarak jauh, dan evaluasi kinerja MPS secara akurat.

Kata kunci : Web-SCADA, Multi Kontroller, *Gateway* Multi-Protokol, *Modular Production System* (MPS), *Overall Equipment Effectiveness* (OEE).

(Halaman ini sengaja dikosongkan)

WEB-SCADA SYSTEM DESIGN FOR A MODULAR PRODUCTION SYSTEM USING MULTIPLE CONTROLLERS

Muhammad Reza Alfiano

ABSTRACT

The development of Industry 4.0 requires automation systems that can control production processes while providing integrated operational information to support data driven decision making. This study develops a multi-controller Web-SCADA system integrating an Omron CP2E-N30 DRA PLC via FINS UDP and a Siemens S7-1200 PLC via S7 TCP/IP in a Modular Production System, particularly the distribution and sorting stations. A Node.js based multi-protocol gateway connects the PLCs and transmits data to local and cloud based Web-SCADA systems. Operational data are presented through process monitoring, operating status, alarms, data logging, and Overall Equipment Effectiveness (OEE) calculations covering Availability, Performance, and Quality. Test results show average FINS UDP read delays of 1.7–2.0 ms and S7 TCP/IP read delays of 12–13 ms, while average gateway to cloud latency is 315.2 ms. Production testing yielded an average cycle time of 0.31 minutes per product as the Performance parameter. The system automatically calculated OEE values of 43.92%, 70.27%, and 85.25% in three scenarios, while the actual plant achieved 92.47% Performance, 100% Availability, 100% Quality, and 92.47% OEE. The system matched manual calculations with 0% error, demonstrating accurate supervision, data acquisition, multi-protocol integration, remote monitoring, and MPS performance evaluation.

Keywords : Web-SCADA, Multi Controller, Multi-Protocol Gateway, Modular Production System (MPS), Overall Equipment Effectiveness (OEE).

(Halaman ini sengaja dikosongkan)

DAFTAR ISI

HALAMAN JUDUL.....	i
LEMBAR PENGESAHAN.....	iii
PERNYATAAN BEBAS PLAGIAT.....	v
KATA PENGANTAR.....	vi
ABSTRAK.....	ix
ABSTRACT.....	xi
DAFTAR ISI	xiii
DAFTAR TABEL	xvii
DAFTAR GAMBAR.....	xix
DAFTAR NOTASI.....	xxiii
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	2
1.3 Tujuan Penelitian.....	2
1.4 Manfaat Penelitian.....	2
1.5 Batasan Penelitian	3
BAB 2 TINJAUAN PUSTAKA	5
2.1 Penelitian Terdahulu	5
2.2 Kajian Pustaka.....	9
2.2.1 <i>Modular Production System</i>	9
2.2.2 <i>Supervisory Control And Data Acquisition (SCADA)</i>	10
2.2.3 <i>Overall Equipment Effectiveness (OEE)</i>	10
2.2.4 <i>Latency Komunikasi Data</i>	12
2.3 <i>Software yang digunakan</i>	13
2.3.1 <i>CX Progammer version 9.7</i>	13
2.3.2 <i>TIA Portal V18</i>	14
2.3.3 <i>Node JS</i>	14
2.3.4 <i>PostgreSQL</i>	15
2.3.5 <i>Visual Studio Code</i>	16
2.3.6 <i>React JS</i>	17
2.3.7 <i>FUXA SCADA</i>	17

2.4	<i>Hardware yang digunakan</i>	18
2.4.1	<i>Programmable Logic Controller (PLC Siemens S7-1200)</i>	18
2.4.2	<i>Programmable Logic Controller (PLC Omron CP2E N30 DRA)</i>	19
2.4.3	<i>Sensor Proximity Kapasitif</i>	21
2.4.4	<i>Sensor Proximity Induktif</i>	22
2.4.5	<i>Sensor Limit Switch</i>	23
2.4.6	<i>Cylinder Pneumatic</i>	24
2.4.7	<i>Rotary Pneumatic</i>	24
2.4.8	<i>Pilot Lamp</i>	26
2.4.9	<i>Motor DC 12V</i>	27
2.4.10	<i>Sensor Infrared</i>	28
2.4.11	<i>Mini PC</i>	29
2.4.12	<i>Pneumatic Solenoid Valve 5/2</i>	30
2.4.13	<i>Vacuum Valve Pneumatic</i>	31
BAB 3 METODOLOGI PENELITIAN		33
3.1	<i>Tahapan Penelitian</i>	33
3.1.1	<i>State of The Art (Keterbaruan Dari Penelitian)</i>	36
3.2	<i>Tahap Identifikasi Awal</i>	36
3.2.1	<i>Studi Literatur</i>	37
3.2.2	<i>Identifikasi Masalah</i>	37
3.3	<i>Analisa Kebutuhan Sistem</i>	38
3.4	<i>Perancangan Sistem</i>	40
3.4.1	<i>Hardware</i>	41
3.4.2	<i>Gateway Connector</i>	42
3.4.3	<i>Web-SCADA (Local) dan Web Monitoring dan Analisis (Cloud)</i>	42
3.5	<i>Pembuatan Plant Distribution dan Sorting station</i>	43
3.6	<i>Wiring Diagram Sistem</i>	48
3.7	<i>Perencanaan dan Pembuatan Gateway Connector Pada Mini PC</i>	49
3.7.1	<i>Arsitektur Protokol FINS UDP pada Gateway</i>	51
3.7.1.1	<i>Parameter Koneksi FINS UDP</i>	51
3.7.1.2	<i>Memory Area (FINS UDP)</i>	52
3.7.1.3	<i>Mekanisme Komunikasi FINS UDP</i>	54
3.7.1.4	<i>Mekanisme Pembacaan dan Penulisan Data PLC Menggunakan Omron fins</i>	56

3.7.2	Arsitektur Protokol S7 TCP/IP pada <i>Gateway</i>	63
3.7.2.1	Parameter Koneksi S7 TCP/IP	63
3.7.2.2	Memory Area (S7 TCP/IP).....	64
3.7.2.3	Mekanisme Komunikasi S7 TCP/IP.....	65
3.7.2.4	Mekanisme Pembacaan dan Penulisan Data PLC Menggunakan NodeS7.....	68
3.7.3	<i>Gateway Connector</i> Pada Web <i>Monitoring</i> dan Analisis OEE (<i>Cloud</i>)...	75
3.7.4	<i>Gateway Connector</i> Pada Web SCADA (<i>Local</i>).....	77
3.8	Perancangan Sistem Web-SCADA	80
3.8.1	Perancangan Web <i>Monitoring</i> & Analisis OEE (<i>Cloud</i>).....	82
3.8.1.1	Use Case Diagram	86
3.8.2	Perancangan Web SCADA (<i>Local</i>)	87
3.9	Konsep Perhitungan OEE Pada <i>Distribution</i> dan <i>Sorting station</i>	88
BAB 4 HASIL DAN PEMBAHASAN.....		93
4.1	Hasil Perancangan <i>Hardware</i>	93
4.1.1	Hasil Perancangan Kontruksi Mekanik	93
4.2	Hasil Pengujian Sensor dan Aktuator.....	97
4.2.1	Hasil Pengujian Sensor Limit Switch	97
4.2.2	Hasil Pengujian Sensor Infrared	102
4.2.3	Hasil Pengujian Sensor <i>Proximity</i> Induktif	104
4.2.4	Hasil Pengujian Sensor <i>Proximity</i> Kapasitif.....	106
4.2.5	Hasil Pengujian Lampu Indikator Hijau, Merah dan Kuning.....	108
4.2.6	Hasil Pengujian Motor DC	113
4.2.7	Hasil Pengujian Selenoid Valve	114
4.2.8	Hasil Pengujian Sensor Photoelectric Fiber.....	118
4.3	Konfigurasi Jaringan Perangkat	120
4.3.1	Topologi Jaringan	120
4.3.2	Konfigurasi IP Address.....	121
4.3.3	Konfigurasi FINS UDP.....	122
4.3.4	Konfigurasi S7 TCP/IP	124
4.4	Pengujian <i>Gateway</i> Multi-protocol	125
4.4.1	Pengujian <i>Delay Read</i> Alamat PLC (<i>Local</i>).....	126
4.4.2	Pengujian <i>Delay</i> pada <i>Write</i> Alamat PLC (<i>Local</i>).....	129
4.4.3	Pengujian <i>Latency</i> Komunikasi <i>Gateway</i> ke <i>Cloud</i>	132

4.5	Hasil Perancangan <i>Software</i>	135
4.5.1	Progam dan Konfigurasi PLC Omron CP2E-N30 DRA.....	135
4.5.2	Progam dan Konfigurasi PLC Siemens S7 – 1200	140
4.5.3	Database PostgreSQL	147
4.5.4	Web SCADA (<i>Local</i>).....	149
4.5.5	Web SCADA (<i>Cloud</i>)	153
4.6	Pengujian OEE	164
4.6.1	Parameter Pengujian OEE.....	164
4.6.2	Pengujian <i>Cycle Time</i>	166
4.6.3	Skenario Pengujian dan Perhitungan OEE	169
4.6.4	Pengujian OEE Aktual pada <i>Plant</i>	177
BAB 5 KESIMPULAN DAN SARAN		179
5.1	Kesimpulan.....	179
5.2	Saran.....	180
DAFTAR PUSTAKA.....		183
LAMPIRAN BIODATA MAHASISWA		187

DAFTAR TABEL

Tabel 2.1 Penelitian Terdahulu	5
Tabel 2. 2 Standar TIPHON <i>Delay</i>	12
Tabel 2.3 Spesifikasi PLC Siemens S7-1200.....	19
Tabel 2.4 Spesifikasi PLC Omron CP2E N30 DRA	20
Tabel 2.5 Spesifikasi Sensor <i>Proximity</i> Kapasitif	22
Tabel 2.6 Spesifikasi Sensor <i>Proximity</i> Induktif	23
Tabel 2. 7 Spesifikasi <i>Cylinder Pneumatic</i>	24
Tabel 2.8 Spesifikasi <i>Rotary Pneumatic</i>	25
Tabel 2.9 Spesifikasi <i>Pilot Lamp</i>	26
Tabel 2.10 Spesifikasi Motor DC 12V	27
Tabel 2.11 Spesifikasi Sensor <i>Infrared</i>	28
Tabel 2.12 Spesifikasi Mini PC	29
Tabel 2.13 Spesifikasi <i>Pneumatic Selenoid Valve 5/2</i>	30
Tabel 3.1 Kebutuhan Sistem.....	39
Tabel 3.2 Parameter Koneksi FINS UDP pada <i>Gateway</i>	51
Tabel 3.3 Memory Area (FINS UDP)	52
Tabel 3. 4 Parameter pada <i>FINS Memory Area Read</i> dan <i>FINS Memory Area Write</i>	59
Tabel 3. 5 Parameter Pembacaan dan Penulisan pada Alamat D0	62
Tabel 3.6 Parameter Koneksi S7 TCP / IP pada <i>Gateway</i>	64
Tabel 3.7 Memory Area (S7 TCP/IP).....	65
Tabel 3. 8 Parameter pada <i>S7 Read Request</i> , <i>S7 Write Request</i> , dan <i>Respons PLC</i>	71
Tabel 3. 9 Parameter Pembacaan dan Penulisan pada Alamat MW0	74
Tabel 3.10 Sumber Data Pengujian OEE.....	89
Tabel 3.11 Contoh Data Pengujian	90
Tabel 4. 1 Hasil Pengujian Sensor Limit Switch	98
Tabel 4. 2 Pengujian Sensor Infrared.....	102
Tabel 4. 3 Jarak Deteksi Sensor Infrared	104
Tabel 4. 4 Pengujian Sensor <i>Proximity</i> Induktif.....	105
Tabel 4. 5 Jarak Deteksi Sensor <i>Proximity</i> Induktif	106
Tabel 4. 6 Pengujian Sensor <i>Proximity</i> Kapasitif.....	107
Tabel 4. 7 Jarak Deteksi <i>Proximity</i> Kapasitif	108
Tabel 4. 8 Pengujian Lampu Indikator Hijau, Merah dan Kuning	109

Tabel 4. 9 Pengujian Motor DC	113
Tabel 4. 10 Pengujian Selenoid Valve	115
Tabel 4. 11 Pengujian Sensor Photoelectric Fiber	119
Tabel 4. 12 Konfigurasi IP Address pada masing-masing perangkat	122
Tabel 4. 13 Konfigurasi FINS UDP	122
Tabel 4. 14 Alamat Memori FINS UDP	123
Tabel 4. 15 S7 TCP / IP	124
Tabel 4. 16 Alamat Memori S7/TCP/IP	125
Tabel 4. 17 Pengujian <i>Delay Read</i> Alamat PLC	128
Tabel 4. 18 Pengujian <i>Delay Write</i> Alamat PLC	131
Tabel 4. 19 Pengujian <i>Latency</i> Komunikasi <i>Gateway</i> ke <i>Cloud</i>	134
Tabel 4. 20 I/O List PLC Omron	136
Tabel 4. 21 Memory Address PLC Omron	136
Tabel 4. 22 I/O List PLC Siemens	141
Tabel 4. 23 Memory Address PLC Siemens	141
Tabel 4. 24 Kondisi Alarm	162
Tabel 4. 25 Parameter Data Pengujian OEE	165
Tabel 4. 26 Pengujian <i>Cycle Time</i> Menggunakan Stopwatch	167
Tabel 4. 27 Pengujian <i>Cycle Time</i> Menggunakan <i>Gateway</i>	167
Tabel 4. 28 Rata-rata Pengujian Cycle Time	168
Tabel 4. 29 Skenario Pengujian OEE	169
Tabel 4. 30 Kategori <i>Overall Equipment Effectiveness (OEE)</i>	170
Tabel 4. 31 Data untuk Perhitungan OEE	171
Tabel 4. 32 Hasil Perbandingan OEE Manual dengan OEE WEB-SCADA	175
Tabel 4. 33 Hasil Perbandingan <i>Availability</i> Manual dengan Web-SCADA	175
Tabel 4. 34 Hasil Perbandingan <i>Performance</i> Manual dengan Web-SCADA	176
Tabel 4. 35 Hasil Perbandingan <i>Quality</i> Manual dengan Web-SCADA	176
Tabel 4. 36 Data Pengujian Aktual pada <i>Plant</i>	177

DAFTAR GAMBAR

Gambar 2.1 Tampilan <i>Software CX Progammer</i>	13
Gambar 2.2 Tampilan <i>Software TIA PORTAL V18</i>	14
Gambar 2.3 <i>Node JS</i>	15
Gambar 2.4 <i>PostgreSQL</i>	16
Gambar 2.5 <i>Visual Studio Code</i>	16
Gambar 2.6 <i>React JS</i>	17
Gambar 2.7 Fuxa SCADA	18
Gambar 2.8 PLC Siemens S7-1200	19
Gambar 2.9 PLC Omron CP2E N30 DRA	20
Gambar 2.10 Sensor <i>Proximity</i> Kapasitif	21
Gambar 2.11 Sensor <i>Proximity</i> Induktif	22
Gambar 2.12 Sensor <i>Limit Switch</i>	23
Gambar 2.13 <i>Cylinder Pneumatic</i>	24
Gambar 2.14 <i>Rotary Pneumatic</i>	25
Gambar 2.15 <i>Pilot Lamp</i>	26
Gambar 2.16 Motor DC 12V	27
Gambar 2.17 Sensor <i>Infrared</i>	28
Gambar 2.18 Mini PC	29
Gambar 2.19 <i>Pneumatic Selenoid Valve 5/2</i>	30
Gambar 2.20 <i>Vacuum Valve Pneumatic</i>	32
Gambar 3.1 <i>Flow Chart</i> Tahapan Penelitian	33
Gambar 3.2 Diagram Blok Sistem.....	41
Gambar 3.3 Desain Tampak Depan	43
Gambar 3.4 Desain Tampak Depan	44
Gambar 3.5 Desain Tampak Isometri Kiri.....	44
Gambar 3.6 <i>Distribution</i> Dan <i>Sorting station</i>	45
Gambar 3.7 <i>Object</i> , A. <i>Good Product</i> , B. <i>Bad Product</i>	45
Gambar 3.8 <i>Flow Chart Distribution</i> dan <i>Sorting station</i>	46
Gambar 3.9 <i>Wiring Diagram</i>	49
Gambar 3.10 Arsitektur Sistem <i>Gateway</i>	50
Gambar 3. 11 <i>Flowchart</i> Komunikasi FINS UDP.....	55
Gambar 3. 12 Struktur FINS <i>Frame</i>	57

Gambar 3. 13 <i>Sequence Diagram</i> Mekanisme Pembacaan Data PLC Menggunakan Omron-fins.....	57
Gambar 3. 14 <i>Sequence Diagram</i> Mekanisme Penulisan Data PLC Menggunakan Omron-fins.....	58
Gambar 3. 15 Contoh Progam Pembacaan dan Penulisan Alamat D0 Menggunakan Omron-fins.....	61
Gambar 3. 16 <i>Flowchart</i> Komunikasi S7 TCP/IP	66
Gambar 3. 17 <i>Mekanisme TCP Three-Way Handshake antara Mini PC dan PLC Siemens S7-1200</i>	67
Gambar 3. 18 <i>Sequence Diagram</i> Mekanisme Pembacaan Data PLC Menggunakan NodeS7.....	69
Gambar 3. 19 <i>Sequence Diagram</i> Mekanisme Penulisan Data PLC Menggunakan NodeS7.....	70
Gambar 3. 20 Contoh Progam Pembacaan dan Penulisan Alamat MW0 Menggunakan NodeS7.....	73
Gambar 3.21 Struktur Sistem <i>Gateway Connector</i> Pada Web <i>Monitoring & Analisis OEE</i>	75
Gambar 3.22 Diagram Alur <i>Gateway Connector</i> Pada Web <i>Monitoring & Analisis OEE</i>	76
Gambar 3.23 Struktur Sistem <i>Gateway Connector</i> Pada Web SCADA (<i>Local</i>).....	78
Gambar 3.24 Diagram Alur <i>Gateway Connector</i> Pada Web SCADA (<i>Local</i>).....	79
Gambar 3.25 <i>Flowchart</i> Sistem Web-SCADA.....	81
Gambar 3.26 <i>Flowchart</i> Perancangan Web <i>Monitoring & Analisis OEE</i>	82
Gambar 3.27 Tampilan Halaman Login	83
Gambar 3.28 Tampilan Halaman Monitoring.....	84
Gambar 3.29 Tampilan Halaman Alarm Sistem.....	85
Gambar 3.30 Tampilan Halaman data <i>logger</i>	85
Gambar 3.31 Tampilan Halaman Analisis OEE	86
Gambar 3 32 Use Case Diagram Web <i>Monitoring & Analisis OEE</i>	86
Gambar 3.33 <i>Flowchart</i> Web SCADA (<i>Local</i>)	87
Gambar 3.34 Tampilan Halaman <i>Monitoring</i> dan Kontrol	88
Gambar 4. 1 Desain <i>Distribution</i> Dan <i>Sorting station</i>	94
Gambar 4. 2 Hasil Perancangan <i>Plant Modular Production System</i> Tampak Depan	94

Gambar 4. 3 Hasil Perancangan <i>Plant Modular Production System</i> Tampak Sisi Kanan	95
Gambar 4. 4 Hasil Perancangan <i>Plant Modular Production System</i> Tampak Sisi Kiri	95
Gambar 4. 5 Hasil Perancangan <i>Plant Modular Production System</i> (Tata letak Komponen)	96
Gambar 4. 6 Hasil Perancangan <i>Plant Modular Production System</i> (Pembagian <i>Station</i>)	96
Gambar 4. 7 Topologi Jaringan	121
Gambar 4. 8 Flowchart Pengujian <i>Delay Read</i>	127
Gambar 4. 9 Pengujian <i>Delay Read</i> pada Protocol FINS UDP	128
Gambar 4. 10 Pengujian <i>Delay Read</i> pada Protocol S7 TCP/IP	129
Gambar 4. 11 Flowchart Pengujian <i>Delay Write</i>	130
Gambar 4. 12 Pengujian <i>Delay Write</i> pada Protocol FINS UDP	132
Gambar 4. 13 Pengujian <i>Delay Write</i> pada Protocol S7 TCP/IP.....	132
Gambar 4. 14 Flowchart Pengujian <i>Latency Komunikasi Gateway ke Cloud</i>	133
Gambar 4. 15 Pengujian <i>Latency Komunikasi Gateway ke Cloud</i>	134
Gambar 4. 16 Konfigurasi Tipe PLC Omron	137
Gambar 4. 17 Konfigurasi Alamat IP PLC Omron	137
Gambar 4. 18 Program Ladder PLC Omron (1)	138
Gambar 4. 19 Program Ladder PLC Omron (2)	139
Gambar 4. 20 Program Ladder PLC Omron (2)	140
Gambar 4. 21 Konfigurasi Tipe PLC Siemens	142
Gambar 4. 22 Konfigurasi Alamat IP PLC Siemens	142
Gambar 4. 23 Program Ladder PLC Siemens (1)	143
Gambar 4. 24 Program Ladder PLC Siemens (2)	144
Gambar 4. 25 Program Ladder PLC Siemens (3)	145
Gambar 4. 26 Program Ladder PLC Siemens (4)	146
Gambar 4. 27 Tabel Users pada Database	147
Gambar 4. 28 Tabel Data OEE MPS pada Database	148
Gambar 4. 29 Tabel Data Alarm pada Database	148
Gambar 4. 30 Tabel Data Logger pada Database	149
Gambar 4. 31 Diagram Arsitektur Web-SCADA (<i>Local</i>) Berbasis FUXA	150
Gambar 4. 32 Koneksi dan Konfigurasi pada FUXA.....	150
Gambar 4. 33 <i>Mapping Address</i> FUXA	151

Gambar 4. 34 Progam <i>Gateway</i> Web-SCADA FUXA (<i>Local</i>)	152
Gambar 4. 35 <i>Interface</i> Web-SCADA (<i>Local</i>)	153
Gambar 4. 36 Diagram Arsitektur Web-SCADA (<i>Cloud</i>).....	154
Gambar 4. 37 Use Case Diagram peran Manager.....	155
Gambar 4. 38 Use Case Diagram peran Supervisor	155
Gambar 4. 39 Use Case Diagram peran Maintanance	156
Gambar 4. 40 Use Case Diagram peran Operator.....	156
Gambar 4. 41 Progam <i>Gateway</i> Web-SCADA (<i>Cloud</i>)	157
Gambar 4. 42 Progam <i>Backend Express</i> Web-SCADA (<i>Cloud</i>)	158
Gambar 4. 43 Interface Login Page	158
Gambar 4. 44 Interface Main Menu.....	159
Gambar 4. 45 Interface <i>Monitoring</i> Plant.....	160
Gambar 4. 46 Interface OEE Page.....	161
Gambar 4. 47 Interface Alarm Page	162
Gambar 4. 48 Interface Data Logger Page.....	163
Gambar 4. 49 <i>Flowchart</i> Pengujian <i>Cycle Time</i>	166
Gambar 4. 50 <i>Payload</i> Pengujian <i>Cycle Time</i> Pada <i>Gateway</i>	168
Gambar 4. 51 Analisis OEE Pada OEE <i>Page</i> Data ke-1	172
Gambar 4. 52 Analisis OEE Pada OEE <i>Page</i> Data ke-2	173
Gambar 4. 53 Analisis OEE Pada OEE <i>Page</i> Data ke-3	174
Gambar 4. 54 Analisis OEE Pada OEE <i>Page</i> Data Aktual.....	178

DAFTAR NOTASI

OEE	= <i>Overall Equipment Effectiveness</i>
A	= <i>Availability</i>
P	= <i>Performance</i>
Q	= <i>Quality</i>
MPS	= <i>Modular Production System</i>
SCADA	= <i>Supervisory Control and Data Acquisition</i>
PLC	= <i>Programmable Logic Controller</i>
FINS	= <i>Factory Interface Network Service</i>
UDP	= <i>User Datagram Protocol</i>
TCP/IP	= <i>Transmission Control Protocol / Internet Protocol</i>
API	= <i>Application Programming Interface</i>
HTTP	= <i>Hypertext Transfer Protocol</i>
PDU	= <i>Protocol Data Unit</i>
IP	= <i>Internet Protocol</i>

(Halaman ini sengaja dikosongkan)

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Perkembangan industri 4.0 mendorong kebutuhan akan sistem otomasi yang tidak hanya berfokus pada pelaksanaan proses produksi secara otomatis, tetapi juga mampu menyajikan informasi yang akurat, terintegrasi, dan tersedia secara *real-time* untuk mendukung pengambilan keputusan berbasis data. Salah satu teknologi yang banyak dimanfaatkan untuk tujuan tersebut adalah *Supervisory Control and Data Acquisition* (SCADA). Pemanfaatan teknologi jaringan dan *Internet of Things* turut mendorong peningkatan fleksibilitas sistem serta memungkinkan pengawasan jarak jauh. Sehingga proses pemantauan dan pengendalian dapat dilakukan melalui antarmuka berbasis web tanpa bergantung pada lokasi maupun jenis perangkat tertentu (Karyatanti, 2024). Pendekatan ini membuat aktivitas supervisi menjadi lebih fleksibel dan efisien, sekaligus sesuai dengan kebutuhan industri modern yang semakin terintegrasi.

Dalam penerapan sistem otomasi industri modern, salah satu tantangan utama yang dihadapi adalah proses integrasi antar-*Programmable Logic Controller* (PLC) dari berbagai produsen yang memiliki karakteristik dan protokol komunikasi berbeda. Ketidaksamaan arsitektur dan sistem kendali pada setiap jenis PLC sering menimbulkan kesulitan dalam melakukan pertukaran data secara sinkron dan efisien. Kondisi ini berpotensi menyebabkan keterlambatan dalam proses pemantauan serta mendorong dilakukannya pencatatan data secara manual pada beberapa lini produksi. Sebagai solusi, beberapa penelitian menyarankan penggunaan sistem terpusat berbasis *Supervisory Control and Data Acquisition* (SCADA) untuk mengintegrasikan komunikasi antarperangkat kendali tersebut. Melalui pendekatan ini, berbagai PLC dengan protokol yang berbeda dapat saling berkomunikasi dan bertukar data dalam satu platform yang terkoordinasi, sehingga sistem otomasi menjadi lebih efisien (Ardaneshwara, 2024).

Selain aspek komunikasi, industri manufaktur juga membutuhkan metode evaluasi kinerja yang terstruktur dan dapat diukur secara objektif. *Overall Equipment Effectiveness* (OEE) merupakan salah satu indikator yang umum digunakan untuk menilai kinerja mesin berdasarkan *Availability*, *Performance*, dan *Quality* (Siburian & Arvianto, 2025). Integrasi Web-SCADA dengan perhitungan OEE memungkinkan pemantauan dan analisis kinerja mesin dilakukan secara otomatis, karena data diperoleh

langsung dari perangkat lapangan melalui *gateway* multi-protokol. Dalam penelitian ini, pendekatan tersebut diterapkan pada *Modular Production System* (MPS), khususnya pada *Distribution* dan *sorting station*, sebagai representasi sistem otomasi industri skala kecil. Penelitian ini bertujuan merancang prototipe sistem otomasi berbasis Web-SCADA yang terintegrasi dengan *gateway* multi-protokol dan perhitungan OEE, sehingga dapat digunakan sebagai pengembangan otomasi industri.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah di uraikan, dirumuskan beberapa rumusan masalah penelitian sebagai berikut :

1. Bagaimana merancang sistem Web-SCADA yang berfungsi untuk melakukan pemantauan dan pengendalian terhadap proses pada *Distribution* dan *Sorting station*?
2. Bagaimana membangun *gateway* multi-protokol yang mampu menghubungkan komunikasi antara PLC Omron CP2E N30 DRA (FINS UDP) dan PLC Siemens S7 1200 (S7 TCP/IP) agar data dapat diterima dan ditampilkan dalam sistem Web-SCADA?
3. Bagaimana memvisualisasikan data operasional dan nilai OEE ke dalam Web-SCADA sehingga operator dan manajemen dapat melakukan *Monitoring* dan analisis?

1.3 Tujuan Penelitian

Adapun tujuan yang ingin dicapai pada tugas akhir ini adalah :

1. Merancang sistem Web-SCADA untuk melakukan proses pemantauan dan pengendalian pada *Distribution* dan *sorting station*.
2. Membangun *gateway* multi-protokol yang mampu mengintegrasikan berbagai media dan protokol komunikasi PLC sehingga data dapat disajikan Web-SCADA.
3. Membangun Web-SCADA untuk fungsi supervisi, akuisisi data, serta visualisasi OEE sebagai evaluasi kinerja mesin.

1.4 Manfaat Penelitian

Dengan adanya penelitian Tugas Akhir ini diharapkan dapat memberikan manfaat diantaranya adalah :

1. Bagi penulis : Penelitian ini sebagai sarana untuk memperdalam pemahaman dan penerapan Web-SCADA, penggunaan *gateway* multi-protokol, dan perhitungan OEE sebagai alat ukur performa mesin.
2. Bagi PPNS : Menjadi referensi tambahan untuk penelitian di bidang industri dan evaluasi peforma mesin, serta mendukung pengembangan ilmu di lingkungan akademik.
3. Bagi industri : Penelitian ini diharapkan dapat memberikan pendekatan sistem integrasi pada multi protokol yang memungkinkan integrasi antara protokol PLC, sehingga sistem Web-SCADA dapat berjalan lebih stabil dan konsisten. Selain itu, sistem ini dapat membantu industri dalam memantau kinerja mesin secara terpusat serta mendukung pengambilan Keputusan berbasis data melalui penerapan OEE.

1.5 Batasan Penelitian

Penelitian ini memiliki Batasan masalah terkait beberapa parameter yang diuji maupun diperoleh, yang tidak dapat dibandingkan dengan penelitian lain karena perbedaan batasan masalah.

1. Penelitian Tugas Akhir di implementasikan pada *Distribution* dan *Sorting Station* dengan mengacu pada konsep *Modular Production System* (MPS).
2. Sistem Web-SCADA dimanfaatkan sebagai media untuk pemantauan dan pengendalian proses produksi.
3. *Gateway* multi-protokol diterapkan untuk menghubungkan PLC Omron CP2E N30 DRA (FINS UDP) dan PLC Siemens S7 1200 (S7 TCP/IP).
4. Penelitian ini hanya berfokus pada perancangan arsitektur sistem, tidak membahas pada mekanikal alat.
5. OEE digunakan sebagai evaluasi kinerja mesin, bukan sebagai metode pengendalian atau optimasi proses.
6. Penelitian ini tidak membahas aspek keamanan sistem (*security*) pada *website* berbasis *cloud*, seperti enkripsi data, autentikasi tingkat lanjut, *firewall*, maupun mekanisme perlindungan terhadap serangan siber.

(Halaman ini sengaja dikosongkan)

BAB 2

TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Bagian ini akan membahas penelitian-penelitian terdahulu yang dijadikan referensi dalam pelaksanaan tugas akhir ini.

Tabel 2.1 Penelitian Terdahulu

Nama dan Tahun Penelitian	Judul Penelitian	Hasil Penelitian
(Ardaneshwara, 2024b)	Optimisasi SCADA di MPS dengan FINS UDP vs Modbus TCP via Node-RED	Dalam hasil penelitiannya, Akhtar Muhammad Ardaneshwara (2024) mengembangkan sistem SCADA berbasis Node-RED yang ditujukan untuk pemantauan dan pengendalian <i>Modular Production System</i> (MPS) pada skala laboratorium. Sistem tersebut telah mengintegrasikan PLC Siemens S7-300 dan Omron CP1L dengan memanfaatkan dua protokol komunikasi, yakni FINS UDP dan Modbus TCP. Evaluasi dilakukan dengan mengukur performa kedua protokol melalui parameter <i>delay</i> dan <i>throughput</i> pada jaringan Ethernet dan WiFi. Hasilnya menunjukkan FINS UDP memiliki performa lebih baik dibandingkan Modbus TCP. Data yang divisualisasikan melalui <i>dashboard</i> platform Node-RED yang terhubung ke InfluxDB.
(ADAM RISKY MUHARAM, 2025)	Integrasi PLC Multi Vendor dalam Pengendalian <i>Real-</i>	Dalam penelitiannya, Adam Risky Maulana (2025) merancang sistem kontrol industri skala laboratorium

Nama dan Tahun Penelitian	Judul Penelitian	Hasil Penelitian
	<i>time</i> dengan HMI dan VSD	yang mengintegrasikan dua PLC berbeda Siemens S7-1200 dan PLC Schneider TM221CE16 yang dapat menggunakan komunikasi protokol Modbus TCP. Sistem ini turut melibatkan perangkat HMI dan VSD yang terkoneksi dalam jaringan lokal untuk menunjang operasi dua <i>plant</i> secara <i>real-time</i> . Evaluasi performa menunjukkan komunikasi antar-PLC berjalan stabil dan sinkron sesuai rancangan.
(Idris et al., 2023)	Sistem SCADA berbasis <i>Internet of Things</i>	Dalam hasil penelitiannya, penulis mengembangkan sistem SCADA berbasis web dengan memanfaatkan PLC Schneider TM241CE24R dan modul ESP32 sebagai media komunikasi untuk pemantauan dan pengendalian perangkat secara jarak jauh. Sistem tersebut menggunakan protokol Modbus RTU sebagai sarana komunikasi antara PLC dan ESP32, sedangkan pengiriman data ke sisi <i>server</i> dilakukan melalui protokol MQTT dengan dukungan platform berbasis Node.js. Pengujian yang difokuskan kepada performa komunikasi data antara PLC, ESP32, dan web, meliputi pengukuran <i>delay</i> ,

Nama dan Tahun Penelitian	Judul Penelitian	Hasil Penelitian
		packet loss, serta kestabilan sistem pada berbagai kondisi jaringan.
(MUHAMMAD NAUFAL HENDRI, 2025)	INTEGRASI SCADA SMARTICS MELALUI LOCALHOST PADA <i>SORTING STATION</i> BERBASIS MODBUS TCP/IP	Hasil penelitian menunjukkan bahwa sistem SCADA SmartICS berhasil diimplementasikan secara integrasi untuk mendukung tahap proses pemantauan dan pengendalian pada mesin sorting yang berbasis PLC Schneider M221. Sistem mampu mengelola proses pemindahan objek secara otomatis dan manual dengan memanfaatkan sensor warna, sensor <i>proximity</i>, serta aktuator berupa lengan robot enam sumbu. Seluruh proses yang dikendalikan melalui komunikasi Modbus TCP/IP yang terhubung melalui jaringan WLAN antara PLC dan sistem SCADA. Berdasarkan hasil pengujian, sistem telah mampu menjalankan fungsi <i>Monitoring</i> dan kontrol secara otomatis dengan tingkat keberhasilan mencapai 100% pada mode operasi otomatis. Pengujian terhadap akurasi gerakan lengan robot menunjukkan nilai rata-rata kesalahan sudut sebesar 1,53°, yang menunjukkan tingkat presisi yang cukup baik dalam proses pemindahan objek. Selain itu, pengujian komunikasi data

Nama dan Tahun Penelitian	Judul Penelitian	Hasil Penelitian
		menunjukkan rata-rata waktu tunda (<i>delay</i>) sebesar 496,5 ms, yang disebabkan oleh karena keterbatasan kualitas jaringan WLAN yang digunakan. Meskipun demikian, sistem tetap mampu beroperasi secara stabil dan konsisten dalam menjalankan fungsi <i>Monitoring</i> dan kontrol. Hasil ini menunjukkan bahwa integrasi SCADA SmartICS dengan sistem sorting berbasis PLC dapat berjalan dengan baik serta sudah layak dan bisa diterapkan sebagai solusi otomasi industri berbasis jaringan.
(Asita et al., 2023)	Pengiriman Data Realtime PLC ke ERP Odoo dengan Dilengkapi Analisa OEE	Dalam penelitian ini, dilakukan pengembangan sistem yang telah dirancang yaitu mengintegrasikan teknologi informasi dan teknologi operasional dan analisis otomatis untuk mendukung peningkatan kinerja proses produksi. Integrasi tersebut telah diwujudkan melalui penerapan <i>softwareconnector</i> yang menghubungkan PLC dengan sistem ERP, sehingga data operasional dapat diproses dan dianalisis secara terpusat. Sistem dirancang untuk mendukung komunikasi melalui beberapa protokol industri, seperti

Nama dan Tahun Penelitian	Judul Penelitian	Hasil Penelitian
		Modbus TCP/IP, Ethernet/IP, dan Profinet. Implementasi dilakukan pada plant sortir botol yang dikendalikan oleh PLC Schneider, di mana data hasil produksi dikirim secara <i>real-time</i> ke sistem ERP untuk dianalisis menggunakan metode <i>Overall Equipment Effectiveness</i> (OEE). Hasil dari pengujian telah menunjukkan bahwa sistem mampu bekerja dengan baik dan memberikan tingkat akurasi yang tinggi, sehingga dapat mendukung secara otomatis proses pengambilan keputusan yang berbasis data secara lebih efektif.

2.2 Kajian Pustaka

Bagian ini akan mengulas teori-teori yang menjadi acuan dalam pelaksanaan tugas akhir ini.

2.2.1 *Modular Production System*

Pembelajaran dan praktikum otomasi industri berbasis *Programmable Logic Controller* (PLC) dapat dilakukan dengan memanfaatkan *Modular Production System* (MPS) yang dikembangkan oleh produsen seperti FESTO. Sistem ini dirancang sebagai media pembelajaran yang menampilkan simulasi proses industri secara terstruktur, sehingga peserta didik dapat memahami cara kerja sistem otomasi dengan tingkat keamanan yang lebih tinggi. Dengan kompleksitas sistem yang relatif sederhana, MPS tetap mampu merepresentasikan kondisi industri nyata serta membantu meminimalkan risiko kecelakaan kerja yang umumnya terjadi pada tahap perancangan sistem otomasi (Abizari et al., 2025).

Dalam implementasinya, MPS memiliki beberapa jenis modul dengan fungsi yang berbeda, di antaranya *Distribution Station*, *Process Station*, dan *Sorting station* yang digunakan pada proyek akhir ini. Modul distribusi berfungsi memindahkan material menggunakan sistem vacuum, modul proses menjalankan aktivitas pengeboran, sedangkan modul sortir melakukan identifikasi material logam dan non-logam untuk selanjutnya dipindahkan sesuai kriteria tertentu. Melalui modul distribusi, mahasiswa dapat mempelajari penerapan sensor dan aktuator seperti *fiber optic sensor*, *limit switch*, dan vacuum yang umum dijumpai pada mesin produksi manufaktur, sehingga pembelajaran yang diperoleh menjadi lebih aplikatif dan relevan dengan kebutuhan industri (Abizari et al., 2025).

2.2.2 Supervisory Control And Data Acquisition (SCADA)

Sistem *Supervisory Control and Data Acquisition* (SCADA) merupakan sistem yang digunakan untuk melakukan pengawasan, pengendalian, serta pengumpulan data dari suatu proses industri secara terpusat. SCADA memungkinkan operator untuk memantau kondisi proses, mengendalikan peralatan, serta memperoleh informasi operasional secara *real-time* melalui antarmuka yang terintegrasi. Sistem ini banyak diterapkan pada berbagai sektor industri seperti manufaktur, energi, dan utilitas karena mampu meningkatkan efisiensi, keandalan, serta keselamatan proses produksi (Karyatanti, 2024).

Perkembangan teknologi pada era industri 4.0 mendorong pemanfaatan sistem otomasi yang mampu meningkatkan efisiensi dan efektivitas proses industri. Salah satu teknologi yang berperan penting dalam hal ini adalah sistem *Supervisory Control and Data Acquisition* (SCADA), yang berfungsi untuk memantau, mengendalikan, serta mengumpulkan data dari proses industri secara terpusat. Dengan dukungan teknologi *Internet of Things* (IoT), sistem SCADA dapat diintegrasikan dengan jaringan komunikasi sehingga memungkinkan pemantauan dan pengendalian dilakukan secara jarak jauh. Penerapan sistem ini memberikan kemudahan dalam pengawasan proses, meningkatkan keandalan operasional, serta mendukung pengambilan keputusan berbasis data secara real time (Idris et al., 2023).

2.2.3 Overall Equipment Effectiveness (OEE)

OEE atau *Overall Equipment Effectiveness* merupakan salah satu metode yang digunakan untuk menilai efektivitas dan produktivitas mesin dalam kegiatan produksi. Melalui pendekatan ini, perusahaan dapat menilai tingkat optimalisasi penggunaan mesin

serta mengidentifikasi kehilangan waktu yang terjadi akibat gangguan, proses perawatan, maupun penurunan kecepatan kerja. Nilai OEE sendiri terdiri atas tiga elemen utama, yaitu *Availability* (ketersediaan mesin), *Performance* (tingkat kecepatan kerja), dan *Quality* (kualitas produk). Dengan mempertimbangkan ketiga elemen tersebut, perusahaan dapat memahami efisiensi proses produksi secara komprehensif sekaligus menjadikannya dasar evaluasi peningkatan produktivitas (Wibowo, 2023).

Selain sebagai alat ukur, OEE juga berperan sebagai indikator yang memberikan gambaran menyeluruh mengenai performa mesin dalam jangka waktu tertentu seperti per shift, harian, atau bulanan. Melalui hasil analisis OEE, perusahaan dapat mengidentifikasi mesin dengan performa rendah dan mengetahui bagian proses produksi yang masih dapat ditingkatkan efektivitasnya. Hasil evaluasi ini kemudian digunakan sebagai acuan dalam menentukan prioritas perbaikan agar aktivitas produksi berjalan lebih efisien. Dengan demikian, OEE tidak hanya berfungsi sebagai pengukur efektivitas, tetapi juga menjadi panduan untuk pelaksanaan perbaikan berkelanjutan di lini produksi (Wibowo, 2023)

Secara umum, *Overall Equipment Effectiveness* (OEE), dihitung menggunakan komponen utama, yaitu *Availability*, *Performance*, dan *Quality*. Ketiga komponen ini saling berkaitan dan menggambarkan efektivitas mesin secara menyeluruh. Rumus umum OEE dapat dituliskan sebagai berikut :

$$OEE = Availability \times Performance \times Quality \times 100\% \quad (2.1)$$

a. *Availability* (A)

Pada analisis OEE, *Availability* merepresentasikan kesiapan mesin dalam mendukung proses produksi sesuai jadwal, termasuk waktu henti akibat perawatan maupun gangguan operasional (Kirana & Widiasih, 2024). Untuk menghitung nilai *Availability* yaitu :

$$A \% = \frac{Operating Time}{Loading Time} \times 100 \% \quad (2.2)$$

b. *Performance Rate* (P)

Pada analisis OEE, *Performance* menunjukkan kemampuan mesin menjalankan operasi pada kecepatan ideal yang diharapkan berdasarkan standar produksi (Kirana & Widiasih, 2024). Untuk menghitung nilai *Performance* yaitu :

$$P \% = \frac{Cycle Time \times Actual Output}{Operating Time} \times 100 \% \quad (2.3)$$

c. *Quality Rate* (Q)

Pada analisis OEE, *Quality* berfungsi sebagai ukuran efektivitas proses produksi dalam menghasilkan produk yang sesuai dengan spesifikasi, dengan menilai rasio

antara jumlah produk layak terhadap total hasil produksi (Kirana & Widiasih, 2024). Untuk menghitung nilai *Performance* yaitu :

$$Q \% = \frac{\text{Acceptable Product}}{\text{Total Production}} \times 100 \% \quad (2.4)$$

Dengan menggunakan rumus OEE, efektivitas mesin dapat dinilai secara periodik, baik per shift, harian, maupun bulanan. Analisis ini tidak hanya menampilkan tingkat kinerja mesin, tetapi juga mengidentifikasi mesin yang belum mencapai performa optimal serta bagian yang masih dapat ditingkatkan untuk meningkatkan efisiensi proses produksi.

2.2.4 Latency Komunikasi Data

Latency atau *delay* merupakan waktu yang dibutuhkan data untuk menempuh perjalanan dari perangkat sumber menuju perangkat tujuan dalam suatu jaringan komunikasi. Pengukuran *delay* dilakukan untuk mengetahui lama waktu yang diperlukan oleh satu paket data hingga diterima pada sisi tujuan. Nilai *delay* umumnya dinyatakan dalam satuan milidetik (ms), dan semakin kecil nilainya maka semakin cepat respons komunikasi yang dihasilkan oleh sistem (Alzi dan Haeruddin, 2022).

Pada penelitian ini, standar TIPHON digunakan sebagai acuan untuk menilai kualitas *latency* komunikasi antara *gateway* dan Web-SCADA berbasis *cloud*. Penggunaan acuan tersebut bertujuan untuk menentukan kategori waktu respons berdasarkan hasil pengujian yang diperoleh. Kategori nilai *delay* berdasarkan standar TIPHON ditunjukkan pada Tabel 2.2.

Tabel 2. 2 Standar TIPHON *Delay*

Kategori	Besar <i>Delay</i> (ms)
Sangat Bagus	<150
Bagus	150 s/d 300
Sedang	300 s/d 450
Buruk	>450

(Sumber : (Alzi dan Haeruddin, 2022)

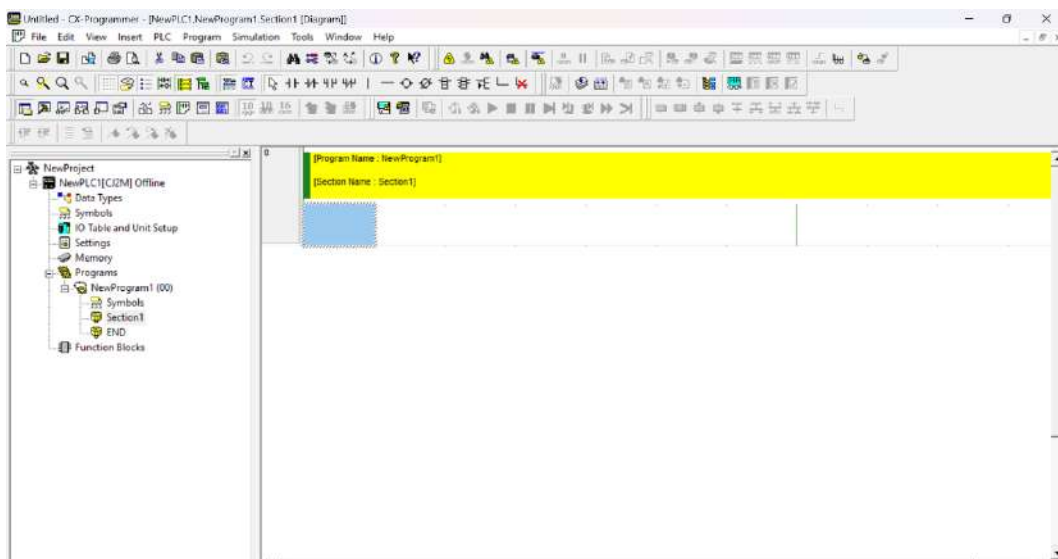
Berdasarkan Tabel 2.2, nilai *delay* kurang dari 150 ms termasuk kategori sangat bagus, nilai 150–300 ms termasuk kategori bagus, nilai 300–450 ms termasuk kategori sedang, sedangkan nilai lebih dari 450 ms termasuk kategori buruk. Kategori tersebut digunakan sebagai acuan dalam menganalisis hasil pengujian *latency* komunikasi pada bagian hasil dan pembahasan.

2.3 *Software yang digunakan*

Dalam pelaksanaan tugas akhir ini, beberapa *software* akan digunakan untuk mendukung desain, implementasi, dan pengujian sistem. Berikut merupakan *software* yang akan digunakan dalam pelaksanaan tugas akhir ini:

2.3.1 *CX Programmer version 9.7*

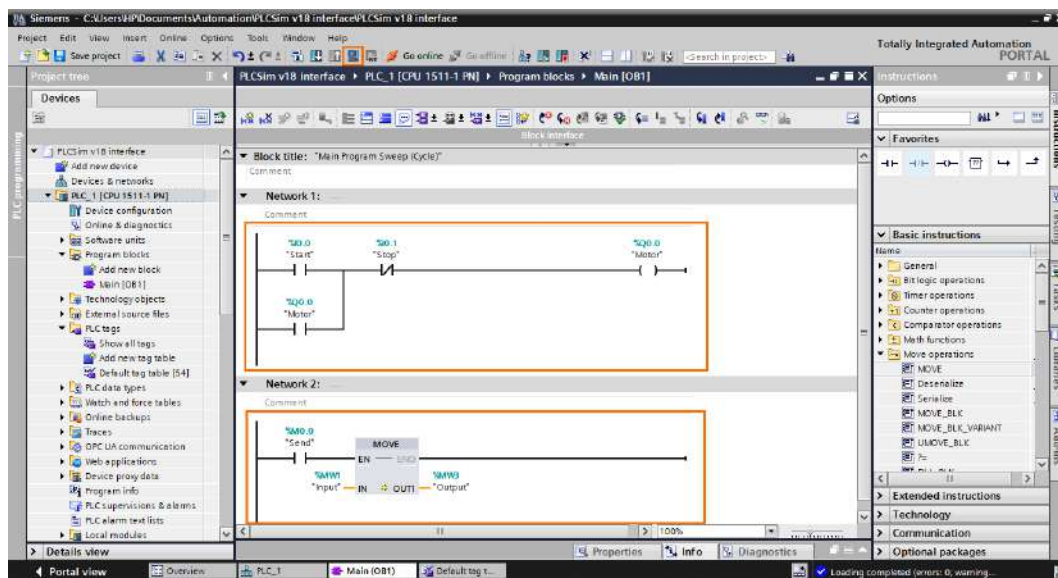
CX-Programmer merupakan perangkat lunak yang dikembangkan oleh Omron Automation untuk melakukan pemrograman, pemantauan, serta simulasi pada berbagai jenis PLC Omron, di antaranya seri CP, CJ, CS, dan CQM1. Perangkat ini menggunakan format pemrograman *ladder diagram* yang telah menjadi standar dalam sistem otomasi industri, sehingga memungkinkan perancangan logika kontrol yang kompleks melalui antarmuka yang intuitif. Selain fungsi dasar pemrograman, perangkat lunak ini juga dilengkapi fitur pemantauan online dan debugging *real-time*, yang mempermudah pengguna dalam melakukan analisis serta pengujian program secara langsung tanpa menghentikan sistem. Dengan kemampuan tersebut, CX-Programmer banyak digunakan dalam berbagai aplikasi otomasi seperti kontrol proses, sistem konveyor, dan sistem modular termasuk *Distribution* dan *sorting station* yang membutuhkan kestabilan dan kecepatan respon tinggi. Penggunaan perangkat lunak ini juga terbukti dapat mempercepat proses pengembangan logika kontrol serta mendukung integrasi dengan sistem SCADA untuk kebutuhan pemantauan data secara langsung (Kholif, 2025).



Gambar 2.1 Tampilan *Software CX Programmer*
(Sumber: Dokumentasi Pribadi, 2026)

2.3.2 TIA Portal V18

Totally Integrated Automation (TIA) Portal adalah perangkat lunak otomasi terpadu yang dikembangkan oleh Siemens untuk mendukung proses pemrograman dan konfigurasi berbagai PLC seri S7, termasuk S7-1200. Aplikasi ini menyatukan fungsi pemrograman PLC, pengaturan HMI, serta manajemen jaringan industri dalam satu platform yang terintegrasi, sehingga mempercepat dan mempermudah proses pengembangan sistem kontrol. Dengan adanya integrasi ini, pengguna tidak perlu lagi menggunakan beberapa perangkat lunak terpisah. TIA Portal juga mendukung beragam bahasa pemrograman standar IEC 61131-3, seperti *Ladder Diagram*, *Function Block Diagram*, dan *Structured Text*, yang memberikan fleksibilitas dalam perancangan logika kontrol. Selain itu, kemampuan sinkronisasi otomatis antara tag PLC dan HMI memungkinkan akses data secara *real-time* tanpa konfigurasi tambahan. Fitur simulasi dan pemantauan online yang tersedia di dalamnya menjadikan TIA Portal efektif digunakan pada sistem otomasi industri modern, termasuk pada *Distribution* dan *sorting station* yang memerlukan kecepatan serta stabilitas tinggi. Perangkat ini juga telah terbukti efisien dalam mengintegrasikan komunikasi antar-PLC dan sistem SCADA melalui dukungan protokol Ethernet dan OPC (Muchtar et al., 2024).



Gambar 2.2 Tampilan Software TIA PORTAL V18
(Sumber: <https://share.google/UbMFJchsp7l9PwhSj>)

2.3.3 Node JS

Node.js adalah lingkungan *runtime* berbasis JavaScript yang dijalankan di sisi *server* dan dikembangkan dengan konsep *event-driven* serta mekanisme *non-blocking I/O*. Desain tersebut membuat Node.js mampu memproses berbagai permintaan secara

bersamaan tanpa mengganggu performa sistem. Teknologi ini banyak dimanfaatkan dalam pengembangan sistem pemantauan berbasis web karena kemampuannya dalam menangani komunikasi data dengan cepat dan efisien. Penelitian ini menunjukkan bahwa Node.js digunakan sebagai *backend server* pada sistem pemantauan dan prediksi kualitas air sungai berbasis web. Dalam implementasinya, Node.js berfungsi untuk mengirimkan data sensor secara *real-time* ke antarmuka pengguna, sehingga proses pengumpulan dan analisis data dapat dilakukan secara langsung. Hasil penelitian tersebut juga menjelaskan bahwa penggunaan Node.js mampu meningkatkan efisiensi transfer data serta mempercepat proses komunikasi pada sistem pemantauan berbasis jaringan (Samudra et al., 2025).



Gambar 2.3 Node JS
(Sumber: <https://en.wikipedia.org/wiki/Node.js>)

2.3.4 PostgreSQL

PostgreSQL merupakan sistem manajemen basis data relasional (*Relational Database Management System / RDBMS*) bersifat *open-source* yang memiliki kemampuan tinggi dalam mengelola transaksi data berukuran besar secara terstruktur dan konsisten. Platform ini dilengkapi dengan beragam fitur seperti *concurrent access*, *data integrity*, serta eksekusi kueri yang efisien, menjadikannya pilihan andal dalam pengembangan aplikasi berbasis web. Dalam penerapannya pada sistem pemantauan daring, PostgreSQL kerap digunakan karena kestabilannya, fleksibilitas terhadap berbagai tipe data kompleks, serta kemampuannya berintegrasi dengan beragam bahasa pemrograman modern. Berdasarkan penelitian yang dilakukan oleh Fitriani dan Alfarisi (2025), PostgreSQL diimplementasikan sebagai basis data utama pada sistem *Monitoring* kebutuhan masyarakat berbasis web dan *mobile*, yang dikembangkan menggunakan *framework* Next.js dan React Native. Penelitian tersebut menunjukkan bahwa integrasi

PostgreSQL dengan arsitektur aplikasi modern mampu meningkatkan efisiensi penyimpanan data, kestabilan sistem, serta mendukung akses informasi secara *real-time* (Fitriani & Alfaris, 2025).



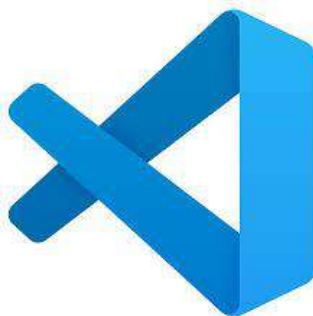
Gambar 2.4 *PostgreSQL*

(Sumber: <https://www.optimadata.nl/en/nieuws/postgresql-16-released/>)

2.3.5 *Visual Studio Code*

Visual Studio Code (VS Code) adalah perangkat lunak penyunting kode sumber yang dikembangkan oleh Microsoft dan dirancang untuk mendukung berbagai bahasa pemrograman seperti JavaScript, PHP, Python, dan C++. Lingkungan pengembangan ini menawarkan antarmuka yang ringan, intuitif, serta dilengkapi berbagai fitur seperti *syntax highlighting*, *auto-completion*, dan *integrated terminal* yang berfungsi untuk mempermudah proses penulisan dan pengujian kode program.

Dalam pengembangan aplikasi berbasis web, VS Code banyak digunakan karena memiliki kompatibilitas tinggi dengan framework modern serta kemudahan dalam pemasangan ekstensi tambahan seperti *Live Server*, *Prettier*, dan *Git Control*. Fitur-fitur tersebut memberikan fleksibilitas bagi pengembang untuk melakukan integrasi dan debugging dengan lebih cepat (Dian et al., 2025).



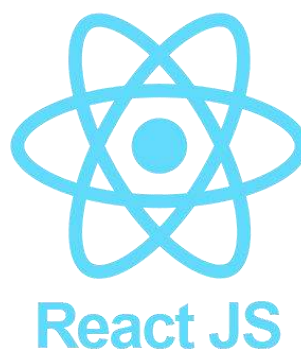
Gambar 2.5 *Visual Studio Code*

(Sumber: <https://share.google/4WJPhPkChBefWvpzK>)

2.3.6 React JS

React JS merupakan sebuah library JavaScript yang digunakan untuk membangun antarmuka pengguna (*user interface*) pada aplikasi berbasis web. React dirancang untuk mempermudah pengembang dalam membuat tampilan web yang dinamis dan interaktif melalui konsep *component-based architecture*, di mana setiap bagian tampilan dapat dikembangkan secara terpisah dan digunakan kembali di berbagai bagian aplikasi. Pendekatan ini menjadikan proses pengembangan lebih efisien dan terstruktur, terutama untuk proyek berskala besar. Selain itu, React JS juga menggunakan teknologi *virtual DOM* yang memungkinkan pembaruan elemen halaman dilakukan dengan cepat tanpa perlu memuat ulang seluruh tampilan, sehingga meningkatkan kinerja dan responsivitas aplikasi (Iswari & Nasution, 2021).

Selain itu, React banyak dipilih dalam pengembangan aplikasi web karena kemampuannya dalam meningkatkan performa akses data dan antarmuka melalui sistem *rendering* yang efisien. Penerapan React pada *Frontend* sebuah sistem informasi dapat mempercepat proses tampilan data, meminimalkan waktu tunggu pengguna, dan memudahkan pengembang dalam menggunakan paket-paket pendukung lainnya untuk memperluas fungsionalitas aplikasi. Hal ini membuat React cocok digunakan pada berbagai jenis aplikasi web yang membutuhkan antarmuka yang interaktif, cepat, dan mudah dipelihara (Iswari & Nasution, 2021).



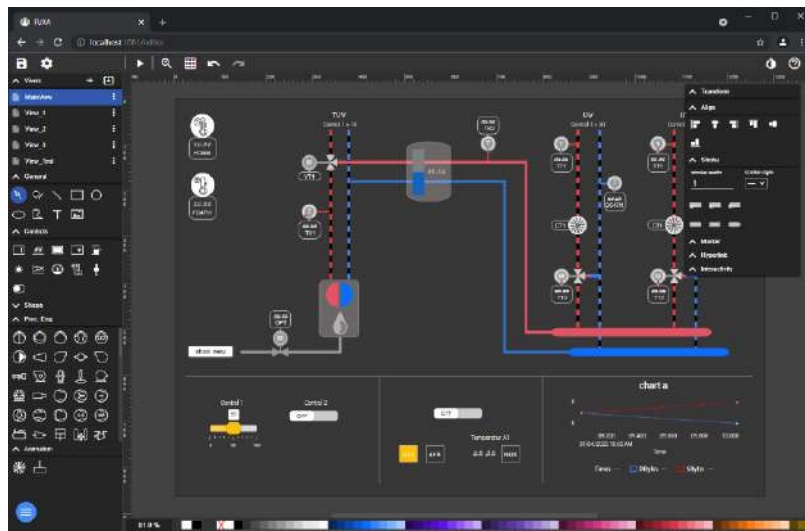
Gambar 2.6 *React JS*
(Sumber: <https://share.google/n0JI1EkpOwLUSnGK0>)

2.3.7 FUXA SCADA

FUXA merupakan perangkat lunak SCADA/HMI berbasis web yang digunakan untuk melakukan visualisasi, monitoring, dan kontrol proses industri secara *real-time*.

FUXA dikembangkan sebagai perangkat lunak open-source yang berjalan pada platform *server* dan dapat diakses melalui antarmuka web tanpa memerlukan instalasi aplikasi klien khusus. Sistem ini mendukung berbagai protokol komunikasi industri, seperti Modbus TCP, Siemens S7, OPC-UA, dan MQTT, sehingga dapat diintegrasikan dengan berbagai jenis PLC dan perangkat otomasi.

Dalam implementasinya, FUXA berperan sebagai SCADA lokal pada level supervisi yang menyediakan tampilan visual proses, pemantauan status peralatan, serta manajemen alarm. Proses kontrol utama tetap dijalankan oleh PLC, sedangkan FUXA berfungsi sebagai media interaksi antara operator dan sistem kendali melalui antarmuka berbasis web.



Gambar 2.7 Fuxa SCADA

(Sumber: <https://share.google/ucq1zq0nlVvfgWvau>)

2.4 Hardware yang digunakan

Berikut ini adalah *hardware* yang akan digunakan dalam pelaksanaan tugas akhir ini :

2.4.1 Programmable Logic Controller (PLC Siemens S7-1200)

Siemens S7-1200 merupakan salah satu *Programmable Logic Controller (PLC)* modern yang dikembangkan untuk memenuhi tuntutan otomasi industri dengan tingkat keandalan serta presisi yang tinggi. PLC ini memiliki kemampuan pemrosesan data yang cepat dan fleksibilitas tinggi dalam mengendalikan berbagai proses produksi. Pemrograman sistem dilakukan melalui perangkat lunak Totally Integrated Automation (TIA) Portal, yang memfasilitasi proses konfigurasi, pembuatan program logika, dan

pemantauan sistem secara terpadu dalam satu platform. Selain itu, Siemens S7-1200 juga mendukung berbagai protokol komunikasi industri seperti Profinet, Ethernet, dan Modbus TCP/IP, yang memungkinkannya terintegrasi secara mudah dengan perangkat otomasi maupun sistem pemantauan berbasis jaringan. Dukungan tersebut menjadikan PLC ini mampu beroperasi secara optimal dalam lingkungan industri yang kompleks serta kompatibel dengan sistem berbasis *Internet of Things* (IoT). Dengan performa yang stabil, akurasi tinggi dalam pemrosesan logika kontrol, dan kemampuan adaptif terhadap beragam kondisi industri, Siemens S7-1200 menjadi salah satu PLC yang banyak digunakan dalam sistem otomasi masa (Maulida et al., 2025).



Gambar 2.8 PLC Siemens S7-1200

. (Sumber: <https://www.rs-online.id/p/siemens-s7-1200-plc-cpu-14-digital-input-2-switch-as-analogue-input-inputs-10-digital-output-transistor-output/>)

Tabel 2.3 Spesifikasi PLC Siemens S7-1200

Parameter	Keterangan
Type	SIMATIC S7-1200
Supply Voltage	24 V DC
Load Voltage	24 V DC
Input Current	500mA
Output Current	1600 mA ; Max. 5 V DC for SM and CM
Data Memory	50 kb
CPU Models	1214C
DI / AI / AO	14 DI / 2 AI / 10 DO

2.4.2 Programmable Logic Controller (PLC Omron CP2E N30 DRA)

PLC Omron CP2E N30 DRA merupakan salah satu produk pengendali logika terprogram yang dirancang oleh Omron untuk mendukung kebutuhan otomasi industri

berskala kecil hingga menengah dengan tingkat kestabilan dan keandalan yang tinggi. Perangkat ini memiliki bentuk yang ringkas namun tetap mampu menyediakan jumlah *input* dan *output* digital yang cukup untuk menangani berbagai proses kontrol. Arsitekturnya bersifat modular sehingga pengguna dapat menambahkan modul I/O tambahan sesuai dengan kebutuhan sistem.

Pemrograman PLC dilakukan melalui perangkat lunak CX-Programmer, yang menggunakan pendekatan *ladder diagram* serta dilengkapi fitur pemantauan langsung (*online monitoring*), pengujian logika, dan *debugging* untuk memastikan keakuratan sistem kontrol. PLC ini juga dilengkapi port komunikasi RS-232C, yang berfungsi sebagai jalur koneksi antara PLC dan komputer pemrograman, memungkinkan proses pengendalian dan pemantauan berjalan efisien. Selain itu, fitur seperti *high-speed counter* dan *pulse output* menjadikan CP1E N40 sangat sesuai untuk aplikasi yang membutuhkan ketelitian tinggi, misalnya dalam sistem konveyor atau pengendalian motor. Dengan kemampuan eksekusi instruksi yang cepat, efisiensi daya yang baik, serta kestabilan operasi tinggi, PLC Omron CP1E N40 menjadi salah satu pengendali yang diandalkan dalam berbagai penerapan sistem otomasi industri (Fachrurrozi & Damayanti, n.d.).



Gambar 2.9 PLC Omron CP2E N30 DRA
(Sumber: <https://share.google/2V4zSrjOEaq9mOoMY>)

Tabel 2.4 Spesifikasi PLC Omron CP2E N30 DRA

Parameter	Keterangan
Type	Omron CP2E N30 DRA
Series	CP2E Series

Parameter	Keterangan
<i>Supply Voltage</i>	220 V AC
Jumlah I/O	<i>Input</i> : 18 <i>Output</i> :12
<i>Progam Capacity</i>	10k <i>Steps</i>
<i>Data Memory</i>	32k <i>Words</i>

2.4.3 Sensor *Proximity* Kapasitif

Sensor *proximity* kapasitif merupakan salah satu jenis sensor non-kontak (*non-contact sensor*) yang berfungsi mendeteksi keberadaan objek dengan memanfaatkan perubahan nilai kapasitansi antara pelat elektroda sensor dan benda di sekitarnya. Prinsip kerjanya didasarkan pada perubahan medan listrik akibat perbedaan permitivitas bahan ketika suatu objek mendekat ke area sensitif sensor. Ketika objek berada dalam jarak tertentu, kapasitansi meningkat hingga melewati ambang batas tertentu, sehingga menghasilkan sinyal keluaran digital atau analog. Tidak seperti sensor induktif yang hanya dapat mendeteksi material logam, sensor kapasitif memiliki kemampuan untuk mengenali berbagai jenis bahan, termasuk material non-konduktor seperti kayu, plastik, kaca, dan cairan. Keunggulan utama sensor ini terletak pada fleksibilitas dan keandalannya dalam mendeteksi objek tanpa memerlukan kontak langsung, sehingga meminimalkan keausan pada sistem mekanis. Dalam dunia industri, sensor kapasitif banyak diterapkan pada sistem *sorting*, deteksi level bahan, hingga proses pengisian otomatis. Efisiensi dan kecepatan responsnya membuat sensor ini ideal untuk aplikasi yang membutuhkan pendeteksian material logam dan non-logam secara simultan dengan akurasi tinggi dan waktu respon yang cepat (S. C. Putra & Saripudin, 2025).



Gambar 2.10 Sensor *Proximity* Kapasitif
(Sumber: <https://share.google/66vVBGYrjaxmfgduR>)

Tabel 2.5 Spesifikasi Sensor *Proximity* Kapasitif

Parameter	Keterangan
<i>Voltage</i>	DC 6~36V
<i>Current</i>	300mA
<i>Detection Distance</i>	0-10mm

2.4.4 Sensor *Proximity* Induktif

Sensor *proximity* induktif merupakan salah satu jenis sensor non-kontak yang bekerja berdasarkan prinsip induksi elektromagnetik untuk mendeteksi keberadaan objek logam di sekitarnya. Sensor ini menghasilkan medan elektromagnetik pada kumparan internalnya, dan ketika benda logam mendekat, terjadi perubahan medan tersebut yang kemudian diubah menjadi sinyal listrik sebagai indikator deteksi. Melalui prinsip kerja ini, sensor mampu mengenali berbagai jenis logam seperti besi, tembaga, aluminium, dan baja dengan tingkat ketelitian yang tinggi. Keunggulan utama sensor ini terletak pada kemampuannya beroperasi di lingkungan industri yang keras tanpa terganggu oleh debu, suhu, atau kelembapan, karena proses pendeteksian dilakukan tanpa kontak fisik. Oleh sebab itu, sensor induktif sering diaplikasikan pada sistem otomasi industri untuk mendeteksi posisi benda logam pada konveyor, mesin produksi, serta sistem keamanan berbasis kontrol logika. Penggunaannya terbukti mampu meningkatkan kecepatan dan efisiensi proses pemantauan karena mampu merespons perubahan posisi material logam secara cepat dan akurat, sekaligus menjaga kestabilan sistem tanpa memerlukan perawatan intensif (Ilmiah et al., 2025).



Gambar 2.11 Sensor *Proximity* Induktif
(Sumber: <https://share.google/cc3CTIdPGfzrbpyuQ>)

Tabel 2.6 Spesifikasi Sensor *Proximity* Induktif

Parameter	Keterangan
<i>Voltage</i>	DC 6~36V
<i>Current</i>	300mA
<i>Detection Distance</i>	0-8mm

2.4.5 Sensor Limit Switch

Sensor *limit switch* adalah jenis saklar elektromekanik yang digunakan untuk mendeteksi posisi atau gerakan suatu bagian mekanis melalui kontak langsung antara aktuator dan objek yang diamati. Prinsip kerjanya didasarkan pada perubahan kondisi rangkaian listrik internal ketika bagian mekanis sistem menyentuh tuas atau penekan pada sensor. Saat kontak terjadi, rangkaian akan berubah dari *normally open* menjadi *closed* atau sebaliknya, menghasilkan sinyal listrik yang dikirim ke sistem kendali sebagai penanda posisi. Tidak seperti sensor non-kontak semacam *proximity sensor*, limit switch bekerja secara mekanis sehingga mampu memberikan hasil deteksi yang akurat untuk menentukan batas gerak linear maupun rotasional. Dalam dunia otomasi industri, sensor ini sering digunakan sebagai komponen pengaman (*safety interlock*) dan batas akhir gerakan pada perangkat seperti konveyor, robotik, dan sistem pintu otomatis. Keunggulannya meliputi konstruksi yang kokoh, ketahanan terhadap tekanan fisik, serta kemampuan beroperasi di lingkungan ekstrem dengan tingkat keandalan tinggi. Berdasarkan penelitian yang dilakukan penerapan limit switch pada sistem *cross transfer* di area *cooling bed* industri baja terbukti meningkatkan kestabilan dan keamanan sistem melalui deteksi posisi yang presisi dan respons waktu nyata (Aziz & Aribowo, 2025).



Gambar 2.12 Sensor *Limit Switch*
(Sumber: <https://share.google/X8egVtNT70Sbea2uR>)

2.4.6 *Cylinder Pneumatic*

Silinder pneumatik merupakan aktuator yang berfungsi mengubah energi udara bertekanan menjadi gerakan linear mekanis. Prinsip kerjanya didasarkan pada pemanfaatan tekanan udara di dalam ruang silinder yang mendorong piston untuk bergerak maju atau mundur sesuai arah aliran udara. Komponen ini banyak digunakan dalam sistem otomasi industri karena memiliki struktur sederhana, waktu respons yang cepat, serta kebutuhan perawatan yang relatif rendah. Silinder pneumatik umumnya dibedakan menjadi dua tipe, yaitu *single acting cylinder* yang hanya bekerja dalam satu arah, dan *double acting cylinder* yang memungkinkan gerakan bolak-balik menggunakan dua saluran udara. Dalam implementasinya, silinder ini sering dipadukan dengan katup solenoid serta sensor posisi untuk mendapatkan pengendalian gerak yang presisi dan efisien. Selain itu, perangkat ini dikenal andal dalam kondisi industri yang berat serta hemat energi dalam pengoperasiannya. Penerapan sistem pneumatik dengan dua silinder yang dikendalikan melalui PLC dan HMI terbukti mampu meningkatkan efisiensi dan ketepatan kontrol pada sistem otomasi berbasis pelatihan industri (Polii et al., 2025).



Gambar 2.13 *Cylinder Pneumatic*

(Sumber: <https://share.google/KxWmgGEOU78UqTLGN>)

Tabel 2. 7 Spesifikasi *Cylinder Pneumatic*

Parameter	Keterangan
Action	Double
Media	Pneumatic
Speed Range	50 – 800mm/s

2.4.7 *Rotary Pneumatic*

Rotary pneumatic actuator merupakan jenis aktuator yang digunakan untuk mengubah energi dari udara bertekanan menjadi gerakan rotasi dengan sudut tertentu.

Mekanisme kerjanya memanfaatkan sistem *rack and pinion*, di mana pergerakan linear piston dikonversi menjadi gerak putar pada poros keluaran. Jenis aktuator ini banyak diaplikasikan pada sistem otomasi industri yang memerlukan rotasi terbatas, biasanya pada sudut 90°, 120°, atau 180° sesuai dengan kebutuhan proses.

Keunggulan utama aktuator jenis ini terdapat pada desainnya yang kompak, torsi keluaran yang besar, serta efisiensi energi yang tinggi. Selain itu, penggunaan udara bertekanan membuat aktuator ini mampu memberikan respon cepat dan presisi tanpa memerlukan sistem mekanik yang rumit. Dalam penerapan di dunia industri, rotary pneumatic actuator umumnya digunakan untuk sistem penyortiran, pengaturan orientasi benda kerja, hingga sistem *pick and place*. Kajian terbaru menunjukkan bahwa penerapan aktuator pneumatik tipe rotary dapat meningkatkan efisiensi dan kestabilan sistem otomasi karena waktu respon yang lebih singkat serta kendali gerak yang lebih akurat.



Gambar 2.14 *Rotary Pneumatic*
(Sumber: <https://share.google/BFaku0YjrCfyHD2yK>)

Tabel 2.8 Spesifikasi *Rotary Pneumatic*

Parameter	Keterangan
Action	Double
Sudut Putar	180 derajat
Maximum Input Pressure	8 bar
Maximum Operating Pressure	8 bar

2.4.8 Pilot Lamp

Pilot lamp merupakan komponen berupa lampu indikator yang berfungsi memberikan tanda visual adanya aliran listrik yang masuk ke dalam panel kontrol. Ketika lampu menyala, hal tersebut menandakan bahwa sistem dalam kondisi aktif atau menerima suplai daya. Komponen ini memiliki peran penting sebagai indikator kerja panel, sehingga operator dapat mengetahui status operasional perangkat. Tegangan kerja pilot lamp bervariasi, umumnya tersedia dalam rentang 24 V AC/DC, 110 V, hingga 220 V, disesuaikan dengan kebutuhan sistem. Selain itu, warna cahaya pada lampu indikator juga digunakan untuk membedakan status operasi, misalnya merah untuk peringatan, hijau untuk kondisi normal, kuning atau jingga untuk proses sedang berjalan, dan putih untuk penanda umum. Keberadaan pilot lamp sangat membantu dalam sistem kendali industri karena memberikan indikasi yang cepat, jelas, dan mudah dikenali oleh operator di lapangan (Teknik et al., 2025).



Gambar 2.15 Pilot Lamp

(Sumber: <https://www.telesindo.id/blog/post/apa-itu-pilot-lamp-atau-lampu-indikator-listrik>)

Tabel 2.9 Spesifikasi Pilot Lamp

Parameter	Keterangan
Voltage	24VDC
Colour	Red, Yellow, Green
Ampere	<20mA

2.4.9 Motor DC 12V

Motor DC 12 volt merupakan motor arus searah yang umum digunakan pada sistem otomasi industri berskala kecil hingga menengah karena memiliki kemudahan pengendalian, kecepatan respon tinggi, serta kemampuan menghasilkan torsi besar pada tegangan rendah. Prinsip kerjanya didasarkan pada konversi energi listrik arus searah menjadi energi mekanik berupa gerakan rotasi melalui interaksi medan magnet pada bagian stator dengan arus yang mengalir pada rotor. Pengendalian kecepatan motor ini dapat dilakukan dengan mengatur besar tegangan masukan atau menggunakan teknik *Pulse Width Modulation* (PWM) untuk menghasilkan kecepatan yang lebih stabil. Motor DC 12V dikenal memiliki efisiensi tinggi dan kemampuan integrasi yang baik dengan berbagai sistem kendali seperti PLC maupun mikrokontroler. Dalam penelitian terbaru, motor DC jenis ini dikontrol menggunakan metode *Proportional-Integral-Derivative* (PID) yang diterapkan pada platform Blynk berbasis *Internet of Things* (IoT), dan hasilnya menunjukkan peningkatan efisiensi serta kestabilan kecepatan dibandingkan metode pengendalian konvensional (Auditiya Marizka et al., 2025).



Gambar 2.16 Motor DC 12V
(Sumber : <https://share.google/ieLvMIho4qIeF6VQg>)

Tabel 2.10 Spesifikasi Motor DC 12V

Parameter	Keterangan
<i>Voltage</i>	12VDC
<i>Speed</i>	6000 Rpm
Torsi Maksimum	7.6-8 kg.m

2.4.10 Sensor *Infrared*

Sensor *infrared* merupakan perangkat optoelektronik yang bekerja dengan prinsip pemancaran dan penerimaan gelombang inframerah untuk mendeteksi keberadaan suatu objek tanpa kontak fisik. Mekanismenya melibatkan pemancar yang mengirimkan cahaya inframerah ke arah objek, kemudian pantulan dari cahaya tersebut diterima oleh detektor dan diubah menjadi sinyal listrik yang dapat diolah sistem kontrol untuk menentukan posisi atau jarak objek. Dalam penerapannya di bidang otomasi industri, sensor ini banyak digunakan pada sistem deteksi benda, penghitung produk, serta pengendali posisi pada konveyor maupun sistem keamanan. Keunggulan utama sensor infrared terletak pada kecepatan respon, efisiensi konsumsi daya, dan kemampuan mendeteksi berbagai jenis material termasuk logam dan non-logam. Selain itu, sensor ini tetap mampu bekerja secara stabil pada kondisi lingkungan yang berdebu atau dinamis, sehingga sesuai digunakan pada sistem otomasi yang memerlukan akurasi tinggi dan respon *real-time*. Implementasi sensor infrared dalam sistem otomatis berbasis *Internet of Things* (IoT) juga terbukti meningkatkan efisiensi dan ketepatan deteksi objek (Hilmy, Aprilianto, Setiowati, & Sudradjat, 2025).



Gambar 2.17 Sensor *Infrared*
(Sumber: <https://share.google/yMRWoqxUwWkt7LAa9>)

Tabel 2.11 Spesifikasi Sensor *Infrared*

Parameter	Keterangan
<i>Voltage</i>	DC 6-36V
<i>Current</i>	200mA
<i>Detection Distance</i>	30mm

2.4.11 Mini PC

Mini PC adalah sebuah komputer berukuran kecil yang dirancang untuk menjalankan fungsi komputasi seperti layaknya PC desktop, tetapi dalam bentuk yang lebih ringkas dan hemat ruang. Perangkat ini biasanya dilengkapi dengan semua komponen dasar seperti prosesor, penyimpanan, dan konektivitas jaringan dalam satu unit kecil sehingga cocok digunakan di lingkungan yang membutuhkan efisiensi ruang dan konsumsi daya rendah. Dalam konteks implementasi teknologi modern, Mini PC juga dimanfaatkan sebagai perangkat yang dapat dipantau dan dioptimalkan kinerjanya secara *real-time*, terutama ketika terintegrasi dengan sistem berbasis *Internet of Things (IoT)* untuk pemeliharaan dan pemantauan performa secara terus-menerus (G. W. Putra, 2025).

Selain efisiensi ukuran dan fleksibilitas penggunaan, Mini PC juga dapat diintegrasikan dengan berbagai sistem teknologi untuk mendukung operasional di lingkungan kerja atau industri. Penggunaan sistem berbasis IoT memungkinkan Mini PC memonitor parameter penting seperti suhu, beban kerja, dan status komponen lainnya secara *real-time*, sehingga proses pemeliharaan bisa dilakukan lebih cepat dan akurat tanpa harus melakukan pemeriksaan manual secara berkala. Pendekatan ini menunjukkan bahwa Mini PC tidak hanya berfungsi sebagai komputer biasa, tetapi juga dapat menjadi bagian dari solusi *smart maintenance* untuk meningkatkan keandalan perangkat teknologi di berbagai aplikasi praktis (G. W. Putra, 2025).



Gambar 2.18 Mini PC

(Sumber: <https://share.google/q2UjdkmTvFTasHvf7>)

Tabel 2.12 Spesifikasi Mini PC

Parameter	Keterangan
Type	Mini PC GK3 Pro
Processor	Intel Celeron N5105

Parameter	Keterangan
Ram & SSD	8 Gb Ram & 256 Gb
Konektivitas	Wifi, RJ45, Bluetooth 4.2, Port HDMI, VGA, Port USB 4x

2.4.12 *Pneumatic Solenoid Valve 5/2*

Solenoid Valve 5/2 merupakan salah satu jenis *directional control valve* yang digunakan dalam sistem pneumatik untuk mengatur arah aliran udara bertekanan. Katup ini berfungsi menghubungkan sumber udara dengan aktuator, seperti silinder kerja ganda, yang memerlukan dua arah aliran berbeda untuk menggerakkan piston ke posisi maju dan mundur. Penamaan “5/2” menandakan bahwa valve ini memiliki lima port dan dua kondisi kerja yang mengatur arah aliran udara. Pergantian posisi valve dikontrol oleh *solenoid*, yaitu kumparan elektromagnetik yang bekerja ketika dialiri arus listrik untuk menggeser spool di dalam katup (Gazali et al., 2022).

Selain mengatur arah aliran udara, Solenoid Valve 5/2 juga berperan sebagai komponen utama dalam sistem kontrol elektro-pneumatik. Energi listrik dari *solenoid* diubah menjadi energi mekanik yang menggeser spool sehingga jalur udara dapat dialihkan sesuai kebutuhan sistem. Proses ini memungkinkan sistem bekerja secara otomatis tanpa intervensi manual, seperti pada alat pengepres atau sistem transfer material. Dengan respon yang cepat dan efisiensi tinggi, katup ini menjadi bagian penting dalam sistem otomasi yang menggunakan udara bertekanan sebagai media penggeraknya (Gazali et al., 2022).



Gambar 2.19 *Pneumatic Solenoid Valve 5/2*
(Sumber : <https://share.google/PgRJ7oxehdtzcKAYJ>)

Tabel 2.13 Spesifikasi *Pneumatic Solenoid Valve 5/2*

Parameter	Keterangan
Airtac	5/2 Way Solenoid Valve Type Double Coil

Parameter	Keterangan
<i>Voltage</i>	220 VAC & 24 VDC
<i>Pressure Range</i>	0,15 ~ 0,8 MPA
Max pressure	1 Mpa / 10 Bar

2.4.13 Vacuum Valve Pneumatic

Vacuum valve pada sistem pneumatik berfungsi sebagai komponen pengatur aliran udara untuk menghasilkan tekanan negatif (vakum) yang diperlukan dalam proses penghisapan benda kerja. Dalam sistem otomasi berbasis *electro-pneumatic*, komponen seperti *vacuum ejector* dan *vacuum valve* bekerja secara terintegrasi dengan rangkaian pneumatik yang dikontrol secara elektrik guna menciptakan gaya hisap yang stabil pada benda kerja yang diangkat atau dipindahkan. Proses ini terjadi karena valve dan komponen vakum mengatur aliran udara bertekanan menuju ruang vakum sehingga tekanan di dalamnya menurun sesuai kebutuhan sistem. Pengaturan tekanan tersebut berpengaruh langsung terhadap kestabilan dan efektivitas sistem *pick and place* otomatis (Gazali et al., 2022).

Selain mengatur tekanan, vacuum valve juga berperan penting dalam memastikan sistem pneumatik otomatis berjalan lebih efisien dan konsisten ketika dikombinasikan dengan kontrol elektro-pneumatik. Hasil penelitian mengenai mesin *vacuum press* menunjukkan bahwa penerapan sistem vakum otomatis memberikan peningkatan efektivitas dan efisiensi dibandingkan dengan metode manual. Hal ini membuktikan bahwa komponen pengatur tekanan vakum—termasuk valve dan ejector—mendukung kinerja sistem pneumatik secara otomatis dengan respon yang cepat dalam setiap siklusnya. Temuan tersebut relevan dengan penggunaan *vacuum valve* dalam aplikasi *pick and place* pada sistem otomasi industri (Gazali et al., 2022).



Gambar 2.20 *Vacuum Valve Pneumatic*
(Sumber : <https://share.google/qzaapGsmp6oNNxyu0>)

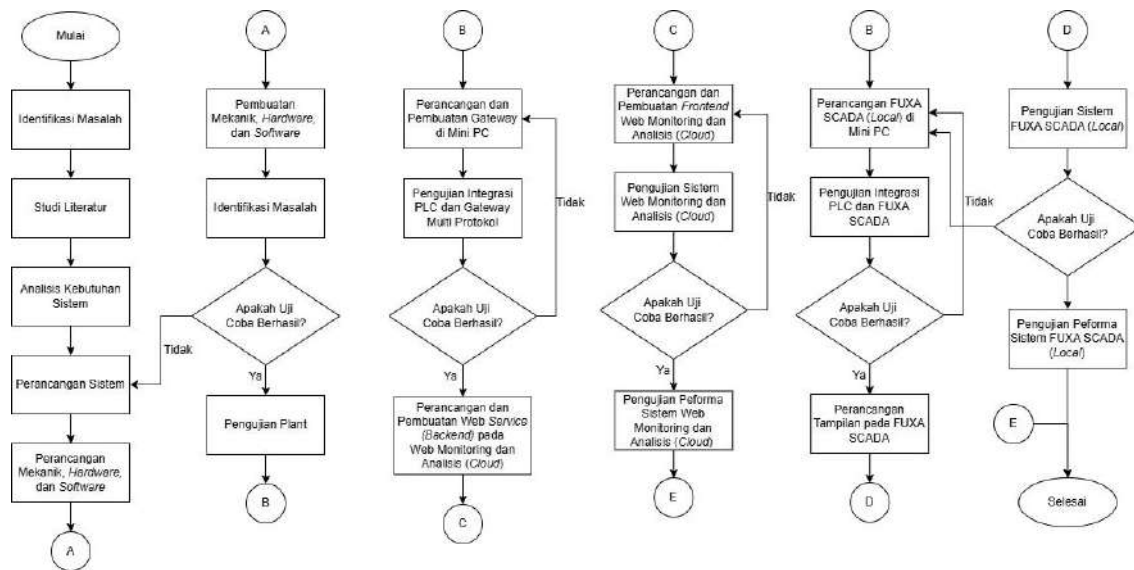
BAB 3

METODOLOGI PENELITIAN

Penelitian ini bertujuan untuk merancang arsitektur integrasi Web-SCADA berbasis *gateway* multi-protokol pada sistem otomasi industri yang terdiri dari beberapa PLC dengan karakteristik berbeda. Tahapan penelitian meliputi identifikasi permasalahan, perancangan arsitektur sistem, implementasi multi-protokol dan Web-SCADA, serta pengujian performa komunikasi. Data yang diperoleh selanjutnya digunakan untuk evaluasi kinerja mesin menggunakan metode *Overall Equipment Effectiveness* (OEE).

3.1 Tahapan Penelitian

Pada pengerjaan tugas akhir ini, terdapat konsep tahapan penelitian yang mencakup perencanaan dan perancangan sistem. Tahapan tersebut disajikan dalam bentuk diagram alir yang dapat dilihat pada Gambar 3.1 berikut.



Gambar 3.1 *Flow Chart* Tahapan Penelitian
(Sumber: Dokumentasi Pribadi, 2026)

dengan penjelasan setiap tahap dijabarkan sebagai berikut :

1. Mulai, Penelitian dimulai dengan penetapan fokus dan ruang lingkup Tugas Akhir, yaitu perancangan dan implementasi sistem supervisi otomasi industri berbasis *gateway* multi-protokol yang mengintegrasikan sistem supervisi lokal dan sistem *Monitoring* berbasis *cloud*.
2. Identifikasi masalah, Tahap ini dilakukan untuk mengkaji dan mengidentifikasi permasalahan yang terjadi pada sistem otomasi industri, khususnya terkait

integrasi data dari beberapa PLC dengan protokol komunikasi yang berbeda, keterbatasan sistem supervisi lokal, serta kebutuhan *Monitoring* dan analisis data secara terpusat.

3. Studi literatur, dilakukan pengumpulan referensi dari jurnal, buku, dan artikel ilmiah yang berkaitan dengan sistem otomasi industri, *SCADA* lokal, *Web-SCADA* berbasis *cloud*, *gateway* multi protokol, *REST API*, serta metode evaluasi kinerja sistem menggunakan *Overall Equipment Effectiveness (OEE)*.
4. Analisis kebutuhan sistem, tahap ini dilakukan untuk menentukan kebutuhan perangkat keras dan perangkat lunak yang digunakan dalam penelitian, meliputi PLC, *Mini PC* sebagai *local gateway*, *FUXA SCADA* untuk supervisi lokal, layanan *web service* berbasis *REST API*, *database*, serta sistem *Web Monitoring* berbasis *cloud*.
5. Perancangan sistem, Pada tahap ini dilakukan penyusunan rancangan arsitektur sistem secara keseluruhan yang menggambarkan alur komunikasi data dari PLC menuju *Mini PC*, sistem supervisi lokal, serta integrasi dengan sistem *cloud* untuk kebutuhan *Monitoring* dan analisis data.
6. Perancangan mekanik, *hardware*, dan *software*, dilakukan perancangan serta pengembangan perangkat mekanik, perangkat keras, dan perangkat lunak yang mendukung sistem otomasi dan arsitektur integrasi yang telah dirancang.
7. Pengujian Integrasi Mekanik, *Hardware*, dan *Software*, Seluruh komponen yang telah dirancang kemudian diintegrasikan dan diuji untuk memastikan sistem dapat berfungsi sesuai dengan perancangan. Hasil pengujian digunakan sebagai dasar penyempurnaan sistem sebelum dilakukan pengujian pada *plant*.
8. Pengujian *Plant*, Pengujian *plant* dilakukan untuk memastikan sistem kendali dan supervisi dapat beroperasi dengan baik pada kondisi kerja nyata atau simulasi proses industri.
9. Perancangan dan Pembuatan *Gateway* pada *Mini PC*, Pada tahap ini dilakukan perancangan dan implementasi *gateway* berbasis *Mini PC* yang berfungsi sebagai penghubung antara PLC dan sistem lain dengan mendukung komunikasi multi protokol.
10. Pengujian Integrasi PLC dan *Gateway* Multi Protokol, Tahap ini bertujuan untuk memastikan proses akuisisi data dari PLC melalui *gateway* dapat berjalan dengan baik dan stabil sesuai dengan spesifikasi sistem.

11. Perancangan dan Pembuatan Web Service (Backend) pada Sistem *Cloud*, Setelah integrasi *gateway* berjalan dengan baik, dilakukan pengembangan web service atau backend sebagai media pertukaran data antara *gateway* dan sistem Web *Monitoring* dan Analisis berbasis *cloud*.
12. Perancangan dan Pembuatan *Frontend* Web *Monitoring* dan Analisis (*Cloud*), Tahap ini meliputi pengembangan antarmuka web *Monitoring* yang digunakan untuk menampilkan data operasional secara *real-time* maupun historis melalui sistem *cloud*.
13. Pengujian Sistem Web *Monitoring* dan Analisis (*Cloud*), Sistem web *Monitoring* yang telah dikembangkan diuji untuk memastikan seluruh fungsi berjalan dengan baik, data ditampilkan secara akurat, serta sistem dapat beroperasi secara stabil.
14. Pengujian Performa Sistem Web *Monitoring* dan Analisis (*Cloud*), Tahap ini dilakukan untuk mengevaluasi kinerja sistem web monitoring, meliputi kecepatan akses, kestabilan layanan, serta keandalan sistem dalam menangani pertukaran data.
15. Perancangan Sistem FUXA SCADA (Local) pada Mini PC, Pada tahap ini dilakukan perancangan sistem supervisi lokal menggunakan FUXA SCADA yang dijalankan pada Mini PC sebagai media *Monitoring* dan kontrol proses.
16. Pengujian Integrasi PLC dan FUXA SCADA, Tahap ini bertujuan untuk memastikan sistem supervisi lokal dapat berkomunikasi dengan PLC dan menampilkan data proses secara *real-time* sesuai dengan kondisi aktual.
17. Perancangan Tampilan pada FUXA SCADA, Perancangan tampilan dilakukan untuk menyusun antarmuka pengguna yang informatif, mudah dipahami, dan sesuai dengan kebutuhan operasional sistem.
18. Pengujian Sistem FUXA SCADA (Local), Sistem FUXA SCADA yang telah dirancang diuji secara menyeluruh untuk memastikan fungsi *Monitoring* dan kontrol berjalan dengan baik.
19. Pengujian Performa Sistem FUXA SCADA (Local), Tahap ini dilakukan untuk mengevaluasi performa sistem supervisi lokal, meliputi respon sistem, kestabilan komunikasi, serta keandalan operasional secara keseluruhan.
20. Selesai, Setelah seluruh tahapan penelitian dilaksanakan dan sistem berfungsi sesuai dengan tujuan penelitian, proses penelitian Tugas Akhir dinyatakan selesai dan dilanjutkan dengan penyusunan laporan Tugas Akhir.

3.1.1 *State of The Art* (Keterbaruan Dari Penelitian)

Beberapa penelitian sebelumnya telah mengangkat tema pengembangan sistem SCADA berbasis web serta integrasi komunikasi antara perangkat otomasi industri. Salah satu contohnya adalah karya Ardaneshwara (2024), yang membangun sistem SCADA menggunakan Node-RED untuk menghubungkan dua jenis PLC dari vendor berbeda, yakni Siemens dan Omron, melalui protokol komunikasi Modbus TCP dan FINS UDP. Namun, sistem tersebut hanya mampu melakukan pemantauan dalam jaringan lokal (LAN) dan jaringan internet.

Penelitian lainnya oleh Adam Risky Muharam (2025) pada integrasi dua PLC berbeda Siemens S7-1200 dan Schneider TM221CE16R dengan protokol komunikasi Modbus TCP. Sistem ini juga telah melibatkan HMI dan VSD guna mendukung operasi secara *real-time* dalam satu jaringan kontrol lokal. Walaupun telah menggabungkan dua merek PLC, sistem ini masih terbatas pada satu jenis media komunikasi, yakni ethernet, dengan protokol yang seragam.

Sebagai bentuk pengembangan lebih lanjut, penelitian ini menawarkan pendekatan yang berbeda baik dari sisi arsitektur sistem, basis *gateway*, maupun pemisahan fungsi supervisi dan analisis data. Pada penelitian ini, integrasi antar-PLC dilakukan menggunakan *gateway* multi-protokol berbasis Node.js yang dijalankan pada perangkat Mini PC. *Gateway* ini berfungsi sebagai *middleware* yang mengakuisisi data dari PLC Omron CP2E N30 DRA dengan protokol FINS UDP dan PLC Siemens S7-1200 dengan protokol S7 TCP/IP, kemudian menyajikannya dalam bentuk REST API.

Selain itu, sistem supervisi dan pengendalian menggunakan FUXA SCADA yang berperan sebagai SCADA lokal untuk *Monitoring* dan kontrol proses secara *real-time*. Di sisi lain, penelitian ini juga mengembangkan Web *Monitoring* berbasis *cloud* yang berfokus pada pemantauan jarak jauh dan analisis kinerja mesin menggunakan *Overall Equipment Effectiveness* (OEE). Dengan pemisahan fungsi antara SCADA lokal (kontrol proses) dan sistem *cloud* (*Monitoring* dan analisis), arsitektur yang diusulkan menjadi lebih fleksibel dan terstruktur.

3.2 Tahap Identifikasi Awal

Pada tahap awal ini dilakukan pengumpulan informasi untuk memahami permasalahan utama yang akan diselesaikan dalam penelitian Tugas Akhir. Analisis difokuskan pada kebutuhan sistem integrasi data pada lingkungan otomasi industri yang terdiri dari beberapa PLC dengan karakteristik dan protokol komunikasi yang berbeda.

Tahap ini mencakup identifikasi kebutuhan perangkat keras seperti PLC dan perangkat jaringan, serta perangkat lunak berupa *gateway* dan *web service*. Selain itu, dilakukan pemahaman terhadap alur pertukaran data dari sistem otomasi menuju sistem supervisi berbasis web serta keterkaitannya dengan evaluasi kinerja mesin menggunakan metode *Overall Equipment Effectiveness* (OEE). Hasil dari tahap ini menjadi dasar dalam perancangan arsitektur sistem yang terintegrasi dan sesuai dengan kebutuhan industri.

3.2.1 Studi Literatur

Tahap studi literatur dilakukan dengan mengkaji berbagai referensi berupa jurnal ilmiah, buku, dan artikel yang berkaitan dengan sistem otomasi industri, Web-SCADA, integrasi data multi-PLC, *gateway* multi protokol, serta metode evaluasi kinerja mesin menggunakan *Overall Equipment Effectiveness* (OEE). Studi literatur ini bertujuan untuk memperoleh pemahaman umum mengenai konsep dan teknologi yang relevan dengan penelitian yang dilakukan.

Selain itu, studi literatur juga mencakup pemahaman mengenai konsep supervisi berbasis web serta integrasi dalam lingkungan industri. Pemahaman terhadap teknologi komunikasi industri dan alur pertukaran data diperlukan agar sistem yang dirancang dapat mendukung kebutuhan supervisi dan pengolahan data secara terpusat.

Selain studi literatur, dilakukan juga observasi terhadap sistem otomasi yang menjadi objek penelitian. Observasi ini bertujuan untuk memahami kondisi sistem yang ada serta kebutuhan integrasi data yang akan dirancang. Informasi yang diperoleh dari observasi digunakan sebagai acuan dalam mengembangkan solusi integrasi berbasis *gateway* dan Web-SCADA, termasuk memastikan bahwa proses pertukaran data antar perangkat dapat berjalan dengan baik. Dengan demikian, studi literatur dan observasi digunakan sebagai dasar dalam memberikan arahan yang jelas terhadap pelaksanaan penelitian Tugas Akhir ini agar berjalan secara efektif dan terstruktur.\

3.2.2 Identifikasi Masalah

Proses identifikasi masalah dilakukan dengan menganalisis kondisi sistem otomasi industri yang memiliki beberapa *plant* atau stasiun produksi yang dikendalikan oleh PLC dengan merek dan protokol komunikasi yang berbeda. Perbedaan karakteristik antar PLC tersebut menyebabkan data operasional mesin belum terintegrasi secara optimal dalam satu sistem pemantauan terpusat. Akibatnya, proses supervisi dan analisis kinerja mesin berbasis web menjadi kurang efisien dan sulit dilakukan secara *real-time*.

Permasalahan utama yang ditemukan adalah keterbatasan integrasi data antar PLC, sehingga data operasional mesin masih tersebar dan belum dimanfaatkan secara maksimal untuk kebutuhan analisis kinerja. Kondisi ini berdampak pada keterbatasan informasi yang dapat digunakan oleh operator maupun manajemen dalam mengevaluasi performa mesin secara menyeluruh dan berkelanjutan.

Berdasarkan permasalahan tersebut, penelitian ini difokuskan pada perancangan arsitektur integrasi Web-SCADA berbasis *gateway* multi protokol sebagai solusi. *Gateway* dirancang untuk berfungsi sebagai penghubung antara PLC dan sistem supervisi berbasis web, sehingga perbedaan karakteristik komunikasi antar PLC dapat dinormalisasi. Dengan adanya arsitektur ini, data operasional mesin dapat dikirim secara lebih stabil dan konsisten ke Web-SCADA serta digunakan sebagai dasar perhitungan dan analisis *Overall Equipment Effectiveness* (OEE) dalam mengevaluasi kinerja mesin secara objektif.

3.3 Analisa Kebutuhan Sistem

Pada tahap ini dilakukan analisa terhadap kebutuhan sistem yang meliputi perangkat keras, perangkat lunak, dan infrastruktur pendukung yang diperlukan untuk membangun arsitektur integrasi Web-SCADA berbasis *gateway* multi protokol. Fokus utama dari analisis ini adalah memastikan bahwa setiap komponen yang digunakan dapat saling terintegrasi dengan baik sehingga data operasional dari sistem otomasi dapat dikumpulkan dan ditampilkan secara terpusat melalui Web-SCADA. Sistem yang dirancang ditujukan untuk lingkungan industri yang terdiri dari beberapa PLC dengan karakteristik dan protokol komunikasi yang berbeda.

Pada sisi perangkat keras, sistem membutuhkan PLC sebagai pengendali proses pada masing-masing *plant* atau stasiun produksi, perangkat jaringan sebagai media komunikasi data, serta satu PLC atau perangkat *gateway* yang berfungsi sebagai penghubung antar sistem otomasi dan sistem supervisi berbasis web. PLC *gateway* dipilih karena kemampuannya dalam melakukan pertukaran data dengan PLC lain serta mendukung komunikasi berbasis jaringan yang stabil. Perangkat jaringan digunakan untuk memastikan proses pengiriman data dapat berjalan secara *real-time* dan andal.

Pada sisi perangkat lunak, sistem menggunakan perangkat lunak pemrograman PLC untuk mengatur logika kontrol proses dan penyediaan data operasional pada masing-masing PLC. Data proses yang dihasilkan oleh PLC tidak dikirimkan langsung ke sistem Web-SCADA, melainkan terlebih dahulu dikumpulkan dan diolah oleh *gateway* multi

protokol. *Gateway* berperan sebagai lapisan integrasi yang melakukan akuisisi, normalisasi, dan pengiriman data antar sistem otomasi dengan sistem supervisi berbasis web. Proses pengiriman data dari *gateway* menuju Web-SCADA dilakukan melalui web service berbasis REST API, sehingga pertukaran data dapat berjalan secara fleksibel dan mudah diintegrasikan dengan aplikasi berbasis web. Web-SCADA berfungsi sebagai antarmuka supervisi untuk menampilkan data operasional mesin secara *real-time* melalui web browser. Data yang disajikan pada Web-SCADA selanjutnya dimanfaatkan sebagai dasar evaluasi kinerja mesin menggunakan metode *Overall Equipment Effectiveness* (OEE), sehingga operator dan pihak terkait dapat memantau performa sistem secara lebih terstruktur dan konsisten.

Tabel 3.1 Kebutuhan Sistem

Pembuatan	Analisa Kebutuhan
<i>Hardware</i>	PLC Siemens S7 – 1200
	PLC Omron CP2E – N30 DRA
	Sensor <i>Proximity</i> Kapasitif
	Sensor <i>Limit Switch</i>
	Sensor <i>Infrared</i>
	<i>Rotary Pneumatic</i>
	<i>Piston Pneumatic</i>
	Motor DC
	<i>Pilot Lamp</i>
	Mini PC
	Sensor <i>Proximity</i> Induktif
<i>Gateway Connector</i>	PostgreSQL
	<i>Visual Studio Code</i>
	NodeJS
Web-SCADA	ReactJS
	FUXA SCADA

(Sumber: Dokumentasi Pribadi, 2026)

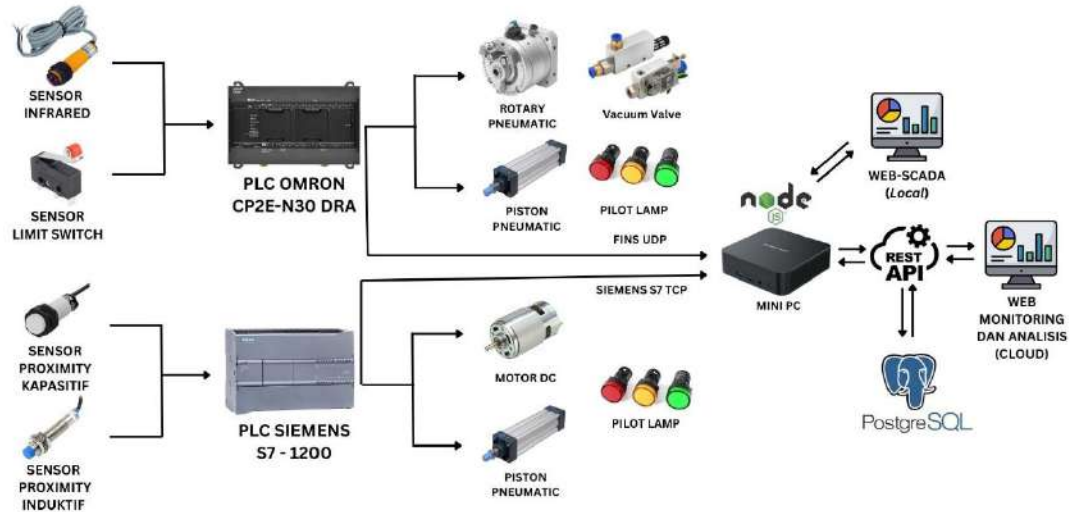
Tabel 3.1 menyajikan daftar komponen perangkat keras dan perangkat lunak yang digunakan dalam perancangan sistem integrasi Web-SCADA berbasis *gateway* multi protokol.

3.4 Perancangan Sistem

Sistem yang dikembangkan pada penelitian Tugas Akhir ini menerapkan arsitektur *layered architecture* yang membagi sistem ke dalam dua bagian utama, yaitu sistem lokal (*local system*) dan sistem *cloud* (*cloud system*). Pada lapisan *process control*, *Distribution and Sorting station* dikendalikan oleh dua PLC yang memiliki karakteristik berbeda, yaitu PLC Omron CP2E-N30 DRA dan PLC Siemens S7-1200. Kedua PLC tersebut berfungsi sebagai pengendali utama proses produksi serta menghasilkan data operasional berupa status sensor, status aktuator, *cycle time*, dan informasi produksi. PLC dirancang untuk beroperasi secara mandiri dan tidak terhubung langsung dengan sistem *web* maupun *cloud*, sehingga kontinuitas dan keandalan pengendalian proses tetap terjaga.

Proses integrasi data dari kedua PLC dilakukan menggunakan *Mini PC* yang berfungsi sebagai *multi-protocol gateway* pada lapisan *supervisory control*. *Gateway* ini bertugas mengakuisisi data dari PLC dengan protokol komunikasi yang berbeda, melakukan penyesuaian dan normalisasi format data, serta menyediakan jalur komunikasi terpusat. Pada sistem lokal (*local system*), *Mini PC* terhubung dengan *Web-SCADA* lokal yang berperan sebagai antarmuka supervisi dan *Monitoring* kondisi operasional *Distribution and Sorting station* secara *real-time* melalui jaringan lokal. Mekanisme kontrol yang tersedia pada *Web-SCADA* bersifat *supervisory* dan tidak menggantikan fungsi kendali utama yang telah ditanamkan pada PLC.

Selain mendukung kebutuhan sistem lokal, *Mini PC* juga berfungsi sebagai penghubung menuju sistem *cloud* (*cloud system*) melalui *web service* berbasis *REST API*. Data operasional yang telah diproses kemudian disimpan pada database *PostgreSQL* dan digunakan untuk keperluan *Monitoring* lanjutan serta analisis kinerja sistem menggunakan metode *Overall Equipment Effectiveness (OEE)*. Perhitungan *OEE* didasarkan pada parameter *Availability*, *Performance*, dan *Quality* yang merepresentasikan kinerja keseluruhan *Distribution and Sorting station* sebagai satu kesatuan sistem terintegrasi. Dengan adanya pemisahan antara sistem lokal dan *cloud*, sistem yang dirancang mampu menyediakan *Monitoring real-time* yang stabil sekaligus mendukung analisis kinerja produksi secara terstruktur dan terpusat. Konsep kerja sistem secara keseluruhan dapat dijelaskan lebih lanjut melalui diagram blok pada Gambar 3.2, yang menggambarkan alur data dari kedua station produksi menuju *gateway* dan *Web-SCADA*.



Gambar 3.2 Diagram Blok Sistem
(Sumber: Dokumentasi Pribadi, 2026)

3.4.1 Hardware

Perangkat keras yang digunakan dalam penelitian Tugas Akhir ini terdiri dari dua station produksi yang masing-masing dikendalikan oleh PLC dengan karakteristik dan protokol komunikasi yang berbeda. Station pertama dikendalikan oleh PLC Omron CP2E-N30 DRA, sedangkan station kedua dikendalikan oleh PLC Siemens S7-1200. Masing-masing PLC berfungsi sebagai pengendali proses lokal yang mengelola *input* dan output sesuai dengan kebutuhan station.

Pada station yang dikendalikan oleh PLC Omron CP2E-N30 DRA, sistem menerima *input* dari beberapa sensor, seperti sensor *proximity* dan limit switch, serta menghasilkan output berupa aktuator rotary pneumatic, piston pneumatic, dan pilot lamp sebagai indikator status proses. PLC Omron berperan dalam menjalankan logika kontrol proses serta menyediakan data operasional yang akan dikirimkan ke sistem integrasi.

Pada station yang dikendalikan oleh PLC Siemens S7-1200, sistem menerima *input* dari sensor photoelectric dan sensor *proximity*, serta menghasilkan output berupa motor DC, piston pneumatic, dan pilot lamp. PLC Siemens berfungsi sebagai pengendali proses pada station kedua dan menyediakan data operasional yang merepresentasikan kondisi dan kinerja station tersebut. Kedua PLC bekerja secara independen pada level kontrol, namun saling terhubung pada level data melalui sistem integrasi yang dirancang.

3.4.2 *Gateway Connector*

Gateway dalam penelitian Tugas Akhir ini direalisasikan menggunakan Mini PC yang menjalankan aplikasi berbasis Node.js. *Gateway* berfungsi sebagai lapisan integrasi antara dua PLC yang memiliki perbedaan media dan protokol komunikasi. PLC Omron CP2E-N30 berkomunikasi dengan *gateway* menggunakan Modbus RTU, sedangkan PLC Siemens S7-1200 berkomunikasi menggunakan Siemens S7 TCP/IP.

Gateway bertugas melakukan akuisisi data dari masing-masing PLC, melakukan normalisasi dan pengemasan data, serta mengelola pertukaran data antar sistem. Dengan adanya *gateway*, perbedaan karakteristik komunikasi antar PLC dapat disatukan dalam satu arsitektur integrasi terpusat, sehingga sistem supervisi tidak perlu berkomunikasi langsung dengan masing-masing PLC.

Selain itu, *gateway* juga berperan dalam menyimpan data operasional ke dalam database PostgreSQL. Database ini digunakan untuk menyimpan data historis, log operasional, serta data yang dibutuhkan untuk analisis performa sistem. Arsitektur ini memungkinkan sistem menjadi lebih fleksibel, mudah dikembangkan, dan stabil dalam menangani data dari multi-station produksi

3.4.3 *Web-SCADA (Local) dan Web Monitoring dan Analisis (Cloud)*

Sistem supervisi pada penelitian ini dirancang dalam dua lapisan utama, yaitu Web-SCADA (Local) dan Web *Monitoring dan Analisis (Cloud)*. Kedua sistem tersebut diintegrasikan untuk mendukung proses monitoring, supervisi, serta analisis data secara menyeluruh pada sistem otomasi industri yang dikembangkan.

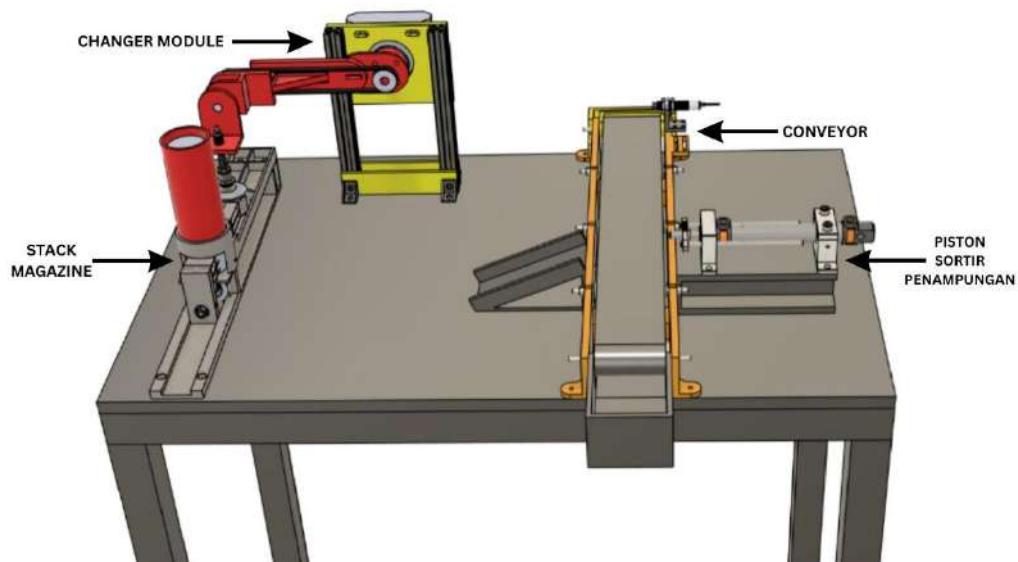
Web-SCADA (Local) diimplementasikan menggunakan FUXA SCADA yang dijalankan pada Mini PC sebagai sistem supervisi berbasis jaringan lokal. Sistem ini berfungsi untuk melakukan *Monitoring* dan kontrol proses secara *real-time* terhadap PLC tanpa memerlukan akses langsung ke perangkat PLC. Informasi yang ditampilkan meliputi status proses, kondisi aktuator, serta parameter operasional lainnya, sehingga operator dapat melakukan pengawasan langsung di area *plant*. Sistem supervisi lokal ini juga memastikan proses *Monitoring* dan kontrol tetap berjalan meskipun koneksi ke sistem *cloud* tidak tersedia.

Web *Monitoring dan Analisis* berbasis *cloud* dikembangkan untuk menyediakan akses pemantauan jarak jauh dan analisis data operasional. Data proses dikirimkan dari *gateway* pada Mini PC menuju sistem *cloud* melalui web service berbasis REST API, sehingga data dapat diakses secara *real-time* maupun historis melalui web browser. Data

yang tersimpan pada sistem *cloud* selanjutnya dimanfaatkan untuk evaluasi kinerja sistem menggunakan metode *Overall Equipment Effectiveness* (OEE), sehingga efektivitas proses dapat dianalisis secara kuantitatif dan berkelanjutan guna mendukung pengambilan keputusan berbasis data.

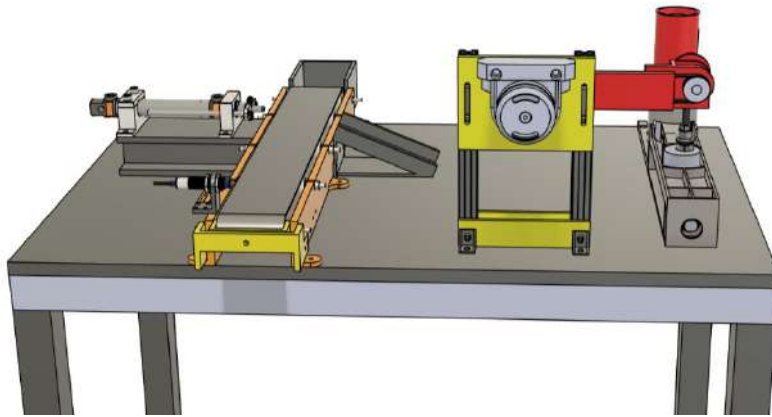
3.5 Pembuatan *Plant Distribution* dan *Sorting station*.

Penelitian Tugas Akhir ini mengimplementasikan konsep *Modular Production System* (MPS) pada *Distribution and Sorting station*. MPS merupakan pendekatan dalam sistem manufaktur yang membagi proses produksi ke dalam beberapa modul atau stasiun kerja yang berdiri secara mandiri namun tetap saling terhubung. Setiap modul memiliki fungsi tertentu, seperti proses distribusi, pemindahan, dan penyortiran benda kerja, sehingga sistem dapat disusun dan dikembangkan sesuai kebutuhan proses produksi. Penerapan konsep MPS pada penelitian ini bertujuan untuk merepresentasikan sistem produksi berskala industri dalam bentuk prototipe yang terstruktur. Melalui pendekatan modular, setiap bagian dari *Distribution and Sorting station* dapat bekerja secara independen namun tetap terkoordinasi melalui sistem kendali terpusat, sehingga keseluruhan proses dapat berjalan secara terintegrasi. Gambar 3.3 untuk desain tampak depan, Gambar 3.4 untuk desain tampak belakang, Gambar 3.5 untuk desain tampak isometri kiri.



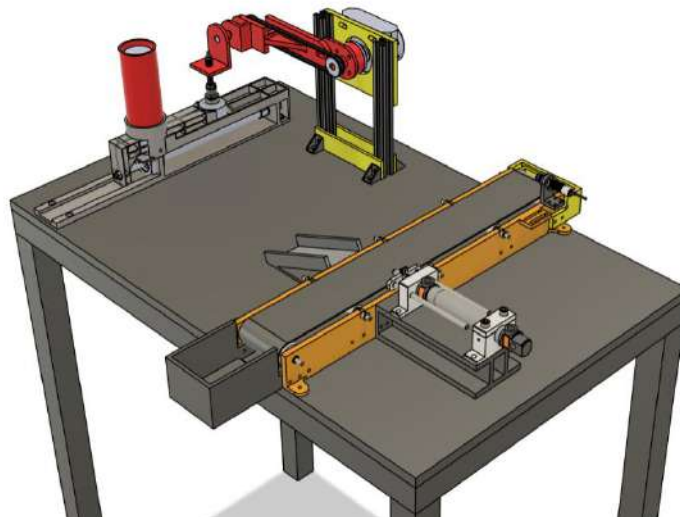
Gambar 3.3 Desain Tampak Depan
(Sumber: Dokumentasi Pribadi, 2026)

Gambar 3.3 menunjukkan desain MPS dari tampak depan yang terdiri atas *stack magazine*, *changer module*, konveyor, dan piston penyortiran. Tampilan ini memperlihatkan posisi masing-masing komponen utama pada plant.



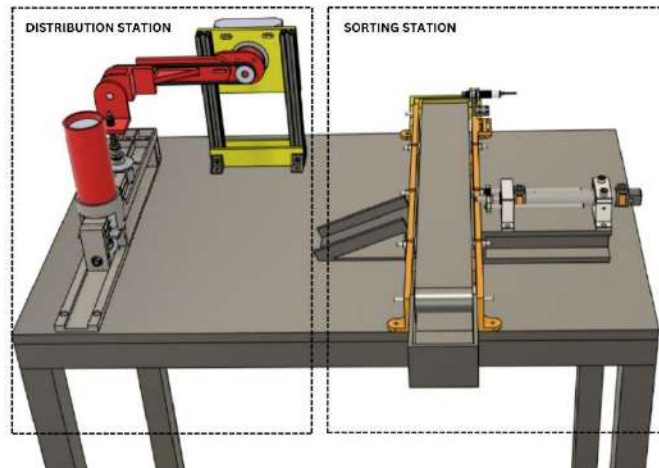
Gambar 3.4 Desain Tampak Depan
(Sumber: Dokumentasi Pribadi, 2026)

Gambar 3.4 menunjukkan desain MPS dari tampak belakang, sehingga posisi bagian belakang *changer module*, konveyor, serta mekanisme pneumatik dapat terlihat dengan lebih jelas. Tampilan ini juga memperlihatkan susunan komponen dari sisi yang tidak tampak pada gambar depan.



Gambar 3.5 Desain Tampak Isometri Kiri
(Sumber: Dokumentasi Pribadi, 2026)

Gambar 3.5 menunjukkan desain MPS dari tampak isometri kiri yang memperlihatkan keseluruhan susunan komponen dan posisi kedua *station* pada meja. Sudut pandang ini memberikan gambaran bentuk plant secara lebih menyeluruh.



Gambar 3.6 *Distribution Dan Sorting station*
(Sumber: Dokumentasi Pribadi, 2026)

Pada gambar 3.6 menunjukkan rancangan *Distribution Station* dan *Sorting station* yang digunakan sebagai objek penelitian pada sistem otomasi dengan representasi station industri. Kedua station ditempatkan pada satu meja kerja yang sama, namun dipisahkan secara fungsional berdasarkan tahapan proses produksi. *Distribution Station* berada pada sisi kiri dan berfungsi sebagai tahap awal distribusi benda kerja, sedangkan *Sorting station* berada pada sisi kanan dan berfungsi sebagai tahap pemilahan benda kerja berdasarkan kriteria tertentu.

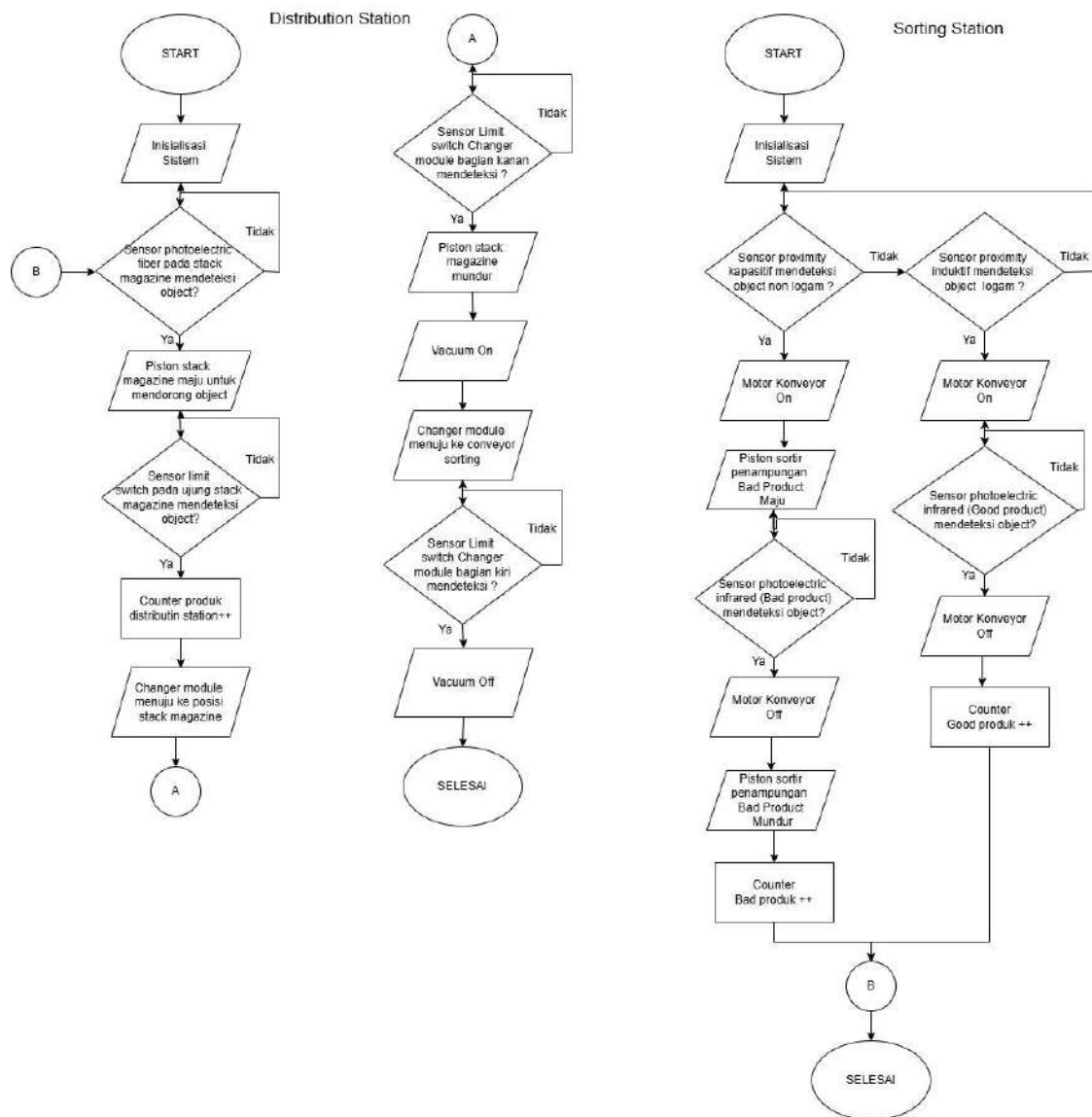


Gambar 3.7 *Object, A. Good Product, B. Bad Product*
(Sumber Dokumentasi Pribadi, 2026)

Gambar 3.7 menunjukkan benda kerja (*material*) yang digunakan pada sistem *distribution station* dan *sorting station*. Benda kerja tersebut dibedakan menjadi dua kategori, yaitu material A sebagai *good product* dan material B sebagai *bad product*.

Material A merupakan benda kerja logam, sedangkan material B merupakan benda kerja non-logam

Kedua benda kerja memiliki bentuk dan ukuran yang relatif serupa, tetapi memiliki karakteristik material yang berbeda. Perbedaan tersebut digunakan dalam proses pengujian untuk mengetahui kemampuan sensor dan sistem penyortiran dalam membedakan serta mengarahkan benda kerja sesuai dengan kategorinya. Alur kerja dari *plant Distribution dan sorting station* ditampilkan pada gambar 3.8 berikut.



Gambar 3.8 Flow Chart Distribution dan Sorting station
(Sumber: Dokumentasi Pribadi, 2026)

Proses kerja sistem *Distribution dan Sorting station* diawali dengan tahapan inisialisasi, yang bertujuan untuk memastikan seluruh komponen sistem berada dalam kondisi siap digunakan. Pada tahap ini, PLC, sensor, aktuator, modul pneumatik, serta motor konveyor dipersiapkan dan diperiksa fungsinya. Selain itu, sistem juga melakukan

pengecekan terhadap sensor infrared, sensor *proximity* induktif, sensor *proximity* kapasitif, dan limit switch guna memastikan seluruh sensor beroperasi secara normal sebelum proses dijalankan.

Pada *Distribution Station*, proses dimulai ketika sensor infrared yang terpasang pada *stack magazine* mendeteksi keberadaan objek. Setelah objek terdeteksi, piston *stack magazine* akan bergerak maju untuk mendorong objek keluar dari *magazine*. Selanjutnya, sistem memantau limit switch yang berada di ujung *stack magazine* untuk memastikan objek telah mencapai posisi yang ditentukan. Apabila posisi objek telah terkonfirmasi, maka nilai counter produk pada *Distribution station* akan bertambah sebagai indikasi bahwa satu objek berhasil didistribusikan.

Setelah objek berhasil didorong keluar, *changer module* bergerak menuju posisi *stack magazine* untuk pengambilan objek. Setelah *changer module* telah sampai di posisi *stack magazine* dan sensor limit switch bagian kanan mendeteksi, sistem akan mengaktifkan vacuum untuk mengambil objek. Selanjutnya, *changer module* bergerak menuju konveyor dengan memanfaatkan limit switch sisi kiri sebagai indikator posisi tujuan. Setelah objek berada di atas konveyor, vacuum dinonaktifkan sehingga objek dilepaskan dan proses distribusi dinyatakan selesai.

Objek yang telah dilepaskan kemudian memasuki *Sorting station* untuk menjalani proses penyortiran. Pada tahap ini, sensor *proximity* induktif berfungsi untuk mengidentifikasi apakah objek yang masuk merupakan material logam. Jika objek terdeteksi sebagai logam, motor konveyor akan diaktifkan untuk mengarahkan objek menuju penampungan produk baik (*good product*). Ketika sensor *proximity* induktif pada penampungan pertama mendeteksi keberadaan objek, motor konveyor akan berhenti dan counter *good product* akan bertambah.

Sebaliknya, apabila objek tidak terdeteksi sebagai logam oleh sensor *proximity* induktif, sistem akan melanjutkan proses pemeriksaan menggunakan sensor *proximity* kapasitif untuk mendeteksi objek non-logam. Jika objek non-logam terdeteksi, motor konveyor tetap beroperasi hingga objek mencapai penampungan *reject*. Ketika sensor *proximity* kapasitif pada penampungan *reject* mendeteksi objek, motor konveyor akan berhenti dan counter *bad product* akan bertambah, yang menandakan objek tersebut dikategorikan sebagai produk tidak sesuai.

Setelah proses penyortiran selesai, sistem akan kembali ke kondisi awal dan siap untuk memproses objek berikutnya. Data jumlah produk yang terdistribusi, produk baik, dan produk *reject* selanjutnya digunakan sebagai dasar pemantauan kinerja sistem dan

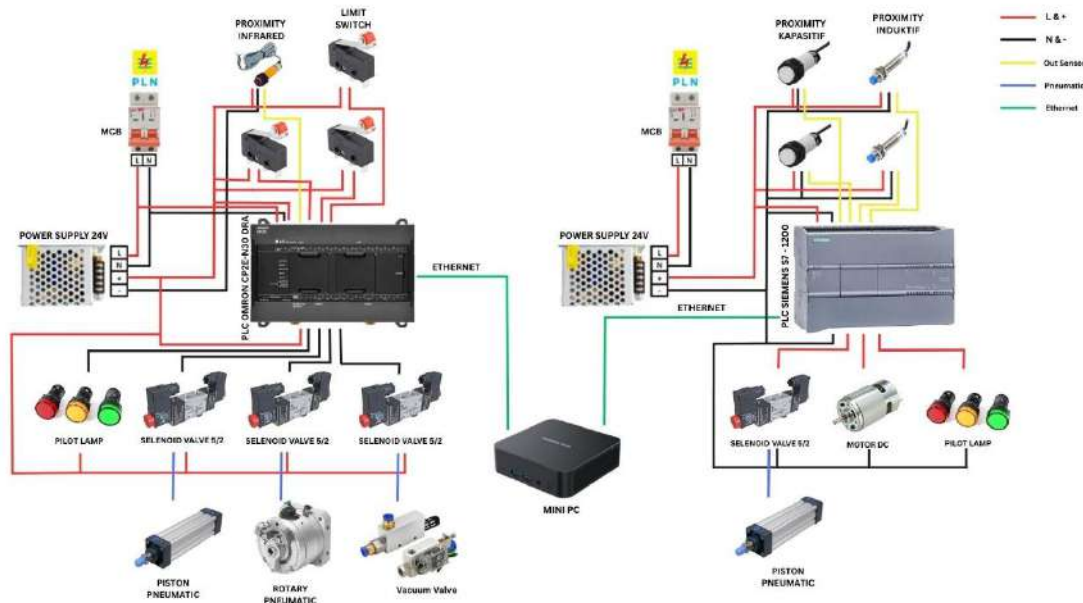
dikirimkan ke Web-SCADA melalui *gateway* untuk keperluan monitoring, pencatatan data, serta analisis performa mesin.

3.6 *Wiring Diagram Sistem*

Wiring diagram pada penelitian ini menggambarkan susunan serta keterkaitan antar komponen utama yang membentuk sistem *Distribution and Sorting Machine* berbasis konsep *Modular Production System (MPS)*. Sistem ini terdiri atas dua unit pengendali utama, yaitu PLC Omron CP2E-N30 DRA dan PLC Siemens S7-1200, yang masing-masing berfungsi mengontrol modul berbeda sesuai dengan peranannya dalam sistem. PLC Omron CP2E-N30 DRA digunakan untuk mengendalikan modul distribusi, dengan menerima sinyal masukan dari sensor *proximity* kapasitif dan limit switch sebagai pendeteksi keberadaan serta posisi benda kerja. Berdasarkan informasi tersebut, PLC mengaktifkan aktuator berupa silinder pneumatik dan rotary pneumatic untuk melakukan proses pemindahan dan pendistribusian objek. Selain itu, sistem dilengkapi dengan pilot lamp yang berfungsi sebagai indikator status kerja. Seluruh data dari PLC Omron kemudian dikirimkan ke sistem *gateway* melalui komunikasi FINS UDP.

Sementara itu, PLC Siemens S7-1200 berperan dalam pengendalian modul penyortiran. PLC ini menerima sinyal dari sensor *proximity* induktif untuk mengidentifikasi karakteristik benda, kemudian mengendalikan aktuator berupa motor DC dan silinder pneumatik guna melakukan proses penyortiran. Data operasional dari PLC Siemens selanjutnya dikirimkan ke sistem *gateway* melalui protokol komunikasi Siemens S7 berbasis TCP/IP.

Sebagai pusat integrasi sistem, mini PC digunakan sebagai *gateway* yang menjalankan layanan berbasis Node.js untuk mengakuisisi data dari kedua PLC. Data yang diperoleh kemudian disimpan ke dalam basis data PostgreSQL dan diteruskan ke sistem Web-SCADA melalui REST API guna mendukung pemantauan dan visualisasi proses secara *real-time*. Dengan konfigurasi tersebut, sistem ini merepresentasikan arsitektur otomasi industri yang terintegrasi, mulai dari perangkat lapangan hingga sistem informasi tingkat atas, sesuai dengan konsep *Modular Production System*.

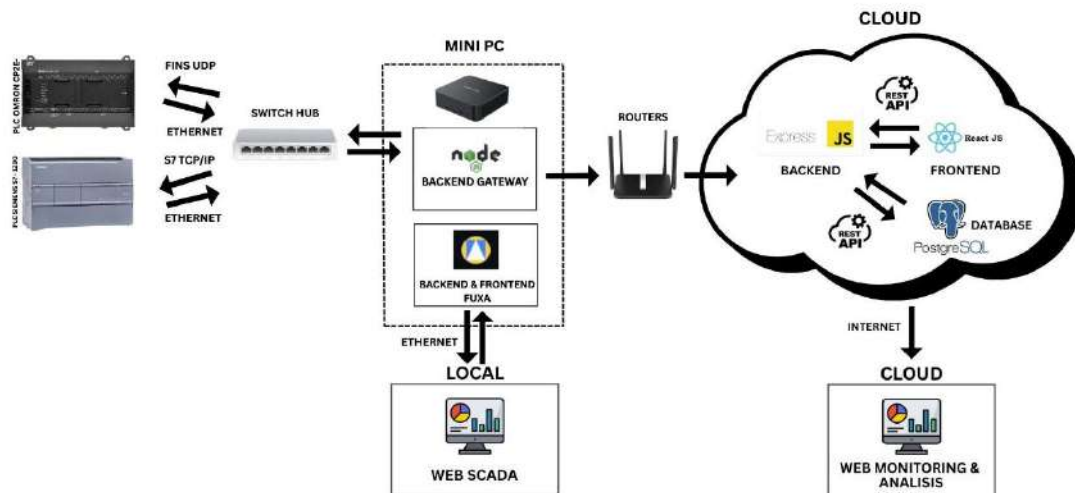


Gambar 3.9 *Wiring Diagram*
(Sumber: Dokumentasi Pribadi, 2026)

3.7 Perencanaan dan Pembuatan *Gateway Connector* Pada Mini PC

Gateway Connector merupakan komponen utama dalam sistem yang berfungsi sebagai penghubung antara PLC dengan sistem supervisi, baik pada level lokal maupun *cloud*. Pada penelitian ini, *Gateway Connector* dirancang dan diimplementasikan pada Mini PC untuk memenuhi kebutuhan integrasi data dari PLC serta mendukung komunikasi lintas sistem secara terpusat dan terstruktur. Perencanaan *gateway* diawali dengan penentuan fungsi utama, yaitu melakukan akuisisi data dari PLC, mengelola komunikasi data, serta meneruskan data ke sistem supervisi lokal dan sistem *Web Monitoring* dan Analisis berbasis *cloud*. *Gateway* dirancang agar mampu menangani komunikasi multi protokol sesuai dengan karakteristik PLC yang digunakan serta menjamin data yang diperoleh bersifat *real-time* dan akurat.

Selain berfungsi sebagai penghubung komunikasi, *Gateway Connector* juga berperan dalam menyesuaikan format data dari masing-masing PLC sebelum diteruskan ke sistem supervisi. Data yang diterima dari PLC Omron melalui protokol FINS UDP dan dari PLC Siemens melalui protokol S7 TCP/IP diproses pada Mini PC, kemudian disusun dalam format yang dapat digunakan oleh Web-SCADA lokal maupun sistem berbasis *cloud*. Dengan mekanisme tersebut, perbedaan protokol komunikasi pada kedua PLC dapat ditangani dalam satu pusat pengolahan data. Arsitektur Sistem *Gateway* dapat dilihat pada gambar 3.10 berikut



Gambar 3.10 Arsitektur Sistem *Gateway*
(Sumber: Dokumentasi Pribadi, 2026)

Gambar 3.10 menunjukkan rancangan sistem yang mengintegrasikan sistem kontrol dan *Monitoring* lokal dengan sistem *Monitoring* berbasis *cloud* melalui sebuah *gateway*. Pada sisi lapangan (OT), PLC Omron CP2E dan PLC Siemens S7-1200 berfungsi sebagai pengendali proses dan sumber data. PLC Omron menggunakan protokol FINS UDP, sedangkan PLC Siemens menggunakan S7 TCP/IP. Kedua PLC terhubung ke jaringan lokal melalui switch hub sehingga memungkinkan komunikasi data yang stabil dalam lingkungan lokal.

Gateway sistem direpresentasikan oleh sebuah Mini PC yang berperan sebagai pusat integrasi antara jaringan lokal dan jaringan internet. Mini PC menjalankan Backend *Gateway* berbasis Node.js untuk melakukan akuisisi data dari PLC. Selain itu, Mini PC juga menjalankan sistem SCADA FUXA sebagai Web SCADA lokal yang menyediakan fungsi *Monitoring* dan kontrol secara langsung melalui jaringan Ethernet lokal. Dengan demikian, sistem kontrol dan *Monitoring* lokal dapat tetap berjalan secara mandiri tanpa ketergantungan pada koneksi internet.

Untuk kebutuhan *Monitoring* jarak jauh, Mini PC mengirimkan data hasil akuisisi ke sistem *cloud* melalui router internet yang berfungsi sebagai *gateway* jaringan. Pada sisi *cloud*, sistem terdiri dari Backend berbasis Node.js dan Express yang menyediakan REST API, database PostgreSQL sebagai media penyimpanan data, serta *Frontend* berbasis React JS sebagai antarmuka pengguna. Data yang tersimpan di *cloud*

ditampilkan dalam bentuk *dashboard* web untuk keperluan *Monitoring* dan analisis jarak jauh, sementara fungsi kontrol utama tetap dipertahankan pada sistem lokal.

3.7.1 Arsitektur Protokol FINS UDP pada *Gateway*

Komunikasi antara *Mini PC* sebagai *gateway* dan PLC Omron CP2E-N30 DRA pada sistem ini menggunakan protokol *Factory Interface Network Service* (FINS) dengan media transport *User Datagram Protocol* (UDP). FINS UDP merupakan protokol komunikasi *native* yang disediakan oleh Omron untuk pertukaran data antar perangkat otomasi pada jaringan lokal berbasis *Ethernet*. Penggunaan FINS UDP memungkinkan *gateway* untuk mengakses area memori PLC secara langsung tanpa memerlukan perangkat lunak tambahan pada sisi PLC.

Pada arsitektur komunikasi ini, *Mini PC* berperan sebagai pihak aktif yang menginisiasi komunikasi (*master*), sedangkan PLC Omron bertindak sebagai pihak yang merespons permintaan (*response node*). Proses komunikasi dilakukan dengan cara *gateway* mengirimkan paket permintaan FINS untuk membaca atau menulis data pada alamat memori tertentu di PLC, kemudian PLC mengirimkan paket balasan yang berisi data sesuai permintaan tersebut. Mekanisme ini menerapkan pola *request-response* yang berjalan di atas protokol UDP.

3.7.1.1 Parameter Koneksi FINS UDP

Agar komunikasi antara PLC Omron CP2E-N30 DRA dan *Mini PC* sebagai *gateway* dapat berjalan dengan baik menggunakan protokol FINS UDP, diperlukan beberapa parameter koneksi yang harus dikonfigurasi dengan benar. Parameter-parameter ini menentukan identitas perangkat, alamat tujuan komunikasi, serta jalur pertukaran data pada jaringan lokal. Parameter Koneksi FINS UDP pada *Gateway* dapat dilihat pada tabel 3.2 berikut.

Tabel 3.2 Parameter Koneksi FINS UDP pada *Gateway*

Parameter	Keterangan
IP Address PLC	Alamat IP PLC Omron sebagai tujuan pengiriman paket FINS UDP.
Port UDP	Nomor port pada layer transport yang digunakan untuk komunikasi FINS UDP sebagai <i>endpoint</i> pengiriman dan penerimaan paket data. Nilai port disesuaikan dengan konfigurasi yang

Parameter	Keterangan
	ditetapkan pada PLC dan <i>gateway</i> .
<i>ICF (Information Control Field)</i>	Field kontrol FINS yang menentukan tipe frame komunikasi.
<i>DNA (Destination Network Address)</i>	Alamat jaringan tujuan FINS
<i>DA1 (Destination Node Address)</i>	Alamat node FINS PLC tujuan.
<i>DA2 (Destination Unit Address)</i>	Alamat unit CPU PLC tujuan.
<i>SNA (Source Network Address)</i>	Alamat jaringan sumber FINS milik <i>gateway</i> .
<i>SA1 (Source Node Address)</i>	Alamat node FINS milik <i>gateway</i> (Mini PC) sebagai pengirim <i>request</i> .

(Sumber: Dokumentasi Pribadi, 2026)

3.7.1.2 Memory Area (FINS UDP)

Memory area pada komunikasi FINS UDP merupakan bagian dari struktur memori PLC yang digunakan untuk menyimpan data operasional dan informasi proses. Setiap *memory area* memiliki fungsi tertentu, tipe data, panjang data, serta hak akses yang menentukan bagaimana data tersebut dapat dibaca atau ditulis melalui mekanisme komunikasi. Struktur *memory area* ini memungkinkan pengelompokan data secara sistematis sesuai dengan kebutuhan pengendalian dan pemantauan proses.

Akses terhadap *memory area* dilakukan menggunakan perintah komunikasi yang mengacu pada alamat memori yang telah ditentukan. Dengan adanya pembagian *memory area* dan pengaturan hak akses, pertukaran data dapat dilakukan secara terkontrol dan terstruktur, sehingga informasi yang tersimpan di dalam PLC dapat dimanfaatkan secara konsisten tanpa mempengaruhi proses internal yang sedang berjalan. Memory Area (FINS UDP) dapat dilihat pada tabel 3.3 berikut.

Tabel 3.3 Memory Area (FINS UDP)

Memory Area	Data Type	Length	Akses Data
<i>Auxiliary area</i>	Bit	1	R/W
	Word	16	
	Dword	32	
	Real	32	

Memory Area	Data Type	Length	Akses Data
	Double	64	
	String	8	
<i>Core Input/Core output area</i>	Bit	1	R/W
	Word	16	
	Dword	32	
<i>Counter</i>	Value (Word)	16	R/W
	Contact (Bit)	1	
<i>Data Memory area</i>	Word	16	R/W
	Dword	32	
	Real	32	
	Double	64	
	String	8	
<i>Data Register</i>	Word	16	R/W
<i>Extended Memory Data area</i>	Word	16	R/W
	Dword	32	
	Real	32	
	Double	64	
	String	8	
<i>Holding area</i>	Bit	1	R/W
	Word	16	
	Dword	32	
	Real	32	
	Double	64	
	String	8	
<i>Index Register</i>	Dword	32	R/W
<i>Task Flag area</i>	Bit	1	R/W
<i>Timer</i>	Value (Word)	16	R/W
	Contact (Bit)	1	
<i>Work area</i>	Bit	1	R/W
	Word	16	
	Dword	32	
	Real	32	

Memory Area	Data Type	Length	Akses Data
	Double	64	
	String	8	

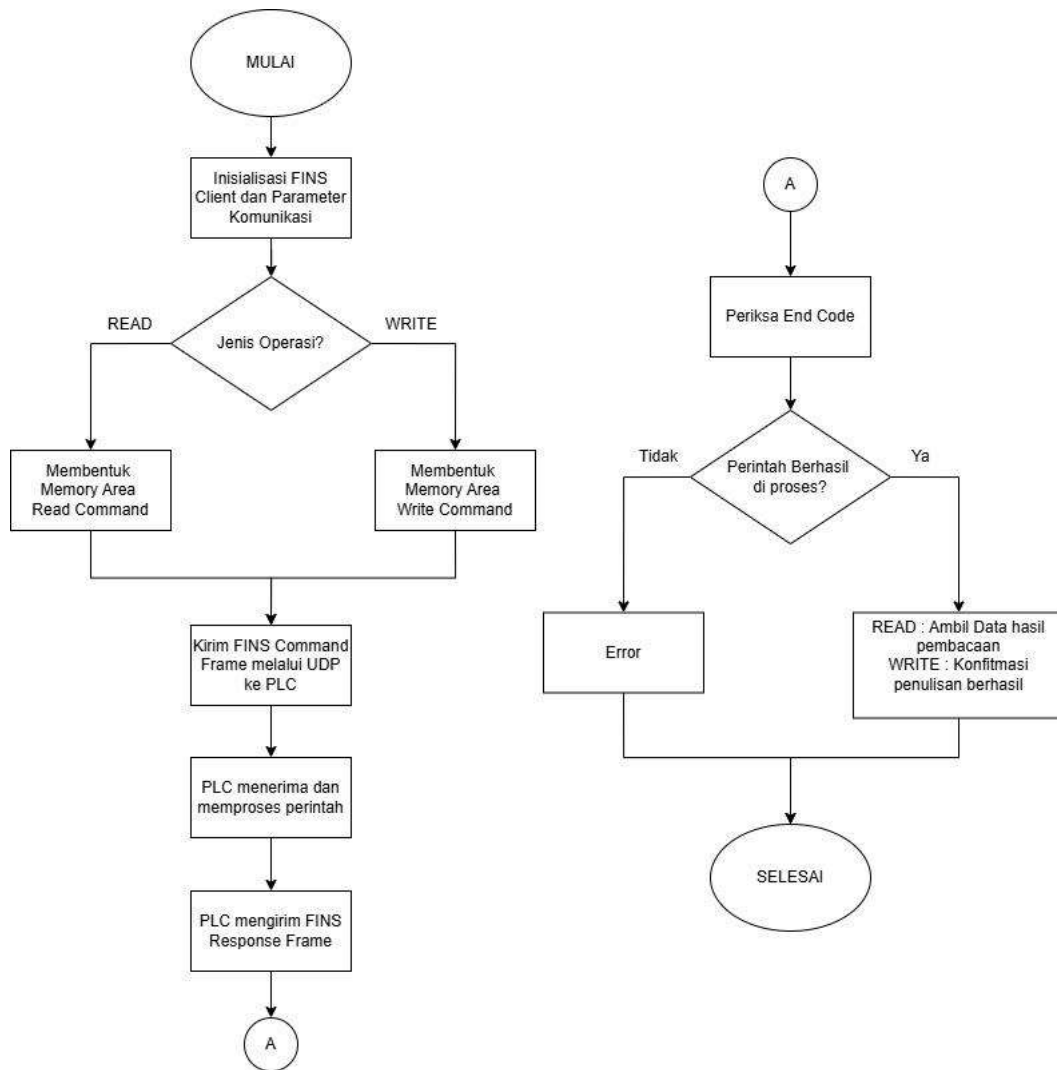
(Sumber: Dokumentasi Pribadi, 2026)

Tabel 3.3 menunjukkan jenis *memory area* yang dapat diakses pada komunikasi FINS UDP beserta tipe data, panjang data, dan hak aksesnya. Area memori yang digunakan meliputi *Auxiliary Area*, *Core Input/Core Output Area*, *Counter*, *Data Memory Area*, *Data Register*, *Extended Memory Data Area*, *Holding Area*, *Index Register*, dan *Task Flag Area*. Setiap area memiliki karakteristik data yang berbeda, seperti *bit*, *word*, *dword*, *real*, *double*, *string*, serta nilai *counter*, dengan panjang data yang menyesuaikan tipe data tersebut. Sebagian besar area memori mendukung proses pembacaan dan penulisan (*read/write*), sehingga aplikasi dapat mengakses data sesuai kebutuhan sistem. Informasi pada tabel ini menjadi dasar dalam menentukan area memori, tipe data, serta ukuran data yang akan digunakan pada proses pertukaran data antara aplikasi Node.js di Mini PC dan PLC Omron melalui komunikasi FINS UDP.

3.7.1.3 Mekanisme Komunikasi FINS UDP

Arsitektur komunikasi FINS UDP merupakan mekanisme pertukaran data antara PLC Omron dan perangkat eksternal melalui jaringan Ethernet berbasis protokol UDP. Pada sistem ini, Mini PC menjalankan aplikasi Node.js yang menggunakan *library* *omron-fins* sebagai antarmuka komunikasi dengan PLC Omron. *Library* tersebut digunakan untuk membentuk *FINS command frame*, menerjemahkan alamat area memori PLC, serta menjalankan operasi pembacaan dan penulisan data menggunakan protokol FINS yang dibawa melalui UDP.

Dalam hubungan komunikasi tersebut, Mini PC berperan sebagai *client* yang membentuk dan mengirimkan permintaan FINS, sedangkan PLC Omron berperan sebagai perangkat tujuan yang menerima serta memproses permintaan tersebut. Mini PC mengirimkan datagram UDP yang berisi *FINS command frame* sesuai operasi yang dijalankan. PLC kemudian memproses perintah tersebut dan mengirimkan *FINS response frame* yang memuat *end code* serta data hasil pembacaan apabila operasi yang dilakukan berupa pembacaan. Alur komunikasi FINS UDP ditunjukkan pada Gambar 3.11.



Gambar 3. 11 *Flowchart* Komunikasi FINS UDP
(Sumber: Dokumentasi Pribadi, 2026)

Gambar 3.11 menunjukkan alur komunikasi FINS UDP antara Mini PC dan PLC Omron. Proses komunikasi diawali dengan inisialisasi *FINS Client* pada aplikasi Node.js beserta parameter komunikasi yang diperlukan, seperti alamat IP PLC, *port*, *Source Node Address* (SA1), *Destination Node Address* (DA1), dan *timeout*. Setelah proses inisialisasi selesai, aplikasi menentukan jenis operasi yang akan dilakukan, yaitu *read* atau *write*. Berdasarkan operasi tersebut, *library* omron-fins membentuk *FINS Command Frame* yang selanjutnya dikirimkan melalui UDP menuju PLC Omron.

PLC menerima *FINS Command Frame* dan memproses perintah sesuai dengan operasi yang diterima. Setelah proses selesai, PLC mengirimkan *FINS Response Frame* kembali ke Mini PC. Respons tersebut kemudian diperiksa melalui *End Code* untuk mengetahui hasil pemrosesan perintah. Apabila perintah berhasil diproses, aplikasi melanjutkan pengolahan respons sesuai jenis operasi yang dijalankan, sedangkan apabila

terjadi kesalahan, kondisi tersebut dapat diidentifikasi berdasarkan *End Code* yang diterima.

Pada sistem ini, pertukaran data antara Mini PC dan PLC Omron dilakukan menggunakan mekanisme *FINS command-response*. Mini PC mengirimkan *FINS Command Frame* sesuai operasi yang dijalankan, kemudian PLC memproses perintah tersebut dan mengembalikan *FINS Response Frame*. Respons yang diterima memuat informasi hasil pemrosesan perintah melalui *End Code* serta data hasil pembacaan apabila operasi yang dilakukan berupa *read*. Parameter *timeout* pada *library omron-fins* digunakan untuk menentukan batas waktu aplikasi dalam menunggu respons dari PLC. Seluruh proses pembentukan *frame*, pengiriman *command*, penerimaan *response*, dan pengolahan hasil komunikasi ditangani oleh *library omron-fins* pada Mini PC.

Setelah mekanisme komunikasi antara Mini PC dan PLC Omron terbentuk, proses pertukaran data dilanjutkan melalui operasi pembacaan dan penulisan pada area memori PLC. Pada tahap ini, *library omron-fins* digunakan untuk menjalankan perintah *Memory Area Read* dan *Memory Area Write* sesuai alamat dan data yang dibutuhkan oleh aplikasi. Penjelasan mengenai mekanisme pembacaan dan penulisan data tersebut dibahas pada subbab berikutnya.

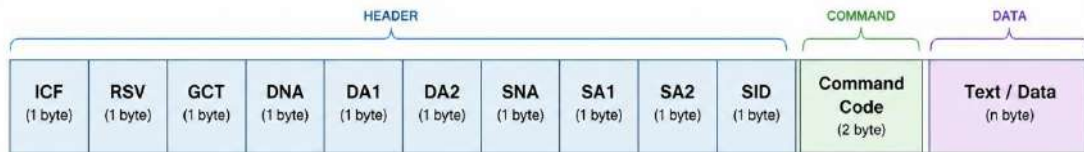
3.7.1.4 Mekanisme Pembacaan dan Penulisan Data PLC Menggunakan Omron-fins

Proses pembacaan dan penulisan data PLC pada sistem dilakukan menggunakan *library omron-fins* yang dijalankan pada aplikasi Node.js di Mini PC. *Library* tersebut menyediakan fungsi untuk mengakses area memori PLC melalui protokol FINS tanpa mengharuskan aplikasi membentuk setiap bagian *frame* secara manual. Operasi pembacaan dilakukan menggunakan perintah *Memory Area Read*, sedangkan operasi penulisan dilakukan menggunakan perintah *Memory Area Write*.

Pada saat aplikasi menjalankan fungsi pembacaan atau penulisan, parameter seperti area memori, alamat awal, jumlah data, serta nilai yang akan ditulis diterjemahkan oleh *library omron-fins* menjadi struktur perintah FINS yang sesuai. Perintah tersebut kemudian dikemas ke dalam *FINS Command Frame* dan dikirimkan menuju PLC Omron. PLC memproses perintah yang diterima dan mengembalikan *FINS Response Frame* sebagai hasil dari operasi yang dijalankan.

Mekanisme pembacaan dan penulisan memiliki struktur data yang berbeda sesuai dengan fungsi masing-masing. *Memory Area Read Command* digunakan untuk menentukan lokasi dan jumlah data yang akan dibaca dari PLC, sedangkan *Memory Area*

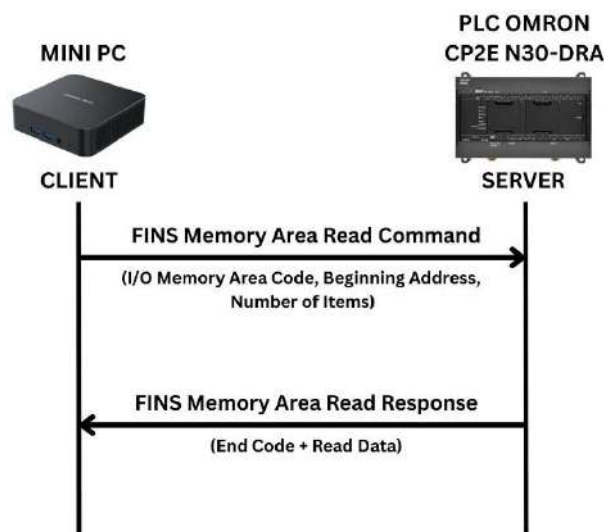
Write Command selain memuat informasi alamat juga membawa data yang akan dituliskan. Struktur kedua perintah tersebut dijelaskan pada bagian berikutnya untuk menunjukkan bagaimana parameter pada aplikasi diterjemahkan menjadi perintah FINS.



Gambar 3. 12 Struktur FINS Frame
(Sumber: Dokumentasi Pribadi, 2026)

Gambar 3.12 menunjukkan struktur umum *FINS Frame* yang digunakan dalam pertukaran data antara Mini PC dan PLC Omron. Struktur tersebut terdiri atas bagian *header*, *Command Code*, dan *Text/Data*. Bagian *header* memuat informasi kontrol komunikasi, alamat jaringan serta node sumber dan tujuan, serta *Service ID* (SID). *Field* SA1 dan DA1 digunakan untuk menunjukkan alamat node sumber dan tujuan, sedangkan SID digunakan untuk membantu mengidentifikasi transaksi antara perintah yang dikirim dan respons yang diterima.

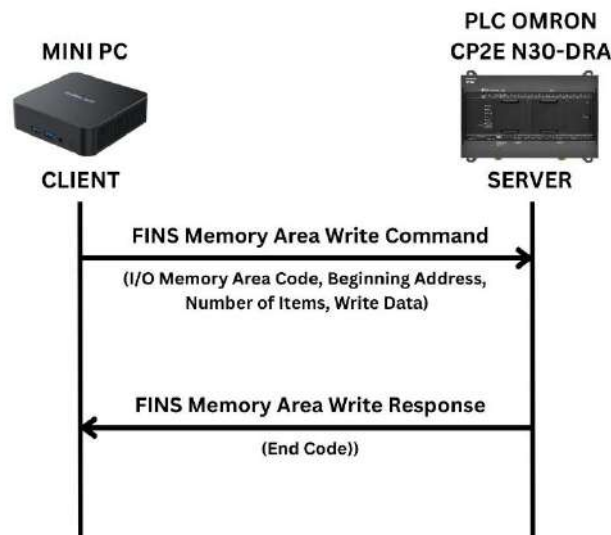
Bagian *Command Code* digunakan untuk menentukan jenis perintah FINS yang dijalankan, seperti *Memory Area Read* atau *Memory Area Write*. Selanjutnya, bagian *Text/Data* memuat parameter yang diperlukan oleh perintah tersebut, seperti *Memory Area Code*, *Start Address*, *Bit Position*, dan *Item Count*. Pada operasi *write*, bagian *Text/Data* juga membawa nilai yang akan dituliskan ke PLC. Pembentukan *frame* beserta parameter tersebut dilakukan secara otomatis oleh *library* omron-fins berdasarkan operasi yang dijalankan oleh aplikasi Node.js.



Gambar 3. 13 *Sequence Diagram* Mekanisme Pembacaan Data PLC Menggunakan Omron-fins
(Sumber: Dokumentasi Pribadi, 2026)

Gambar 3.13 menunjukkan *sequence diagram* mekanisme pembacaan data PLC menggunakan *library* omron-fins. Mini PC sebagai *client* mengirimkan *FINS Memory Area Read Command* menuju PLC Omron sebagai *server*. Perintah tersebut memuat *I/O Memory Area Code* untuk menentukan area memori yang diakses, *Beginning Address* untuk menentukan alamat awal pembacaan, serta *Number of Items* untuk menentukan jumlah data yang akan dibaca.

Setelah menerima perintah, PLC membaca data dari area memori sesuai parameter yang diberikan kemudian mengirimkan *FINS Memory Area Read Response* kembali ke Mini PC. Respons tersebut memuat *End Code* sebagai informasi hasil pemrosesan perintah serta *Read Data* sebagai data hasil pembacaan apabila perintah berhasil dijalankan. Selanjutnya, *library* omron-fins mengolah respons tersebut agar data dapat digunakan oleh aplikasi Node.js.



Gambar 3. 14 *Sequence Diagram* Mekanisme Penulisan Data PLC Menggunakan Omron-fins
(Sumber: Dokumentasi Pribadi, 2026)

Gambar 3.14 menunjukkan *sequence diagram* mekanisme penulisan data PLC menggunakan *library* omron-fins. Mini PC sebagai *client* mengirimkan *FINS Memory Area Write Command* menuju PLC Omron. Perintah tersebut memuat *I/O Memory Area Code*, *Beginning Address*, *Number of Items*, serta *Write Data*. Parameter tersebut digunakan untuk menentukan area dan alamat awal memori tujuan, jumlah data yang akan ditulis, serta nilai yang akan disimpan pada PLC.

Setelah menerima perintah, PLC melakukan proses penulisan sesuai parameter dan data yang diterima, kemudian mengirimkan *FINS Memory Area Write Response* kembali ke Mini PC. Respons penulisan memuat *End Code* yang digunakan untuk

menunjukkan hasil pemrosesan perintah. Apabila *End Code* menunjukkan kondisi berhasil, aplikasi dapat mengonfirmasi bahwa perintah penulisan telah diproses oleh PLC. Seluruh proses pembentukan perintah dan pengolahan respons tersebut ditangani oleh *library* omron-fins pada aplikasi Node.js.

Parameter yang digunakan pada proses pembacaan dan penulisan data FINS mencakup informasi area memori, alamat awal, jumlah data, serta data yang dipertukarkan antara Mini PC dan PLC. Pada *FINS Memory Area Read Command*, parameter utama terdiri atas *I/O Memory Area Code*, *Beginning Address*, dan *Number of Items*, sedangkan pada *FINS Memory Area Write Command* terdapat tambahan *Write Data* sebagai nilai yang akan dituliskan ke PLC. Pada implementasi menggunakan *library* omron-fins, parameter tersebut tidak seluruhnya dimasukkan secara terpisah oleh aplikasi karena alamat seperti D0 atau D0.0 diterjemahkan oleh *library* menjadi informasi area memori dan alamat awal yang sesuai dengan format FINS. Selanjutnya, *library* membentuk *FINS Memory Area Read Command* atau *FINS Memory Area Write Command* sebelum perintah dikirimkan menuju PLC. Penjelasan masing-masing parameter ditunjukkan pada Tabel 3.4.

Tabel 3. 4 Parameter pada *FINS Memory Area Read* dan *FINS Memory Area Write*

Parameter	Digunakan pada	Penjelasan
<i>I/O Memory area code</i>	<i>Read dan write</i>	Menentukan area memori PLC yang akan diakses, seperti DM, CIO, WR, HR, atau area memori lain yang didukung oleh PLC.
<i>Beginning Address</i>	<i>Read dan write</i>	Menentukan alamat awal pada area memori yang akan dibaca atau ditulis. Alamat ini digunakan sebagai titik awal proses akses data pada PLC.
<i>Number of Items</i>	<i>Read dan write</i>	Berisi data yang akan dituliskan ke area memori PLC sesuai jumlah item yang ditentukan.
<i>Write Data</i>	<i>Write</i>	Menentukan posisi bit tertentu di dalam satu alamat <i>byte</i> . Parameter ini digunakan ketika data yang diakses berupa status logika atau Boolean.

Parameter	Digunakan pada	Penjelasan
<i>End Code</i>	<i>Read response</i> dan <i>write response</i>	Menunjukkan hasil pemrosesan perintah oleh PLC, baik berhasil maupun terjadi kondisi kesalahan.
<i>Read Data</i>	<i>Read response</i>	Berisi data hasil pembacaan dari area memori PLC sesuai alamat dan jumlah item yang diminta.

(Sumber: Diolah oleh penulis berdasarkan OMRON Communications Commands Reference Manual, 2026)

Berdasarkan Tabel 3.4, parameter pada proses pembacaan dan penulisan data FINS terdiri atas *I/O Memory Area Code*, *Beginning Address*, *Number of Items*, *Write Data*, *End Code*, dan *Read Data*. Parameter *I/O Memory Area Code*, *Beginning Address*, dan *Number of Items* digunakan pada operasi *read* maupun *write* untuk menentukan area memori, alamat awal, dan jumlah data yang akan diakses. Pada operasi *write*, perintah juga membawa *Write Data* sebagai nilai yang akan dituliskan ke PLC, sedangkan pada operasi *read* data hasil pembacaan dikembalikan melalui *Read Data*.

Setelah perintah diproses oleh PLC, *End Code* digunakan untuk menunjukkan hasil eksekusi perintah, baik pada operasi *read* maupun *write*. Pada implementasi menggunakan *library* omron-fins, parameter tersebut tidak seluruhnya dimasukkan secara terpisah oleh aplikasi, karena alamat dan jumlah data yang diberikan pada fungsi *read()* atau *write()* diterjemahkan oleh *library* menjadi struktur parameter FINS yang sesuai. Untuk memperjelas proses tersebut, contoh penerapan pembacaan dan penulisan data pada alamat D0 dijelaskan pada bagian berikutnya.

Pada implementasi menggunakan *library* omron-fins, aplikasi tidak membentuk parameter FINS tersebut secara langsung. *Library* menerjemahkan alamat dan data yang diberikan melalui fungsi *read()* atau *write()* menjadi *FINS Memory Area Read Command* atau *FINS Memory Area Write Command*. Sebagai contoh penerapan, digunakan alamat D0 pada PLC Omron untuk menunjukkan proses pembacaan dan penulisan data pada area DM (*Data Memory*).

```

1  const fins = require('omron-fins');
2
3  const client = fins.FinsClient(
4    9600,
5    '192.168.1.120',
6    {
7      SA1: 36,
8      DA1: 120,
9      timeout: 2000
10   }
11 );
12
13 // Membaca 1 word dari D0
14 client.read('D0', 1, (error, msg) => {
15   if (error) {
16     console.error('Pembacaan D0 gagal');
17     return;
18   }
19
20   console.log('Nilai D0:', msg.response.values);
21 });
22
23 // Menulis nilai 100 ke D0
24 client.write('D0', [100], (error, msg) => {
25   if (error) {
26     console.error('Penulisan D0 gagal');
27     return;
28   }
29
30   console.log('Penulisan D0 berhasil');
31 });

```

Gambar 3. 15 Contoh Program Pembacaan dan Penulisan Alamat D0 Menggunakan Omron-fins
(Sumber: Dokumentasi Pribadi, 2026)

Gambar 3.15 menunjukkan contoh implementasi pembacaan dan penulisan data PLC pada alamat D0 menggunakan *library* omron-fins. Aplikasi terlebih dahulu membentuk *FINS Client* dengan menggunakan *port* 9600, alamat IP PLC 192.168.0.5, SA1 sebesar 1, DA1 sebesar 5, serta *timeout* 2000 ms. Parameter tersebut digunakan oleh *library* untuk menentukan konfigurasi komunikasi antara Mini PC sebagai *client* dan PLC Omron sebagai perangkat tujuan

Pada proses pembacaan, fungsi `client.read('D0', 1, ...)` digunakan untuk membaca satu *word* mulai dari alamat D0. Parameter D0 menunjukkan alamat memori yang akan diakses, sedangkan nilai 1 menunjukkan jumlah item yang dibaca. Setelah respons diterima, hasil pembacaan diambil dari `msg.response.values` dan ditampilkan pada *console*. Apabila proses pembacaan mengalami kesalahan, aplikasi menampilkan informasi bahwa pembacaan D0 gagal melalui fungsi *callback*.

Pada proses penulisan, fungsi `client.write('D0', [100], ...)` digunakan untuk menuliskan nilai 100 ke alamat D0. Alamat D0 menentukan lokasi memori tujuan, sedangkan [100] merupakan data yang dikirimkan untuk ditulis ke PLC. Setelah perintah

diproses, fungsi *callback* digunakan untuk mengetahui apakah operasi penulisan berhasil atau mengalami kesalahan. Seluruh proses penerjemahan alamat, pembentukan *FINS Memory Area Read/Write Command*, serta pengolahan respons dilakukan oleh *library omron-fins* sehingga aplikasi tidak perlu membentuk *FINS frame* secara manual.

Alamat D0 menunjukkan data bertipe *word* pada area DM (*Data Memory*) PLC Omron. Karena operasi dilakukan pada level *word*, area DM direpresentasikan menggunakan *I/O Memory Area Code* 82 hex sesuai format FINS. Alamat D0 menjadi *Beginning Address* dengan alamat *word* 0, sedangkan nilai 1 pada fungsi *read()* menunjukkan bahwa jumlah data yang diminta adalah satu *word*. Pada proses penulisan, alamat yang digunakan tetap D0 dengan nilai 100 sebagai data yang akan dituliskan. Parameter pembacaan dan penulisan pada alamat D0 ditunjukkan pada Tabel 3.5

Tabel 3. 5 Parameter Pembacaan dan Penulisan pada Alamat D0

Parameter	Pembacaan	Penulisan
<i>Address</i>	D0	D0
<i>Memory area</i>	<i>DM (Data Memory)</i>	<i>DM (Data Memory)</i>
<i>I/O Memory Area Code</i>	82 hex (<i>word</i>)	82 hex (<i>word</i>)
<i>Beginning Address</i>	<i>D0 (word 0)</i>	<i>D0 (word 0)</i>
<i>Number of Items</i>	1 <i>word</i>	1 <i>word</i>
<i>Write Data</i>	-	100
<i>End Code</i>	Menunjukkan hasil proses pembacaan	Menunjukkan hasil proses penulisan
<i>Read Data</i>	Nilai D0	-
Hasil pada aplikasi	Nilai D0 diperoleh melalui <i>msg.response.values</i>	Informasi keberhasilan atau kegagalan penulisan diperoleh melalui <i>callbac</i>

(Sumber: Diolah oleh penulis berdasarkan dokumentasi OMRON FINS dan implementasi *omron-fins*, 2026)

Berdasarkan Tabel 3.5, proses pembacaan dan penulisan menggunakan alamat yang sama, yaitu D0 pada area DM (*Data Memory*). Pada akses *word*, area DM menggunakan *I/O Memory Area Code* 82 hex, sedangkan D0 menjadi *Beginning Address* dengan alamat *word* 0. Pada operasi pembacaan, *Number of Items* bernilai 1 sehingga PLC membaca satu *word* dari alamat D0. Hasil pembacaan dikembalikan sebagai *Read Data* dan dapat diakses pada aplikasi melalui *msg.response.values*.

Pada operasi penulisan, parameter area memori, *Beginning Address*, dan *Number of Items* tetap sama, tetapi ditambahkan *Write Data* bernilai 100 sebagai data yang akan disimpan pada D0. Setelah perintah diproses, PLC mengembalikan *End Code* untuk menunjukkan hasil operasi. Pada implementasi omron-fins, hasil proses penulisan selanjutnya ditangani melalui fungsi *callback* untuk mengetahui apakah operasi berhasil atau mengalami kesalahan

3.7.2 Arsitektur Protokol S7 TCP/IP pada Gateway

Perancangan Komunikasi antara Mini PC sebagai *gateway* dan PLC Siemens S7-1200 pada sistem ini menggunakan protokol S7 TCP/IP yang berjalan di atas media transport Transmission Control Protocol (TCP). S7 TCP/IP merupakan protokol komunikasi native Siemens yang digunakan untuk pertukaran data antar PLC dan perangkat eksternal pada jaringan Ethernet industri. Penggunaan protokol ini memungkinkan *gateway* untuk mengakses area memori PLC secara langsung melalui koneksi jaringan tanpa memerlukan perangkat tambahan pada sisi PLC.

Pada arsitektur komunikasi ini, Mini PC berperan sebagai pihak aktif yang menginisiasi koneksi (*client*), sedangkan PLC Siemens bertindak sebagai pihak yang menerima dan merespons permintaan (*server*). Proses komunikasi dilakukan melalui koneksi TCP yang bersifat *connection-oriented*, di mana *gateway* membuka koneksi ke PLC sebelum melakukan pembacaan atau penulisan data pada alamat memori tertentu. Setelah koneksi terbentuk, pertukaran data dilakukan secara terstruktur dan berkelanjutan hingga koneksi ditutup, sehingga mekanisme komunikasi berlangsung dengan pola *request-response* yang andal.

3.7.2.1 Parameter Koneksi S7 TCP/IP

Agar komunikasi antara PLC Siemens dan *Mini PC* sebagai *gateway* dapat berjalan dengan baik menggunakan protokol S7 TCP/IP, diperlukan konfigurasi beberapa parameter koneksi yang sesuai. Parameter-parameter ini berfungsi untuk menentukan identitas perangkat, alamat tujuan komunikasi, serta mekanisme pertukaran data pada jaringan Ethernet. Konfigurasi yang tepat memastikan *gateway* dapat membangun koneksi dengan PLC secara andal sehingga proses pembacaan dan penulisan data dapat berlangsung secara stabil tanpa mengganggu operasi internal PLC. Parameter Koneksi S7 TCP / IP pada *Gateway* dapat dilihat pada tabel 3.6 berikut.

Tabel 3.6 Parameter Koneksi S7 TCP / IP pada *Gateway*

Parameter	Keterangan
<i>Transport</i>	Media komunikasi yang digunakan untuk pertukaran data antara <i>gateway</i> dan PLC Siemens melalui jaringan Ethernet berbasis protokol ISO-on-TCP.
<i>IP Address</i>	Alamat tujuan koneksi berupa alamat IP PLC yang digunakan oleh <i>gateway</i> untuk membangun komunikasi jaringan.
<i>Port</i>	Nomor port pada layer transport yang digunakan untuk komunikasi S7 TCP/IP sebagai <i>endpoint</i> pertukaran data antara <i>gateway</i> dan PLC.
<i>Mode</i>	Metode identifikasi perangkat PLC
<i>Rack</i>	Parameter yang menunjukkan posisi rak CPU PLC dalam sistem
<i>Slot</i>	Parameter yang menunjukkan posisi slot CPU PLC dalam rak,

(Sumber: Dokumentasi Pribadi, 2026)

3.7.2.2 Memory Area (S7 TCP/IP)

Memory area pada komunikasi S7 TCP/IP merupakan bagian dari struktur memori PLC yang digunakan untuk menyimpan data operasional dan informasi proses. Setiap *memory area* memiliki fungsi, tipe data, kapasitas, serta hak akses tertentu yang menentukan bagaimana data tersebut dapat diakses melalui mekanisme komunikasi S7. Pembagian *memory area* ini memungkinkan pengelolaan data dilakukan secara sistematis sesuai dengan kebutuhan pengendalian dan integrasi data pada sistem otomasi.

Akses terhadap *memory area* dilakukan melalui perintah komunikasi yang mengacu pada alamat memori yang telah ditentukan. Dengan adanya pengelompokan *memory area* dan pengaturan hak akses yang jelas, pertukaran data antara PLC dan perangkat eksternal dapat berlangsung secara terkontrol dan terstruktur, sehingga data yang tersimpan di dalam PLC dapat dimanfaatkan secara konsisten tanpa mengganggu proses internal yang sedang berjalan. Memory Area (S7 TCP/IP) dapat dilihat pada tabel 3.7 berikut.

Tabel 3.7 Memory Area (S7 TCP/IP)

Memory Area	Data Type	Length	Akses Data
<i>Process Image Input (I)</i>	Bit / Byte / Word / Dword	1–32 bit	R
<i>Process Image Output (Q)</i>	Bit / Byte / Word / Dword	1–32 bit	R/W
<i>Bit Memory (M)</i>	Bit / Byte / Word / Dword	1–32 bit	R/W
<i>Data Block (DB)</i>	Byte / Word / Dword / Struktur	Variabel	R/W
<i>Timer (T)</i>	Time	32 bit	R
<i>Counter (C)</i>	Integer	16 bit	R

(Sumber: Dokumentasi Pribadi, 2026)

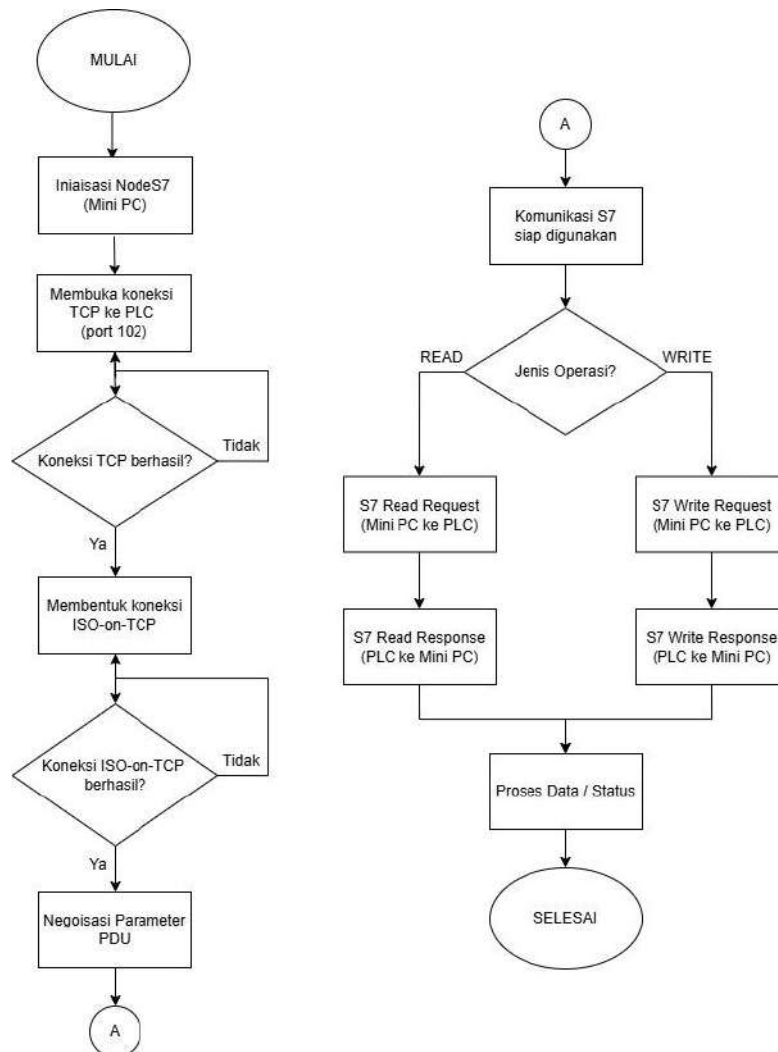
Tabel 3.8 menunjukkan jenis *memory area* pada komunikasi S7 TCP/IP beserta tipe data, panjang data, dan hak aksesnya. *Process Image Input (I)* digunakan untuk membaca kondisi masukan PLC, sedangkan *Process Image Output (Q)*, *Bit Memory (M)*, dan *Data Block (DB)* dapat digunakan untuk proses pembacaan maupun penulisan data. Selain itu, area *Timer (T)* dan *Counter (C)* digunakan untuk menyimpan nilai waktu dan pencacahan yang dapat dibaca oleh aplikasi. Perbedaan karakteristik setiap *memory area* tersebut menjadi dasar dalam menentukan alamat dan tipe data yang digunakan pada proses komunikasi antara aplikasi Node.js dan PLC Siemens S7-1200.

3.7.2.3 Mekanisme Komunikasi S7 TCP/IP

Arsitektur komunikasi S7 TCP/IP merupakan mekanisme pertukaran data antara PLC Siemens dan perangkat eksternal melalui jaringan Ethernet berbasis protokol TCP/IP. Pada sistem ini, Mini PC menjalankan aplikasi Node.js yang menggunakan *library* NodeS7 sebagai antarmuka komunikasi dengan PLC Siemens S7-1200. NodeS7 berfungsi untuk membentuk koneksi, menerjemahkan alamat data PLC, serta menjalankan operasi pembacaan dan penulisan menggunakan S7 Communication yang berjalan di atas ISO-on-TCP.

Dalam hubungan komunikasi tersebut, Mini PC berperan sebagai *client* yang menginisiasi koneksi dan mengirimkan permintaan, sedangkan PLC Siemens S7-1200 berperan sebagai *server* yang menerima serta memproses permintaan komunikasi.

Pertukaran data baru dapat dilakukan setelah koneksi TCP, ISO-on-TCP, dan tahapan persiapan komunikasi S7 berhasil dibentuk.

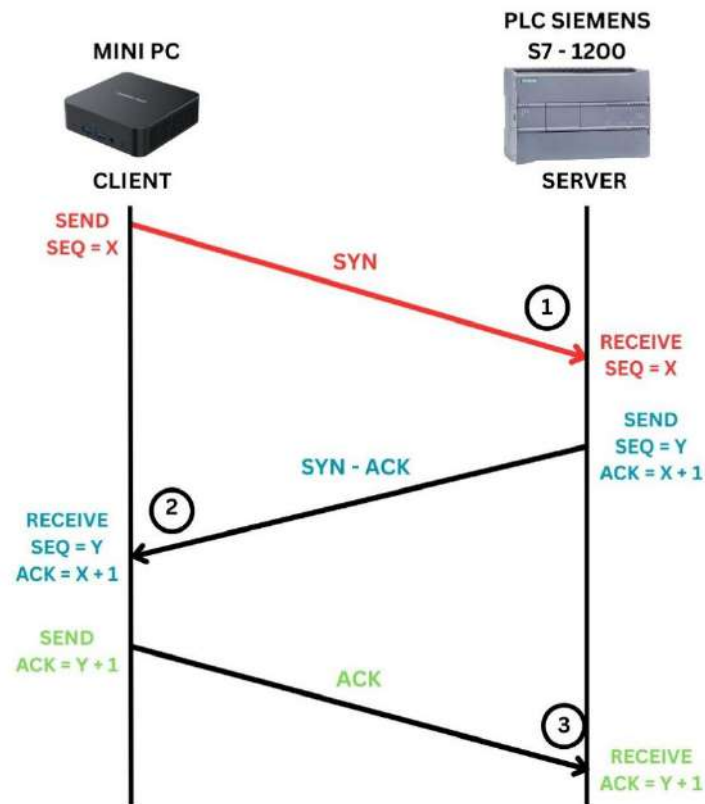


Gambar 3. 16 *Flowchart* Komunikasi S7 TCP/IP
(Sumber: Dokumentasi Pribadi, 2026)

Berdasarkan Gambar 3.16, proses komunikasi diawali dengan inisialisasi NodeS7 pada Mini PC. NodeS7 kemudian membuka koneksi TCP menuju PLC melalui port 102, membentuk koneksi ISO-on-TCP, dan melakukan negosiasi parameter PDU. Setelah koneksi siap digunakan, NodeS7 dapat mengirimkan permintaan pembacaan atau penulisan data kepada PLC dan memproses respons yang diterima.

Proses komunikasi diawali dengan pembentukan koneksi TCP melalui mekanisme *three-way handshake*. Setelah koneksi TCP mencapai kondisi *established*, komunikasi dilanjutkan dengan pembentukan koneksi ISO-on-TCP dan negosiasi parameter PDU. Setelah seluruh tahapan koneksi berhasil, pertukaran data dilakukan menggunakan mekanisme *request-response*. Pada operasi pembacaan, Mini PC mengirimkan permintaan yang memuat spesifikasi alamat data, kemudian PLC

mengembalikan status dan data hasil pembacaan. Sementara itu, pada operasi penulisan, Mini PC mengirimkan spesifikasi alamat tujuan beserta data yang akan ditulis, kemudian PLC mengembalikan status keberhasilan atau kesalahan penulisan.



Gambar 3. 17 Mekanisme TCP Three-Way Handshake antara Mini PC dan PLC Siemens S7-1200
(Sumber: Dokumentasi Pribadi, 2026)

Gambar 3.17 menunjukkan mekanisme pembentukan koneksi TCP antara Mini PC sebagai *client* dan PLC Siemens S7-1200 sebagai *server* melalui proses *three-way handshake*. Pada tahap pertama, Mini PC mengirimkan segmen SYN kepada PLC dengan *initial sequence number* yang dilambangkan sebagai X. Setelah segmen SYN dikirim, Mini PC berada pada kondisi *SYN-SENT* karena sedang menunggu respons dari PLC.

Pada tahap kedua, PLC menerima segmen SYN dengan $SEQ = X$, kemudian membalas menggunakan segmen SYN-ACK. Dalam respons tersebut, PLC mengirimkan *initial sequence number* miliknya sendiri yang dilambangkan sebagai Y, serta *acknowledgment number* sebesar $X + 1$. Nilai $X + 1$ menunjukkan bahwa PLC telah menerima SYN dengan nomor urut X dan mengharapkan nomor urut berikutnya. Pada tahap ini, PLC berada pada kondisi *SYN-RECEIVED* karena telah mengirim SYN-ACK tetapi masih menunggu ACK terakhir dari Mini PC.

Pada tahap ketiga, Mini PC menerima SYN-ACK dari PLC, kemudian mengirimkan segmen ACK dengan *acknowledgment number* sebesar $Y + 1$. Nilai

tersebut menunjukkan bahwa Mini PC telah menerima SYN milik PLC dengan $SEQ = Y$ dan mengharapkan nomor urut berikutnya. Secara teknis, segmen ACK terakhir juga memiliki $SEQ = X + 1$ karena nomor urut X sebelumnya telah digunakan oleh SYN dari Mini PC.

Setelah ACK terakhir diterima oleh PLC, kedua perangkat berada pada kondisi *ESTABLISHED*. Kondisi ini menunjukkan bahwa koneksi TCP telah berhasil dibentuk, kedua perangkat telah saling bertukar dan mengakui *initial sequence number*, serta jalur komunikasi dua arah telah siap digunakan. Selanjutnya, koneksi TCP tersebut digunakan sebagai media transport bagi pertukaran data S7 antara Mini PC dan PLC Siemens S7-1200.

Setelah koneksi TCP berhasil dibentuk, NodeS7 melanjutkan proses dengan membentuk koneksi ISO-on-TCP. Pada tahap ini, NodeS7 mengirimkan ISO-on-TCP *Connection Request* kepada PLC. Permintaan tersebut digunakan untuk membentuk koneksi logis di atas koneksi TCP dan memuat parameter utama berupa TSAP sumber, TSAP tujuan, serta parameter transport. TSAP digunakan untuk menentukan titik akses komunikasi dan tujuan logis CPU PLC. PLC kemudian memeriksa parameter koneksi tersebut dan mengirimkan ISO-on-TCP *Connection Confirm* apabila koneksi dapat diterima. Apabila respons konfirmasi berhasil diterima dan dinyatakan valid, koneksi ISO-on-TCP dianggap telah terbentuk. Tahap ini belum memuat perintah pembacaan atau penulisan alamat PLC, tetapi hanya menyediakan koneksi logis yang digunakan untuk membawa pesan komunikasi S7.

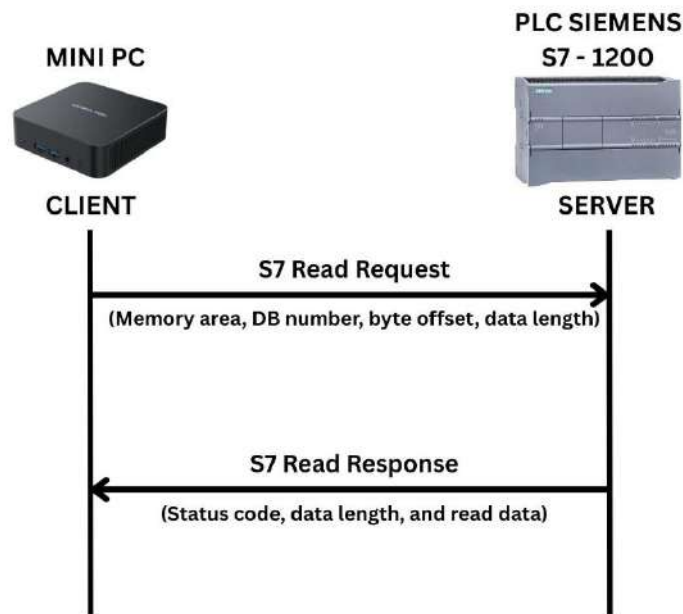
Setelah koneksi ISO-on-TCP terbentuk, NodeS7 dan PLC melakukan negosiasi parameter PDU. Tahap ini bertujuan untuk menentukan batas ukuran pesan komunikasi yang dapat digunakan selama proses pertukaran data. NodeS7 mengirimkan parameter yang diminta, kemudian PLC mengembalikan nilai yang didukung. Hasil negosiasi tersebut digunakan sebagai batas dalam penyusunan permintaan pembacaan dan penulisan data. Apabila proses negosiasi berhasil, koneksi komunikasi S7 dinyatakan siap digunakan untuk menjalankan operasi pembacaan dan penulisan data PLC. Seluruh tahapan tersebut ditangani secara otomatis oleh *library* NodeS7 pada Mini PC.

3.7.2.4 Mekanisme Pembacaan dan Penulisan Data PLC Menggunakan NodeS7

Setelah koneksi TCP dan ISO-on-TCP terbentuk serta negosiasi PDU berhasil dilakukan, Mini PC dapat melakukan pembacaan dan penulisan data pada PLC Siemens S7-1200 melalui *library* NodeS7. Pada implementasi penelitian ini, aplikasi Node.js tidak

menyusun paket komunikasi S7 secara manual. Aplikasi hanya menentukan alamat data dan jenis operasi yang akan dilakukan, sedangkan NodeS7 menerjemahkan alamat tersebut, membentuk permintaan komunikasi, mengirimkannya kepada PLC, serta memproses respons yang diterima.

Pada proses pembacaan, NodeS7 mengirimkan S7 Read Request yang memuat spesifikasi alamat dan jumlah data yang akan dibaca. PLC kemudian mengakses data pada alamat yang ditentukan dan mengirimkan S7 Read Response yang berisi status serta data hasil pembacaan. Pada proses penulisan, NodeS7 mengirimkan S7 Write Request yang memuat alamat tujuan dan data yang akan ditulis. Setelah data diproses, PLC mengirimkan S7 Write Response yang berisi status keberhasilan atau kesalahan penulisan. Mekanisme pembacaan dan penulisan data tersebut ditunjukkan pada Gambar 3.18 dan 3.19.

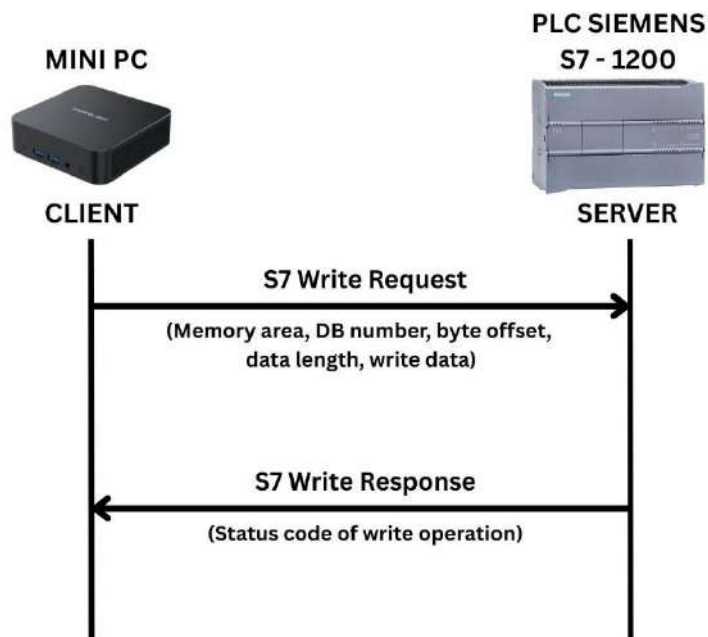


Gambar 3. 18 *Sequence Diagram* Mekanisme Pembacaan Data PLC Menggunakan NodeS7
(Sumber: Dokumentasi Pribadi, 2026)

Gambar 3.18 menunjukkan mekanisme pembacaan data antara Mini PC sebagai *client* dan PLC Siemens S7-1200 sebagai *server*. Proses dimulai ketika Mini PC melalui NodeS7 mengirimkan S7 Read Request kepada PLC. Permintaan tersebut memuat informasi alamat data yang akan dibaca, yaitu *area*, *DB number* apabila data berada pada *data block*, *byte offset*, dan *data length*.

Setelah menerima permintaan, PLC mengakses area memori sesuai alamat yang diberikan. PLC kemudian mengirimkan S7 Read Response kepada Mini PC. Respons tersebut memuat *status code* yang menunjukkan keberhasilan atau kegagalan proses

pembacaan, *data length*, serta *read data* sebagai hasil pembacaan dari memori PLC. Selanjutnya, NodeS7 memproses data yang diterima dan mengubahnya menjadi nilai sesuai tipe data yang digunakan oleh aplikasi.



Gambar 3. 19 *Sequence Diagram* Mekanisme Penulisan Data PLC Menggunakan NodeS7
(Sumber: Dokumentasi Pribadi, 2026)

Gambar 3.19 menunjukkan mekanisme penulisan data antara Mini PC sebagai *client* dan PLC Siemens S7-1200 sebagai *server*. Proses dimulai ketika Mini PC melalui NodeS7 mengirimkan S7 Write Request kepada PLC. Permintaan tersebut memuat informasi alamat tujuan, yaitu *area*, *DB number* apabila data berada pada *data block*, *byte offset*, *data length*, serta *write data* sebagai nilai yang akan dituliskan.

Setelah menerima permintaan, PLC memproses data dan menuliskannya ke area memori sesuai alamat yang ditentukan. Setelah proses penulisan selesai, PLC mengirimkan S7 Write Response kepada Mini PC. Respons tersebut berisi *status code of write operation* yang menunjukkan apakah proses penulisan berhasil atau mengalami kesalahan. NodeS7 kemudian meneruskan status tersebut kepada aplikasi.

Parameter komunikasi pada proses pembacaan dan penulisan data tidak hanya terdiri atas alamat memori, tetapi juga mencakup informasi area, nomor *data block*, alamat awal, jumlah data, tipe transport, serta nilai yang ditulis pada operasi penulisan. Pada implementasi NodeS7, parameter tersebut diperoleh dari hasil penerjemahan alamat yang dimasukkan oleh aplikasi. NodeS7 kemudian menyusunnya menjadi spesifikasi

alamat dan data pada paket S7 sebelum dikirimkan kepada PLC. Penjelasan masing-masing parameter ditunjukkan pada Tabel 3.8.

Tabel 3. 8 Parameter pada S7 Read Request, S7 Write Request, dan Respons PLC

Parameter	Digunakan pada	Penjelasan
<i>Memory area</i>	<i>Read request</i> dan <i>write request</i>	Menentukan jenis area memori PLC yang akan diakses, misalnya area <i>input</i> , <i>output</i> , <i>marker</i> , atau <i>data block</i> .
<i>DB number</i>	<i>Read request</i> dan <i>write request</i>	Menentukan nomor <i>data block</i> apabila data yang diakses berada di area DB. Parameter ini tidak digunakan untuk area <i>input</i> , <i>output</i> , atau <i>marker</i> .
<i>Byte offset</i>	<i>Read request</i> dan <i>write request</i>	Menentukan posisi awal data di dalam area memori PLC. Dengan parameter ini, NodeS7 mengetahui lokasi data yang akan dibaca atau ditulis.
<i>Bit offset</i>	Akses data bit	Menentukan posisi bit tertentu di dalam satu alamat <i>byte</i> . Parameter ini digunakan ketika data yang diakses berupa status logika atau Boolean.
<i>Data type</i>	Pemrosesan NodeS7	Menentukan bentuk data yang digunakan, seperti <i>Boolean</i> , <i>integer</i> , <i>WORD</i> , atau <i>REAL</i> . Tipe data membantu NodeS7 membaca dan mengubah data dengan benar.
<i>Data length</i>	<i>Read request</i> , <i>write request</i> , dan <i>read response</i>	Menentukan jumlah data yang akan dibaca, ditulis, atau dikembalikan oleh PLC.
<i>Write data</i>	<i>Write request</i>	Merupakan nilai yang dikirim dari Mini PC untuk dituliskan ke alamat memori PLC.
<i>Status code</i>	<i>Read response</i> dan <i>write response</i>	Menunjukkan apakah proses pembacaan atau penulisan berhasil atau mengalami kesalahan.

Parameter	Digunakan pada	Penjelasan
<i>Read data</i>	<i>Read response</i>	Merupakan data hasil pembacaan yang dikirimkan PLC kembali ke Mini PC.
<i>Write operation status</i>	<i>Write response</i>	Menunjukkan hasil akhir proses penulisan data, yaitu berhasil atau gagal.

(Sumber: Diolah oleh penulis berdasarkan dokumentasi komunikasi Siemens S7 dan kode sumber *library NodeS7*, 2026)

Berdasarkan Tabel 3.8, parameter komunikasi S7 dapat dikelompokkan menjadi parameter penentu alamat, karakteristik data, dan hasil komunikasi. Parameter *area*, *DB number*, *byte offset*, dan *bit offset* digunakan untuk menentukan lokasi data di dalam memori PLC, sedangkan *data type* dan *data length* digunakan oleh NodeS7 untuk mengenali bentuk serta jumlah data yang harus diproses.

Pada proses pembacaan, parameter alamat dan panjang data dikirimkan melalui *S7 Read Request*. PLC kemudian mengakses lokasi memori yang diminta dan mengirimkan *S7 Read Response* yang berisi *status code* serta *read data*. Pada proses penulisan, *S7 Write Request* memuat parameter alamat, panjang data, dan *write data* sebagai nilai yang akan dituliskan ke PLC. Setelah proses selesai, PLC mengirimkan *S7 Write Response* yang berisi status hasil penulisan.

Seluruh parameter tersebut digunakan oleh NodeS7 untuk menerjemahkan perintah dari aplikasi Node.js menjadi pesan komunikasi yang dapat diproses oleh PLC Siemens S7-1200. Setelah respons diterima, NodeS7 mengubah data atau status tersebut ke bentuk yang dapat digunakan kembali oleh aplikasi.

Sebagai contoh penerapan parameter komunikasi tersebut, digunakan alamat MW0 pada PLC Siemens S7-1200. Contoh ini digunakan untuk menjelaskan proses penerjemahan alamat oleh NodeS7, pembentukan permintaan komunikasi, pemrosesan data pada PLC, hingga penerimaan respons oleh aplikasi Node.js. Mekanisme yang dijelaskan meliputi proses pembacaan dan penulisan data pada alamat yang sama.

```

1  conn.addItem('MW0');
2
3  // Membaca nilai MW0
4  conn.readAllItems((error, values) => {
5      if (error || !values) {
6          console.error('Pembacaan MW0 gagal');
7          return;
8      }
9
10     console.log('Nilai MW0:', values.MW0);
11 });
12
13 // Menulis nilai 100 ke MW0
14 conn.writeItems('MW0', 100, (error) => {
15     if (error) {
16         console.error('Penulisan MW0 gagal');
17         return;
18     }
19
20     console.log('Penulisan MW0 berhasil');
21 });

```

Gambar 3. 20 Contoh Program Pembacaan dan Penulisan Alamat MW0 Menggunakan NodeS7
(Sumber: Dokumentasi Pribadi, 2026)

Berdasarkan Gambar 3.20, aplikasi Node.js terlebih dahulu menggunakan fungsi `addItem()` untuk mendaftarkan alamat MW0 sebagai data yang akan dibaca. Selanjutnya, fungsi `readAllItems()` digunakan untuk menjalankan proses pembacaan terhadap alamat tersebut. Hasil pembacaan diterima melalui fungsi *callback* dalam objek `values`, kemudian nilai MW0 diakses melalui `values.MW0` dan ditampilkan pada *console*. Apabila proses pembacaan mengalami kesalahan, aplikasi menampilkan informasi bahwa pembacaan MW0 gagal.

Pada proses penulisan, fungsi `writeItems()` digunakan untuk mengirimkan nilai 100 ke alamat MW0. Setelah proses penulisan dijalankan, hasil operasi diterima melalui fungsi *callback*. Apabila terjadi kesalahan, aplikasi menampilkan informasi bahwa penulisan gagal. Sebaliknya, apabila tidak terdapat kesalahan, aplikasi menampilkan informasi bahwa penulisan MW0 berhasil.

Pada kedua operasi tersebut, aplikasi tidak membentuk paket komunikasi S7 secara langsung. NodeS7 menerjemahkan alamat MW0 menjadi spesifikasi *memory area*, *byte offset*, *data type*, dan *data length*. Pada operasi pembacaan, NodeS7 membentuk *S7 Read Request*, mengirimkannya kepada PLC, kemudian memproses *S7 Read Response* hingga nilai hasil pembacaan dimasukkan ke dalam objek `values`. Pada operasi penulisan, NodeS7 membentuk *S7 Write Request* yang memuat alamat dan nilai

100, kemudian memproses *S7 Write Response* untuk menentukan apakah penulisan berhasil atau mengalami kesalahan.

Alamat MW0 menunjukkan data bertipe *WORD* pada area memori *marker* yang dimulai dari *byte* ke-0. Karena satu *WORD* berukuran dua *byte*, nilai MW0 dibentuk oleh dua *byte* berurutan, yaitu MB0 dan MB1. Berdasarkan alamat tersebut, NodeS7 menentukan parameter komunikasi berupa area *marker*, *DB number* 0 karena alamat tidak berada pada *data block*, *byte offset* 0, serta *data length* dua *byte*. Nilai parameter tersebut ditunjukkan pada Tabel 3.9.

Tabel 3. 9 Parameter Pembacaan dan Penulisan pada Alamat MW0

Parameter	Pembacaan	Penulisan
<i>Address</i>	MW0	MW0
<i>Memory area</i>	<i>Marker</i>	<i>Marker</i>
<i>DB number</i>	0	0
<i>Data type</i>	<i>WORD</i>	<i>WORD</i>
<i>Byte offset</i>	0	0
<i>Data length</i>	2 <i>byte</i>	2 <i>byte</i>
<i>Request</i>	<i>S7 Read Request</i>	<i>S7 Write Request</i>
Isi Permintaan	<i>Memory Area, DB number, byte offset, dan data length</i>	<i>Memory Area, DB number, byte offset, data length, dan nilai yang ditulis</i>
<i>Respons PLC</i>	<i>Status code, panjang data, dan data hasil pembacaan</i>	<i>Status code</i> hasil penulisan
<i>Hasil pada aplikasi</i>	Nilai <i>WORD</i> yang dibaca dari MW0	Informasi berhasil atau gagal

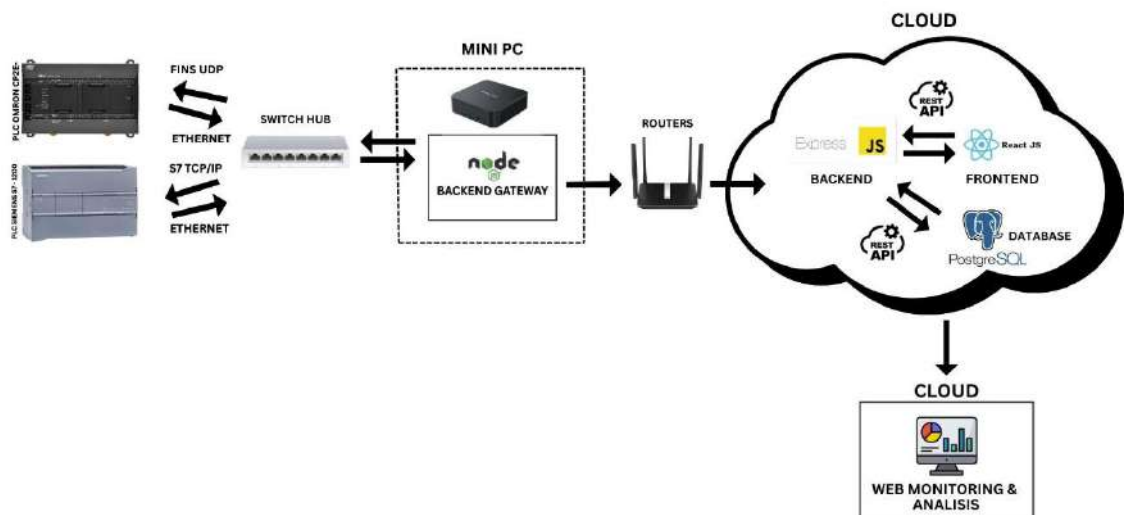
(Sumber: Diolah oleh penulis berdasarkan implementasi NodeS7, 2026)

Berdasarkan Tabel 3.9, operasi pembacaan dan penulisan pada alamat MW0 menggunakan parameter alamat yang sama, yaitu area memori *marker*, *DB number* 0, *byte offset* 0, dan panjang data dua *byte*. Perbedaannya terdapat pada isi permintaan dan respons. Pada pembacaan, PLC mengembalikan status serta data dari MW0, sedangkan pada penulisan, permintaan juga membawa nilai yang akan ditulis dan PLC hanya mengembalikan status hasil operasi.

3.7.3 Gateway Connector Pada Web Monitoring dan Analisis OEE (Cloud)

Perancangan dan pembuatan *Gateway Connector* merupakan komponen penting dalam penelitian Tugas Akhir ini karena berperan sebagai penghubung antara sistem kendali mesin produksi dan sistem *Web Monitoring* dan Analisis OEE berbasis *cloud*. *Gateway Connector* dirancang untuk mengintegrasikan data dari beberapa PLC yang memiliki perbedaan protokol komunikasi, sehingga data operasional dapat dikumpulkan, diolah, dan dikirimkan secara terpusat ke sistem berbasis web untuk keperluan *Monitoring* dan analisis kinerja.

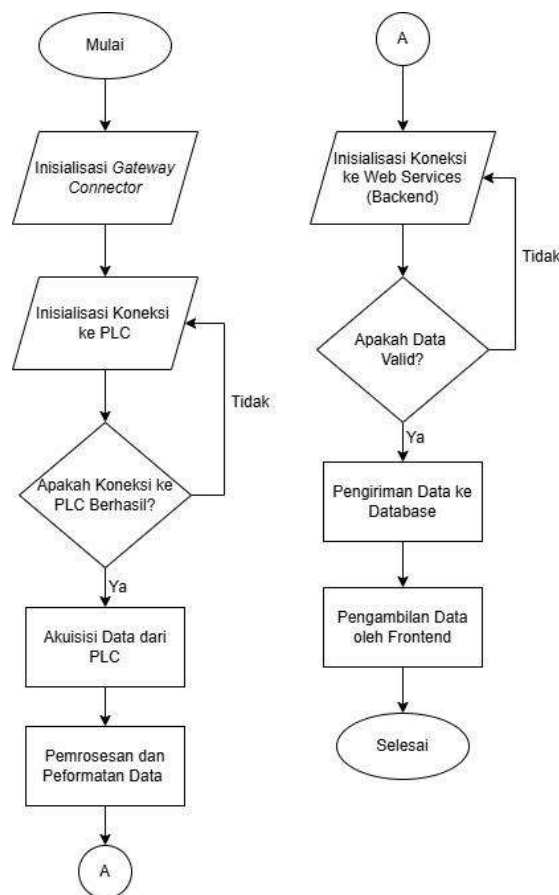
Pada implementasinya, *Gateway Connector* dijalankan pada Mini PC dan dikembangkan menggunakan Node.js. Sistem ini menerima data dari PLC Omron CP2E dengan protokol FINS UDP dan PLC Siemens S7-1200 dengan protokol S7 TCP/IP. Data yang diperoleh, seperti status proses, kondisi sensor, status aktuator, dan parameter operasional, selanjutnya diproses dan dikonversi ke dalam format REST API untuk dikirimkan ke sistem *Web Monitoring* dan Analisis OEE melalui web service backend berbasis Express.js. Data tersebut disimpan pada database PostgreSQL dan dimanfaatkan untuk monitoring, pencatatan data historis, serta perhitungan nilai *Overall Equipment Effectiveness* (OEE).



Gambar 3.21 Struktur Sistem *Gateway Connector* Pada *Web Monitoring* & Analisis OEE
(Sumber: Dokumentasi Pribadi, 2026)

Gambar 3.21 menunjukkan struktur integrasi antara PLC, *gateway connector*, dan sistem *Web Monitoring* serta Analisis OEE berbasis *cloud*. Data dari PLC Omron CP2E melalui protokol FINS UDP dan PLC Siemens S7-1200 melalui protokol S7 TCP/IP

dikirimkan menuju *gateway* yang dijalankan pada Mini PC. Selanjutnya, data diproses dan diteruskan melalui REST API menuju *backend* berbasis Express.js untuk disimpan pada basis data PostgreSQL. Data tersebut kemudian digunakan oleh bagian *frontend* untuk menampilkan informasi monitoring, pencatatan data, dan analisis OEE melalui antarmuka web. Alur kerja integrasi antara PLC, Web *Monitoring & Analisis OEE* dan Database melalui *Gateway Connector* dapat dilihat pada gambar 3.22 berikut.



Gambar 3.22 Diagram Alur *Gateway Connector* Pada Web *Monitoring & Analisis OEE*
(Sumber: Dokumentasi Pribadi, 2026)

dengan penjelasan setiap tahap dijabarkan sebagai berikut :

1. Mulai, Tahap ini menandai dimulainya proses kerja *Gateway Connector* pada Mini PC sebagai bagian dari sistem Web *Monitoring* dan Analisis.
2. Inisialisasi *Gateway Connector*, Pada tahap ini dilakukan inisialisasi sistem *gateway* yang meliputi pemuatan konfigurasi awal, seperti parameter komunikasi PLC, pengaturan protokol yang digunakan, serta konfigurasi koneksi ke sistem backend. Tahap ini bertujuan untuk memastikan *gateway* siap menjalankan proses akuisisi dan pengiriman data.

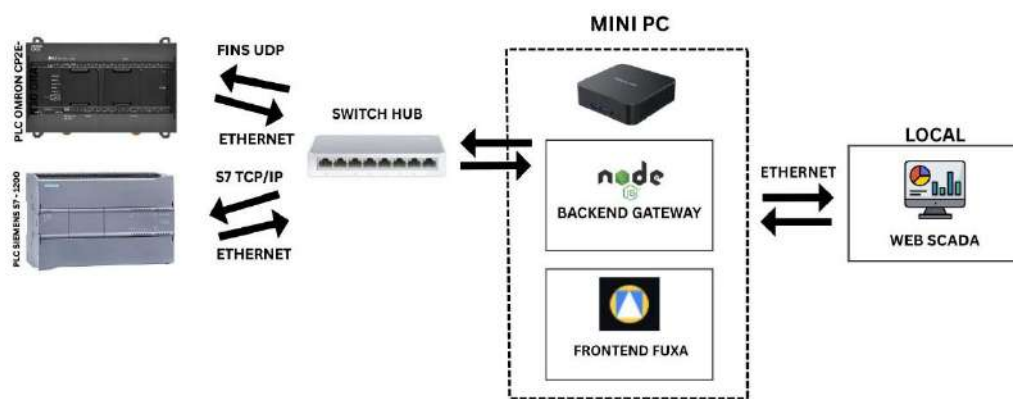
3. Inisialisasi Koneksi ke PLC, *Gateway Connector* melakukan proses inisialisasi koneksi ke PLC melalui protokol komunikasi yang telah ditentukan. Tahap ini memastikan *gateway* dapat terhubung dengan PLC untuk membaca data operasional sistem.
4. Akuisisi Data dari PLC, Pada tahap ini, *Gateway Connector* melakukan pembacaan data dari PLC. Data yang diakuisisi meliputi status proses, kondisi sensor, status aktuator, serta parameter operasional yang berkaitan dengan proses produksi.
5. Pemrosesan dan Pemformatan Data, Data yang diperoleh dari PLC selanjutnya diproses dan diformat agar sesuai dengan struktur data yang digunakan pada sistem backend. Proses ini bertujuan untuk memastikan data siap dikirim melalui web service berbasis REST API.
6. Inisialisasi Koneksi ke *Web Services (Backend)*, Pada tahap ini, *Gateway Connector* menyiapkan koneksi ke web service backend sebagai bagian dari proses pengiriman data.
7. Validasi Data, Sebelum data dikirimkan ke sistem backend, dilakukan proses validasi untuk memastikan data yang dikirim sesuai dengan format dan kebutuhan sistem. Validasi ini bertujuan untuk menjaga keakuratan dan konsistensi data.
8. Pengiriman Data ke Database, Data yang telah tervalidasi kemudian dikirimkan ke sistem backend untuk disimpan ke dalam database. Data ini disimpan sebagai data *real-time* maupun data historis yang akan digunakan pada tahap selanjutnya.
9. Pengambilan Data oleh *Frontend*, Data yang tersimpan dalam database selanjutnya diakses oleh sistem *Frontend Web Monitoring* dan Analisis untuk ditampilkan kepada pengguna melalui antarmuka web.
10. Selesai, Tahap ini menandai berakhirnya satu siklus proses *Gateway Connector*. Proses ini berjalan secara berulang sesuai dengan interval waktu yang telah ditentukan untuk mendukung *Monitoring* dan analisis data secara berkelanjutan.

3.7.4 Gateway Connector Pada Web SCADA (Local)

Perancangan dan implementasi *Gateway Connector* pada sistem Web SCADA (*Local*) merupakan bagian penting dalam penelitian Tugas Akhir ini karena berfungsi sebagai penghubung antara sistem kendali pada mesin *Distribution Station* dan *Sorting station* dengan sistem supervisi lokal di area *plant*. *Gateway Connector* dirancang untuk mengintegrasikan data dari PLC yang menggunakan protokol komunikasi berbeda

sehingga informasi proses dapat diakses secara terpusat melalui sistem Web SCADA berbasis web untuk kebutuhan operasional langsung.

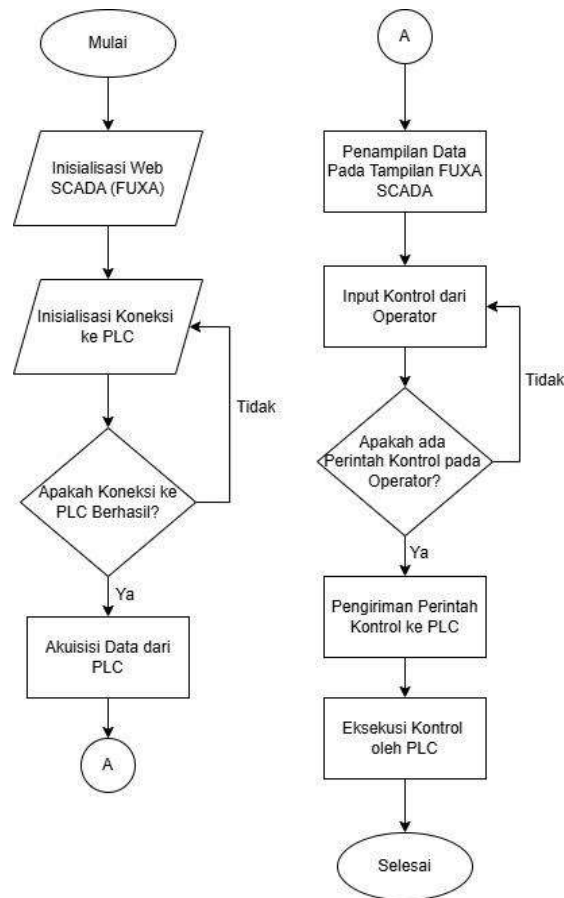
Pada sistem ini, *Gateway Connector* dijalankan pada Mini PC dan berfungsi menerima data dari PLC Omron CP2E dengan protokol FINS UDP serta PLC Siemens S7-1200 dengan protokol S7 TCP/IP. Data yang diperoleh meliputi kondisi sensor, status aktuator, serta parameter proses pada masing-masing station. Data tersebut kemudian diproses dan disesuaikan agar dapat ditampilkan pada sistem Web SCADA (Local) menggunakan FUXA SCADA, sehingga operator dapat melakukan *Monitoring* dan kontrol proses secara langsung di dekat *plant* tanpa harus mengakses PLC secara manual.



Gambar 3.23 Struktur Sistem *Gateway Connector* Pada Web SCADA (Local)
(Sumber: Dokumentasi Pribadi, 2026)

Gambar 3.23 menunjukkan struktur integrasi antara PLC dan Web-SCADA lokal melalui *gateway connector* yang dijalankan pada Mini PC. Data dari PLC Omron CP2E dikirim menggunakan protokol FINS UDP, sedangkan data dari PLC Siemens S7-1200 dikirim menggunakan protokol S7 TCP/IP melalui jaringan Ethernet. Seluruh data tersebut diterima dan diproses oleh *backend gateway*, kemudian diteruskan ke FUXA SCADA sebagai antarmuka Web-SCADA lokal. Melalui sistem ini, operator dapat melakukan *monitoring*, melihat status sensor dan aktuator, serta menjalankan perintah kontrol secara langsung melalui jaringan lokal tanpa harus mengakses PLC satu per satu. Arsitektur ini juga membuat proses pengawasan dan pengendalian menjadi lebih terpusat, sehingga komunikasi antara kedua PLC dan antarmuka Web-SCADA dapat berjalan

dalam satu sistem yang terintegrasi. Alur integrasi antara PLC dan Web SCADA (Local) melalui *Gateway Connector* ditunjukkan pada Gambar 3.24.



Gambar 3.24 Diagram Alur *Gateway Connector* Pada Web SCADA (Local)
(Sumber: Dokumentasi Pribadi, 2026)

dengan penjelasan setiap tahap dijabarkan sebagai berikut :

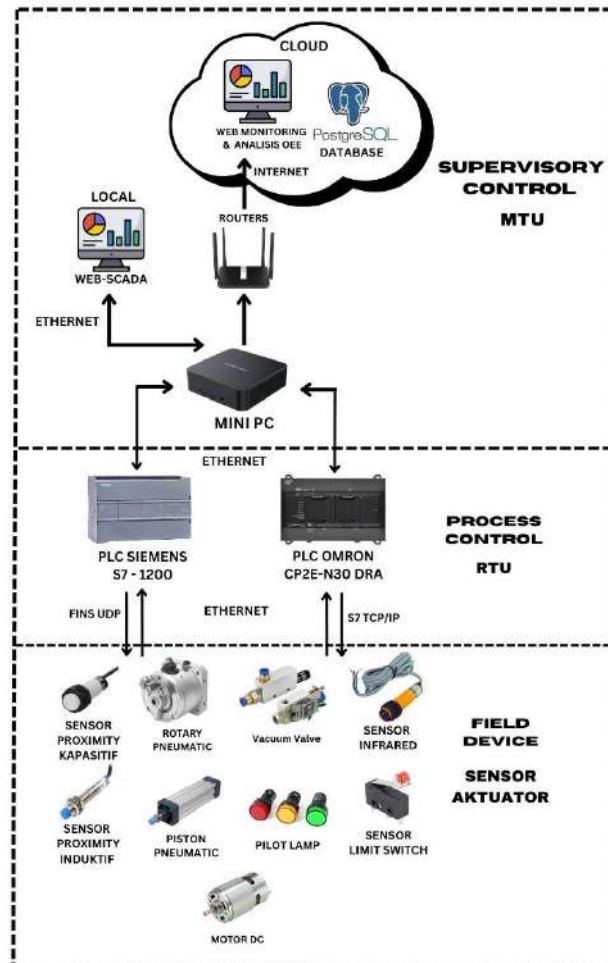
1. Mulai, Tahap ini menandai dimulainya proses kerja sistem Web-SCADA (Local) yang dijalankan menggunakan FUXA SCADA pada Mini PC di area plant.
2. Inisialisasi Web-SCADA (FUXA), Pada tahap ini, sistem FUXA SCADA memuat konfigurasi proyek yang telah dibuat, meliputi pengaturan perangkat PLC, tag data, serta tampilan antarmuka. Proses ini bertujuan untuk menyiapkan sistem agar siap melakukan *Monitoring* dan kontrol.
3. Inisialisasi Koneksi ke PLC, FUXA SCADA melakukan inisialisasi koneksi langsung ke PLC menggunakan protokol komunikasi yang sesuai, yaitu FINS UDP untuk PLC Omron CP2E dan S7 TCP/IP untuk PLC Siemens S7-1200.
4. Pemeriksaan Koneksi ke PLC, Pada tahap ini dilakukan pengecekan untuk memastikan koneksi antara FUXA SCADA dan PLC telah berhasil terjalin.

Koneksi yang berhasil menjadi dasar untuk melanjutkan proses pembacaan data dari PLC.

5. Akuisisi Data dari PLC, Setelah koneksi berhasil, FUXA SCADA melakukan pembacaan data dari PLC. Data yang diakuisisi meliputi status sensor, kondisi aktuator, serta parameter proses yang berkaitan dengan mesin *Distribution Station* dan *Sorting station*.
6. Penampilan Data pada Tampilan FUXA SCADA, Data yang diperoleh dari PLC ditampilkan secara *real-time* pada antarmuka Web-SCADA (Local). Informasi ini disajikan dalam bentuk indikator, status, dan elemen visual lainnya untuk memudahkan operator dalam melakukan pemantauan proses.
7. *Input* Kontrol dari Operator, Operator memberikan perintah kontrol melalui antarmuka Web-SCADA, seperti perintah start dan stop sistem, pengaturan mode operasi, pengaktifan atau penonaktifan aktuator, serta reset counter.
8. Pemeriksaan Perintah Kontrol dari Operator, Pada tahap ini sistem memeriksa apakah terdapat perintah kontrol yang diberikan oleh operator melalui Web-SCADA. Pemeriksaan ini dilakukan untuk memastikan bahwa hanya perintah yang valid yang akan diteruskan ke PLC.
9. Pengiriman Perintah Kontrol ke PLC, Perintah kontrol yang diberikan oleh operator selanjutnya dikirimkan oleh FUXA SCADA ke PLC sesuai dengan jenis dan tujuan perintah yang diberikan.
10. Eksekusi Kontrol oleh PLC, PLC mengeksekusi perintah kontrol yang diterima dengan menjalankan logika program dan menggerakkan aktuator sesuai dengan perintah dari Web-SCADA.
11. Selesai, Tahap ini menandai akhir dari satu siklus proses kontrol Web-SCADA (Local). Proses *Monitoring* dan kontrol berjalan secara berulang selama sistem aktif untuk mendukung operasi plant secara *real-time*.

3.8 Perancangan Sistem Web-SCADA

Perancangan sistem Web-SCADA pada penelitian ini dibagi ke dalam tiga lapisan utama, yaitu *Supervisory Control* (MTU), *Process Control* (RTU), dan *Field Device*, sebagaimana ditunjukkan pada Gambar 3.2.



Gambar 3.25 Flowchart Sistem Web-SCADA
(Sumber: Dokumentasi Pribadi, 2026)

Sistem yang dikembangkan terdiri dari dua bagian utama, yaitu Web-SCADA lokal menggunakan FUXA SCADA dan sistem Web *Monitoring* serta Analisis OEE. Pembagian ini dilakukan untuk memisahkan fungsi pengendalian proses dengan fungsi pemantauan dan analisis data produksi.

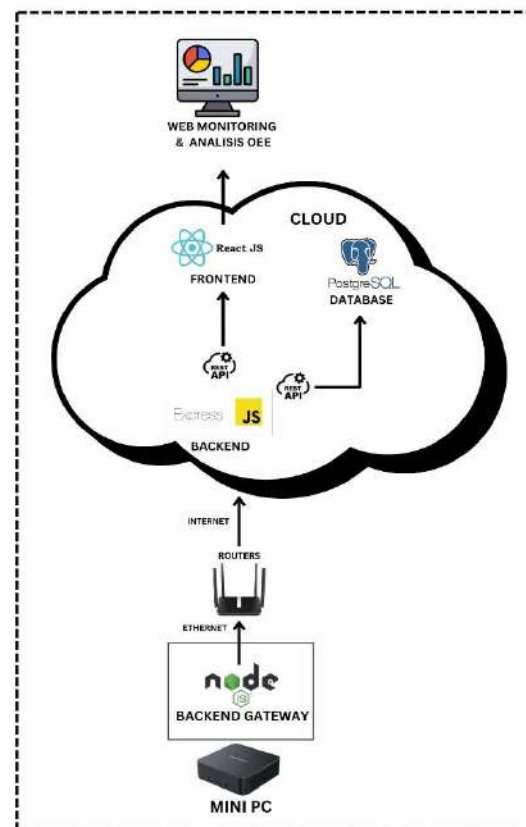
Web-SCADA lokal dijalankan pada jaringan lokal dan berfungsi sebagai sistem supervisory control yang terhubung langsung dengan PLC Siemens S7-1200 dan PLC Omron CP2E-N30 DRA melalui sebuah mini PC sebagai *gateway*. Sistem ini digunakan untuk melakukan *Monitoring* kondisi proses secara *real-time*, pengendalian aktuator, serta penanganan alarm sistem pada *Distribution* dan *Sorting station*. Seluruh fungsi kontrol proses dilakukan secara lokal tanpa bergantung pada koneksi internet guna menjaga keandalan dan keselamatan sistem.

Sementara itu, sistem berbasis *cloud* digunakan khusus untuk keperluan *Monitoring* data produksi dan analisis kinerja mesin. Data operasional dari PLC dikirimkan oleh *gateway* dalam bentuk REST API ke backend dan disimpan pada

database PostgreSQL. Sistem *cloud* ini menyediakan fitur pencatatan data historis (data logger) serta analisis *Overall Equipment Effectiveness* (OEE) yang meliputi nilai *Availability*, *Performance*, dan *Quality*. Sistem ini tidak memiliki akses untuk melakukan pengendalian proses produksi.

3.8.1 Perancangan Web *Monitoring & Analisis* OEE (*Cloud*)

Perancangan Web *Monitoring* dan Analisis OEE pada penelitian ini terdiri dari tiga komponen utama, yaitu *Frontend* berbasis *React JS* sebagai antarmuka monitoring, *backend* berbasis *Node.js* dan *Express* sebagai penyedia layanan *REST API*, serta *database PostgreSQL* sebagai media penyimpanan data operasional dan hasil perhitungan OEE, sebagaimana ditunjukkan pada Gambar 3.26.



Gambar 3.26 *Flowchart* Perancangan Web *Monitoring & Analisis* OEE
(Sumber: Dokumentasi Pribadi, 2026)

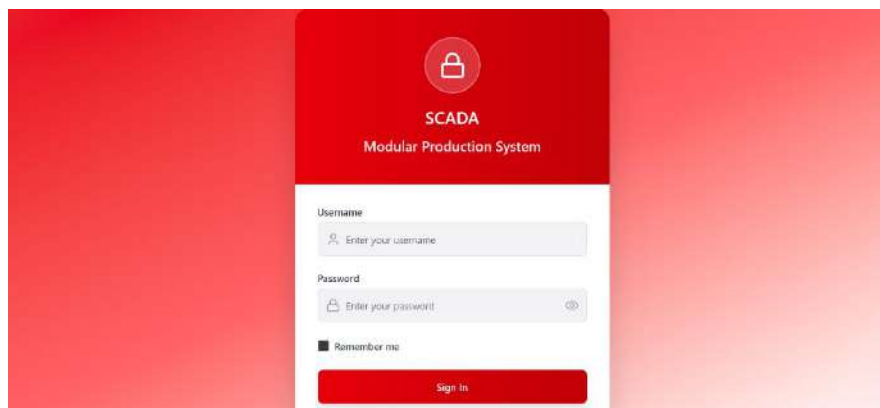
Sistem Web *Monitoring* dan analisis *Overall Equipment Effectiveness* (OEE) pada penelitian ini dirancang sebagai sistem berbasis *cloud* yang berfungsi untuk menampilkan kondisi operasional serta kinerja *Distribution and Sorting station* secara terpusat melalui antarmuka berbasis *web*. Sistem ini berada pada lapisan supervisi dan menerima data operasional dari Mini PC melalui layanan *REST API*, tanpa melakukan komunikasi langsung dengan PLC. Dengan pendekatan tersebut, sistem *Monitoring* dapat

berjalan secara independen dari sistem kendali proses lokal, sehingga meningkatkan keamanan dan keandalan pengendalian

Pada lapisan *backend*, sistem dikembangkan menggunakan *Node.js* dan *Express* sebagai penyedia layanan *REST API* yang berjalan pada lingkungan *cloud*. *Backend* berperan dalam mengelola pertukaran data antara Mini PC, *Frontend*, dan *database*, termasuk proses penerimaan, pengolahan, serta penyimpanan data operasional ke dalam *database PostgreSQL*. Selain itu, *backend* juga berfungsi sebagai *data logger* yang mencatat data historis sistem serta mengelola informasi *alarm* berdasarkan kondisi tertentu yang terdeteksi selama proses operasi.

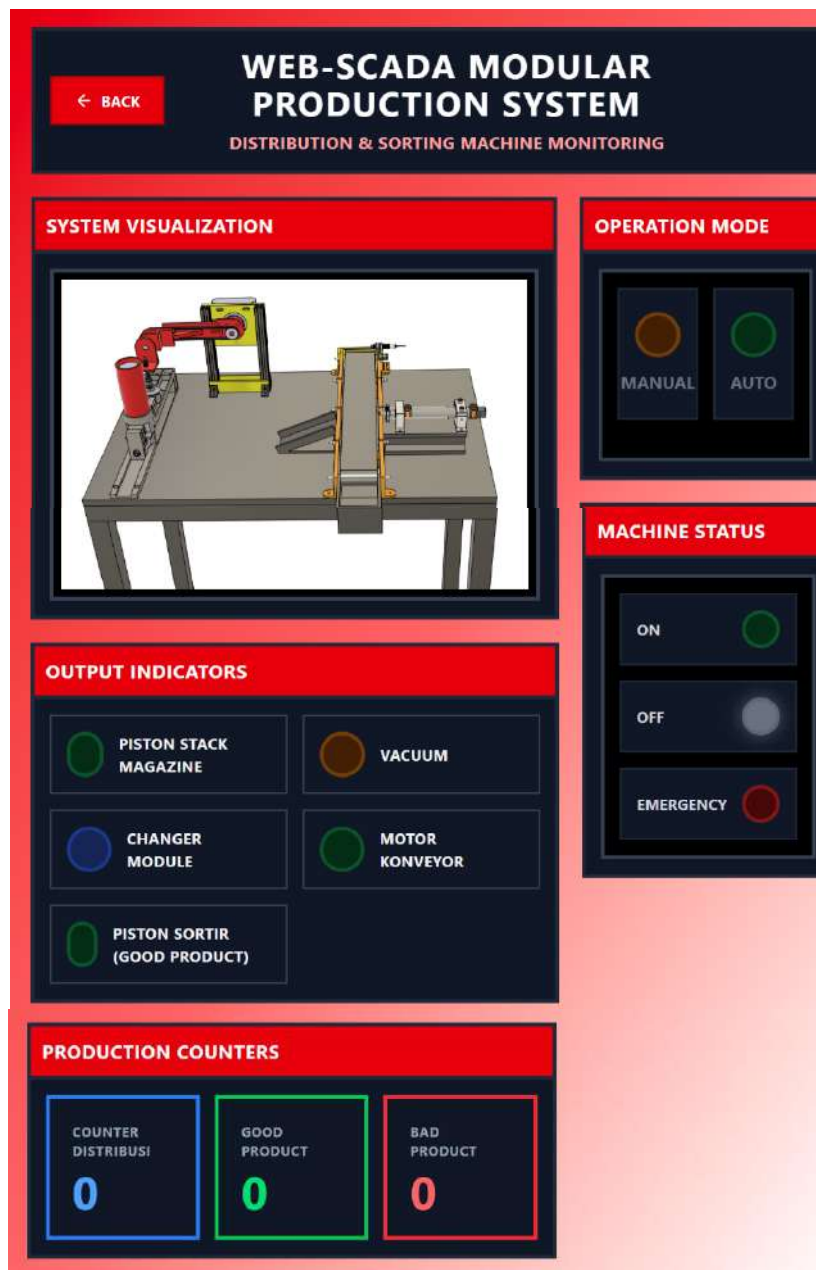
Pada lapisan *Frontend*, sistem *Web Monitoring* berbasis *cloud* dibangun menggunakan *React JS* sebagai antarmuka visualisasi data. *Frontend* menampilkan kondisi sistem secara *real-time*, data historis hasil pencatatan *data logger*, status dan riwayat *alarm*, serta hasil analisis *OEE* yang dihitung berdasarkan parameter *Availability*, *Performance*, dan *Quality*. Melalui integrasi antara *Frontend*, *backend*, dan *database* pada lingkungan *cloud*, sistem ini mampu menyediakan informasi operasional dan evaluasi kinerja *Distribution and Sorting station* secara akurat, terstruktur, dan dapat diakses melalui *web browser* dari berbagai perangkat.

Gambar 3.27 menunjukkan desain halaman login, yang menampilkan kolom *input* bagi operator untuk memasukkan username dan password yang telah terdaftar.



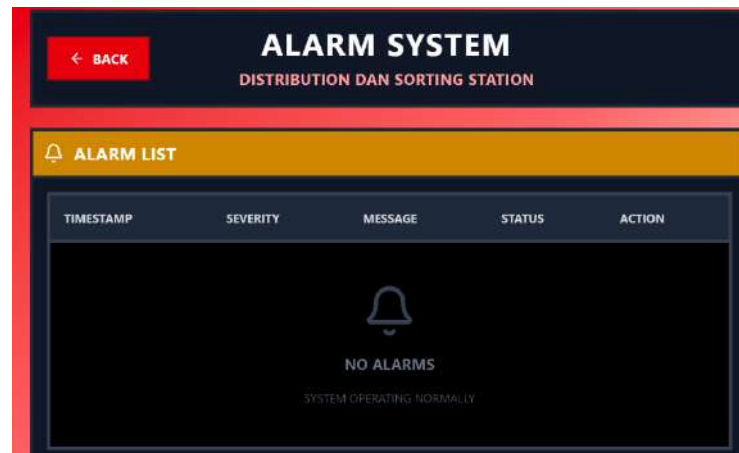
Gambar 3.27 Tampilan Halaman Login
(Sumber: Dokumentasi Pribadi, 2026)

Tampilan halaman *Monitoring* dapat dilihat pada pada Gambar 3.28. Halaman ini berperan sebagai pusat *Monitoring* pada sistem *Distribution* dan *Sorting station*. Tampilan halaman menampilkan visualisasi kondisi mesin secara aktual, indikator status output dan sistem, serta informasi counter produksi yang mencakup jumlah produk baik dan produk cacat.



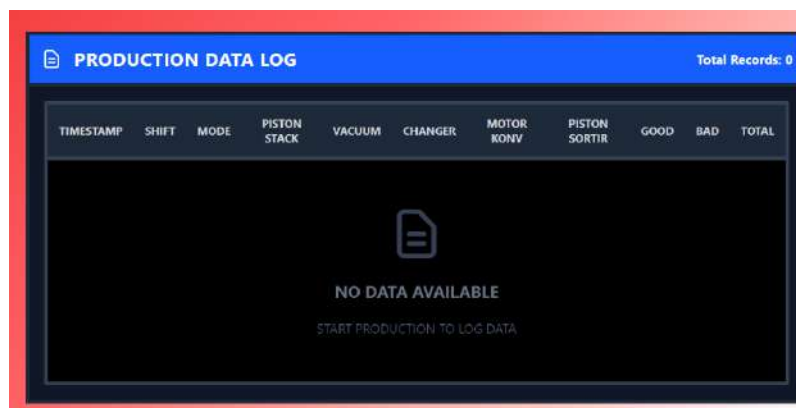
Gambar 3.28 Tampilan Halaman Monitoring
(Sumber: Dokumentasi Pribadi, 2026)

Tampilan halaman *alarm system* ditunjukkan pada Gambar 3.29. Halaman ini menampilkan daftar alarm yang dilengkapi informasi *timestamp*, tingkat keparahan, pesan, status, dan tindakan. Pada kondisi normal, halaman menampilkan keterangan bahwa tidak terdapat alarm dan sistem sedang beroperasi dengan normal.



Gambar 3.29 Tampilan Halaman Alarm Sistem
(Sumber: Dokumentasi Pribadi, 2026)

Tampilan halaman *data logger* ditunjukkan pada Gambar 3.30. Halaman ini digunakan untuk menampilkan data produksi, seperti waktu pencatatan, status proses, kondisi, jumlah produk baik, produk buruk, dan total produksi. Apabila proses belum dijalankan, halaman akan menampilkan keterangan bahwa data belum tersedia.



Gambar 3.30 Tampilan Halaman data *logger*
(Sumber: Dokumentasi Pribadi, 2026)

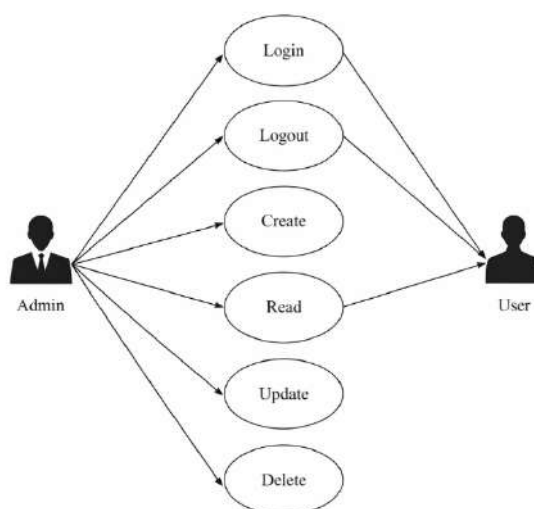
Tampilan halaman analisis OEE dapat dilihat pada pada Gambar 3.31. Halaman *Overall Equipment Effectiveness* (OEE) digunakan untuk memantau performa mesin *Distribution* dan Sorting pada sistem Web-SCADA. Tampilan halaman menampilkan nilai *Availability*, *Performance*, *Quality*, dan OEE yang dihitung dari data produksi dan waktu kerja mesin. Selain itu, ditampilkan pula informasi waktu operasi dan downtime, target produksi, jumlah *good* dan *bad product*, serta data histori OEE berdasarkan shift.



Gambar 3.31 Tampilan Halaman Analisis OEE
(Sumber: Dokumentasi Pribadi, 2026)

3.8.1.1 Use Case Diagram

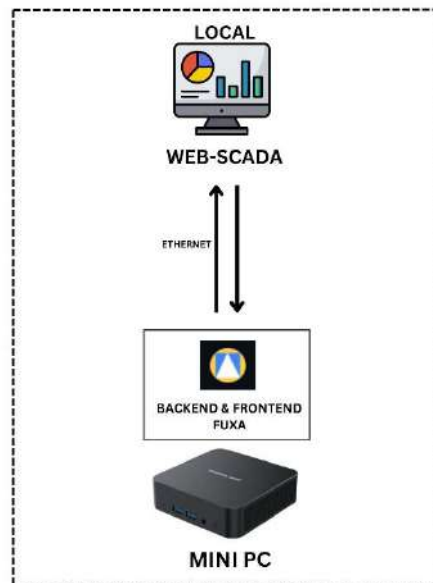
Use Case pada Gambar 3.32 merupakan rangkaian langkah-langkah yang menjelaskan alur kerja sistem pada Website. Sistem ini memiliki dua tingkat pengguna, yaitu user biasa dan admin. User biasa memiliki akses *Monitoring* analisis pada tampilan website, mereka dapat melihat data terkait status mesin, status *counter object*, selain itu user biasa dapat melakukan analisis pada parameter OEE. Sementara itu, admin memiliki hak akses lebih luas yaitu dapat memantau serta mengontrol sistem melalui Website dan memiliki kemampuan untuk menghapus, memperbarui, serta menambahkan elemen atau tampilan baru pada Website sesuai kebutuhan.



Gambar 3 32 Use Case Diagram Web *Monitoring & Analisis OEE*
(Sumber: Dokumentasi Pribadi, 2026)

3.8.2 Perancangan Web SCADA (*Local*)

Perancangan sistem Web SCADA (*Local*) pada penelitian ini diimplementasikan pada sebuah Mini PC yang digunakan sebagai platform sistem *lokal*. Sistem ini memanfaatkan FUXA SCADA sebagai media supervisi untuk melakukan *Monitoring* dan kontrol proses pada lingkungan *plant*, sebagaimana ditunjukkan pada, sebagaimana ditunjukkan pada Gambar 3.33.



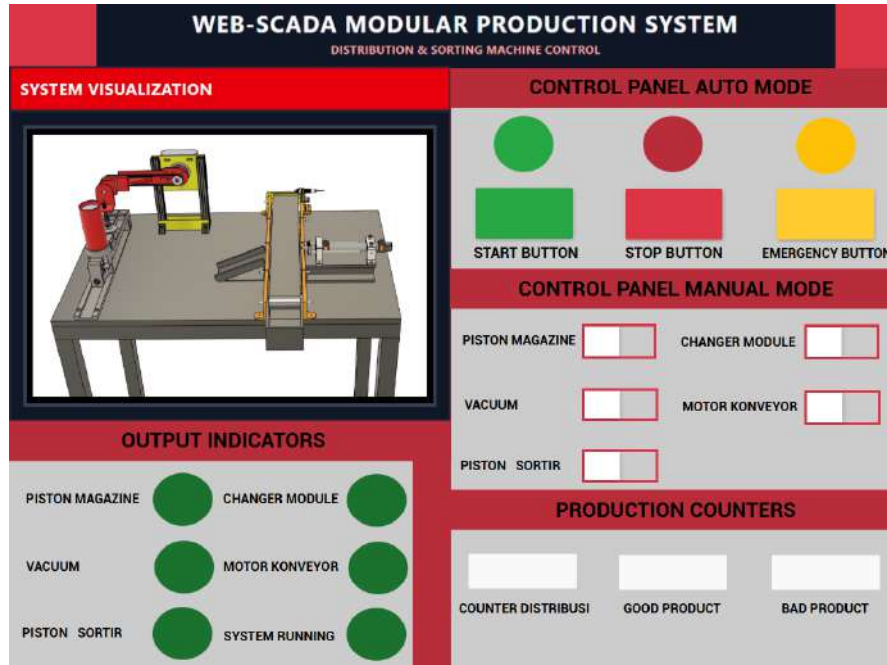
Gambar 3.33 Flowchart Web SCADA (*Local*)
(Sumber: Dokumentasi Pribadi, 2026)

Sistem supervisi lokal dirancang untuk mendukung aktivitas pemantauan dan pengendalian proses secara langsung di area plant. Seluruh komponen sistem ditempatkan pada lingkungan local system, sehingga sistem dapat beroperasi secara mandiri tanpa ketergantungan pada koneksi internet. Mini PC berperan sebagai perangkat komputasi lokal yang menjalankan sistem supervisi serta mendukung kebutuhan operasional, sehingga proses *Monitoring* dan kontrol dapat dilakukan secara stabil dan berkesinambungan di lingkungan industri.

FUXA SCADA digunakan sebagai antarmuka Web SCADA (*Local*) untuk menampilkan kondisi operasional sistem secara *real-time* dan menyediakan media interaksi bagi operator. Informasi yang ditampilkan meliputi status proses, kondisi peralatan, serta parameter operasional yang relevan. Melalui antarmuka ini, operator dapat melakukan *Monitoring* dan kontrol supervisi terhadap *Distribution Station* dan

Sorting station secara langsung di area *plant*, sehingga respon terhadap perubahan kondisi proses dapat dilakukan dengan cepat dan efektif.

Tampilan Web SCADA halaman *Monitoring* dan Kontrol dapat dilihat pada Gambar 3.34.



Gambar 3.34 Tampilan Halaman *Monitoring* dan Kontrol
(Sumber: Dokumentasi Pribadi, 2026)

3.9 Konsep Perhitungan OEE Pada *Distribution* dan *Sorting station*

Perhitungan *Overall Equipment Effectiveness* (OEE) pada sistem *Distribution* dan *Sorting station* dilakukan berdasarkan tiga komponen utama, yaitu *Availability*, *Performance*, dan *Quality*. Ketiga komponen tersebut dihitung menggunakan data operasional yang diperoleh dari sensor dan aktuator yang terpasang pada masing-masing station. Data hasil akuisisi kemudian dikirim melalui *Gateway Connector* untuk selanjutnya diproses dan ditampilkan pada sistem Web-SCADA. Sistem ini dirancang untuk memberikan gambaran menyeluruh mengenai tingkat efisiensi dan efektivitas kinerja *Distribution* dan *Sorting station* selama proses pemindahan dan penyortiran objek.

Pada penelitian ini, penentuan *Good Product* dan *Bad Product* dilakukan berdasarkan skenario pengujian pada *Modular Production System* (MPS) sebagai media simulasi. *Good Product* direpresentasikan oleh *workpiece* standar yang memenuhi spesifikasi pengujian dan diidentifikasi menggunakan sensor *proximity* induktif. Sementara itu, *Bad Product* direpresentasikan oleh *workpiece* yang telah diberikan *Reject*

Marker sebagai identitas produk yang telah diasumsikan tidak memenuhi spesifikasi kualitas (*reject*). Pemberian *Reject Marker* dimaksudkan untuk mensimulasikan hasil proses *Quality inspection* sehingga sistem dapat membedakan *Good Count* dan *Reject Count* sebagai masukan dalam perhitungan *Overall Equipment Effectiveness (OEE)*. Pendekatan ini dipilih agar sistem dapat menghasilkan data *Good Count* dan *Reject Count* sebagai masukan dalam perhitungan *Overall Equipment Effectiveness (OEE)* pada media simulasi *Modular Production System (MPS)*. Tabel 3.10 menyajikan daftar variabel OEE yang digunakan beserta sumber data untuk setiap variabel.

Tabel 3.10 Sumber Data Pengujian OEE

Variabel	Sumber Data
Waktu produksi yang direncanakan (<i>Loading Time</i>)	Data ditentukan berdasarkan durasi pengujian yang telah direncanakan.
Lama waktu mesin menjalankan proses produksi (<i>Operating Time</i>)	Diambil berdasarkan durasi total operasi sistem <i>Distribution</i> dan <i>Sorting station</i> , dimulai sejak tombol start pada mode otomatis diaktifkan hingga sistem dihentikan melalui tombol stop atau emergency stop.
Waktu mesin dalam keadaan berhenti (<i>Downtime</i>)	Data diperoleh dari waktu ketika sistem aktivasi emergency stop.
Lama waktu mesin dalam memproses satu objek (<i>Cycle Time</i>)	Data diambil dari selang waktu sejak objek terdeteksi oleh sensor awal proses hingga objek tersebut mencapai penampungan akhir.
Jumlah total produk yang diproses (<i>Total Product</i>)	Data diperoleh dari nilai <i>counter</i> produk pada <i>Sorting station</i> yang mencatat seluruh objek yang berhasil diproses sistem, baik produk yang memenuhi kriteria kualitas (<i>good product</i>) maupun produk yang ditolak (<i>bad product</i>).
Jumlah produk baik (<i>Good Product</i>)	Data diperoleh dari hasil pembacaan sensor <i>proximity</i> induktif yang mengidentifikasi workpiece standar sebagai produk yang memenuhi spesifikasi pada skenario pengujian, kemudian dicatat sebagai <i>Good Product</i> oleh PLC.
Jumlah produk cacat (<i>Bad Product</i>)	Data diperoleh dari hasil pembacaan sensor <i>proximity</i> kapasitif yang mengidentifikasi

Variabel	Sumber Data
	workpiece yang telah diberi <i>Reject Marker</i> sebagai representasi produk yang tidak memenuhi spesifikasi kualitas (<i>reject</i>) pada skenario pengujian, kemudian dicatat sebagai <i>Bad Product</i> oleh PLC.

(Sumber: Dokumentasi Pribadi, 2026)

Berdasarkan Tabel 3.10, data yang diperoleh akan digunakan dalam perhitungan dengan rumus *Overall Equipment Effectiveness* (OEE) sebagai berikut:

$$A \% = \frac{\text{Operating Time}}{\text{Loading Time}} \times 100 \quad (3.1)$$

$$P \% = \frac{\text{Cycle Time} \times \text{Actual Output}}{\text{Operating Time}} \times 100 \% \quad (3.2)$$

$$Q \% = \frac{\text{Good Product}}{\text{Total Production}} \times 100 \% \quad (3.3)$$

Setelah nilai dari setiap variabel diperoleh, hasil tersebut akan dimasukkan ke dalam persamaan berikut untuk perhitungan lebih lanjut.

$$OEE (\%) = A \times P \times Q \quad (3.4)$$

Untuk memberikan gambaran penerapan metode, analisis dilakukan berdasarkan data yang tersaji pada Tabel 3.11.

Tabel 3.11 Contoh Data Pengujian

Variabel	Sumber Data
Waktu keseluruhan mesin beroperasi (<i>Loading Time</i>)	60 Menit
Waktu mesin dalam keadaan berhenti (<i>Downtime</i>)	3 Menit
Lama waktu mesin dalam memproses satu objek (<i>Cycle Time</i>)	1 Menit
Jumlah total produk yang diproses (<i>Total Product</i>)	50 Pcs
Jumlah produk baik (<i>Good Product</i>)	40 Pcs

(Sumber: Dokumentasi Pribadi, 2026)

Berdasarkan data yang tercantum pada Tabel 3.11, perhitungan OEE akan dilakukan dan dianalisis seperti uraian berikut:

$$\begin{aligned}
 A \% &= \frac{\text{Operating Time}}{\text{Loading Time}} \times 100 \quad (3.5) \\
 &= \frac{57}{60} \times 100 \% \\
 &= 95 \%
 \end{aligned}$$

$$P \% = \frac{\text{Cycle Time} \times \text{Actual Output}}{\text{Operating Time}} \times 100 \% \quad (3.6)$$

$$\begin{aligned}
&= \frac{1 \times 50}{60-3} \times 100 \% \\
&= \frac{50}{57} \times 100 \% \\
&= 87.72 \% \\
Q \% &= \frac{\text{Good Product}}{\text{Total Production}} \times 100 \% \quad (3.7) \\
&= \frac{40}{50} \times 100 \% \\
&= 80 \%
\end{aligned}$$

Setelah diperoleh hasil akhir dari ketiga komponen OEE, yaitu *Availability*, *Performance*, dan *Quality*, maka perhitungan nilai OEE berdasarkan data tersebut dapat dihitung sebagai berikut :

$$\begin{aligned}
OEE &= A \times P \times Q \quad (3.8) \\
&= 95 \times 87.72 \times 80 \\
&= 66.67 \%
\end{aligned}$$

(Halaman ini sengaja dikosongkan)

BAB 4

HASIL DAN PEMBAHASAN

4.1 Hasil Perancangan *Hardware*

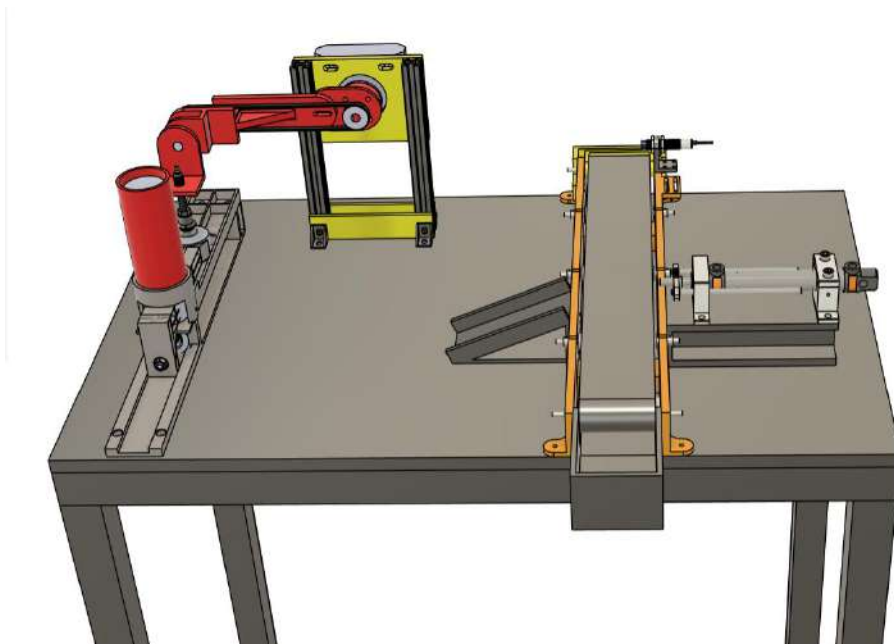
Setelah menyelesaikan tahap perancangan sistem, pada bab ini akan dibahas hasil implementasi serta analisis dari proses perancangan dan pengujian sistem Web-SCADA yang telah dibangun. Pengujian dilakukan untuk memastikan seluruh komponen *plant*, PLC, *gateway*, serta aplikasi *Monitoring* dapat bekerja secara terintegrasi dalam melakukan proses kontrol, akuisisi data, dan pemantauan sistem. Pengujian pada penelitian ini dibagi menjadi dua bagian utama, yaitu:

1. Pengujian *hardware*, mencakup pengujian sensor, aktuator pneumatik, motor, serta berbagai komponen pada *plant Distribution dan Sorting station* yang dikendalikan oleh PLC Omron CP2E N30 DRA dan Siemens S7-1200.
2. Pengujian *software* mencakup pengujian program PLC, komunikasi *Gateway* multi-protokol, sistem Web-SCADA lokal menggunakan FUXA, sistem Web *Monitoring* dan Analisis berbasis *cloud* untuk penyimpanan serta visualisasi data, serta perhitungan nilai *Overall Equipment Effectiveness* (OEE) sebagai evaluasi kinerja mesin.

4.1.1 Hasil Perancangan Kontruksi Mekanik

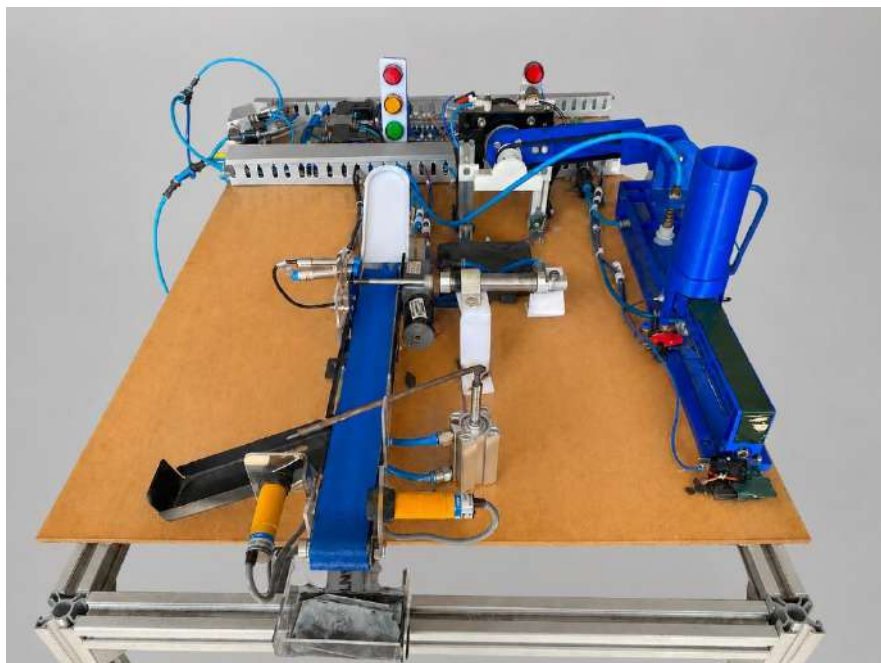
Hasil perancangan sistem mekanik pada tugas akhir ini ditunjukkan pada Gambar 4.1, 4.2 dan Gambar 4.3, yang mengacu pada konsep *Modular Production System* (MPS) berbasis Festo. Sistem dirancang secara modular dengan membagi proses menjadi beberapa stasiun kerja yang saling terintegrasi, seperti *Distribution Station* dan *Sorting station*. Pendekatan modular ini bertujuan untuk meningkatkan fleksibilitas, kemudahan integrasi, serta efisiensi dalam pengoperasian sistem, sehingga setiap modul dapat bekerja secara optimal sesuai dengan fungsinya dalam mensimulasikan sistem otomasi industri skala kecil.

Seluruh komponen mekanik ditempatkan pada satu meja kerja dengan mempertimbangkan posisi antarstasiun, jalur perpindahan benda kerja, serta kemudahan dalam pemasangan sensor dan aktuator. Penempatan tersebut juga disesuaikan agar proses pengamatan, perawatan, dan pengujian sistem dapat dilakukan dengan lebih mudah. Hasil perancangan ini kemudian menjadi dasar dalam proses pemasangan komponen elektrikal, pneumatik, dan sistem kendali pada *plant*



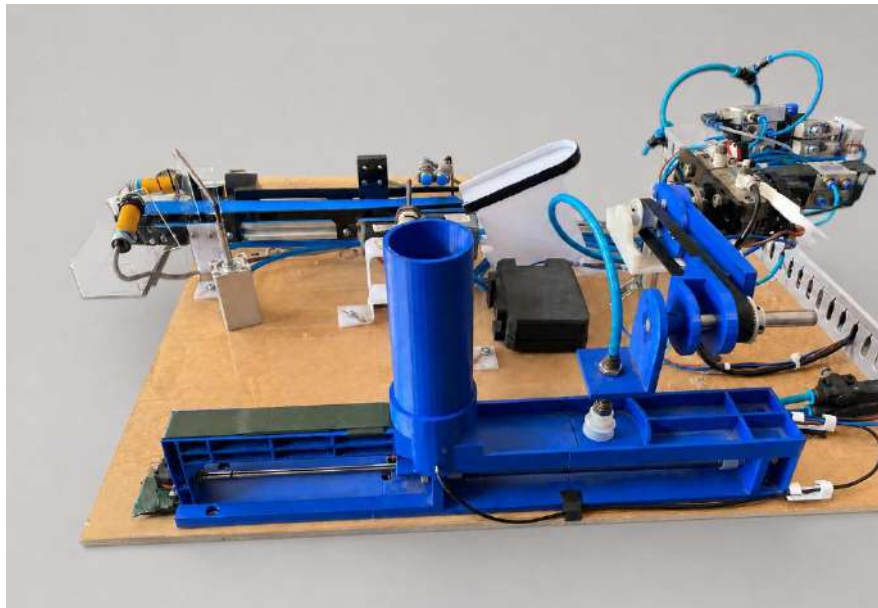
Gambar 4. 1 Desain *Distribution* Dan *Sorting station*
(Sumber: Dokumentasi Pribadi, 2026)

Gambar 4.1 menunjukkan desain awal *distribution station* dan *sorting station* dalam bentuk model tiga dimensi. Desain ini digunakan sebagai acuan penempatan komponen, arah perpindahan benda kerja, serta susunan mekanik sebelum proses pembuatan alat dilakukan.



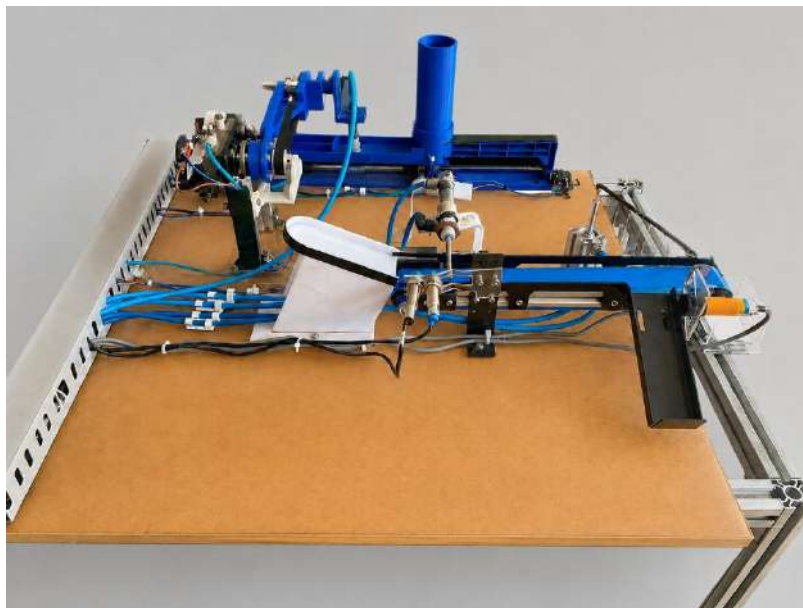
Gambar 4. 2 Hasil Perancangan *Plant Modular Production System* Tampak Depan
(Sumber : Dokumentasi Pribadi, 2026)

Gambar 4.2 menunjukkan hasil akhir perancangan *Modular Production System* dari tampak depan. Dari sudut ini, susunan *distribution station* dan *sorting station* dapat terlihat secara keseluruhan sebagai satu sistem yang saling terhubung.



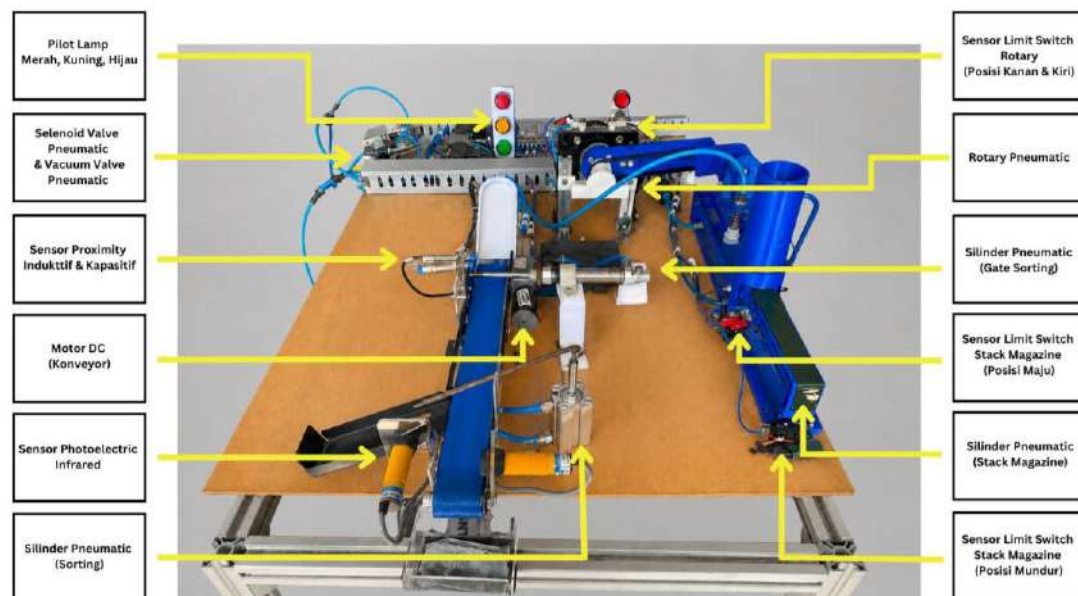
Gambar 4. 3 Hasil Perancangan *Plant Modular Production System* Tampak Sisi Kanan
(Sumber : Dokumentasi Pribadi, 2026)

Gambar 4.3 menampilkan plant dari sisi kanan, sehingga bagian *distribution station* terlihat lebih jelas. Bagian ini berfungsi sebagai tahap awal proses, yaitu menyiapkan dan mendorong benda kerja menuju jalur pemindahan (*changer module*) sebelum diteruskan ke proses penyortiran.



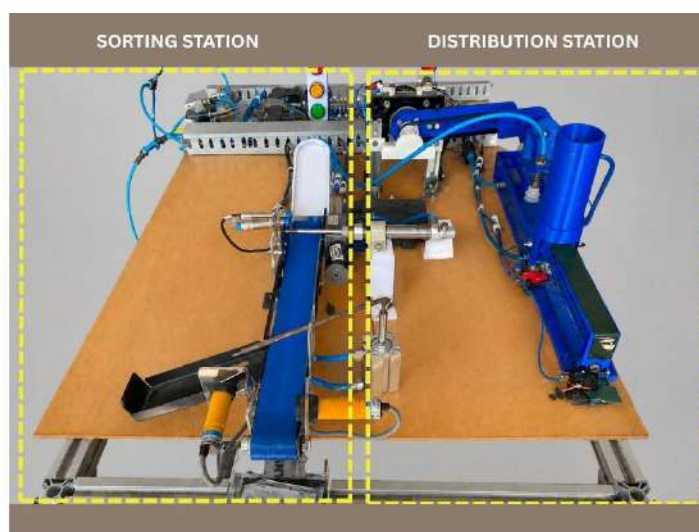
Gambar 4. 4 Hasil Perancangan *Plant Modular Production System* Tampak Sisi Kiri
(Sumber : Dokumentasi Pribadi, 2026)

Gambar 4.4 memperlihatkan plant dari sisi kiri, dengan bagian *sorting station* yang tampak lebih dominan. Pada bagian ini, benda kerja yang telah dikirim dari *distribution station* akan dideteksi, dipindahkan, dan diarahkan sesuai dengan hasil proses penyortiran.



Gambar 4. 5 Hasil Perancangan *Plant Modular Production System* (Tata letak Komponen)
(Sumber : Dokumentasi Pribadi, 2026)

Gambar 4.5 menunjukkan tata letak komponen utama yang digunakan pada plant. Penandaan pada gambar memperjelas posisi sensor, motor DC, lampu indikator, katup pneumatik, silinder, serta mekanisme rotary yang bekerja bersama dalam menjalankan proses distribusi dan penyortiran.



Gambar 4. 6 Hasil Perancangan *Plant Modular Production System* (Pembagian Station)
(Sumber : Dokumentasi Pribadi, 2026)

Gambar 4.6 memperlihatkan pembagian area kerja plant menjadi dua bagian utama, yaitu *sorting station* di sisi kiri dan *distribution station* di sisi kanan. Pembagian ini membantu menunjukkan alur proses, dimulai dari penyaluran benda kerja pada *distribution station* hingga proses identifikasi dan pemisahan pada *sorting station*.

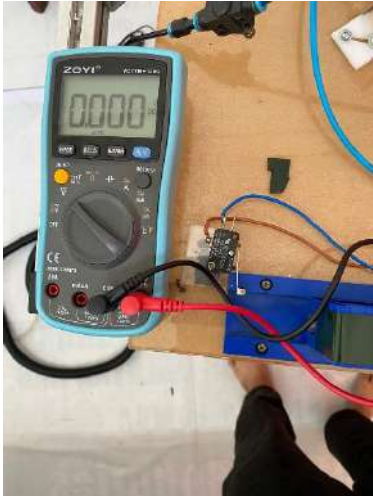
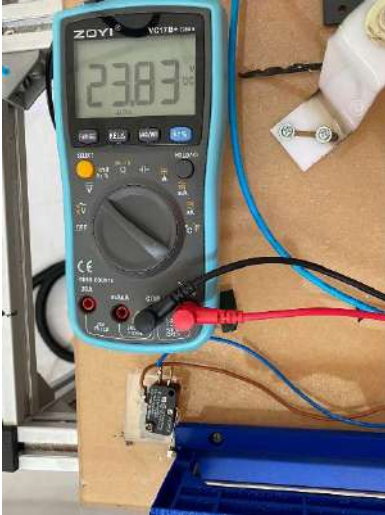
4.2 Hasil Pengujian Sensor dan Aktuator

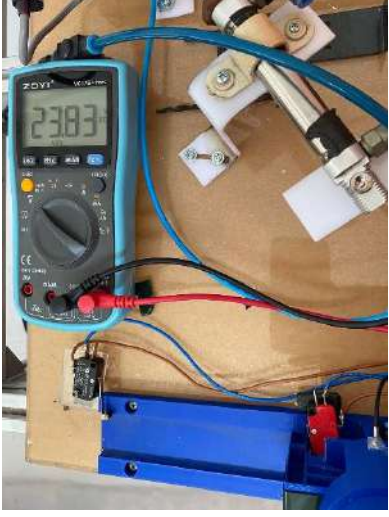
Pengujian sistem merupakan tahap penting untuk memastikan seluruh perangkat keras pada *plant Distribution* dan *Sorting station* dapat berfungsi sesuai dengan perancangan. Pengujian dilakukan pada masing-masing sensor dan aktuator yang terhubung ke sistem kendali berbasis PLC, meliputi sensor *proximity* kapasitif, sensor *proximity* induktif, sensor infrared, limit switch, aktuator pneumatik (silinder dan rotary), serta motor DC konveyor. Seluruh perangkat tersebut dikendalikan oleh PLC Omron CP2E N30 DRA dan Siemens S7-1200, yang berfungsi membaca sinyal *input* dari sensor dan mengaktifkan aktuator sesuai logika program untuk menjalankan proses distribusi dan penyortiran material. Salah satu aspek penting dalam pengujian ini adalah mengevaluasi tingkat keberhasilan kerja setiap komponen melalui pengujian berulang, dengan membandingkan jumlah respon yang sesuai terhadap total percobaan sehingga diperoleh persentase keberhasilan sistem sebagai indikator keandalan sebelum dilanjutkan ke tahap pengujian komunikasi dan sistem Web-SCADA. Salah satu aspek penting dalam pengujian ini adalah melakukan verifikasi fungsi kerja setiap sensor dan aktuator secara langsung. Pengujian dilaksanakan dengan mengamati respon masing-masing komponen terhadap kondisi *input* yang diberikan, kemudian memastikan bahwa keluaran yang dihasilkan telah sesuai dengan logika kendali pada PLC. Hasil pengujian dinyatakan dalam bentuk status berfungsi atau tidak berfungsi sebagai indikator kelayakan perangkat sebelum dilanjutkan ke tahap pengujian komunikasi dan implementasi sistem Web-SCADA.

4.2.1 Hasil Pengujian Sensor Limit Switch

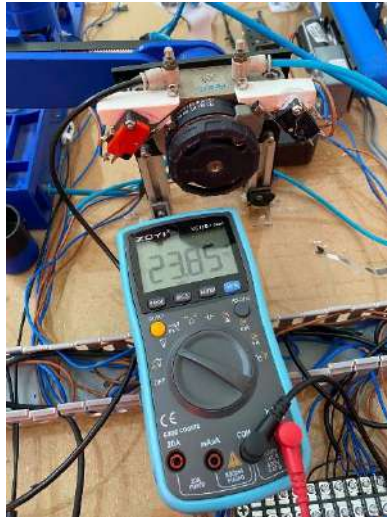
Dalam tugas akhir ini, digunakan Sensor Limit Switch untuk mendeteksi posisi stack magazine dan changer module *Distribution Station*. Dokumentasi hasil pengujian terhadap fungsi kedua tombol tersebut disajikan pada Tabel 4.1.

Tabel 4. 1 Hasil Pengujian Sensor Limit Switch

No	Keterangan	Gambar Pengujian	Hasil Tegangan (V)
1	Sensor Limit switch pada posisi stack magazine mundur tidak terdeteksi aktif.		0V
2	Sensor Limit switch pada posisi stack magazine mundur terdeteksi aktif.		23,83V

No	Keterangan	Gambar Pengujian	Hasil Tegangan (V)
3	Sensor Limit switch pada posisi stack magazine maju tidak terdeteksi aktif.		0V
4	Sensor Limit switch pada posisi stack magazine maju terdeteksi aktif.		23,83V

No	Keterangan	Gambar Pengujian	Hasil Tegangan (V)
5	Sensor Limit switch pada posisi changer module bagian kiri tidak terdeteksi aktif.		0V
6	Sensor Limit switch pada posisi changer module bagian kiri terdeteksi aktif.		23,84V

No	Keterangan	Gambar Pengujian	Hasil Tegangan (V)
7	Sensor Limit switch pada posisi changer module bagian kanan tidak terdeteksi aktif.		0V
8	Sensor Limit switch pada posisi changer module bagian kanan terdeteksi aktif.		23,85V

(Sumber: Dokumentasi Pribadi, 2026)

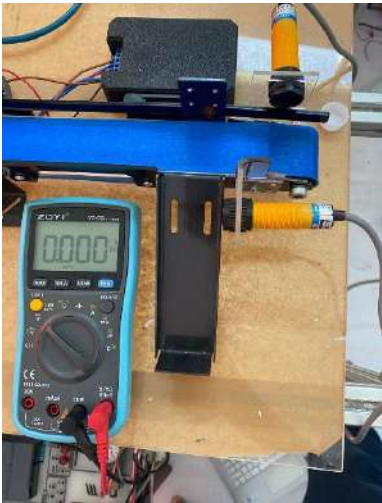
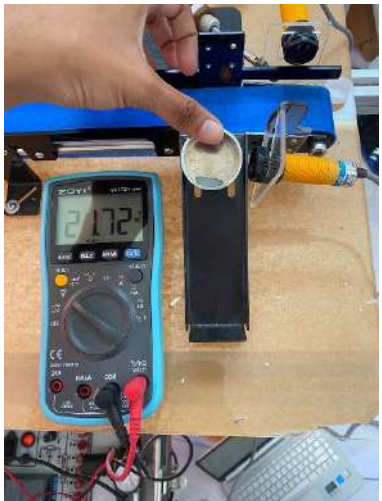
Berdasarkan Tabel 4.1, pengujian terhadap sensor limit switch pada *Distribution Station* menunjukkan bahwa sensor bekerja dengan baik sesuai kondisi yang diberikan. Saat limit switch tidak terdeteksi aktif, tegangan yang terbaca adalah sebesar 0V, yang menandakan bahwa tidak ada sinyal yang dikirim ke PLC. Sebaliknya, ketika limit switch terdeteksi aktif, tegangan meningkat menjadi 24V, yang menunjukkan bahwa sensor berhasil mengirimkan sinyal ke PLC. Dengan demikian, dapat disimpulkan bahwa limit switch mampu mendeteksi kondisi mekanis dengan baik dan memberikan respon sinyal yang sesuai terhadap perubahan posisi.

4.2.2 Hasil Pengujian Sensor Infrared

Dalam tugas akhir ini, digunakan Sensor Infrared untuk mendeteksi object good / bad product pada konveyor sortir. Dokumentasi hasil pengujian terhadap fungsi kedua tombol tersebut disajikan pada Tabel 4.2.

Tabel 4. 2 Pengujian Sensor Infrared

No	Keterangan	Gambar Pengujian	Hasil Tegangan (V)
1	Sensor Infrared (Good Product) tidak mendeteksi Object		0V
2	Sensor Infrared (Good Product) mendeteksi Object		21,75V

No	Keterangan	Gambar Pengujian	Hasil Tegangan (V)
3	Sensor Infrared (Bad Product) tidak mendeteksi Object		0V
4	Sensor Infrared (Bad Product) mendeteksi Object		21,72V

(Sumber: Dokumentasi Pribadi, 2026)

Berdasarkan Tabel 4.2, pengujian terhadap sensor infrared menunjukkan bahwa sensor dapat bekerja dengan baik dalam mendeteksi keberadaan objek. Saat tidak terdapat objek (baik produk good maupun bad), tegangan yang terbaca adalah sebesar 0V, yang menandakan bahwa sensor tidak mengirimkan sinyal ke PLC. Sebaliknya, ketika objek terdeteksi, tegangan meningkat menjadi sekitar 24V, yang menunjukkan bahwa sensor berhasil mengirimkan sinyal *input* ke PLC. Dengan demikian, sensor infrared mampu membedakan kondisi ada dan tidak adanya objek secara akurat sesuai dengan fungsinya pada sistem. Berikut adalah hasil pengujian Sensor Infrared :

Tabel 4. 3 Jarak Deteksi Sensor Infrared

No	Jarak Deteksi (cm)	Pengujian
1	5 cm	Berhasil
2	10 cm	Berhasil
3	15 cm	Berhasil
4	20 cm	Berhasil
5	25 cm	Berhasil
6	30 cm	Berhasil
7	35 cm	Gagal
8	40 cm	Gagal

(Sumber: Dokumentasi Pribadi, 2026)

Berdasarkan Tabel 4,3, pengujian jarak deteksi sensor infrared menunjukkan bahwa sensor mampu mendeteksi objek dengan baik pada jarak 5 cm hingga 30 cm, yang ditandai dengan status pengujian “berhasil”. Namun, pada jarak 35 cm dan 40 cm, sensor tidak mampu mendeteksi objek, sehingga hasil pengujian dinyatakan “gagal”. Hal ini menunjukkan bahwa jarak efektif deteksi sensor infrared berada pada rentang maksimum sekitar 30 cm. Dengan demikian, sensor masih bekerja optimal selama objek berada dalam batas jarak tersebut, dan performa deteksi akan menurun apabila melebihi jarak maksimum yang dimiliki sensor.

4.2.3 Hasil Pengujian Sensor *Proximity* Induktif

Dalam tugas akhir ini, digunakan Sensor *Proximity* Induktif untuk mendeteksi object logam pada konveyor sortir. Dokumentasi hasil pengujian terhadap fungsi kedua tombol tersebut disajikan pada Tabel 4.4.

Tabel 4. 4 Pengujian Sensor *Proximity* Induktif

No	Keterangan	Gambar Pengujian	Hasil Tegangan (V)
1	Sensor <i>Proximity</i> Induktif tidak mendeteksi Object		0V
2	Sensor <i>Proximity</i> Induktif mendeteksi Object		22,31V

(Sumber: Dokumentasi Pribadi, 2026)

Berdasarkan Tabel 4.4, pengujian terhadap sensor *proximity* induktif menunjukkan bahwa sensor dapat bekerja dengan baik dalam mendeteksi keberadaan objek logam. Saat tidak terdapat objek logam di depan sensor, tegangan yang terbaca adalah sebesar 0V, yang menandakan bahwa sensor tidak mengirimkan sinyal ke PLC. Sebaliknya, ketika objek logam terdeteksi, tegangan meningkat menjadi 24V, yang menunjukkan bahwa sensor berhasil mengirimkan sinyal *input* ke PLC. Dengan demikian, sensor *proximity* induktif mampu mendeteksi keberadaan objek logam secara

akurat sesuai dengan fungsinya pada sistem. Berikut adalah hasil pengujian Sensor *Proximity* Induktif :

Tabel 4. 5 Jarak Deteksi Sensor *Proximity* Induktif

No	Jarak Deteksi (cm)	Pengujian
1	0.2 cm	Berhasil
2	0.4 cm	Berhasil
3	0.6 cm	Berhasil
4	0.8 cm	Berhasil
5	1 cm	Gagal
6	1.2 cm	Gagal

(Sumber: Dokumentasi Pribadi, 2026)

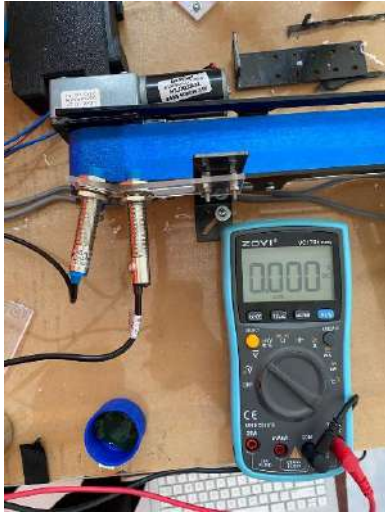
Berdasarkan tabel 4.5, pengujian jarak deteksi, sensor *proximity* induktif menunjukkan bahwa sensor mampu mendeteksi objek logam dengan baik pada jarak 0,2 cm hingga 0,8 cm, yang ditandai dengan hasil pengujian “berhasil”. Namun, pada jarak 1 cm dan 1,2 cm, sensor tidak lagi mampu mendeteksi objek sehingga hasil pengujian dinyatakan “gagal”. Hal ini menunjukkan bahwa kemampuan deteksi sensor menurun seiring bertambahnya jarak antara sensor dan objek.

Dengan demikian, dapat disimpulkan bahwa jarak efektif deteksi sensor *proximity* induktif berada pada rentang kurang dari 1 cm. Sensor bekerja secara optimal pada jarak di bawah batas maksimum tersebut, sehingga dalam implementasi sistem, penempatan sensor harus diperhatikan agar tetap berada dalam jarak deteksi efektif untuk memastikan proses pendeteksian objek logam berjalan dengan stabil dan akurat.

4.2.4 Hasil Pengujian Sensor *Proximity* Kapasitif

Dalam tugas akhir ini, digunakan Sensor *Proximity* Kapasitif untuk mendeteksi object non logam logam pada konveyor sortir. Dokumentasi hasil pengujian terhadap fungsi kedua tombol tersebut disajikan pada Tabel 4.6.

Tabel 4. 6 Pengujian Sensor *Proximity* Kapasitif

No	Keterangan	Gambar Pengujian	Hasil Tegangan (V)
1	Sensor <i>Proximity</i> Kapasitif tidak mendeteksi Objek		0V
2	Sensor <i>Proximity</i> Kapasitif mendeteksi Object		22,39V

(Sumber: Dokumentasi Pribadi, 2026)

Berdasarkan tabel 4.6, sensor *proximity* kapasitif menunjukkan bahwa sensor dapat bekerja dengan baik dalam mendeteksi keberadaan objek non-logam. Saat tidak terdapat objek di depan sensor, tegangan yang terbaca adalah sebesar 0V, yang menandakan bahwa sensor tidak mengirimkan sinyal ke PLC. Sebaliknya, ketika objek terdeteksi, tegangan meningkat menjadi sekitar 24V, yang menunjukkan bahwa sensor berhasil mengirimkan sinyal *input* ke PLC.

Dengan demikian, sensor *proximity* kapasitif mampu mendeteksi keberadaan objek baik logam maupun non-logam sesuai dengan prinsip kerjanya yang berbasis

perubahan kapasitansi. Sensor ini juga menunjukkan respon yang baik terhadap perubahan kondisi objek di sekitarnya, sehingga dapat digunakan secara efektif dalam sistem untuk mendeteksi keberadaan material yang tidak dapat dideteksi oleh sensor *proximity* induktif. Berikut adalah hasil pengujian Sensor *Proximity* Kapasitif :

Tabel 4. 7 Jarak Deteksi *Proximity* Kapasitif

No	Jarak Deteksi (cm)	Pengujian
1	0.2 cm	Berhasil
2	0.4 cm	Berhasil
3	0.6 cm	Berhasil
4	0.8 cm	Berhasil
5	1 cm	Berhasil
6	1.2 cm	Gagal
7	1.4 cm	Gagal

(Sumber: Dokumentasi Pribadi, 2026)

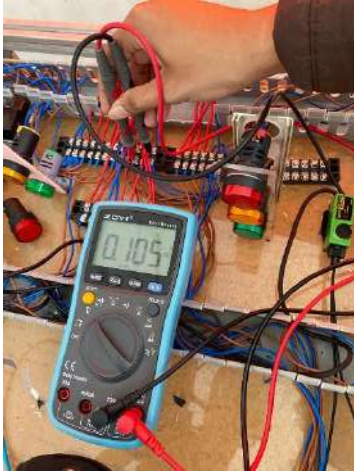
Berdasarkan tabel 4.7, sensor *proximity* kapasitif menunjukkan bahwa sensor mampu mendeteksi objek dengan baik pada jarak 0,2 cm hingga 1 cm, yang ditandai dengan hasil pengujian “berhasil”. Namun, pada jarak 1,2 cm dan 1,4 cm, sensor tidak lagi mampu mendeteksi objek sehingga hasil pengujian dinyatakan “gagal”. Hal ini menunjukkan bahwa kemampuan deteksi sensor menurun seiring bertambahnya jarak antara sensor dan objek.

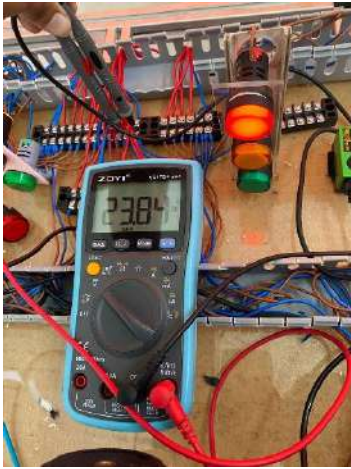
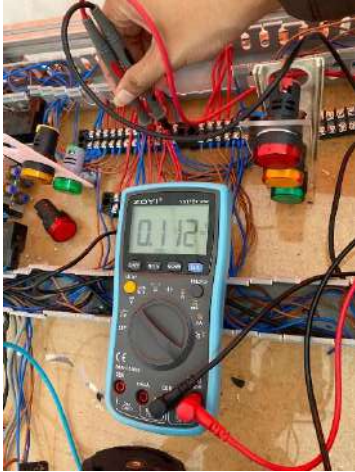
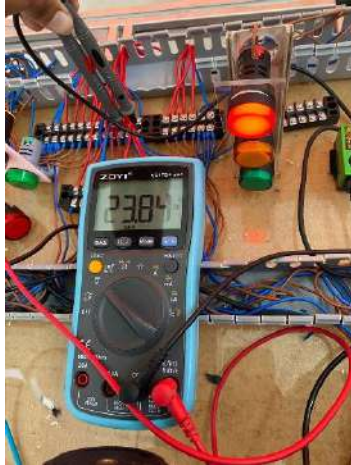
Dengan demikian, dapat disimpulkan bahwa jarak efektif deteksi sensor *proximity* kapasitif berada pada rentang hingga 1 cm. Sensor bekerja secara optimal pada jarak di bawah batas maksimum tersebut, sehingga dalam implementasi sistem, penempatan sensor harus diperhatikan agar tetap berada dalam jarak deteksi efektif untuk memastikan proses pendeteksian objek berjalan dengan stabil dan akurat.

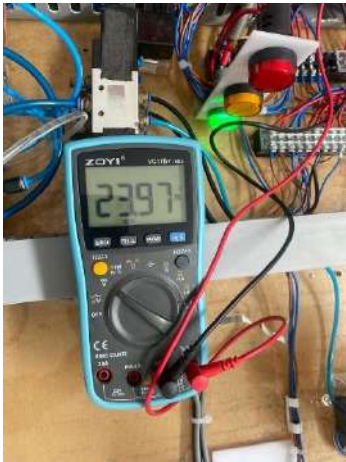
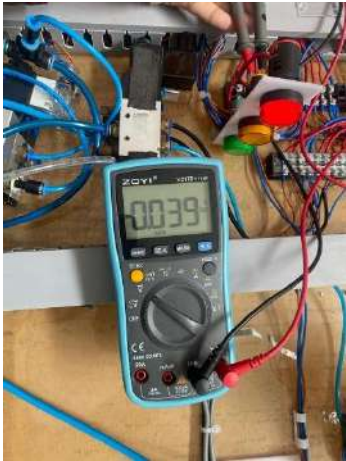
4.2.5 Hasil Pengujian Lampu Indikator Hijau, Merah dan Kuning

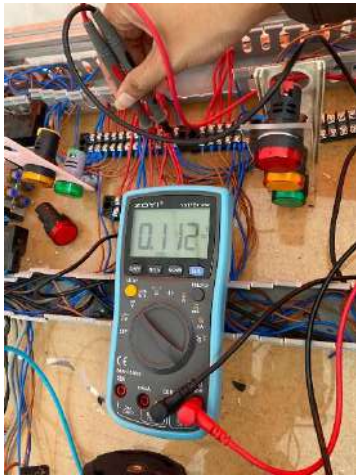
Pengujian lampu indikator hijau, merah dan kuning dilakukan menggunakan power supply 24VDC dan PLC Siemens S7-1200 dan PLC Omron CP2E-N30 DRA sebagai pengendali. Hasil pengujian ditampilkan pada Tabel 4.8.

Tabel 4. 8 Pengujian Lampu Indikator Hijau, Merah dan Kuning

No	Keterangan	Gambar Pengujian	Hasil Tegangan (V)
1	Lampu Hijau pada <i>Distribution Station</i> OFF		
2	Lampu Hijau pada <i>Distribution Station</i> ON		23,73V
3	Lampu Kuning pada <i>Distribution Station</i> OFF		0V

No	Keterangan	Gambar Pengujian	Hasil Tegangan (V)
4	Lampu Kuning pada <i>Distribution</i> Station ON		23,84V
5	Lampu Merah pada <i>Distribution</i> Station OFF		0V
6	Lampu Merah pada <i>Distribution</i> Station ON		23,84V

No	Keterangan	Gambar Pengujian	Hasil Tegangan (V)
7	Lampu Hijau pada <i>Sorting station</i> OFF		0V
8	Lampu Hijau pada <i>Sorting station</i> ON		23,97V
9	Lampu Kuning pada <i>Sorting station</i> OFF		0V

No	Keterangan	Gambar Pengujian	Hasil Tegangan (V)
10	Lampu Kuning pada <i>Sorting station</i> ON		23,96V
11	Lampu Merah pada <i>Sorting station</i> OFF		0V
12	Lampu Merah pada <i>Sorting station</i> ON		23,97V

(Sumber: Dokumentasi Pribadi, 2026)

Berdasarkan Tabel 4.8, pengujian terhadap lampu indikator hijau, merah, dan kuning menunjukkan bahwa ketiganya berfungsi dengan baik sesuai perintah dari PLC. Saat kondisi ON, masing-masing lampu menghasilkan tegangan sebesar 24V, menandakan bahwa sinyal dari PLC berhasil mengaktifkan lampu. Sebaliknya, saat kondisi OFF, tegangan turun menjadi 0V, yang menunjukkan bahwa lampu benar-benar mati dan tidak menerima arus.

4.2.6 Hasil Pengujian Motor DC

Pengujian Motor DC dilakukan menggunakan power supply 24VDC dan PLC Siemens PLC Siemens S7 1200 sebagai pengendali. Hasil pengujian ditampilkan pada Tabel 4.9.

Tabel 4. 9 Pengujian Motor DC

No	Keterangan	Gambar Pengujian	Hasil Tegangan (V)
1	Motor DC OFF		0V

No	Keterangan	Gambar Pengujian	Hasil Tegangan (V)
2	Motor DC ON		23,92V

(Sumber: Dokumentasi Pribadi, 2026)

Berdasarkan Tabel 4.9, pengujian terhadap motor DC menunjukkan bahwa motor berfungsi dengan baik sesuai perintah dari PLC. Saat kondisi OFF, tegangan yang terbaca adalah sebesar 0V, yang menandakan bahwa motor tidak menerima suplai tegangan sehingga tidak berputar. Sebaliknya, saat kondisi ON, tegangan meningkat menjadi sekitar 24V, yang menunjukkan bahwa motor menerima suplai tegangan dari sistem dan dapat beroperasi sebagaimana mestinya.

Dengan demikian, dapat disimpulkan bahwa motor DC mampu bekerja secara optimal dalam sistem sebagai aktuator penggerak. Respon ON dan OFF yang sesuai menunjukkan bahwa kontrol dari PLC terhadap motor berjalan dengan baik. Berikut adalah hasil pengujian Motor DC:

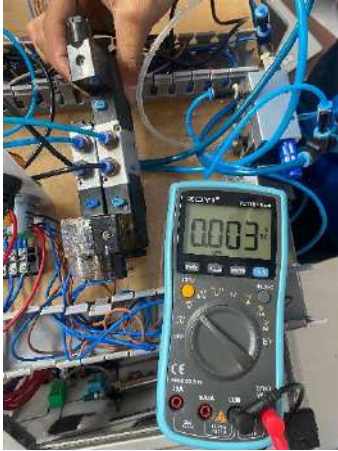
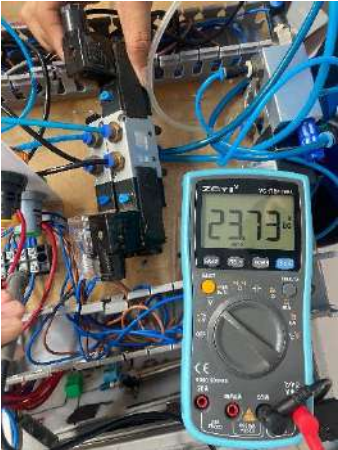
4.2.7 Hasil Pengujian Selenoid Valve

Pengujian selenoid valve dilakukan menggunakan power supply 24VDC dan PLC Siemens PLC Omron CP2E-N30 DRA sebagai pengendali. Hasil pengujian ditampilkan pada Tabel 4.10.

Tabel 4. 10 Pengujian Selenoid Valve

No	Keterangan	Gambar Pengujian	Hasil Tegangan (V)
1	Vacuum Out (5/2 Valve Single Selenoid) OFF		0V
2	Vacuum Out (5/2 Valve Single Selenoid) ON		23,63V
3	Vacuum In (5/2 Valve Single Selenoid) OFF		0V

No	Keterangan	Gambar Pengujian	Hasil Tegangan (V)
4	Vacuum In (5/2 Valve Single Selenoid) ON		23,69V
5	Cylinder Pneumatic (5/2 Valve Single Selenoid) Stack Magazine OFF		0V
6	Cylinder Pneumatic (5/2 Valve Single Selenoid) Stack Magazine ON		23,68V

No	Keterangan	Gambar Pengujian	Hasil Tegangan (V)
7	Rotary Pneumatic (5/2 Valve Double Selenoid) pada posisi kanan OFF		0V
8	Rotary Pneumatic (5/2 Valve Double Selenoid) pada posisi kanan ON		23,73V
9	Rotary Pneumatic (5/2 Valve Double Selenoid) pada posisi kiri OFF		0V

No	Keterangan	Gambar Pengujian	Hasil Tegangan (V)
10	Rotary Pneumatic (5/2 Valve Double Selenoid) pada posisi kiri ON	 //	23,74V

(Sumber: Dokumentasi Pribadi, 2026)

Berdasarkan Tabel 4.10, pengujian terhadap solenoid valve pada sistem pneumatik menunjukkan bahwa komponen tersebut berfungsi dengan baik sesuai perintah dari PLC. Saat kondisi ON, solenoid valve menghasilkan tegangan 24V, yang menandakan bahwa sinyal dari PLC berhasil mengaktifkan aktuator sehingga katup dapat berpindah posisi. Sebaliknya, saat kondisi OFF, tegangan turun menjadi 0V, yang menunjukkan bahwa solenoid valve tidak aktif dan tidak menerima arus dari PLC.

Dengan demikian, dapat disimpulkan bahwa solenoid valve mampu bekerja secara optimal dalam mengontrol aliran udara pada sistem pneumatik sesuai logika kendali yang diberikan. Respon ON dan OFF yang sesuai menunjukkan bahwa integrasi antara PLC dan aktuator berjalan dengan baik, sehingga pergerakan mekanis pada sistem dapat berlangsung secara tepat dan konsisten.

4.2.8 Hasil Pengujian Sensor Photoelectric Fiber

Dalam tugas akhir ini, digunakan Sensor Photoelectric Fiber untuk mendeteksi object pada stack magazine *Distribution* Station. Dokumentasi hasil pengujian terhadap fungsi kedua tombol tersebut disajikan pada Tabel 4.11.

Tabel 4. 11 Pengujian Sensor Photoelectric Fiber

No	Keterangan	Gambar Pengujian	Hasil Tegangan (V)
1	Sensor Photoelectric Fiber tidak mendeteksi Object		0V
2	Sensor Photoelectric Fiber mendeteksi Object		23,85V

(Sumber: Dokumentasi Pribadi, 2026)

Berdasarkan Tabel 4.11, pengujian terhadap sensor photoelectric fiber menunjukkan bahwa sensor mampu bekerja dengan baik sesuai kondisi pengujian yang diberikan. Saat sensor photoelectric fiber tidak mendeteksi objek, tegangan yang terbaca sebesar 0V, yang menandakan bahwa sensor tidak mengirimkan sinyal keluaran ke PLC. Sebaliknya, ketika sensor photoelectric fiber mendeteksi objek, tegangan yang terbaca meningkat menjadi 23,85V, yang menunjukkan bahwa sensor berhasil memberikan sinyal keluaran dengan baik. Dengan demikian, dapat disimpulkan bahwa sensor photoelectric fiber mampu mendeteksi keberadaan objek secara akurat dan memberikan respon tegangan yang sesuai terhadap kondisi deteksi objek pada sistem.

4.3 Konfigurasi Jaringan Perangkat

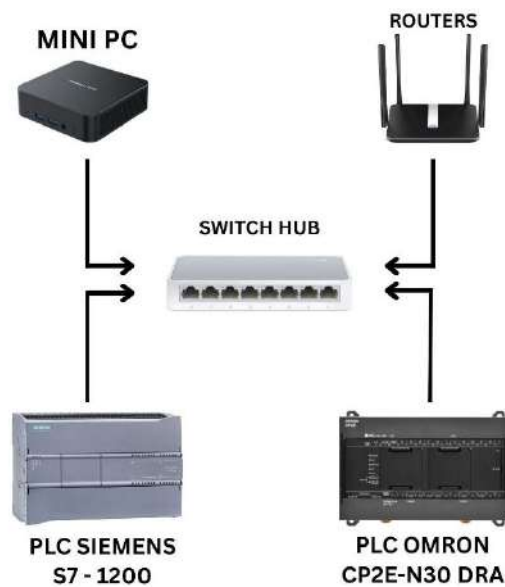
Hasil pengujian jaringan pada penelitian ini menunjukkan bahwa PLC Omron CP2E N30 DRA dan PLC Siemens S7-1200 dapat terhubung dengan Mini PC sebagai *gateway* melalui protokol komunikasi FINS UDP dan S7 TCP/IP. Kedua protokol tersebut digunakan untuk mengirimkan data parameter dari masing-masing PLC ke sistem Web-SCADA secara *real-time*. Mini PC berperan sebagai penghubung (*gateway* multi-protokol) yang menangani proses komunikasi, konversi data, serta pengiriman data ke sistem Web-SCADA.

Dalam implementasinya, sistem Web-SCADA yang digunakan terdiri dari dua jenis, yaitu Web-SCADA Local dan Web-SCADA *Cloud*. Web-SCADA Local digunakan untuk keperluan *Monitoring* dan pengendalian secara langsung dalam jaringan lokal (LAN), sehingga memberikan respon yang cepat dan stabil. Sedangkan Web-SCADA *Cloud* digunakan untuk *Monitoring* dan analisis data yang dapat diakses melalui jaringan internet, sehingga memungkinkan pemantauan sistem dilakukan secara jarak jauh. Dengan konfigurasi yang diterapkan, seluruh data dari sistem *Modular Production System* (MPS), khususnya pada *Distribution* dan *sorting station*, berhasil ditampilkan dan diakses melalui antarmuka berbasis web.

4.3.1 Topologi Jaringan

Topologi jaringan yang digunakan pada sistem ini adalah topologi star, di mana seluruh perangkat terhubung melalui switch sebagai pusat jaringan dalam satu Local Area Network (LAN). Perangkat yang terlibat meliputi PLC Omron CP2E N30 DRA, PLC Siemens S7-1200, Mini PC sebagai *gateway*, serta router sebagai penghubung ke jaringan internet. Seluruh perangkat dalam jaringan lokal dikonfigurasi dalam satu subnet sehingga memungkinkan proses komunikasi data berlangsung secara langsung dan stabil.

Pada implementasinya, PLC Omron dan PLC Siemens terhubung ke Mini PC melalui jaringan lokal untuk proses komunikasi data menggunakan protokol FINS UDP dan S7 TCP/IP. Mini PC kemudian mendistribusikan data ke Web-SCADA Local dalam jaringan LAN untuk kebutuhan *Monitoring* dan pengendalian secara *real-time*, serta ke Web-SCADA *Cloud* melalui router dan koneksi internet untuk keperluan *Monitoring* dan analisis jarak jauh. Dengan topologi ini, sistem mampu mendukung integrasi komunikasi antar perangkat secara terpusat serta memisahkan fungsi kontrol lokal dan *Monitoring* berbasis *cloud*.



Gambar 4. 7 Topologi Jaringan
(Sumber : Dokumentasi Pribadi, 2026)

4.3.2 Konfigurasi IP Address

Konfigurasi IP address pada sistem ini dilakukan untuk memastikan setiap perangkat yang terhubung dalam jaringan memiliki alamat yang unik sehingga proses komunikasi data dapat berjalan dengan baik. Perangkat yang dikonfigurasi meliputi PLC Omron CP2E N30 DRA, PLC Siemens S7-1200, Mini PC sebagai *gateway*, serta router sebagai penghubung ke jaringan internet. Seluruh perangkat tersebut dihubungkan dalam satu jaringan lokal (Local Area Network) menggunakan alamat IP yang berada dalam satu subnet.

Penggunaan IP address statis diterapkan pada seluruh perangkat untuk menjaga kestabilan komunikasi selama sistem beroperasi. Dengan konfigurasi ini, Mini PC dapat melakukan proses pembacaan dan pengiriman data ke masing-masing PLC secara konsisten melalui protokol FINS UDP dan S7 TCP/IP. Selain itu, Mini PC juga dapat mengirimkan data ke sistem Web-SCADA Local melalui jaringan LAN serta ke Web-SCADA *Cloud* melalui router dan koneksi internet. Pada tabel 4.12 adalah konfigurasi IP Address pada masing-masing perangkat pada sistem tugas akhir ini :

Tabel 4. 12 Konfigurasi IP Address pada masing-masing perangkat

No	Perangkat	IP Address	Subnet Mask
1	PLC Siemens S7-1200	192.168.0.4	255.255.255.0
2	PLC Omron CP2E N30 DRA	192.168.0.5	255.255.255.0
3	Mini PC	192.168.0.15	255.255.255.0
4	Router	192.168.0.20	255.255.255.0

(Sumber: Dokumentasi Pribadi, 2026)

4.3.3 Konfigurasi FINS UDP

Konfigurasi komunikasi FINS UDP pada sistem ini dilakukan untuk memungkinkan pertukaran data antara PLC Omron CP2E N30 DRA dengan Mini PC sebagai *gateway* melalui jaringan Ethernet. Protokol FINS (Factory Interface Network Service) digunakan sebagai metode komunikasi yang memungkinkan Mini PC untuk melakukan proses pembacaan (*read*) dan penulisan (*write*) data pada area memori PLC secara langsung.

Pada implementasinya, Mini PC berperan sebagai client yang mengirimkan request data ke PLC Omron sebagai *server* melalui protokol FINS UDP. Parameter konfigurasi yang digunakan meliputi IP address PLC, nomor port, serta pengaturan node address yang terdiri dari Source Node (SA1) dan Destination Node (DA1). Dengan konfigurasi ini, proses komunikasi data antara PLC dan Mini PC dapat berjalan secara *real-time* untuk mendukung sistem *Monitoring* dan pengendalian pada Web-SCADA. Pada tabel 4.13 adalah konfigurasi FINS UDP pada PLC Omron CP2E N30 DRA pada sistem tugas akhir ini :

Tabel 4. 13 Konfigurasi FINS UDP

No	Parameter	Nilai	Keterangan
1	IP Address PLC	192.168.0.5	PLC Omron
2	Port	9600	Default FINS UDP
3	SA1 (Source Node)	15	Node Mini PC
4	DA1 (Destination Node)	5	Node PLC
5	Network Address	0	Default
6	Unit Address	0	CPU Unit

Selain konfigurasi parameter komunikasi, sistem juga menggunakan pengaturan alamat memori PLC sebagai media pertukaran data antara PLC dan Mini PC. Alamat memori yang digunakan meliputi area CIO (Control *Input* Output) untuk *input* dan output digital, serta area DM (Data Memory) untuk penyimpanan data numerik.

Setiap alamat memori dikaitkan dengan fungsi tertentu seperti pembacaan sensor, indikator status, serta pengendalian aktuator. Data pada alamat tersebut dibaca secara periodik oleh. Pada tabel 4.14 adalah alamat memori PLC yang digunakan untuk pembacaan data dalam proses *Monitoring* sistem melalui protokol FINS UDP.

Tabel 4. 14 Alamat Memori FINS UDP

No	Parameter	Fungsi	Address
1	Pilot Lamp Green (ON)	Indikator Status <i>Distribution</i> Station ON	Q: 101.00
2	Pilot Lamp Red (OFF)	Indikator Status <i>Distribution</i> Station OFF	Q: 101.01
3	Pilot Lamp Yellow (EMERGENCY)	Indikator Status <i>Distribution</i> Station Emergency	Q: 101.02
4	Vacuum Out (5/2 Valve Single Selenoid)	Kondisi Operasional Vacuum (Membuang Udara)	Q: 100.02
5	Vacuum In (5/2 Valve Single Selenoid)	Kondisi Operasional Vacuum (Menyedot Udara)	Q: 100.03
6	Cylinder Pneumatic (5/2 Valve Single Selenoid) Stack Magazine	Kondisi Operasional Cylinder penggerak Stack Magazine (Mendorong Object)	Q: 100.04
7	Rotary Pneumatic (5/2 Valve Double Selenoid) pada posisi kanan	Kondisi Operasional Rotary Pneumatic (Bergerak ke sisi kanan)	Q: 100.05
8	Rotary Pneumatic (5/2 Valve Double Selenoid) pada posisi kiri	Kondisi Operasional Rotary Pneumatic (Bergerak ke sisi kiri)	Q: 100.06

No	Parameter	Fungsi	Address
9	Counter Object Stack Magazine	Menghitung jumlah objek yang berada pada stack magazine	D100

(Sumber: Dokumentasi Pribadi, 2026)

4.3.4 Konfigurasi S7 TCP/IP

Konfigurasi komunikasi S7 TCP/IP pada sistem ini dilakukan untuk memungkinkan pertukaran data antara PLC Siemens S7-1200 dengan Mini PC sebagai *gateway* melalui jaringan Ethernet. Protokol S7 digunakan sebagai metode komunikasi yang memungkinkan Mini PC untuk melakukan proses pembacaan (*read*) dan penulisan (*write*) data pada PLC secara langsung melalui alamat data block (DB) yang telah dikonfigurasi.

Pada implementasinya, Mini PC berperan sebagai client yang mengirimkan request data ke PLC Siemens sebagai *server* melalui protokol S7 TCP/IP. Parameter konfigurasi yang digunakan meliputi IP address PLC, nomor rack dan slot, serta alamat data block (DB) yang digunakan sebagai media pertukaran data. Dengan konfigurasi ini, komunikasi data antara PLC dan Mini PC dapat berjalan secara *real-time* untuk mendukung sistem *Monitoring* dan pengendalian pada Web-SCADA. Pada tabel 4.15 adalah konfigurasi S7 TCP / IP pada PLC Siemens S7-1200 pada sistem tugas akhir ini :

No	Parameter	Nilai	Keterangan
1	IP Address PLC	192.168.0.4	PLC Siemens
2	Port	102	Default S7
3	Rack	0	Rack PLC
4	Slot	1	CPU Position

Selain konfigurasi parameter komunikasi, sistem juga menggunakan pengalamatan langsung pada area memori PLC Siemens S7-1200 sebagai media pertukaran data antara PLC dan Mini PC. Area memori yang digunakan meliputi *input* (I), *output* (Q), serta *memory* (M) yang digunakan untuk merepresentasikan kondisi sensor, aktuator, dan variabel internal sistem.

Setiap alamat memori dikaitkan dengan fungsi tertentu seperti pembacaan sensor, indikator status, serta pengendalian aktuator. Data pada alamat tersebut dibaca secara periodik oleh Mini PC menggunakan protokol S7 TCP/IP untuk ditampilkan pada sistem

Web-SCADA. Pada tabel 4.16 adalah alamat memori PLC yang digunakan untuk pembacaan data dalam proses *Monitoring* sistem melalui protokol S7 TCP / IP.

Tabel 4. 16 Alamat Memori S7/TCP/IP

No	Parameter	Fungsi	Address
1	Pilot Lamp Green (ON)	Indikator Status Sorting Station ON	%Q0.3
2	Pilot Lamp Red (OFF)	Indikator Status <i>Distribution</i> Station OFF	%Q0.4
3	Pilot Lamp Yellow (EMERGENCY)	Indikator Status <i>Distribution</i> Station Emergency	%Q0.5
4	Motor DC	Kondisi Operasional Motor Penggerak Konveyor (ON)	%Q0.0
5	Cylinder Pneumatic (5/2 Valve Single Selenoid)	Kondisi Operasional Piston Penyortir (Bergerak Maju)	%Q0.1
7	Counter Good Product	Menghitung jumlah Good Product	%MW100
8	Counter Bad Product	Menghitung jumlah Bad Product	%MW150
9	Total Product	Menghitung jumlah Total Product	%MW200

(Sumber: Dokumentasi Pribadi, 2026)

4.4 Pengujian *Gateway* Multi-protocol

Pengujian performa komunikasi pada sistem ini dilakukan untuk mengetahui kualitas pertukaran data antara PLC Omron CP2E N30 DRA dan PLC Siemens S7-1200 dengan Mini PC sebagai *gateway* multi-protocol dalam sistem Web-SCADA. Komunikasi data dilakukan menggunakan dua protokol, yaitu FINS UDP untuk PLC Omron dan S7 TCP/IP untuk PLC Siemens. Pengujian ini bertujuan untuk memastikan bahwa proses komunikasi data dapat berjalan dengan baik serta untuk mengetahui kinerja sistem dalam mendukung proses *Monitoring* dan pengendalian secara *real-time*.

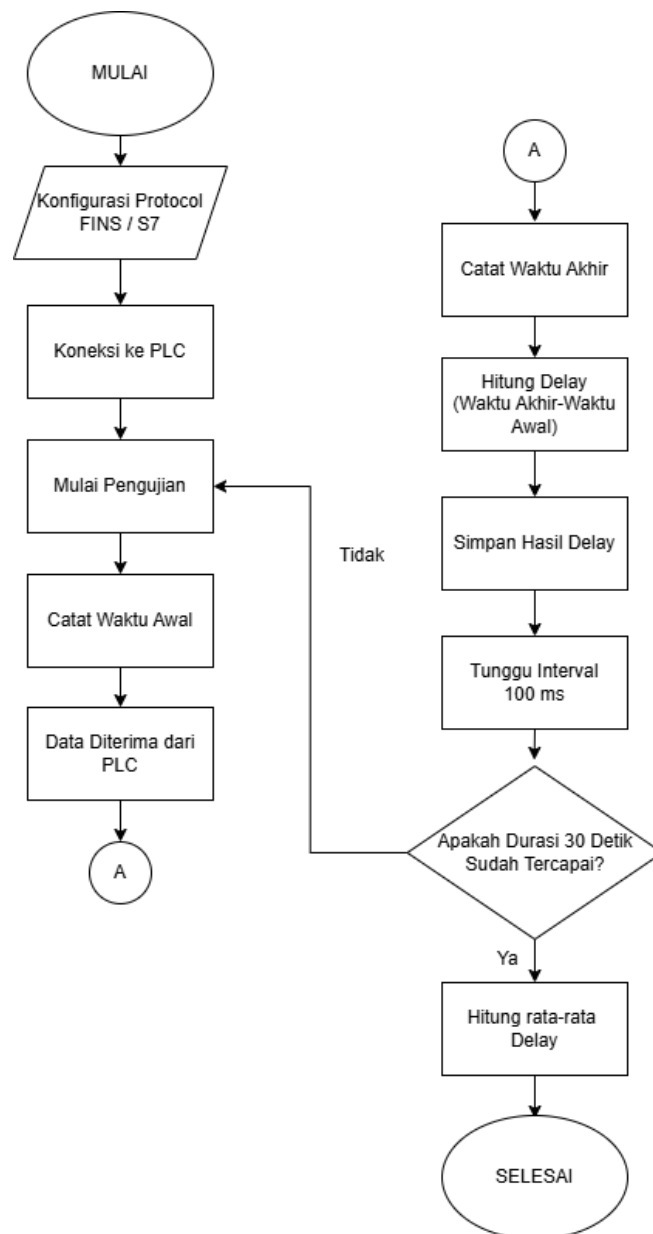
Pengukuran performa komunikasi dilakukan dengan menggunakan parameter waktu tunda (*delay*), yaitu selisih waktu sejak *gateway* mengirimkan permintaan pembacaan data (*read request*) hingga data diterima kembali oleh sistem. Nilai *delay* yang diperoleh digunakan sebagai indikator untuk mengetahui kecepatan respon komunikasi pada masing-masing protokol. Hasil pengujian ini digunakan untuk mengetahui karakteristik komunikasi dari masing-masing protokol, yaitu FINS UDP dan S7 TCP/IP, pada sistem yang dikembangkan.

4.4.1 Pengujian *Delay Read* Alamat PLC (*Local*)

Pengujian *delay read* pada penelitian ini dilakukan untuk mengetahui waktu respon komunikasi data antara PLC dan *gateway* multiprotocol yang tertanam dalam Mini PC pada sistem Web-SCADA. Pengujian dilakukan pada dua protokol komunikasi, yaitu FINS UDP pada PLC Omron CP2E N30-DRA dan S7 TCP/IP pada PLC Siemens S7-1200. Parameter *delay* digunakan sebagai indikator untuk mengetahui kecepatan respon komunikasi data dalam sistem.

Mekanisme pengujian dilakukan dengan cara *gateway* mengirimkan permintaan pembacaan data (*read request*) ke PLC, kemudian sistem mencatat waktu saat permintaan dikirim dan waktu saat data diterima kembali. Selisih antara kedua waktu tersebut dinyatakan sebagai nilai *delay* dalam satu siklus komunikasi. Proses ini dilakukan secara berulang dengan interval pembacaan sebesar 100 ms selama durasi 30 detik, sehingga diperoleh sekitar 300 sampel data untuk setiap skenario pengujian.

Untuk mendapatkan gambaran yang lebih jelas, pengujian alamat memori secara bersamaan (*batch/multiple address*). Pembagian ini dilakukan untuk melihat nilai *delay* pada kondisi sederhana serta kondisi saat sistem membaca beberapa data sekaligus seperti pada penggunaan sebenarnya di Web-SCADA.



Gambar 4. 8 Flowchart Pengujian *Delay Read*
(Sumber : Dokumentasi Pribadi, 2026)

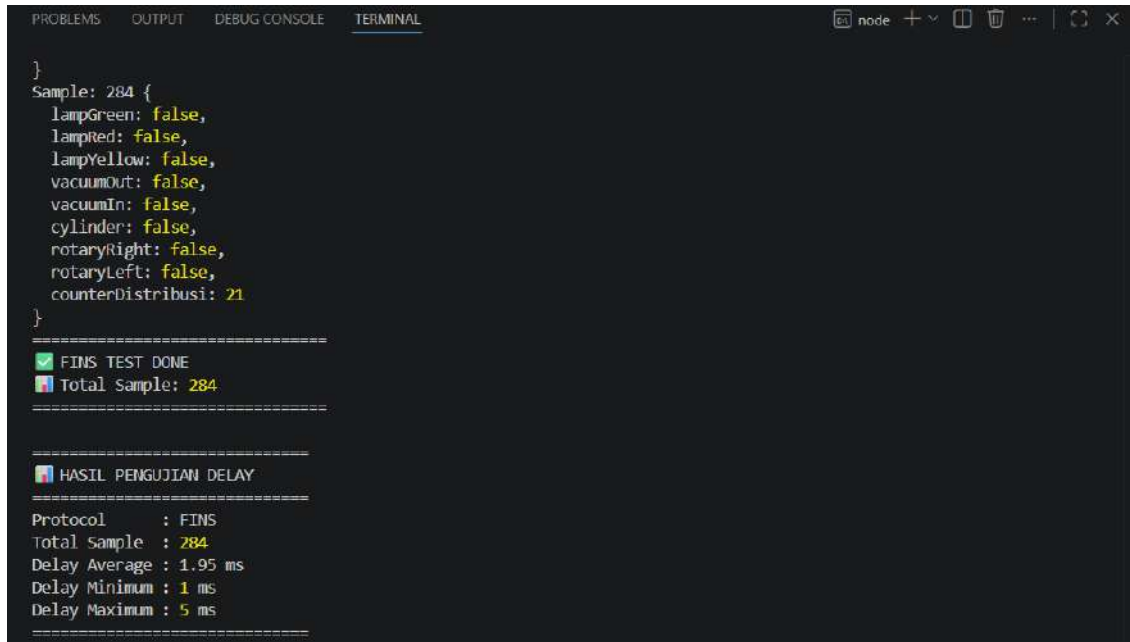
Berdasarkan flowchart pada Gambar 4.8, proses pengujian *delay* dilakukan dengan mengonfigurasi protokol komunikasi FINS UDP atau S7 TCP/IP, kemudian melakukan koneksi ke PLC dan memulai proses pembacaan data secara periodik setiap 100 ms selama 30 detik. Pada setiap siklus pembacaan, sistem mencatat waktu awal sebelum data dibaca dan waktu akhir setelah seluruh data diterima dari PLC, kemudian selisih kedua waktu tersebut digunakan sebagai nilai *delay* komunikasi. Seluruh hasil pengujian *delay* disimpan dan dihitung rata-ratanya untuk mengetahui performa komunikasi data antara *gateway*-multiprotocol dan PLC pada masing-masing protokol yang diuji.

Tabel 4. 17 Pengujian *Delay Read* Alamat PLC

Jumlah Pengujian	FINS UDP (ms)	S7 TCP/IP (ms)
1	1.95	12.67
2	2.00	12.35
3	1.76	12.65
4	1.77	13.36
5	1.77	12.67
6	1.76	12.31
7	1.76	12.61
8	1.80	12.74
9	1.74	13.75
10	1.83	12.80

(Sumber: Dokumentasi Pribadi, 2026)

Tabel 4.17 menyajikan hasil pengujian *delay read* pada protokol FINS UDP dan S7 TCP/IP sebanyak 10 kali pengujian. Hasilnya menunjukkan bahwa FINS UDP memiliki waktu pembacaan yang lebih rendah, yaitu sekitar 1.74–2.00 ms, sedangkan S7 TCP/IP berada pada kisaran 12.31–13.75 ms.



```

}
Sample: 284 {
  lampGreen: false,
  lampRed: false,
  lampYellow: false,
  vacuumOut: false,
  vacuumIn: false,
  cylinder: false,
  rotaryRight: false,
  rotaryLeft: false,
  counterDistribusi: 21
}
=====
✅ FINS TEST DONE
📊 Total Sample: 284
=====

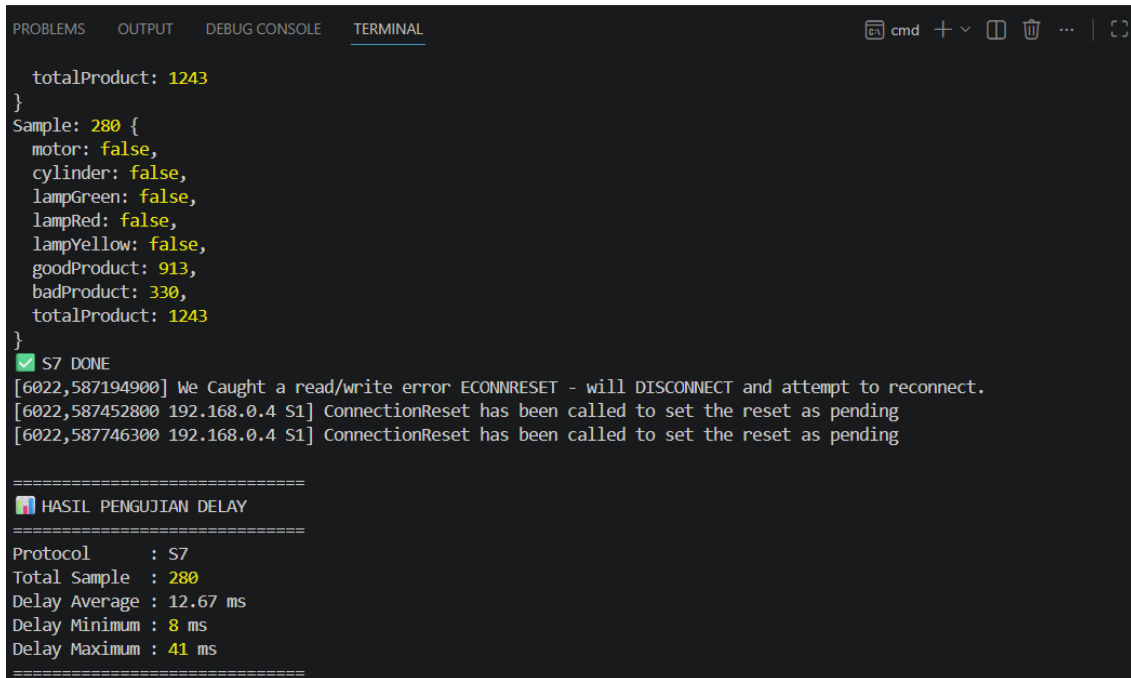
📊 HASIL PENGUJIAN DELAY
=====
Protocol      : FINS
Total Sample  : 284
Delay Average  : 1.95 ms
Delay Minimum  : 1 ms
Delay Maximum  : 5 ms
=====

```

Gambar 4. 9 Pengujian *Delay Read* pada Protocol FINS UDP
(Sumber : Dokumentasi Pribadi, 2026)

Gambar 4.9 menunjukkan proses pengujian *delay read* pada protokol FINS UDP melalui terminal program. Dari hasil pengujian diperoleh rata-rata *delay* sebesar 1,95 ms,

dengan nilai minimum 1 ms dan maksimum 5 ms, sehingga komunikasi pembacaan data dapat berlangsung dengan respons yang cepat.



```
totalProduct: 1243
}
Sample: 280 {
  motor: false,
  cylinder: false,
  lampGreen: false,
  lampRed: false,
  lampYellow: false,
  goodProduct: 913,
  badProduct: 330,
  totalProduct: 1243
}
✓ S7 DONE
[6022,587194900] We Caught a read/write error ECONNRESET - will DISCONNECT and attempt to reconnect.
[6022,587452800 192.168.0.4 S1] ConnectionReset has been called to set the reset as pending
[6022,587746300 192.168.0.4 S1] ConnectionReset has been called to set the reset as pending

=====
🚩 HASIL PENGUJIAN DELAY
=====
Protocol      : S7
Total Sample  : 280
Delay Average  : 12.67 ms
Delay Minimum  : 8 ms
Delay Maximum  : 41 ms
=====
```

Gambar 4. 10 Pengujian *Delay Read* pada Protocol S7 TCP/IP
(Sumber : Dokumentasi Pribadi, 2026)

Gambar 4.10 memperlihatkan hasil pengujian *delay read* pada protokol S7 TCP/IP. Pengujian menghasilkan rata-rata *delay* sebesar 12,67 ms, dengan nilai minimum 9 ms dan maksimum 41 ms, sehingga waktu responsnya lebih besar dibandingkan dengan protokol FINS UDP.

Berdasarkan hasil pengujian, protokol FINS UDP memiliki performa pembacaan data yang lebih cepat dibandingkan S7 TCP/IP. Perbedaan tersebut terlihat dari rata-rata *delay read* FINS UDP yang berada pada kisaran 1,7–2,0 ms, sedangkan S7 TCP/IP berada pada kisaran 12–13 ms. Meskipun memiliki waktu respons yang berbeda, kedua protokol termasuk dalam kategori sangat bagus berdasarkan standar TIPHON karena nilai *delay* yang diperoleh masih berada di bawah 150 ms. Hasil tersebut menunjukkan bahwa FINS UDP lebih responsif untuk proses pembacaan beberapa alamat data PLC secara bersamaan melalui *gateway* multi-protokol, sedangkan S7 TCP/IP tetap mampu menjalankan proses pembacaan data dengan kualitas waktu respons yang sangat baik.

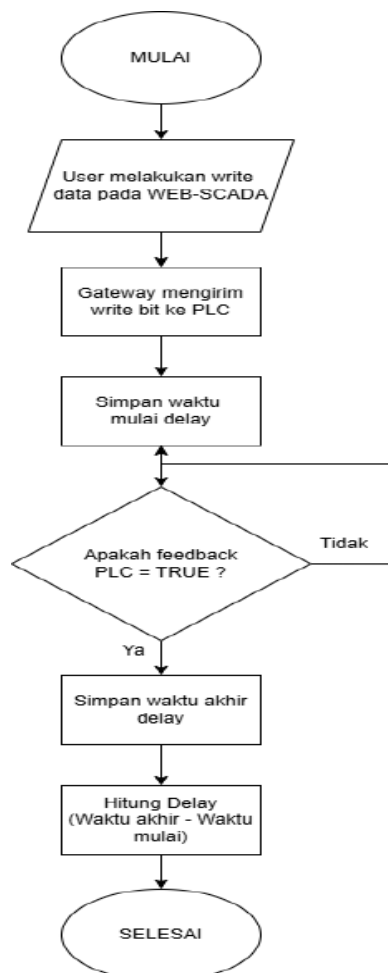
4.4.2 Pengujian *Delay* pada *Write* Alamat PLC (*Local*)

Pengujian *delay write* pada penelitian ini dilakukan untuk mengetahui waktu respon komunikasi data saat sistem Web-SCADA mengirimkan perintah penulisan data (*write*) ke PLC melalui *gateway* multiprotocol yang tertanam pada Mini PC. Pengujian

dilakukan pada dua protokol komunikasi, yaitu FINS UDP pada PLC Omron CP2E N30-DRA dan S7 TCP/IP pada PLC Siemens S7-1200. Parameter *delay* digunakan untuk mengetahui kecepatan respon sistem dalam proses pengiriman data write pada masing-masing protokol komunikasi.

Mekanisme pengujian dilakukan dengan cara Web-SCADA mengirimkan perintah write berupa perubahan nilai bit memori PLC melalui *gateway* multiprotocol. Setelah proses write dilakukan, sistem mencatat waktu awal pengiriman data, kemudian *gateway* melakukan pembacaan status bit memori PLC secara periodik hingga nilai bit berubah menjadi *TRUE*. Selisih antara waktu pengiriman write dan waktu diterimanya perubahan status bit tersebut dinyatakan sebagai nilai *delay* write dalam satu siklus komunikasi.

Pengujian dilakukan secara berulang untuk memperoleh beberapa sampel data *delay* pada masing-masing protokol komunikasi. Data hasil pengujian kemudian digunakan untuk menganalisis karakteristik waktu respon komunikasi write pada sistem Web-SCADA berbasis *gateway* multiprotocol.



Gambar 4. 11 Flowchart Pengujian *Delay Write*
(Sumber : Dokumentasi Pribadi, 2026)

Berdasarkan flowchart pada Gambar 4.11, proses pengujian *delay write* dilakukan dengan cara user melakukan write data melalui sistem Web-SCADA, kemudian *gateway* multiprotocol meneruskan perintah write bit ke PLC menggunakan protokol komunikasi FINS UDP atau S7 TCP/IP. Setelah proses write dilakukan, sistem mencatat waktu awal pengiriman data sebagai waktu mulai *delay*. Selanjutnya *gateway* melakukan pengecekan status feedback bit PLC hingga bernilai *TRUE* sebagai indikasi bahwa data write telah diterima dan diproses oleh PLC. Ketika feedback bernilai *TRUE*, sistem mencatat waktu akhir kemudian menghitung selisih antara waktu akhir dan waktu awal sebagai nilai *delay write* komunikasi. Nilai *delay* tersebut digunakan untuk mengetahui performa komunikasi write antara *gateway* multiprotocol dan PLC pada masing-masing protokol yang diuji.

Tabel 4. 18 Pengujian *Delay Write* Alamat PLC

Jumlah Pengujian	FINS UDP (ms)	S7 TCP/IP (ms)
1	3	38
2	2	21
3	16	23
4	5	52
5	5	38
6	12	23
7	16	29
8	3	27
9	3	23
10	12	21

(Sumber: Dokumentasi Pribadi, 2026)

Tabel 4.18 menyajikan hasil pengujian *delay write* pada protokol FINS UDP dan S7 TCP/IP sebanyak 10 kali percobaan. Nilai *delay* FINS UDP berada pada rentang 2–16 ms dengan rata-rata sekitar 7,7 ms, sedangkan S7 TCP/IP berada pada rentang 21–52 ms dengan rata-rata sekitar 29,5 ms. Hasil ini menunjukkan bahwa proses penulisan data menggunakan FINS UDP berlangsung lebih cepat.



```
D:\BITSMILLAH TUGAS AKHIR\SISTEM TUGAS AKHIR\gateway-fuxa>node gateway-fuxa.js
Modbus TCP Server running on port 502
⚡ DELAY WRITE FINS = 3 ms
⚡ DELAY WRITE FINS = 2 ms
⚡ DELAY WRITE FINS = 16 ms
⚡ DELAY WRITE FINS = 5 ms
⚡ DELAY WRITE FINS = 5 ms
⚡ DELAY WRITE FINS = 12 ms
⚡ DELAY WRITE FINS = 12 ms
⚡ DELAY WRITE FINS = 16 ms
⚡ DELAY WRITE FINS = 3 ms
⚡ DELAY WRITE FINS = 3 ms
⚡ DELAY WRITE FINS = 12 ms
```

Gambar 4. 12 Pengujian *Delay Write* pada Protocol FINS UDP
(Sumber : Dokumentasi Pribadi, 2026)

Gambar 4.12 menunjukkan proses pengujian *delay write* pada protokol FINS UDP melalui terminal program. Nilai yang diperoleh berubah-ubah pada setiap percobaan, tetapi secara keseluruhan masih berada pada kisaran yang rendah, sehingga proses pengiriman perintah dari *gateway* menuju PLC dapat dilakukan dengan respons yang relatif cepat.



```
D:\BITSMILLAH TUGAS AKHIR\SISTEM TUGAS AKHIR\gateway-fuxa>node gateway-fuxa.js
Modbus TCP Server running on port 502
⚡ DELAY WRITE S7 = 38 ms
⚡ DELAY WRITE S7 = 21 ms
⚡ DELAY WRITE S7 = 23 ms
⚡ DELAY WRITE S7 = 52 ms
⚡ DELAY WRITE S7 = 38 ms
⚡ DELAY WRITE S7 = 23 ms
⚡ DELAY WRITE S7 = 29 ms
⚡ DELAY WRITE S7 = 27 ms
⚡ DELAY WRITE S7 = 23 ms
⚡ DELAY WRITE S7 = 21 ms
```

Gambar 4. 13 Pengujian *Delay Write* pada Protocol S7 TCP/IP
(Sumber : Dokumentasi Pribadi, 2026)

Gambar 4.13 memperlihatkan proses pengujian *delay write* pada protokol S7 TCP/IP. Hasil pengujian menunjukkan nilai *delay* yang lebih besar dibandingkan FINS UDP, yaitu antara 21 ms hingga 52 ms, sehingga waktu yang dibutuhkan untuk menuliskan data ke PLC Siemens relatif lebih lama.

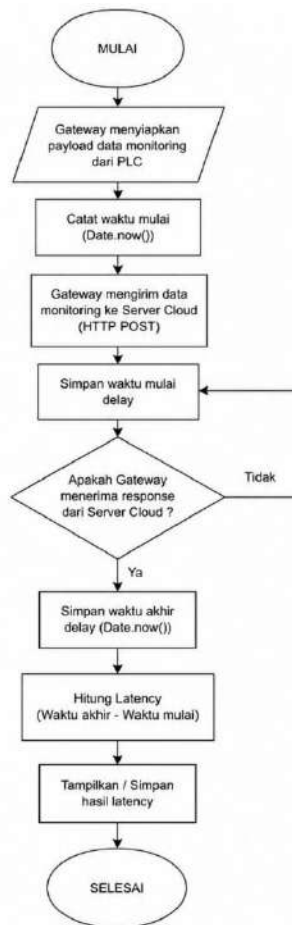
Berdasarkan hasil pengujian, protokol FINS UDP memiliki performa *write* yang lebih baik dibandingkan S7 TCP/IP. Rata-rata *delay write* FINS UDP sebesar 7,7 ms, sedangkan S7 TCP/IP sebesar 29,5 ms. Meskipun terdapat perbedaan waktu respons, kedua protokol masih termasuk dalam kategori sangat bagus berdasarkan standar TIPHON karena seluruh nilai rata-rata *delay* berada di bawah 150 ms.

4.4.3 Pengujian *Latency Komunikasi Gateway ke Cloud*

Pengujian *latency* komunikasi *gateway* ke *cloud* pada penelitian ini dilakukan untuk mengetahui waktu respons komunikasi data saat *gateway* multiprotocol mengirimkan data *Monitoring* menuju *server backend cloud* melalui layanan *REST API*.

Pengujian ini bertujuan untuk mengevaluasi performa komunikasi antara *gateway* yang tertanam pada *Mini PC* dengan *server backend cloud* dalam proses pengiriman data *Monitoring* secara *real-time*. Parameter *latency* digunakan untuk mengetahui waktu yang dibutuhkan sejak data dikirim oleh *gateway* hingga *gateway* menerima *HTTP response* dari *server backend cloud*.

Mekanisme pengujian dilakukan dengan cara *gateway* membentuk *payload* data *Monitoring* hasil akuisisi dari PLC Omron CP2E N30-DRA dan PLC Siemens S7-1200, kemudian mengirimkannya ke *server backend cloud* menggunakan metode *HTTP POST*. Sebelum proses pengiriman dilakukan, sistem mencatat waktu awal menggunakan fungsi *Date.now()*, kemudian menghitung selisih waktu setelah *HTTP response* diterima dari *server backend cloud*. Selisih kedua waktu tersebut dinyatakan sebagai nilai *latency* komunikasi dalam satu siklus pengiriman data. Pengujian dilakukan secara berulang untuk memperoleh beberapa sampel nilai *latency* yang selanjutnya digunakan sebagai dasar analisis performa komunikasi antara *gateway* dan *server backend cloud*.



Gambar 4. 14 Flowchart Pengujian *Latency* Komunikasi *Gateway* ke *Cloud*
(Sumber : Dokumentasi Pribadi, 2026)

Berdasarkan flowchart pada Gambar 4.14, proses pengujian *latency* komunikasi *gateway* ke *cloud* diawali dengan *gateway* membentuk *payload* data *Monitoring* hasil akuisisi dari PLC, kemudian sistem mencatat waktu awal pengiriman menggunakan fungsi *Date.now()*. Selanjutnya *gateway* mengirimkan *payload* tersebut ke *server backend cloud* menggunakan metode *HTTP POST* dan menunggu *HTTP response* dari *server*. Apabila *gateway* telah menerima respons dari *server backend cloud*, sistem mencatat waktu akhir menggunakan fungsi *Date.now()*, kemudian menghitung selisih antara waktu akhir dan waktu awal sebagai nilai *latency* komunikasi. Nilai *latency* tersebut digunakan untuk mengetahui performa komunikasi pengiriman data *Monitoring* antara *gateway* multiprotocol dan *server backend cloud* pada sistem Web-SCADA berbasis *cloud*.

Tabel 4. 19 Pengujian *Latency* Komunikasi *Gateway* ke *Cloud*

Jumlah Pengujian	<i>Latency</i> (ms)
1	284
2	309
3	226
4	308
5	352
6	330
7	352
8	370
9	308
10	313

(Sumber: Dokumentasi Pribadi, 2026)



Gambar 4. 15 Pengujian *Latency* Komunikasi *Gateway* ke *Cloud*
(Sumber : Dokumentasi Pribadi, 2026)

Berdasarkan hasil pengujian pada Tabel 4.19, diperoleh nilai *latency* komunikasi *gateway* ke *cloud* pada 10 kali pengujian dengan rentang 226–370 ms dan nilai rata-rata sebesar 315,2 ms. Jika dibandingkan dengan standar TIPHON, sebanyak dua hasil

pengujian, yaitu 226 ms dan 284 ms, termasuk kategori bagus karena berada pada rentang 150–300 ms. Sementara itu, delapan hasil pengujian lainnya berada pada rentang 300–450 ms sehingga termasuk kategori sedang. Dengan demikian, secara keseluruhan nilai rata-rata *latency* komunikasi *gateway* ke *cloud* termasuk kategori sedang berdasarkan standar TIPHON.

Nilai *latency* tersebut menunjukkan bahwa waktu respons komunikasi antara *gateway* dan *server backend cloud* masih mengalami variasi pada setiap pengujian. Perbedaan nilai ini dapat dipengaruhi oleh kondisi jaringan internet, proses pengiriman data, serta waktu respons dari *server*. Meskipun demikian, seluruh data berhasil dikirim dan memperoleh respons dari *server*, sehingga komunikasi *gateway* ke *cloud* dapat berjalan dengan baik selama proses pengujian. Gambar 4.15 menunjukkan proses pengukuran *latency* yang dihitung sejak *gateway* mengirimkan *payload* data monitoring melalui metode HTTP POST hingga menerima respons dari *server backend cloud*.

4.5 Hasil Perancangan Software

Setelah konfigurasi pengujian pada jaringan, sensor, aktuator, dan komponen perangkat keras lainnya selesai dilakukan, tahap selanjutnya adalah melakukan pengujian terhadap perangkat lunak yang digunakan pada sistem. Pengujian ini meliputi program PLC yang berfungsi mengendalikan proses pada *Modular Production System*, backend *Gateway Connector* berbasis Node.js yang menangani komunikasi data antara PLC dan *server*, WEB-SCADA lokal yang digunakan untuk *Monitoring* dan kontrol sistem, serta website *Monitoring* dan analisis berbasis *cloud* untuk menampilkan data *Monitoring* dan analisis sistem.

4.5.1 Program dan Konfigurasi PLC Omron CP2E-N30 DRA

PLC Omron CP2E-N30 DRA digunakan sebagai kontrol utama pada *Distribution Station* dalam sistem *Modular Production System*. PLC ini bertugas mengendalikan proses distribusi benda kerja berdasarkan pembacaan sensor serta mengaktifkan aktuator yang digunakan pada stasiun tersebut. Program PLC dibuat menggunakan *software CX-Programmer* dengan bahasa pemrograman ladder diagram. Selain menjalankan logika kontrol pada *Distribution Station*, PLC juga mengirimkan data status sensor dan aktuator ke sistem WEB-SCADA melalui backend *server* menggunakan protokol komunikasi FINS. Data tersebut kemudian digunakan untuk keperluan *Monitoring* dan kontrol sistem secara *real-time* melalui antarmuka web. Program PLC juga menyediakan beberapa alamat memori yang digunakan sebagai media pertukaran data antara PLC dan sistem

SCADA. Berikut merupakan daftar alamat PLC yang akan diterapkan dapat dilihat pada Tabel 4.20.

Tabel 4. 20 I/O List PLC Omron

No	PLC	Hardware	Address
1	DI 0	Sensor Photoelectric Fiber	I: 0.00
2	DI 1	Sensor Limit Switch pada posisi stack magazine maju	I: 0.01
3	DI 2	Sensor Limit Switch 1	I: 0.02
4	DI 3	Sensor Limit Switch 2	I: 0.03
5	DI 4	Sensor Limit Switch pada posisi stack magazine mundur	I: 0.04
6	DO 2	Vacuum Out (5/2 Valve Single Selenoid)	Q: 100.02
7	DO 3	Vacuum In (5/2 Valve Single Selenoid)	Q: 100.03
8	DO 4	Cylinder Pneumatic (5/2 Valve Single Selenoid) Stack Magazine	Q: 100.04
9	DO 5	Rotary Pneumatic (5/2 Valve Double Selenoid) pada posisi kanan	Q: 100.05
10	DO 6	Rotary Pneumatic (5/2 Valve Double Selenoid) pada posisi kiri	Q: 100.06
11	DO 17	Pilot Lamp Green (ON)	Q: 101.00
12	DO 18	Pilot Lamp Red (OFF)	Q: 101.01
13	DO 19	Pilot Lamp Yellow (Emergency)	Q: 101.02

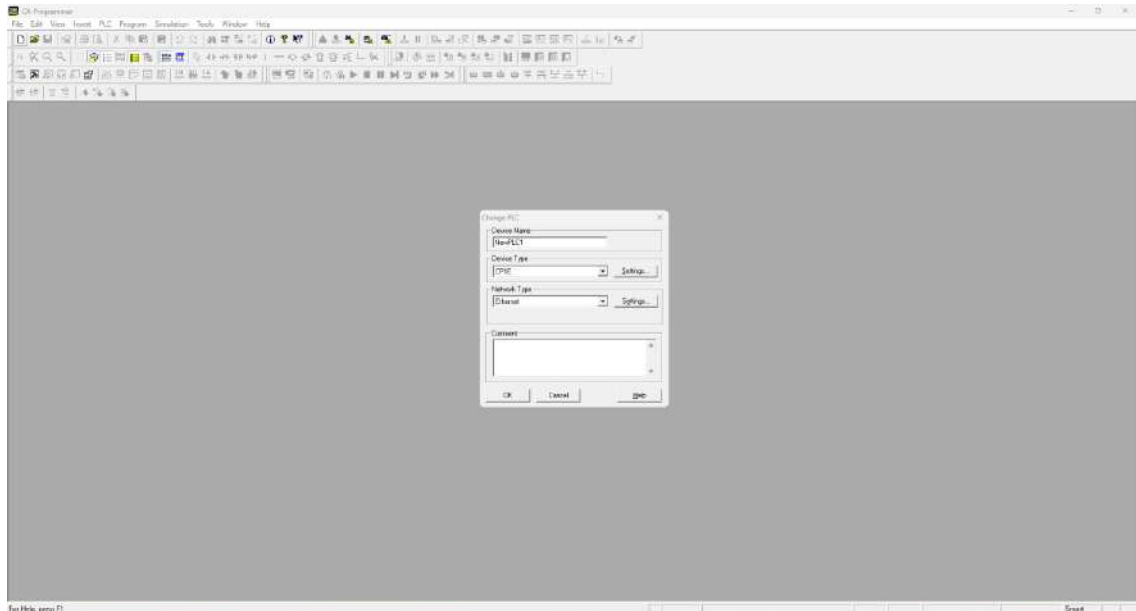
(Sumber: Dokumentasi Pribadi, 2026)

Tabel 4. 21 Memory Address PLC Omron

No	Memory Address	Keterangan
1	D100	Counter Object Pada Stack Magazine

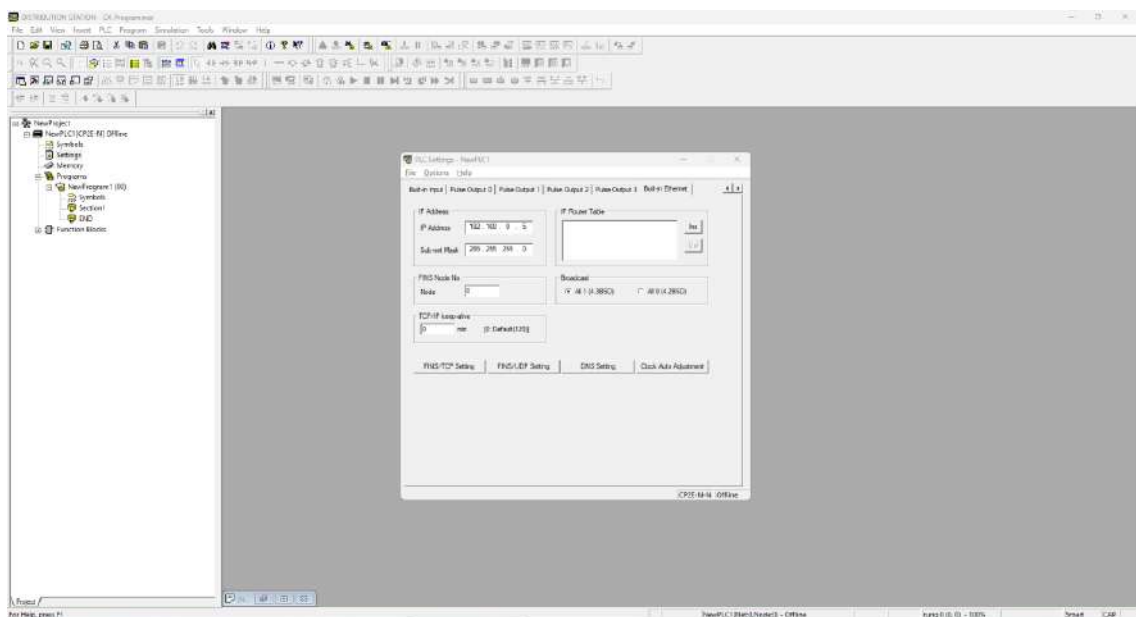
(Sumber: Dokumentasi Pribadi, 2026)

Setelah melakukan addressing pada PLC, langkah selanjutnya adalah mengonfigurasi PLC Omron CP2E-N30 DRA menggunakan *software* CX-Programmer. Aplikasi ini digunakan untuk merancang logika program dengan pemrograman ladder diagram pada PLC Omron CP2E-N30 DRA.



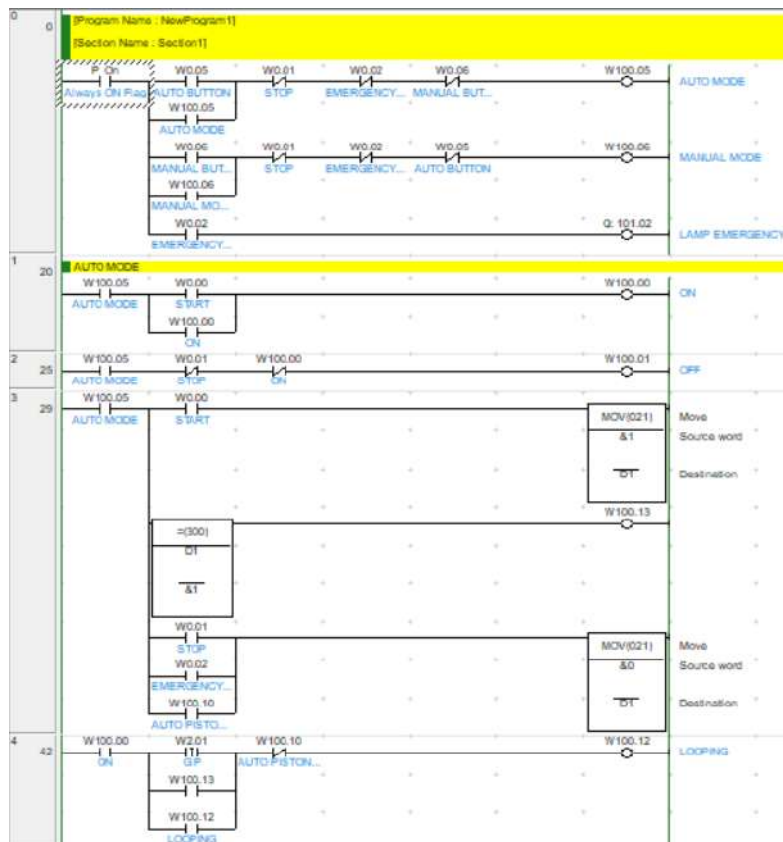
Gambar 4. 16 Konfigurasi Tipe PLC Omron
(Sumber : Dokumentasi Pribadi, 2026)

Gambar 4.16 menunjukkan pemilihan tipe PLC Omron yang akan digunakan dalam penelitian ini, yaitu PLC Omron CP2E-N30 DRA dengan Network Type Ethernet. PLC tipe ini dilengkapi dengan 18 Digital *Input* dan 12 Digital Output.



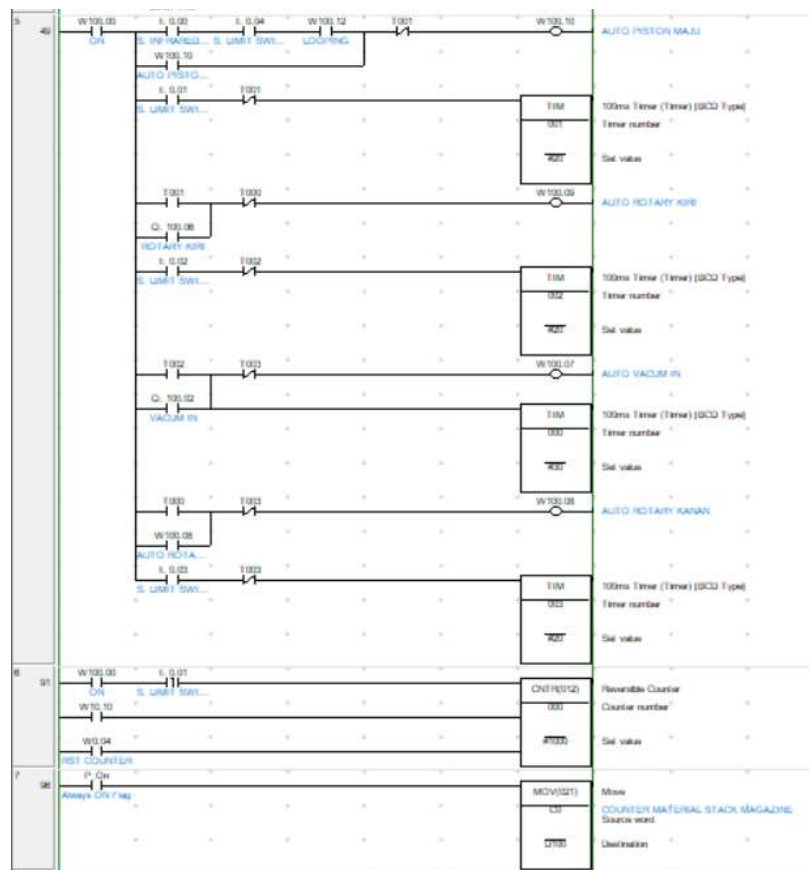
Gambar 4. 17 Konfigurasi Alamat IP PLC Omron
(Sumber : Dokumentasi Pribadi, 2026)

Setelah konfigurasi tipe PLC Omron yang digunakan, langkah selanjutnya adalah melakukan pengaturan komunikasi yang akan digunakan, yaitu melalui koneksi Ethernet dengan IP address yang tercantum pada Gambar 4.17.



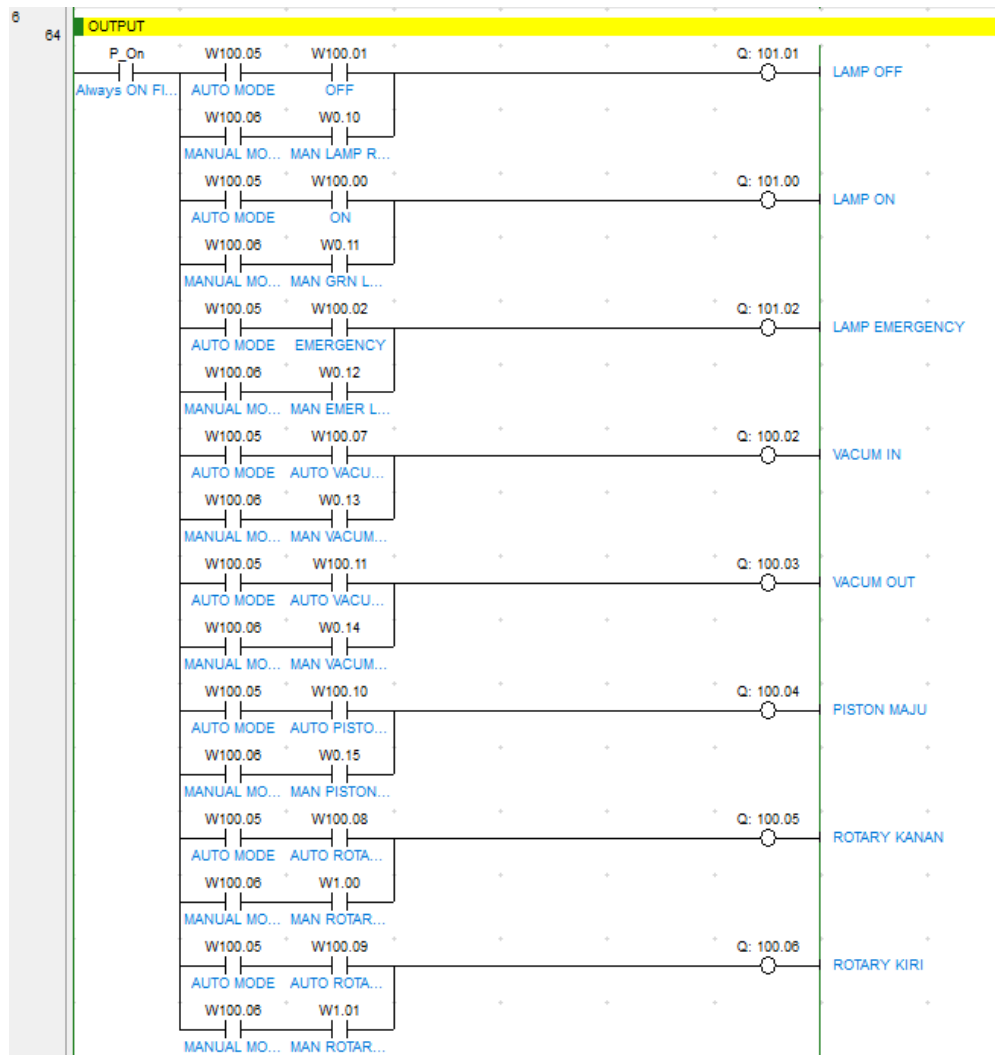
Gambar 4. 18 Progam Ladder PLC Omron (1)
(Sumber : Dokumentasi Pribadi, 2026)

Gambar 4.18 menunjukkan logika pemrograman ladder pada PLC Omron yang digunakan untuk mengendalikan fungsi dasar sistem, meliputi pemilihan mode operasi Auto dan Manual, serta logika Start, Stop, dan *Emergency Stop*. Program ini memastikan bahwa sistem hanya dapat beroperasi pada satu mode yang dipilih, sementara tombol Start digunakan untuk menjalankan sistem, tombol Stop untuk menghentikan proses, dan *Emergency Stop* untuk menghentikan sistem secara langsung sebagai mekanisme keselamatan. Selain itu, program juga menerima sinyal looping dari PLC Siemens sebagai indikator bahwa proses pada Sorting Station telah selesai, sehingga PLC Omron dapat memulai kembali siklus distribusi secara otomatis tanpa memerlukan intervensi operator.



Gambar 4. 19 Progam Ladder PLC Omron (2)
(Sumber : Dokumentasi Pribadi, 2026)

Gambar 4.19 menunjukkan logika pemrograman ladder pada PLC Omron yang digunakan untuk mengendalikan urutan kerja (*sequence*) pada *Distribution Station*. Program ini mengatur proses distribusi benda kerja secara berurutan, dimulai dari pergerakan piston maju, dilanjutkan dengan rotary berputar ke kiri, vacuum aktif untuk mengambil benda kerja, kemudian rotary kembali ke posisi semula, hingga benda kerja berhasil dipindahkan ke proses berikutnya. Setiap tahapan dikendalikan menggunakan sensor sebagai umpan balik dan timer sebagai pengatur waktu perpindahan antarproses, sehingga *sequence* proses dapat berjalan secara otomatis sesuai dengan urutan yang telah dirancang. Selain itu, program juga dilengkapi dengan fungsi counter untuk menghitung jumlah benda kerja yang telah berhasil didistribusikan dari stack magazine.



Gambar 4. 20 Progam Ladder PLC Omron (2)
(Sumber : Dokumentasi Pribadi, 2026)

Pada Gambar 4.20 ditunjukkan logika program *ladder* pada PLC Omron yang digunakan untuk mengendalikan seluruh keluaran (*output*) pada *distribution station*, meliputi lampu indikator, *vacuum*, piston, dan *rotary actuator*. Pada mode otomatis, setiap keluaran diaktifkan sesuai urutan proses (*sequence*) yang telah dirancang, sedangkan pada mode manual setiap keluaran dapat dikendalikan secara individual oleh operator melalui Web-SCADA untuk keperluan pengujian dan pemeliharaan..

4.5.2 Program dan Konfigurasi PLC Siemens S7 – 1200

PLC Siemens S7-1200 digunakan sebagai pengendali pada *sorting station* dalam sistem *Modular Production System*. PLC ini bertugas menjalankan proses berdasarkan pembacaan sensor serta mengaktifkan aktuator pada stasiun tersebut. Program PLC dibuat menggunakan perangkat lunak TIA Portal V18 dengan bahasa pemrograman

ladder diagram. Selain menjalankan logika kendali, PLC juga mengirimkan data status sensor dan aktuator ke sistem Web-SCADA melalui *gateway* dan *backend server* menggunakan protokol S7 TCP/IP pada jaringan Ethernet. Data tersebut digunakan untuk keperluan *monitoring* sistem secara *real-time* melalui antarmuka web. Program PLC juga menyediakan sejumlah alamat memori sebagai media pertukaran data antara PLC dan sistem SCADA. Daftar alamat PLC yang digunakan ditunjukkan pada Tabel 4.22.

Tabel 4. 22 I/O List PLC Siemens

No	PLC	Hardware	Address
1	DI 0	Sensor <i>Proximity</i> Induktif	%I0.0
2	DI 1	Sensor Infrared 1	%I0.1
3	DI 2	Sensor <i>Proximity</i> Kapasitif	%I0.2
4	DI 3	Sensor Infrared 2	%I0.3
5	DO 0	Motor DC	%Q0.0
6	DO 1	Cylinder Pneumatic (5/2 Valve Single Selenoid)	%Q0.1
7	DO 3	Pilot Lamp Green (ON)	%Q0.3
8	DO 4	Pilot Lamp Red (OFF)	%Q0.4
9	DO 5	Pilot Lamp Yellow (Emergency)	%Q0.5

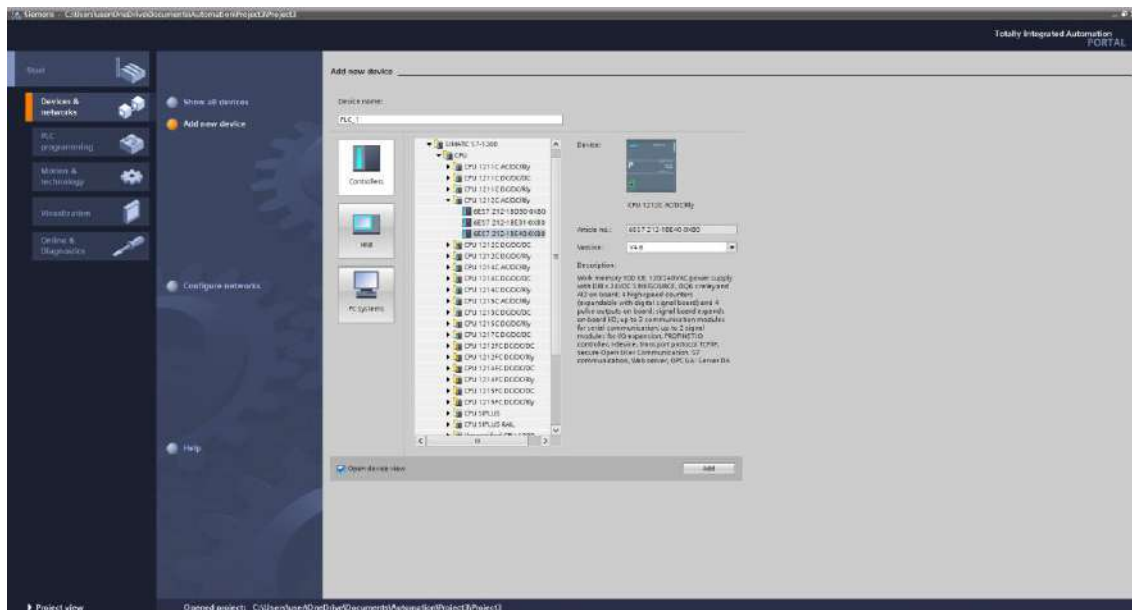
(Sumber: Dokumentasi Pribadi, 2026)

Tabel 4. 23 Memory Address PLC Siemens

No	Memory Address	Keterangan
1	%MW100	Counter Good Product
2	%MW150	Counter Bad Product
3	%MW200	Total Product

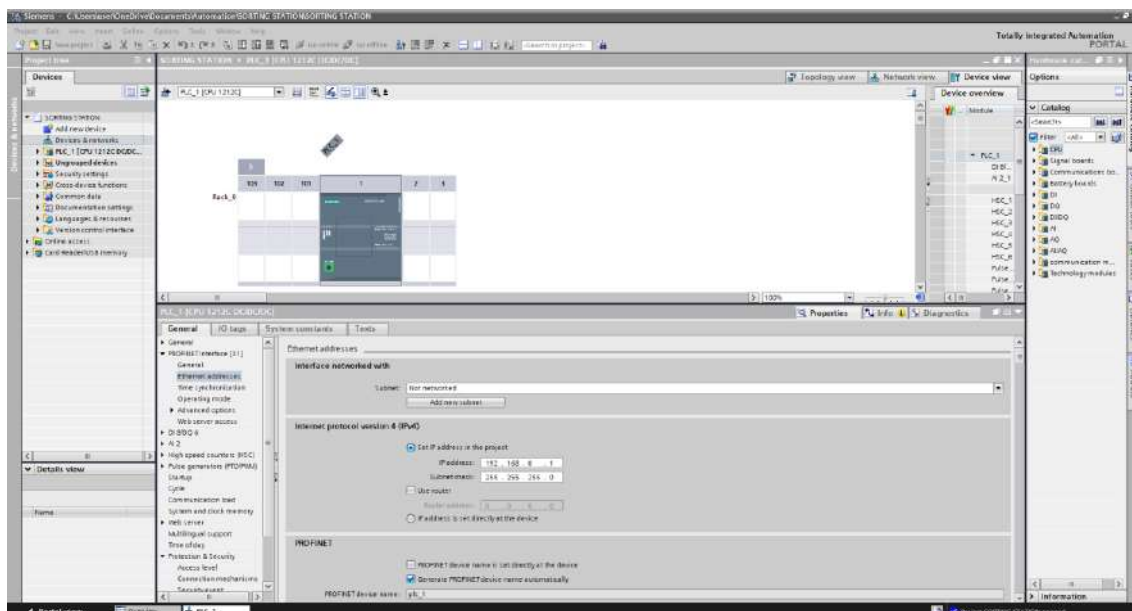
(Sumber: Dokumentasi Pribadi, 2026)

Setelah melakukan addressing pada PLC, langkah selanjutnya adalah mengonfigurasi PLC Siemens S7-1200 menggunakan *software* TIA Portal V18. Aplikasi ini digunakan untuk merancang logika program dengan pemrograman ladder diagram pada PLC Siemens S7-1200.



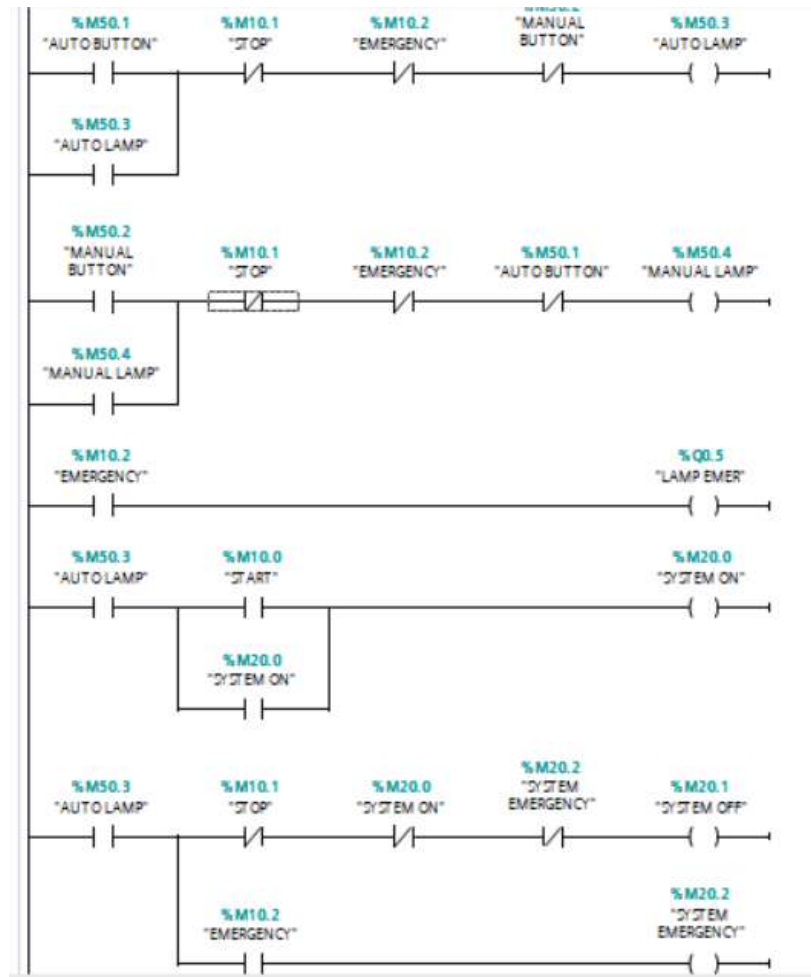
Gambar 4. 21 Konfigurasi Tipe PLC Siemens
(Sumber : Dokumentasi Pribadi, 2026)

Gambar 4.21, menunjukkan pemilihan tipe PLC Siemens yang akan digunakan dalam penelitian ini, yaitu PLC Siemens S7-1200 CPU 1212C AC/DC/Rly seri 6ES7 212-1BE40-0XB0. PLC tipe ini dilengkapi dengan 8 Digital *Input*, 2 Analog *Input*, dan 6 Digital Output.



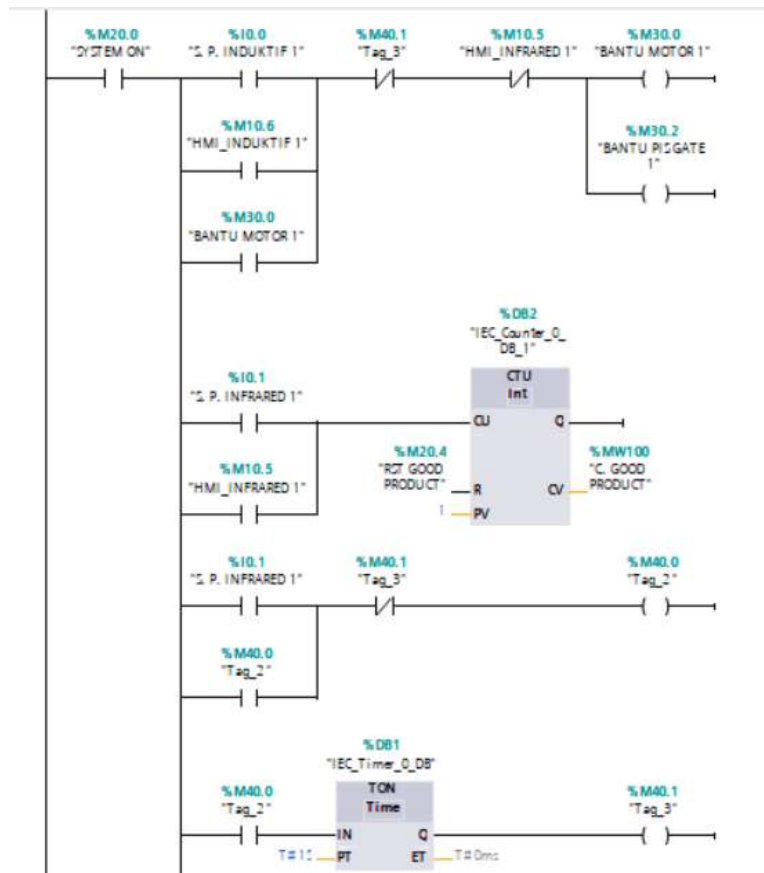
Gambar 4. 22 Konfigurasi Alamat IP PLC Siemens
(Sumber : Dokumentasi Pribadi, 2026)

Setelah konfigurasi tipe PLC Siemens yang digunakan, langkah selanjutnya adalah melakukan pengaturan komunikasi yang akan digunakan, yaitu melalui koneksi Ethernet dengan IP address yang tercantum pada Gambar 4.22.



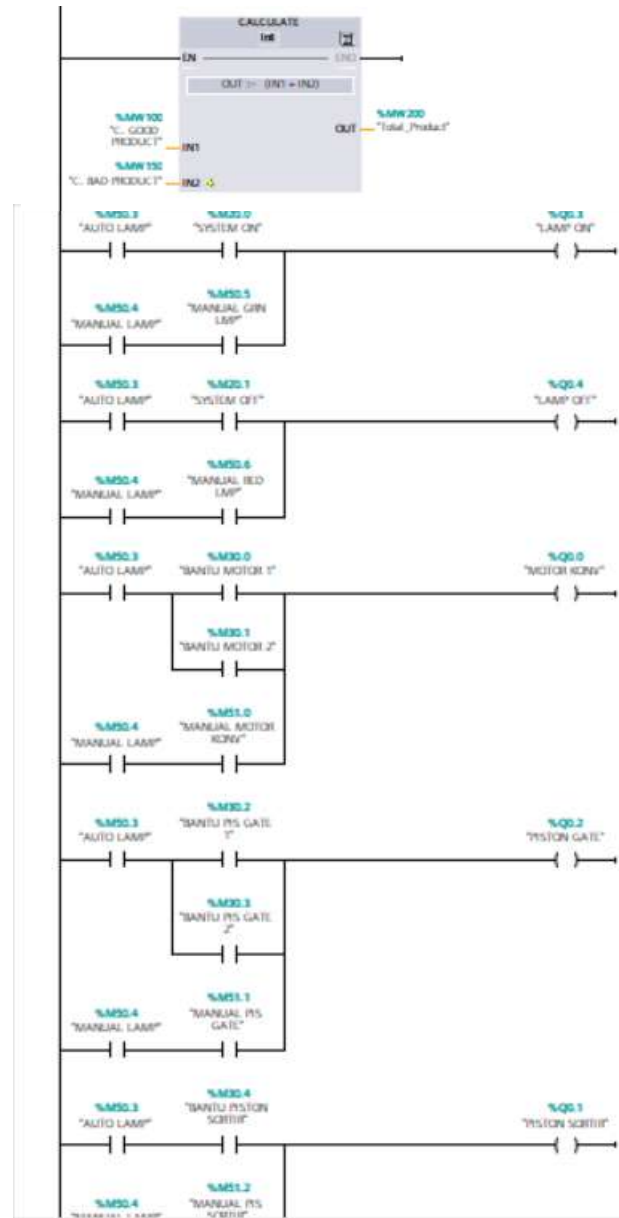
Gambar 4. 23 Progam Ladder PLC Siemens (1)
(Sumber : Dokumentasi Pribadi, 2026)

Gambar 4.23 menunjukkan logika pemrograman ladder pada PLC Siemens yang digunakan untuk mengendalikan fungsi dasar sistem, meliputi pemilihan mode operasi Auto dan Manual, serta logika Start, Stop, dan *Emergency* Stop. Program ini memastikan bahwa sistem hanya dapat beroperasi pada satu mode yang dipilih, sementara tombol Start digunakan untuk menjalankan sistem, tombol Stop untuk menghentikan proses, dan *Emergency* Stop untuk menghentikan sistem secara langsung sebagai mekanisme keselamatan



Gambar 4. 24 Program Ladder PLC Siemens (2)
(Sumber : Dokumentasi Pribadi, 2026)

Gambar 4.24 menunjukkan logika pemrograman ladder pada PLC Siemens yang digunakan untuk mengendalikan *sequence* proses pada kondisi *good product* di *Sorting Station*. Program ini mendeteksi benda kerja yang memenuhi kriteria *good product* berdasarkan pembacaan sensor induktif, kemudian mengaktifkan *motor conveyor* dan *piston gate* untuk mengarahkan benda kerja menuju jalur produk baik. Selain itu, program juga dilengkapi dengan fungsi *counter* untuk menghitung jumlah *good product* yang berhasil diproses, sehingga data jumlah produk baik dapat digunakan sebagai parameter monitoring dan perhitungan *Overall Equipment Effectiveness* (OEE).



Gambar 4. 26 Progam Ladder PLC Siemens (4)
(Sumber : Dokumentasi Pribadi, 2026)

Pada Gambar 4.26 menunjukkan logika pemrograman ladder pada PLC Siemens yang digunakan untuk menghitung *total product* serta mengendalikan fungsi *Manual Mode* dan *output* pada *Sorting Station*. Program ini menjumlahkan nilai *good product* dan *bad product* untuk memperoleh nilai *total product* yang digunakan sebagai salah satu parameter dalam perhitungan *Overall Equipment Effectiveness (OEE)*. Selain itu, program juga mengatur pengoperasian aktuator pada *Manual Mode*, sehingga operator dapat mengendalikan setiap *output*, seperti lampu indikator, motor konveyor, *sorting gate*, dan *piston sortir*, secara langsung untuk keperluan pengujian, pemeliharaan, maupun *troubleshooting* sistem.

4.5.3 Database PostgreSQL

Pada tahap ini dilakukan pengujian terhadap database PostgreSQL yang digunakan sebagai media penyimpanan data pada sistem SCADA. Database ini berfungsi untuk menyimpan data yang dikirimkan oleh backend sehingga dapat ditampilkan kembali pada berbagai halaman Web-SCADA. Dalam implementasinya, database PostgreSQL terdiri dari empat tabel utama, yaitu *users*, *data_mps*, *alarm_history*, dan *production_logs*.

Tabel *users* digunakan untuk menyimpan data pengguna yang meliputi *username*, *password*, dan *role* untuk keperluan autentikasi saat login ke sistem. Tabel *data_mps* digunakan untuk menyimpan data *Overall Equipment Effectiveness* (OEE) pada MPS yang selanjutnya ditampilkan pada halaman *OEE Dashboard* SCADA. Tabel *alarm_history* berfungsi untuk menyimpan riwayat alarm yang terjadi selama sistem beroperasi dan ditampilkan pada halaman *Alarm Dashboard*. Sementara itu, tabel *production_logs* digunakan untuk menyimpan data aktivitas produksi yang kemudian ditampilkan pada halaman *Data Logger Dashboard*. Hasil pengujian menunjukkan bahwa seluruh tabel dapat menyimpan dan menampilkan data sesuai dengan fungsi masing-masing sehingga integrasi antara backend, database, dan sistem WEB-SCADA dapat berjalan dengan baik.



	id [PK] integer	username character varying (50)	password character varying (255)	role character varying (20)
1	1	operator1	123	operator
2	2	tech1	123	maintenance
3	3	spv1	123	supervisor
4	4	boss1	123	manager
5	5	operator2	321	operator2

Gambar 4. 27 Tabel Users pada Database
(Sumber : Dokumentasi Pribadi, 2026)

Berdasarkan Gambar 4.27 Tabel *users* digunakan untuk menyimpan data pengguna yang memiliki akses ke sistem, khususnya pada fitur login dan manajemen pengguna. Tabel ini terdiri dari beberapa field seperti *id* sebagai primary key, *username* sebagai identitas pengguna, *password* sebagai autentikasi, serta *role* yang menentukan hak akses masing-masing pengguna, seperti operator, maintenance, supervisor, dan manager. Dengan adanya pembagian role ini, sistem dapat membatasi akses dan fitur yang dapat digunakan oleh setiap pengguna sesuai dengan tanggung jawabnya.

Data Output Messages Notifications													
Showing rows: 1 to 73 Page No: 1 of 1													
	l_time real	downtime double precision	cycle_time real	total_product integer	good_product integer	bad_product integer	availability real	performance real	quality real	oee real	created_at timestamp without time zone	session_finished boolean	
46	25.04	0	0.31	81	74	7	41.73	100	91.36	88.13	2026-06-02 11:59:52.540952	false	
47	50.04	0	0.31	105	98	7	50.07	100	93.33	46.73	2026-06-02 12:04:52.634695	false	
48	35.04	0	0.31	117	110	7	58.4	100	94.02	54.91	2026-06-02 12:09:52.667858	false	
49	40.05	0	0.31	133	126	7	66.75	100	94.74	63.24	2026-06-02 12:14:52.873296	false	
50	45.06	0	0.31	142	135	7	75.1	97.69	95.07	69.75	2026-06-02 12:19:53.524198	false	
51	50.06	0	0.31	142	135	7	83.43	87.93	95.07	69.75	2026-06-02 12:24:53.526088	false	
52	55.07	0	0.31	142	135	7	91.78	79.93	95.07	69.75	2026-06-02 12:29:54.436609	false	
53	56.98	3.02	0.31	142	135	7	94.97	77.26	95.07	69.75	2026-06-02 12:34:50.349599	true	

Gambar 4. 28 Tabel Data OEE MPS pada Database
(Sumber : Dokumentasi Pribadi, 2026)

Berdasarkan Gambar 4.28. Tabel data_mps digunakan sebagai media penyimpanan utama untuk data hasil *Monitoring* dan analisis performa pada sistem *Modular Production System* (MPS). Tabel ini menyimpan parameter seperti *loading_time*, *operating_time*, *downtime*, dan *cycle_time* yang merepresentasikan kondisi operasi sistem, serta *total_product*, *good_product*, dan *bad_product* untuk mencatat hasil distribusi. Data-data tersebut kemudian digunakan untuk perhitungan nilai OEE yang meliputi *Availability*, *Performance*, dan *Quality*, sehingga tabel ini berperan penting dalam mendukung proses monitoring, analisis kinerja, serta integrasi sistem secara keseluruhan.

Data Output Messages Notifications					
	id [PK] integer	alarm_code character varying (20)	alarm_message text	status character varying (20)	timestamp timestamp without time zone
1	1	ALM001	Emergency Stop Aktif	ACTIVE	2026-06-01 21:02:34.353075
2	2	ALM002	Piston Gate Fault	ACTIVE	2026-06-01 21:08:25.099762
3	3	ALM001	Emergency Stop Aktif	CLEARED	2026-06-01 21:09:35.11885
4	4	ALM002	Silinder Stack Magazine Gagal Maju	FAULT	2026-06-01 21:17:53.454212
5	5	ALM001	Emergency Stop Aktif	ACTIVE	2026-06-02 12:31:50.080744
6	6	ALM001	Emergency Stop Aktif	ACTIVE	2026-06-02 14:31:01.568856
7	7	ALM001	Emergency Stop Aktif	ACTIVE	2026-06-02 14:34:13.331319
8	8	ALM001	Emergency Stop Aktif	CLEARED	2026-06-02 14:34:28.347418
9	9	ALM002	Silinder Stack Magazine Gagal Maju	FAULT	2026-06-06 00:07:46.062402
10	10	ALM003	Silinder Mundur / Rotary Kiri Gagal Beroper...	FAULT	2026-06-06 00:08:15.561774
11	11	ALM002	Silinder Stack Magazine Gagal Maju	CLEARED	2026-06-06 00:08:53.411172

Gambar 4. 29 Tabel Data Alarm pada Database
(Sumber : Dokumentasi Pribadi, 2026)

Berdasarkan Gambar 4.29, tabel data Alarm digunakan sebagai media penyimpanan data riwayat alarm yang terjadi pada sistem *Modular Production System* (MPS). Tabel ini menyimpan informasi berupa *alarm_code* sebagai kode identifikasi alarm, *alarm_message* sebagai deskripsi kondisi atau gangguan yang terjadi, *status* yang menunjukkan kondisi alarm seperti *ACTIVE*, *FAULT*, atau *CLEARED*, serta *timestamp* yang mencatat waktu kejadian alarm. Data yang tersimpan pada tabel ini kemudian ditampilkan pada halaman Alarm *Dashboard* Web-SCADA sehingga operator dapat memantau, menelusuri, dan menganalisis setiap kejadian alarm yang terjadi selama

sistem beroperasi. Dengan demikian, tabel `alarm_history` berperan penting dalam mendukung proses monitoring, troubleshooting, dan evaluasi kinerja sistem secara historis.

Data Output Messages Notifications											
Showing rows: 1 to 48											
	id [PK] integer	timestamp timestamp without time zone	machine_status character varying (20)	good_product integer	bad_product integer	total_product integer	availability double precision	performance double precision	quality double precision	oee double precision	
36	36	2026-06-02 12:53:04.91011	RUNNING	135	7	142	0	0	95.07	0	
37	37	2026-06-02 12:58:05.768296	RUNNING	135	7	142	0	0	95.07	0	
38	38	2026-06-02 13:36:59.142943	RUNNING	10	0	10	8.33	62	100	5.17	
39	39	2026-06-02 13:41:59.490456	RUNNING	22	0	22	16.68	68.13	100	11.37	
40	40	2026-06-02 13:47:00.236019	RUNNING	40	0	40	25.03	82.56	100	20.67	
41	41	2026-06-02 13:52:01.190106	RUNNING	55	0	55	33.4	85.08	100	28.42	
42	42	2026-06-02 13:57:02.162985	RUNNING	71	0	71	41.75	87.86	100	36.68	
43	43	2026-06-02 14:02:03.090151	RUNNING	87	0	87	50.12	89.69	100	44.95	
44	44	2026-06-02 14:07:04.0514	RUNNING	102	0	102	58.47	90.14	100	52.7	
45	45	2026-06-02 14:12:04.593878	RUNNING	116	0	116	66.82	89.7	100	59.93	
46	46	2026-06-02 14:17:05.353816	RUNNING	132	0	132	75.18	90.71	100	68.2	
47	47	2026-06-02 14:22:05.406345	RUNNING	147	0	147	83.52	90.94	100	75.95	
48	48	2026-06-02 14:27:05.433559	RUNNING	163	0	163	91.85	91.69	100	84.22	

Gambar 4. 30 Tabel Data Logger pada Database
(Sumber : Dokumentasi Pribadi, 2026)

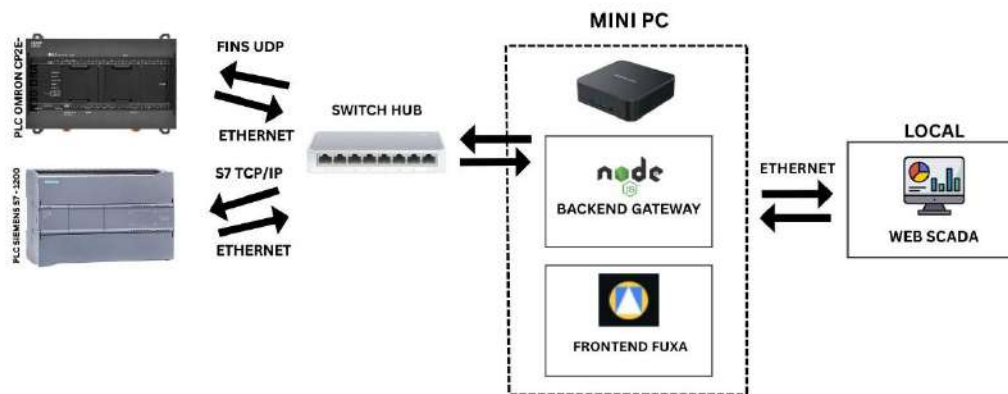
Berdasarkan Gambar 4.30, tabel data logger digunakan sebagai media penyimpanan data historis proses produksi yang ditampilkan pada halaman Data Logger *Dashboard* Web-SCADA. Tabel ini menyimpan informasi berupa timestamp sebagai waktu pencatatan data, `machine_status` yang menunjukkan kondisi mesin saat proses berlangsung, jumlah `good_product`, `bad_product`, dan `total_product` sebagai hasil produksi, serta nilai *Availability*, *Performance*, *Quality*, dan *OEE* yang diperoleh dari hasil perhitungan kinerja sistem. Data yang tersimpan pada tabel ini memungkinkan pengguna untuk melakukan pemantauan dan penelusuran riwayat produksi secara berkala, sehingga dapat digunakan sebagai dasar dalam evaluasi performa sistem, analisis produktivitas, serta pengambilan keputusan untuk peningkatan efektivitas proses produksi.

4.5.4 Web SCADA (Local)

Pada tahap ini dijelaskan implementasi Web-SCADA (*Local*) yang dibangun menggunakan platform FUXA SCADA sebagai antarmuka *Monitoring* dan kontrol pada sistem *Modular Production System* (MPS). Web-SCADA ini digunakan untuk memantau kondisi sistem secara *real-time* serta melakukan pengendalian terhadap aktuator yang terdapat pada *Distribution Station* dan *Sorting station*. Melalui tampilan berbasis web.

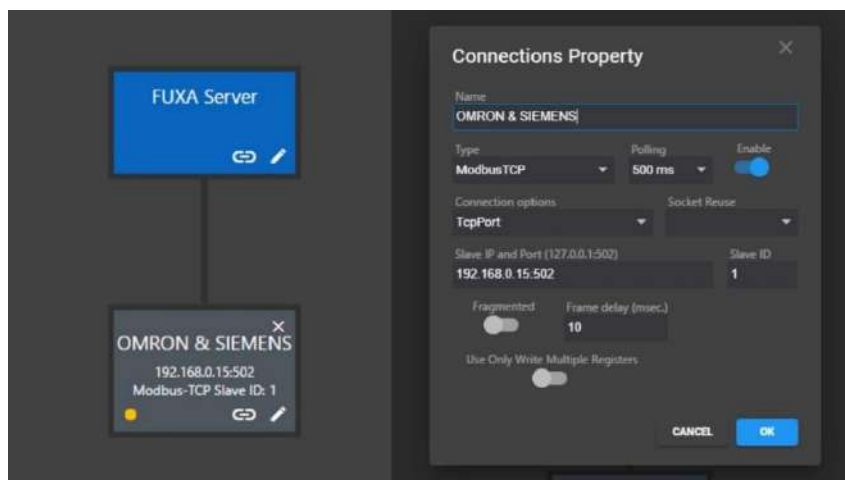
Sistem Web-SCADA dijalankan pada jaringan lokal (LAN) dan terhubung dengan PLC melalui *gateway* yang telah dikonfigurasi pada FUXA. Data yang berasal dari sensor, aktuator, dan variabel proses lainnya dibaca oleh PLC kemudian ditampilkan pada *dashboard* Web-SCADA secara *real-time*. Selain digunakan untuk monitoring,

dashboard ini juga menyediakan fitur kontrol manual yang dapat digunakan untuk mengoperasikan aktuator sesuai kebutuhan pengujian maupun operasional sistem.



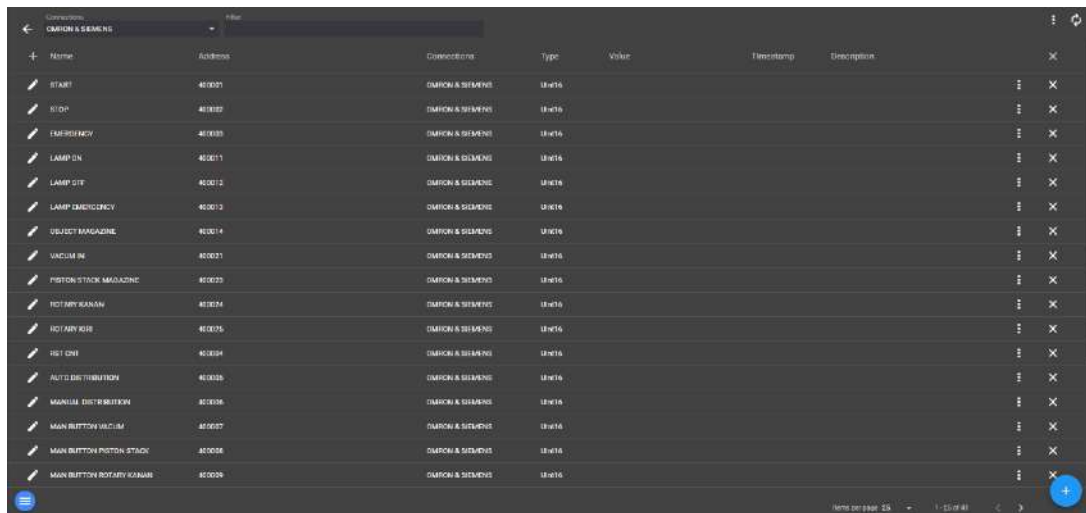
Gambar 4. 31 Diagram Arsitektur Web-SCADA (*Local*) Berbasis FUXA
(Sumber : Dokumentasi Pribadi, 2026)

Gambar 4.31 menunjukkan arsitektur sistem *Web-SCADA* lokal yang digunakan pada penelitian ini. Sistem terdiri dari dua PLC, yaitu PLC Omron dan PLC Siemens, yang terhubung ke *Mini PC* melalui jaringan Ethernet menggunakan *switch hub*. PLC Omron berkomunikasi dengan *backend gateway* menggunakan protokol FINS UDP, sedangkan PLC Siemens menggunakan protokol S7 TCP/IP. Selanjutnya, *backend gateway* yang dibangun menggunakan Node.js berfungsi sebagai penghubung dan penerjemah komunikasi antarprotokol, sehingga data dari kedua PLC dapat diintegrasikan dan diteruskan ke *Frontend FUXA*. Melalui jaringan Ethernet, pengguna dapat mengakses *Web-SCADA* lokal untuk melakukan pemantauan kondisi sistem secara *real-time* serta mengendalikan proses pada *Modular Production System (MPS)*.



Gambar 4. 32 Koneksi dan Konfigurasi pada FUXA
(Sumber : Dokumentasi Pribadi, 2026)

Gambar 4.32 menunjukkan konfigurasi koneksi antara FUXA dan *gateway* yang digunakan pada sistem. Komunikasi data menggunakan protokol Modbus TCP melalui alamat IP 192.168.0.15 dengan port 502. Konfigurasi ini digunakan agar FUXA dapat mengakses data yang terdapat pada *gateway* sebagai penghubung komunikasi antara PLC Omron dan PLC Siemens. Data yang berhasil dibaca kemudian ditampilkan pada halaman Web-SCADA untuk keperluan *Monitoring* dan kontrol sistem.



Name	Address	Connections	Type	Value	Timestamp	Description
START	400001	OMRON & SIEMENS	Unit6			
STOP	400002	OMRON & SIEMENS	Unit6			
EMERGENCY	400003	OMRON & SIEMENS	Unit6			
LAMP ON	400011	OMRON & SIEMENS	Unit6			
LAMP OFF	400012	OMRON & SIEMENS	Unit6			
LAMP EMERGENCY	400013	OMRON & SIEMENS	Unit6			
OBJECT MAGAZINE	400014	OMRON & SIEMENS	Unit6			
UNCLIM IN	400021	OMRON & SIEMENS	Unit6			
PISTON STACK MAGAZINE	400023	OMRON & SIEMENS	Unit6			
ROTARY KANAN	400024	OMRON & SIEMENS	Unit6			
ROTARY KIRI	400025	OMRON & SIEMENS	Unit6			
ROT LEM	400024	OMRON & SIEMENS	Unit6			
AUTO DISTRIBUTION	400026	OMRON & SIEMENS	Unit6			
MANUAL DISTRIBUTION	400026	OMRON & SIEMENS	Unit6			
MAIN BUTTON VALCLIM	400027	OMRON & SIEMENS	Unit6			
MAIN BUTTON PISTON STACK	400028	OMRON & SIEMENS	Unit6			
MAIN BUTTON ROTARY KANAN	400029	OMRON & SIEMENS	Unit6			

Gambar 4. 33 *Mapping Address FUXA*
(Sumber : Dokumentasi Pribadi, 2026)

Gambar 4.33 menunjukkan konfigurasi mapping address pada FUXA yang digunakan untuk menghubungkan data dari *gateway* dengan objek pada halaman Web-SCADA. Setiap tag diberikan nama sesuai dengan fungsi perangkat atau variabel proses yang digunakan pada sistem, kemudian dipetakan ke alamat register Modbus TCP yang telah disediakan oleh *gateway*. Alamat-alamat tersebut digunakan oleh FUXA untuk membaca maupun menulis data yang berasal dari PLC Omron dan PLC Siemens melalui *gateway*. Dengan konfigurasi ini, data *Monitoring* dan kontrol dapat ditampilkan serta diakses pada *dashboard* Web-SCADA sesuai dengan fungsi masing-masing perangkat yang digunakan pada sistem.

```

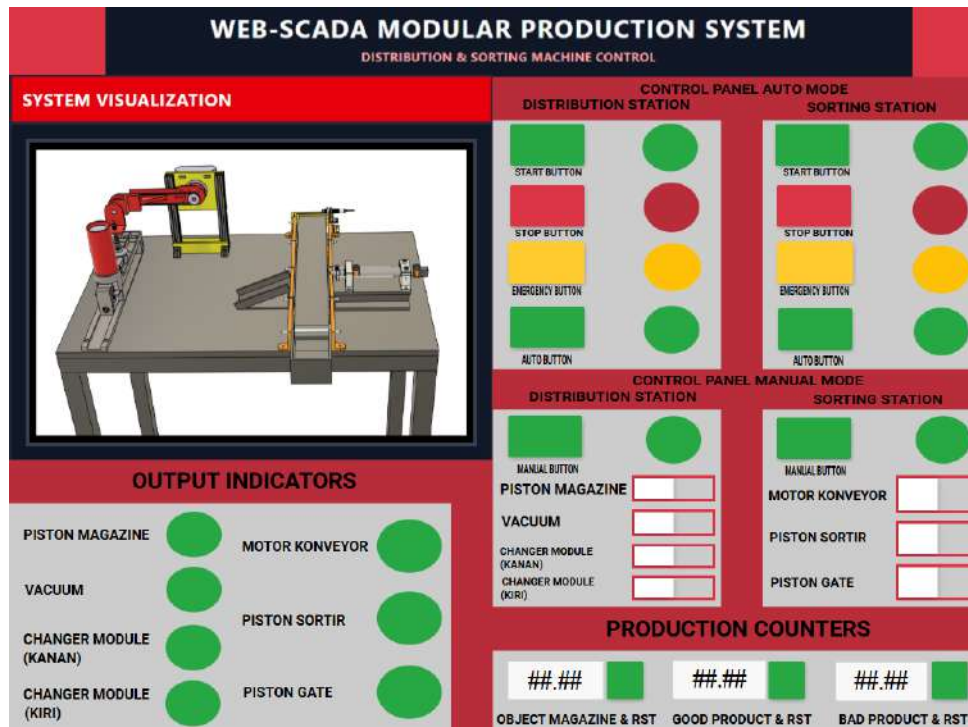
gateway-fuxa.js: X
gateway-fuxa.js: 233
234 // ----- MODBUS SERVER -----
235 const server = new ModbusRTU.ServerTCP({
236
237 // ----- READ -----
238 getHoldingRegisters: function(addr, unitID, callback) {
239
240 // ----- OMRON -----
241 if (addr === 10) return callback(null, getBit(cache.cio101, 0));
242 else if (addr === 11) return callback(null, getBit(cache.cio101, 1));
243 else if (addr === 12) return callback(null, getBit(cache.cio101, 2));
244 else if (addr === 13) return callback(null, cache.d100);
245
246 else if (addr === 20) return callback(null, getBit(cache.cio100, 2));
247 else if (addr === 21) return callback(null, getBit(cache.cio100, 3));
248 else if (addr === 22) return callback(null, getBit(cache.cio100, 4));
249 else if (addr === 23) return callback(null, getBit(cache.cio100, 5));
250 else if (addr === 24) return callback(null, getBit(cache.cio100, 6));
251 else if (addr === 25) return callback(null, getBit(cache.w100, 5));
252 else if (addr === 26) return callback(null, getBit(cache.w100, 6));
253
254 // ----- SIEMENS READ -----
255
256 // OUTPUT
257 else if (addr === 100) return callback(null, cache.MOTOR_RUN);
258 else if (addr === 101) return callback(null, cache.PISTON_SORTER);
259 else if (addr === 102) return callback(null, cache.PISTON_GATE);
260
261 // LAMP
262 else if (addr === 103) return callback(null, cache.LAMP_ON);
263 else if (addr === 104) return callback(null, cache.LAMP_OFF);
264 else if (addr === 105) return callback(null, cache.LAMP_EMERGENCY);
265
266 // COUNTER
267 else if (addr === 110) return callback(null, cache.COUNTER_GOOD);
268 else if (addr === 111) return callback(null, cache.COUNTER_BAD);
269 else if (addr === 112) return callback(null, cache.TOTAL_PRODUCT);
270
271

```

Gambar 4. 34 Program *Gateway* Web-SCADA FUXA (*Local*)
(Sumber : Dokumentasi Pribadi, 2026)

Gambar 4.34 menunjukkan program *gateway* yang digunakan sebagai penghubung komunikasi antara PLC Omron, PLC Siemens, dan Web-SCADA FUXA. *Gateway* ini dikembangkan menggunakan JavaScript dan dijalankan pada platform Node.js. Pada implementasinya, proses akuisisi data dari PLC Omron dilakukan menggunakan protokol FINS UDP, sedangkan akuisisi data dari PLC Siemens dilakukan menggunakan protokol S7 TCP/IP. Melalui kedua protokol tersebut, *gateway* membaca data dari masing-masing PLC sesuai dengan alamat memori yang telah ditentukan.

Data yang berhasil diperoleh dari PLC Omron dan PLC Siemens kemudian diproses dan dipetakan ke dalam register Modbus TCP. Dengan demikian, *gateway* berfungsi sebagai penerjemah komunikasi (*protocol converter*) yang mengubah data dari protokol FINS dan S7 menjadi Modbus TCP. Proses ini dilakukan karena FUXA SCADA menggunakan Modbus TCP sebagai protokol komunikasi utama untuk membaca dan menulis data. Setelah data tersedia pada register Modbus TCP, FUXA dapat mengakses data tersebut untuk keperluan *Monitoring* dan kontrol sistem melalui antarmuka Web-SCADA.



Gambar 4. 35 Interface Web-SCADA (Local)
(Sumber : Dokumentasi Pribadi, 2026)

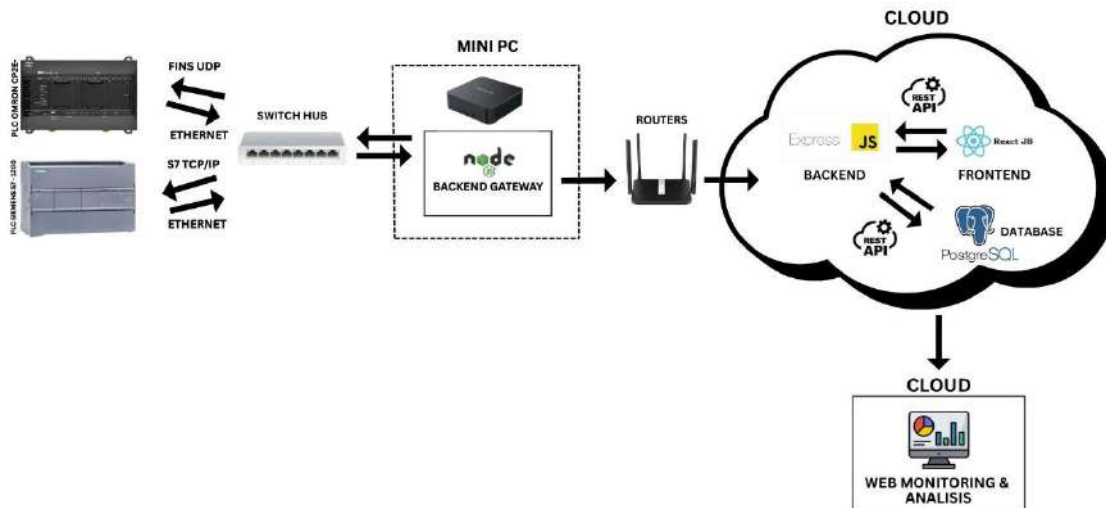
Berdasarkan Gambar 4.35. Tampilan Web-SCADA dirancang dalam bentuk *dashboard* yang terdiri dari beberapa bagian utama, yaitu visualisasi sistem, control panel, indikator output, serta counter distribusi. *Dashboard* ini memungkinkan pengguna untuk memantau kondisi sistem secara langsung sekaligus melakukan pengendalian baik dalam mode otomatis maupun manual sesuai dengan kebutuhan operasional, dengan tampilan yang terstruktur dan mudah digunakan.

4.5.5 Web SCADA (Cloud)

Pada tahap ini akan menjelaskan tentang sistem Web SCADA berbasis *Cloud* yang digunakan sebagai media *Monitoring* dan analisis sistem secara jarak jauh. Web SCADA ini dirancang untuk menampilkan data hasil proses yang telah diolah, sehingga pengguna dapat memantau kondisi sistem tanpa harus berada di lokasi secara langsung. Berbeda dengan Web SCADA lokal, sistem ini dapat diakses melalui jaringan internet, sehingga memberikan fleksibilitas dalam melakukan pengawasan terhadap sistem *Modular Production System* (MPS).

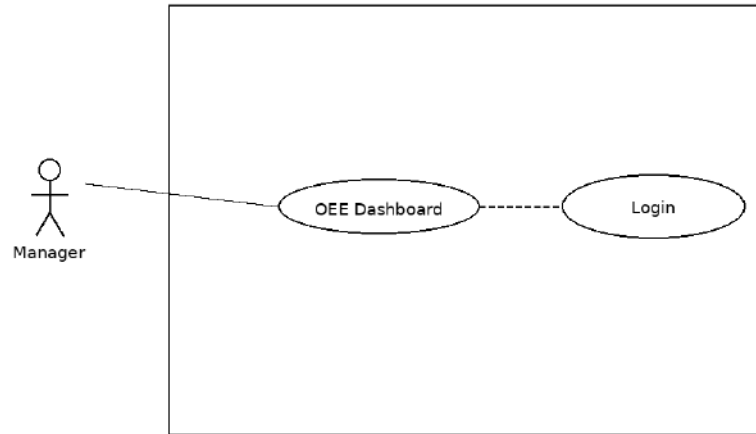
Fungsi utama dari Web SCADA (*Cloud*) adalah untuk menyajikan informasi *Monitoring* dan analisis yang meliputi empat aspek utama, yaitu OEE *dashboard*, *Monitoring* mesin, alarm system, dan data logger. Keempat fitur tersebut memungkinkan pengguna untuk mengetahui kondisi operasional sistem, performa distribusi, kejadian

gangguan, serta riwayat data secara terintegrasi dalam satu tampilan berbasis web. Dengan adanya sistem ini, proses pemantauan menjadi lebih mudah dilakukan dari berbagai lokasi sesuai dengan kebutuhan pengguna.



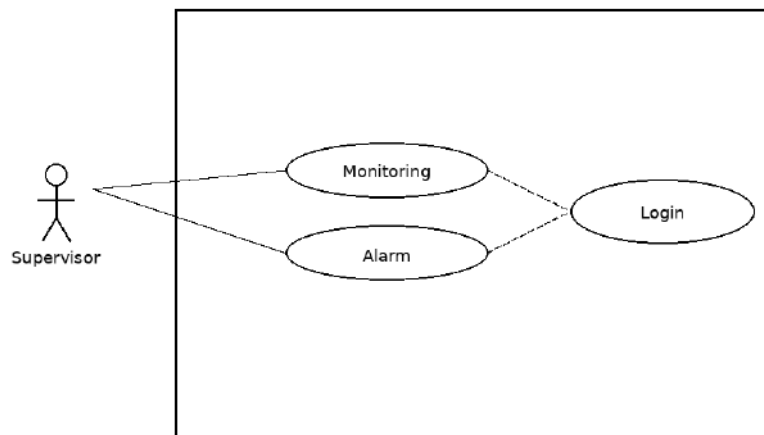
Gambar 4. 36 Diagram Arsitektur Web-SCADA (Cloud)
(Sumber : Dokumentasi Pribadi, 2026)

Gambar 4.36 menunjukkan arsitektur sistem *Web-SCADA* berbasis *cloud* yang digunakan pada penelitian ini. Sistem diawali dengan proses akuisisi data dari PLC Omron menggunakan protokol FINS UDP dan PLC Siemens menggunakan protokol S7 TCP/IP melalui jaringan Ethernet menuju *backend gateway* yang berjalan pada *Mini PC*. Selanjutnya, *backend gateway* mengirimkan data ke *backend* berbasis Express.js melalui jaringan internet menggunakan *REST API*. Data yang diterima kemudian disimpan pada basis data PostgreSQL dan diakses oleh *Frontend* yang dibangun menggunakan React.js untuk menampilkan informasi monitoring dan analisis. Dengan arsitektur ini, pengguna dapat mengakses sistem *Web Monitoring & Analysis* secara *online* untuk memantau kondisi sistem, melihat data historis, serta melakukan analisis *Overall Equipment Effectiveness (OEE)* dari berbagai lokasi yang terhubung ke internet.



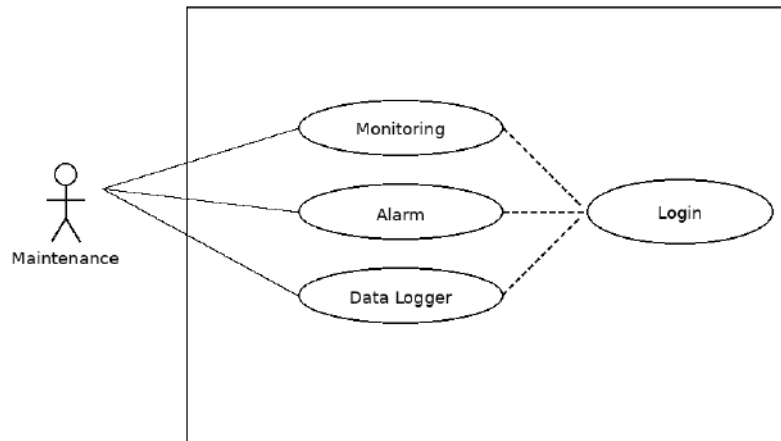
Gambar 4. 37 Use Case Diagram peran Manager
(Sumber: Dokumentasi Pribadi, 2026)

Gambar 4.37 menunjukkan *use case diagram* untuk peran *Manager*. Pengguna dengan peran ini harus melakukan proses *login* terlebih dahulu sebelum dapat mengakses halaman *OEE Dashboard* untuk memantau nilai *Overall Equipment Effectiveness (OEE)* sebagai bahan evaluasi kinerja sistem.



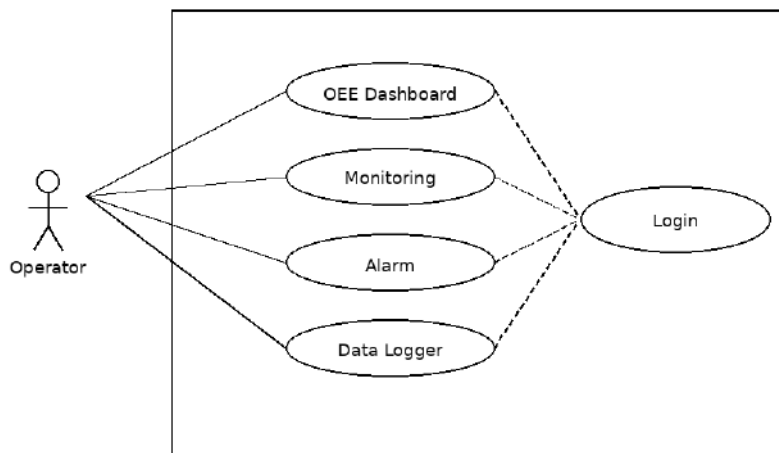
Gambar 4. 38 Use Case Diagram peran Supervisor
(Sumber: Dokumentasi Pribadi, 2026)

Gambar 4.38 menunjukkan *use case diagram* untuk peran *Supervisor*. Setelah berhasil melakukan *login*, *Supervisor* dapat mengakses halaman *Monitoring* untuk memantau kondisi sistem secara *real-time* serta halaman *Alarm* untuk melihat informasi gangguan yang terjadi pada sistem.



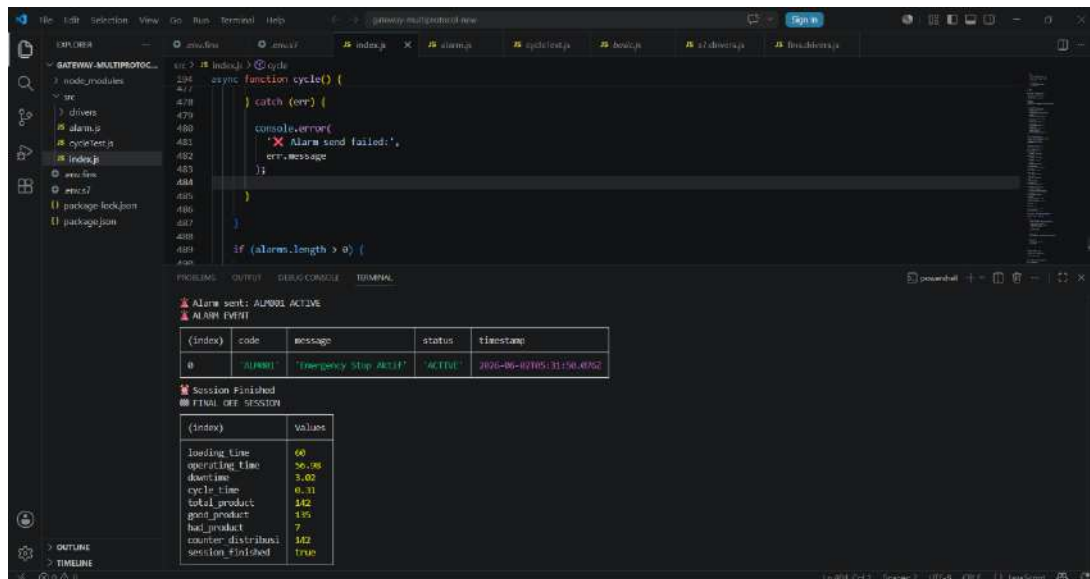
Gambar 4. 39 Use Case Diagram peran Maintenance
(Sumber: Dokumentasi Pribadi, 2026)

Gambar 4.39 menunjukkan *use case diagram* untuk peran *Maintenance*. Setelah melakukan *login*, pengguna dapat mengakses halaman *Monitoring*, *Alarm*, dan *Data Logger* untuk memantau kondisi sistem, melihat riwayat alarm, serta menganalisis data historis sebagai pendukung kegiatan pemeliharaan dan *troubleshooting*



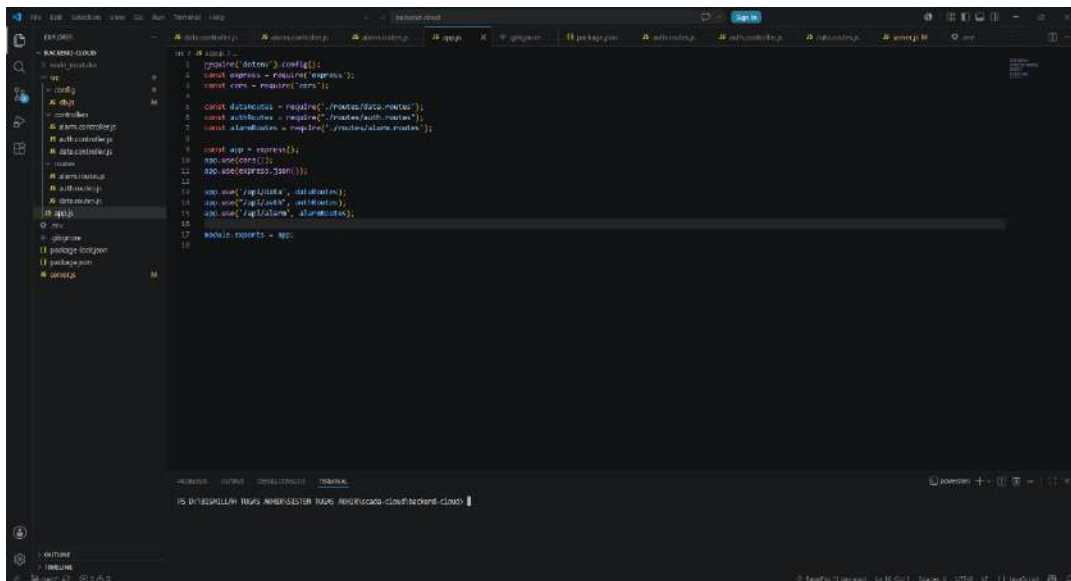
Gambar 4. 40 Use Case Diagram peran Operator
(Sumber: Dokumentasi Pribadi, 2026)

Gambar 4.40 menunjukkan *use case diagram* untuk peran *Operator*. Setelah berhasil melakukan *login*, pengguna dapat mengakses seluruh fitur utama sistem, meliputi *OEE Dashboard*, *Monitoring*, *Alarm*, dan *Data Logger*, sehingga proses monitoring, analisis kinerja mesin, serta pengelolaan data produksi dapat dilakukan melalui satu antarmuka *Web Monitoring & Analysis*.



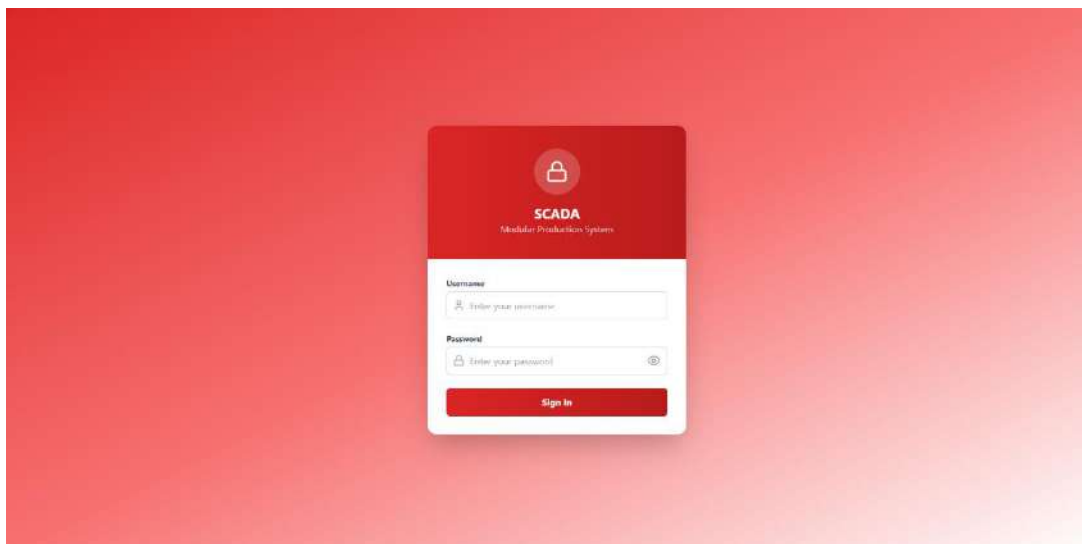
Gambar 4. 41 Progam *Gateway Web-SCADA (Cloud)*
(Sumber : Dokumentasi Pribadi, 2026)

Berdasarkan Gambar 4.41, program *gateway* yang digunakan pada sistem berhasil menghubungkan PLC Omron dan PLC Siemens dengan sistem *Web Monitoring* dan Analisis berbasis *cloud*. *Gateway* dikembangkan menggunakan Node.js dan dijalankan pada *mini PC* untuk membaca data dari PLC Omron melalui protokol FINS UDP serta PLC Siemens melalui protokol S7 TCP/IP. Data yang berhasil dibaca kemudian diproses untuk memperoleh informasi monitoring, data alarm, dan parameter OEE sebelum dikirimkan ke *server cloud* melalui API. Hasil pengujian menunjukkan bahwa *gateway* mampu melakukan akuisisi data dari kedua PLC secara bersamaan dan mengirimkan data tersebut ke *backend* Express.js sehingga dapat disimpan pada database PostgreSQL dan ditampilkan pada halaman *Monitoring* berbasis web.



Gambar 4. 42 Program *Backend Express Web-SCADA (Cloud)*
(Sumber : Dokumentasi Pribadi, 2026)

Berdasarkan Gambar 4.42, program *backend cloud* telah berhasil diimplementasikan menggunakan Express.js. Pada pengujian yang dilakukan, *backend* mampu menerima data yang dikirimkan oleh *gateway* dan menyimpannya ke dalam database PostgreSQL. Selain itu, *backend* juga menangani pengolahan data *monitoring*, data alarm, data OEE, serta autentikasi pengguna. Data yang tersimpan pada database kemudian dapat ditampilkan pada halaman Web *Monitoring* dan Analisis sesuai dengan kebutuhan sistem.

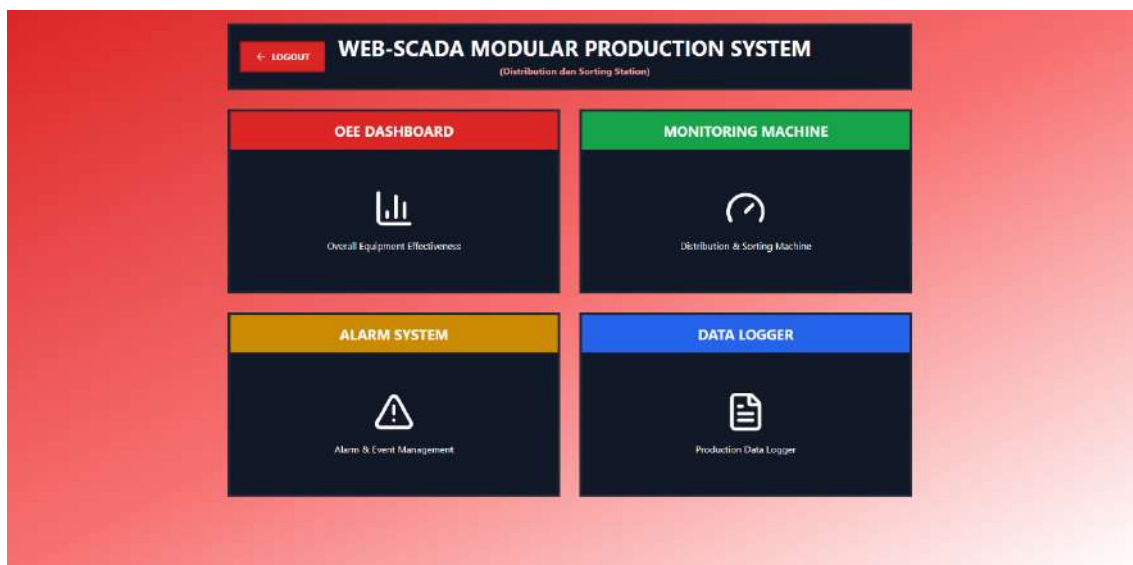


Gambar 4. 43 Interface Login Page
(Sumber : Dokumentasi Pribadi, 2026)

Berdasarkan Gambar 4.43, halaman *Login* telah berhasil diimplementasikan sebagai gerbang akses menuju sistem Web-SCADA berbasis *cloud*. Pada halaman ini,

pengguna diwajibkan memasukkan *username* dan *password* yang telah terdaftar pada database PostgreSQL sebelum dapat mengakses seluruh fitur yang tersedia pada sistem. Implementasi halaman *Login* bertujuan untuk membatasi akses hanya kepada pengguna yang memiliki hak akses sehingga keamanan data dan sistem dapat terjaga.

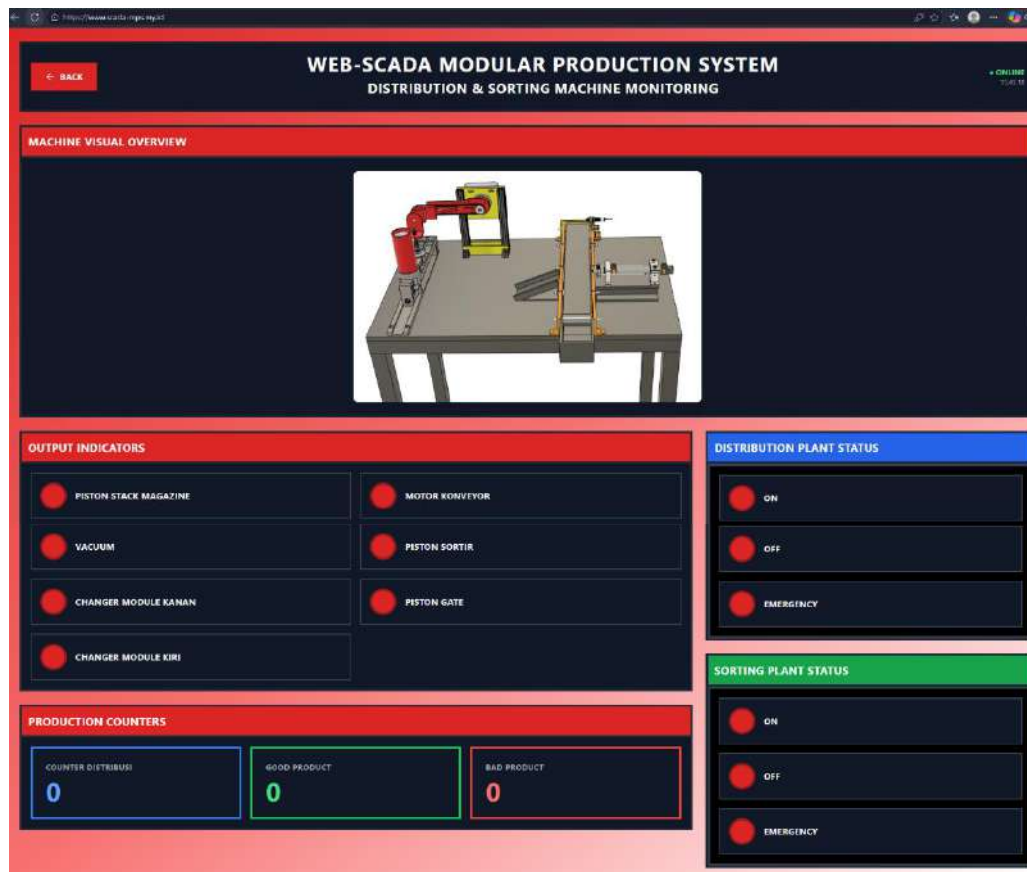
Hasil pengujian menunjukkan bahwa proses autentikasi dapat berjalan dengan baik. Pengguna yang memasukkan *username* dan *password* yang valid berhasil diarahkan ke halaman utama sistem, sedangkan pengguna yang memasukkan data yang tidak sesuai akan ditolak dan tetap berada pada halaman *Login*. Dengan demikian, mekanisme autentikasi yang diterapkan telah berfungsi sesuai dengan kebutuhan sistem.



Gambar 4. 44 Interface Main Menu
(Sumber : Dokumentasi Pribadi, 2026)

Berdasarkan Gambar 4.44, halaman *Main Menu* telah berhasil diimplementasikan sebagai halaman utama yang ditampilkan setelah pengguna berhasil melakukan proses *login*. Halaman ini berfungsi sebagai pusat navigasi untuk mengakses seluruh fitur yang tersedia pada sistem Web-SCADA. Pada halaman ini tersedia empat menu utama, yaitu *OEE Dashboard*, *Monitoring Machine*, *Alarm System*, dan *Data Logger*, yang masing-masing digunakan untuk menampilkan informasi sesuai dengan kebutuhan pengguna.

Hasil pengujian menunjukkan bahwa setiap menu dapat diakses dengan baik dan mengarahkan pengguna ke halaman yang sesuai. Selain itu, tersedia tombol *Logout* yang digunakan untuk mengakhiri sesi pengguna dan kembali ke halaman *Login*. Dengan adanya halaman *Main Menu* ini, proses navigasi antar fitur pada sistem menjadi lebih terstruktur dan memudahkan pengguna dalam mengakses fungsi monitoring, analisis OEE, pengelolaan alarm, serta pencatatan data produksi.



Gambar 4. 45 Interface *Monitoring Plant*
(Sumber : Dokumentasi Pribadi, 2026)

Berdasarkan Gambar 4.45, halaman *Monitoring* telah berhasil diimplementasikan untuk menampilkan kondisi *Distribution Station* dan *Sorting station* secara *real-time*. Pada halaman ini ditampilkan ilustrasi sistem *Modular Production System* (MPS) sebagai representasi visual dari plant yang dimonitor. Selain itu, tersedia indikator status yang menunjukkan kondisi aktuator dan perangkat utama, seperti *piston stack magazine*, *vacuum*, *changer module*, motor konveyor, piston sortir, dan piston gate. Indikator berwarna hijau menunjukkan perangkat dalam kondisi aktif, sedangkan indikator berwarna merah menunjukkan perangkat dalam kondisi tidak aktif.

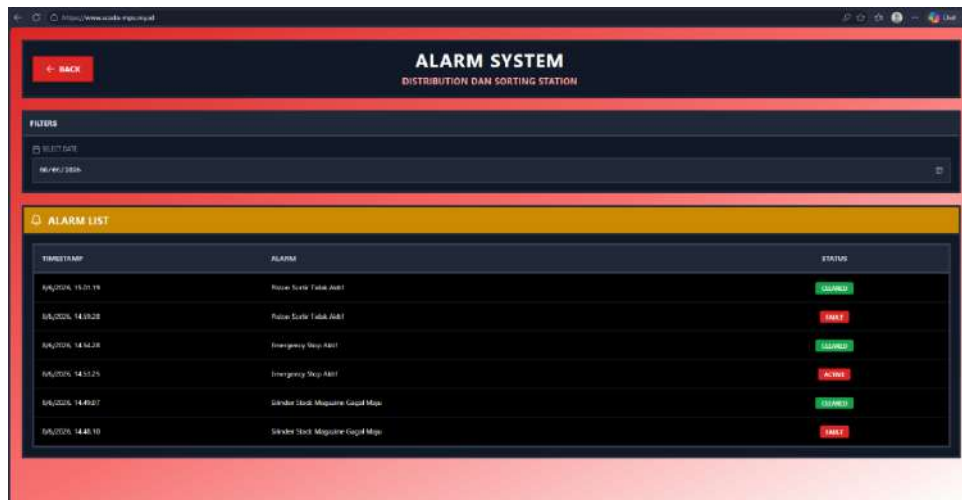
Halaman ini juga menampilkan status operasi *Distribution Station* dan *Sorting station* yang meliputi kondisi *ON*, *OFF*, dan *Emergency*. Pada bagian bawah halaman ditampilkan informasi jumlah objek yang telah diproses (*counter distribusi*), jumlah *good product*, serta jumlah *bad product*. Hasil pengujian menunjukkan bahwa perubahan status yang terjadi pada PLC dapat ditampilkan dengan baik pada halaman *Monitoring*, sehingga pengguna dapat memantau kondisi sistem produksi secara langsung melalui antarmuka Web-SCADA berbasis *cloud*.



Gambar 4. 46 Interface OEE Page
(Sumber : Dokumentasi Pribadi, 2026)

Berdasarkan Gambar 4.46, halaman *Overall Equipment Effectiveness* (OEE) berhasil diimplementasikan untuk menampilkan hasil perhitungan OEE berdasarkan data produksi yang diperoleh dari sistem. Halaman ini menampilkan nilai *Availability*, *Performance*, *Quality*, dan OEE dalam bentuk indikator visual sehingga memudahkan pengguna dalam memantau tingkat efektivitas operasi sistem. Selain itu, ditampilkan pula informasi pendukung seperti *running time*, *downtime*, jumlah *total product*, *good product*, dan *bad product* yang digunakan dalam proses perhitungan OEE.

Pada bagian bawah halaman juga ditampilkan tabel *History OEE* yang berisi riwayat hasil perhitungan OEE yang tersimpan pada database PostgreSQL. Hasil pengujian menunjukkan bahwa data yang dikirimkan oleh *gateway* dapat diolah dan ditampilkan dengan baik pada halaman ini, sehingga pengguna dapat melakukan pemantauan dan evaluasi kinerja sistem produksi berdasarkan nilai OEE yang diperoleh.



Gambar 4. 47 Interface Alarm Page
(Sumber : Dokumentasi Pribadi, 2026)

Berdasarkan Gambar 4.47, halaman *Alarm System* telah berhasil diimplementasikan untuk menampilkan informasi alarm yang terjadi pada *Distribution Station* dan *Sorting station*. Halaman ini menampilkan data alarm yang tersimpan pada database PostgreSQL dalam bentuk tabel yang berisi waktu kejadian (*timestamp*), deskripsi alarm, dan status alarm. Selain itu, tersedia fitur pemilihan tanggal yang digunakan untuk memfilter data alarm berdasarkan tanggal tertentu sehingga memudahkan pengguna dalam melakukan penelusuran riwayat alarm

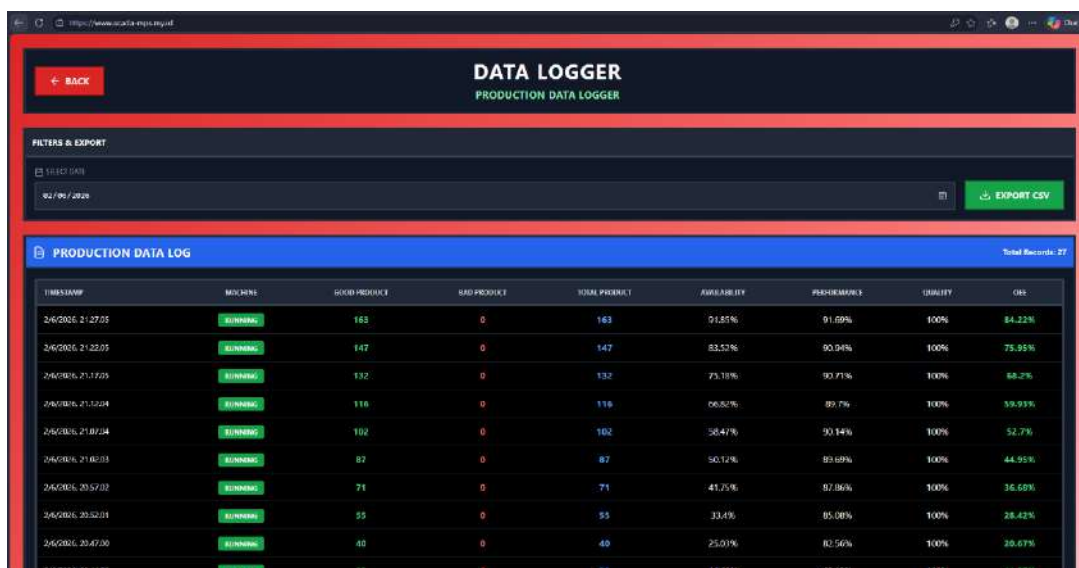
Tabel 4. 24 Kondisi Alarm

No	Deskripsi Alarm	Kondisi Aktivasi
1	<i>Emergency Stop</i> Aktif	Tombol <i>Emergency</i> pada <i>Distribution Station</i> atau <i>Sorting station</i> aktif
2	Silinder <i>Stack Magazine</i> Gagal Maju	Sensor <i>Photoelectric Fiber</i> aktif (mendeteksi objek) tetapi Sensor <i>limit switch</i> maju pada <i>Stack Magazine</i> tidak aktif selama 10 detik
3	Silinder Mundur / <i>Rotary Kiri</i> Gagal Beroperasi	Sensor <i>limit switch</i> maju aktif, tetapi Sensor <i>limit switch</i> mundur dan Sensor <i>limit switch rotary</i> kanan tidak aktif selama 10 detik
4	<i>Vacuum</i> Gagal Mengambil Objek	Sensor <i>limit switch rotary</i> kanan aktif, tetapi aktuator <i>vacuum</i> tidak aktif selama 10 detik
5	<i>Rotary Kanan</i> Tidak Bergerak	<i>Vacuum</i> aktif, tetapi aktuator <i>rotary</i> kanan tidak aktif selama 10 detik
6	Motor <i>Conveyor Sorting</i> Tidak Aktif	Sensor <i>proximity</i> induktif atau sensor <i>proximity</i> kapasitif mendeteksi objek,

No	Deskripsi Alarm	Kondisi Aktivasi
		tetapi motor <i>conveyor</i> tidak aktif selama 10 detik
7	Piston <i>Gate Sorting</i> Tidak Aktif	Sensor <i>proximity</i> induktif atau sensor <i>proximity</i> kapasitif mendeteksi objek, tetapi piston gate tidak aktif selama 10 detik
8	Piston Sortir Tidak Aktif	Sensor <i>proximity</i> kapasitif mendeteksi

(Sumber: Dokumentasi Pribadi, 2026)

. Berdasarkan Tabel 4.24, sistem alarm yang diimplementasikan terdiri dari delapan kondisi alarm yang dibangkitkan berdasarkan pembacaan sensor, *limit switch*, dan status aktuator selama sistem beroperasi. Sebagian besar alarm menggunakan *delay* selama 10 detik untuk menghindari terjadinya alarm akibat perubahan kondisi sesaat. Hasil pengujian menunjukkan bahwa alarm yang terdeteksi oleh *gateway* dapat ditampilkan dengan baik pada halaman ini, baik dalam kondisi *ACTIVE*, *FAULT*, maupun *CLEARED*, serta berhasil tersimpan ke dalam database PostgreSQL sehingga riwayat gangguan yang terjadi selama proses produksi dapat dipantau dan ditelusuri kembali melalui sistem *Web Monitoring and Analysis*.



The screenshot shows a web application titled "DATA LOGGER" with a subtitle "PRODUCTION DATA LOGGER". It features a "FILTERS & EXPORT" section with a date range selector set to "24/09/2024" and an "EXPORT CSV" button. Below this is a "PRODUCTION DATA LOG" table with 9 columns: TIME/STAMP, MACHINES, GOOD PRODUCT, BAD PRODUCT, TOTAL PRODUCT, AVAILABILITY, PERFORMANCE, QUALITY, and OEE. The table contains 10 rows of data, all with a "RUNNING" status in the MACHINES column. The OEE values range from 84.22% to 91.37%.

TIME/STAMP	MACHINES	GOOD PRODUCT	BAD PRODUCT	TOTAL PRODUCT	AVAILABILITY	PERFORMANCE	QUALITY	OEE
24/09/2024 21:27:03	RUNNING	163	0	163	91.65%	91.69%	100%	84.22%
24/09/2024 21:22:03	RUNNING	147	0	147	83.52%	90.94%	100%	75.99%
24/09/2024 21:17:03	RUNNING	132	0	132	73.18%	90.71%	100%	68.2%
24/09/2024 21:12:04	RUNNING	118	0	118	66.82%	89.7%	100%	59.93%
24/09/2024 21:07:04	RUNNING	102	0	102	58.47%	90.14%	100%	52.7%
24/09/2024 21:02:03	RUNNING	87	0	87	50.32%	89.69%	100%	44.95%
24/09/2024 20:57:03	RUNNING	71	0	71	41.75%	87.86%	100%	36.68%
24/09/2024 20:52:03	RUNNING	55	0	55	33.4%	85.08%	100%	28.42%
24/09/2024 20:47:00	RUNNING	40	0	40	25.01%	82.56%	100%	20.67%
24/09/2024 20:42:00	RUNNING	23	0	23	16.65%	69.13%	100%	11.37%

Gambar 4. 48 Interface Data Logger Page
(Sumber : Dokumentasi Pribadi, 2026)

Berdasarkan Gambar 4.48, halaman *Data Logger* telah berhasil diimplementasikan untuk menampilkan data historis hasil produksi yang tersimpan pada database PostgreSQL. Pada halaman ini, pengguna dapat melihat informasi waktu pencatatan data

(*timestamp*), status mesin, jumlah *good product*, *bad product*, dan *total product*, serta nilai *Availability*, *Performance*, *Quality*, dan OEE yang diperoleh selama proses produksi berlangsung. Selain itu, tersedia fitur pemilihan tanggal dan *export* data dalam format CSV untuk memudahkan proses pencarian dan pengolahan data hasil produksi. Hasil pengujian menunjukkan bahwa data yang dikirimkan oleh *gateway* dapat ditampilkan dengan baik pada halaman *Data Logger*, sehingga pengguna dapat melakukan pemantauan dan penelusuran riwayat produksi berdasarkan waktu pencatatan yang tersedia.

4.6 Pengujian OEE

Setelah sistem *Modular Production System* (MPS) berhasil dibangun dan diintegrasikan dengan Web-SCADA, tahap selanjutnya adalah melakukan pengujian *Overall Equipment Effectiveness* (OEE). Pengujian ini bertujuan untuk mengetahui kemampuan sistem dalam mengolah data operasional sistem dan menghitung nilai OEE secara otomatis berdasarkan kondisi operasional mesin. Data pengujian diperoleh secara langsung dari PLC Siemens S7-1200 dan PLC Omron CP2E yang telah terhubung melalui *gateway* multi-protokol, kemudian diproses oleh backend dan ditampilkan pada *dashboard* Web-SCADA.

Pengujian dilakukan dengan beberapa skenario kondisi operasional mesin untuk menghasilkan variasi nilai OEE, mulai dari kondisi efisiensi rendah hingga efisiensi tinggi. Parameter yang digunakan dalam pengujian meliputi *operating time*, *downtime*, *cycle time*, jumlah produk baik (*good product*), jumlah produk cacat (*bad product*), serta total output distribusi yang diperoleh langsung dari proses kerja sistem MPS. Setiap skenario dirancang untuk merepresentasikan kondisi operasional tertentu sehingga sistem dapat dievaluasi berdasarkan perubahan nilai *Availability*, *Performance*, dan *Quality* yang memengaruhi hasil perhitungan OEE.

4.6.1 Parameter Pengujian OEE

Pengujian *Overall Equipment Effectiveness* (OEE) dilakukan menggunakan beberapa parameter yang diperoleh secara langsung dari proses distribusi pada *Modular Production System* (MPS). Parameter tersebut digunakan untuk menghitung nilai *Availability*, *Performance*, dan *Quality* yang kemudian menghasilkan nilai OEE pada sistem Web-SCADA.

Tabel 4. 25 Parameter Data Pengujian OEE

Variabel	Sumber Data
Waktu distribusi yang direncanakan (<i>Loading Time</i>)	Data ditentukan berdasarkan durasi pengujian yang telah direncanakan.
Lama waktu mesin menjalankan proses distribusi (<i>Operating Time</i>)	Diambil berdasarkan durasi total operasi sistem <i>Distribution</i> dan <i>Sorting station</i> , dimulai sejak tombol start pada mode otomatis diaktifkan hingga sistem dihentikan melalui tombol stop atau emergency stop.
Waktu mesin dalam keadaan berhenti (<i>Downtime</i>)	Data diperoleh dari durasi ketika sistem mengalami Emergency Stop atau gangguan pada station yang terdeteksi melalui parameter alarm.
Lama waktu mesin dalam memproses satu objek (<i>Cycle Time</i>)	Data diambil dari selang waktu sejak objek terdeteksi oleh sensor awal proses hingga objek tersebut mencapai penampungan akhir.
Jumlah total produk yang diproses (<i>Total Product</i>)	Data diperoleh dari nilai <i>counter</i> produk pada <i>Sorting station</i> yang mencatat seluruh objek yang berhasil diproses sistem, baik produk yang memenuhi kriteria kualitas (<i>good product</i>) maupun produk yang ditolak (<i>bad product</i>).
Jumlah produk baik (<i>Good Product</i>)	Data diperoleh dari hasil pembacaan sensor <i>proximity</i> induktif yang mengidentifikasi workpiece standar sebagai produk yang memenuhi spesifikasi pada skenario pengujian, kemudian dicatat sebagai <i>Good Product</i> oleh PLC.
Jumlah produk cacat (<i>Bad Product</i>)	Data diperoleh dari hasil pembacaan sensor <i>proximity</i> kapasitif yang mengidentifikasi workpiece yang telah diberi <i>Reject Marker</i> sebagai representasi produk yang tidak memenuhi spesifikasi kualitas (<i>reject</i>) pada skenario pengujian, kemudian dicatat sebagai <i>Bad Product</i> oleh PLC.

(Sumber: Dokumentasi Pribadi, 2026)

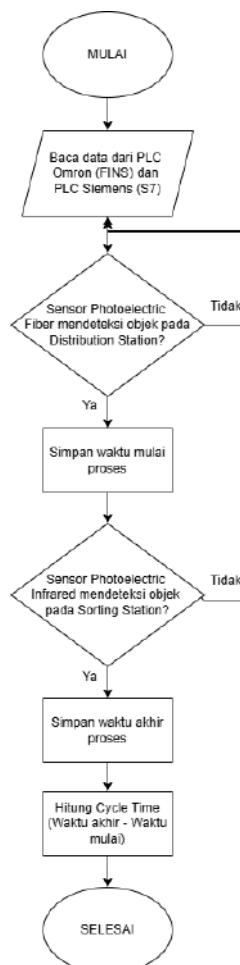
Berdasarkan parameter pengujian pada Tabel 4.25, data yang diperoleh selanjutnya digunakan dalam proses perhitungan *Overall Equipment Effectiveness* (OEE). Perhitungan dilakukan dengan menggunakan tiga komponen utama, yaitu

Availability (A), *Performance (P)*, dan *Quality (Q)*. Masing-masing komponen dihitung berdasarkan data operasional sistem yang diperoleh selama proses distribusi berlangsung.

4.6.2 Pengujian *Cycle Time*

Pengujian *Cycle Time* dilakukan untuk mengetahui waktu yang dibutuhkan sistem dalam menyelesaikan satu siklus proses distribusi pada *Modular Production System (MPS)*. Pengukuran dilakukan berdasarkan selang waktu sejak sensor *photoelectric optic* pada *Distribution Station* mendeteksi objek hingga sensor *photoelectric infrared* pada *Sorting station* mendeteksi objek pada akhir proses distribusi.

Pada pengujian ini digunakan dua metode pengukuran, yaitu pengukuran secara manual menggunakan *stopwatch* dan pengukuran otomatis menggunakan sistem *gateway* yang telah dirancang. Masing-masing metode dilakukan sebanyak 10 kali pengujian dengan parameter pengukuran yang sama. Pengujian ini bertujuan untuk mengetahui kemampuan sistem *gateway* dalam melakukan pencatatan *Cycle Time* secara otomatis pada proses distribusi.



Gambar 4. 49 Flowchart Pengujian *Cycle Time*
(Sumber : Dokumentasi Pribadi, 2026)

Gambar 4.49 menunjukkan *flowchart* pengujian *Cycle Time* pada plant *Modular Production System (MPS)*. Proses diawali dengan pembacaan data dari PLC Omron menggunakan protokol FINS UDP dan PLC Siemens menggunakan protokol S7 TCP/IP. Selanjutnya, sistem memeriksa apakah sensor *photoelectric fiber* pada *Distribution Station* mendeteksi keberadaan benda kerja. Jika objek terdeteksi, sistem akan menyimpan waktu awal proses. Setelah itu, sistem menunggu hingga sensor *photoelectric infrared* pada *Sorting Station* mendeteksi objek yang sama sebagai penanda berakhirnya proses distribusi. Ketika objek berhasil terdeteksi, sistem menyimpan waktu akhir proses, kemudian menghitung nilai *Cycle Time* berdasarkan selisih antara waktu akhir dan waktu awal. Hasil perhitungan *Cycle Time* selanjutnya digunakan sebagai salah satu parameter dalam perhitungan *Overall Equipment Effectiveness (OEE)*.

Tabel 4. 26 Pengujian *Cycle Time* Menggunakan Stopwatch

Jumlah Pengujian	<i>Cycle Time</i> (s)
1	19.59
2	19.38
3	19.23
4	19.63
5	19.38
6	18.93
7	19.53
8	19.84
9	19.16
10	18.71

Tabel 4. 27 Pengujian *Cycle Time* Menggunakan Gateway

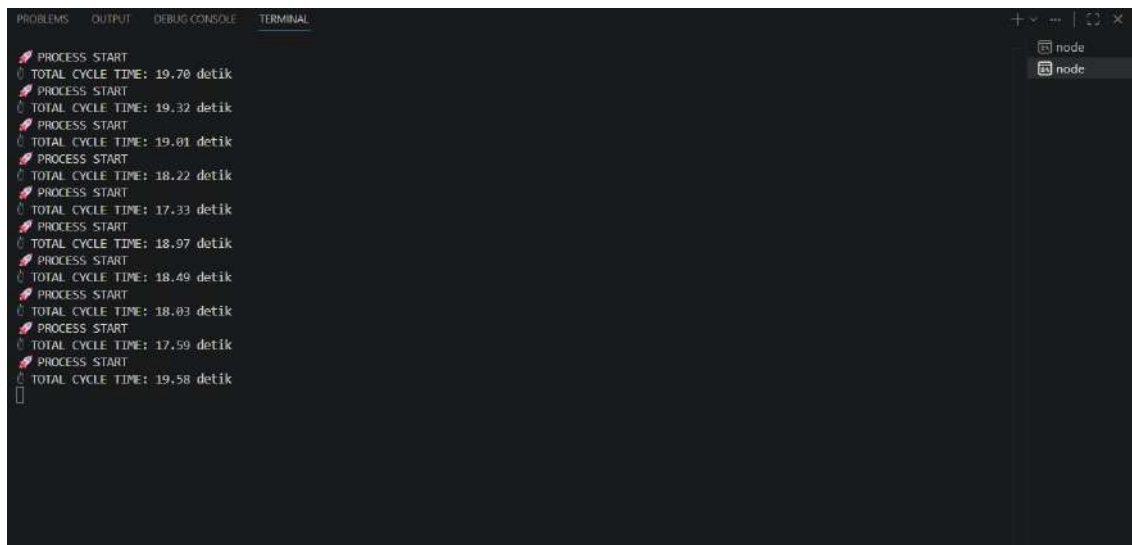
Tabel 4.26 menunjukkan hasil pengujian *Cycle Time* menggunakan *stopwatch* sebagai metode pengukuran manual. Pengujian dilakukan sebanyak sepuluh kali untuk memperoleh nilai *Cycle Time* yang menjadi acuan dalam mengevaluasi hasil pengukuran yang dilakukan oleh sistem.

Jumlah Pengujian	<i>Cycle Time</i> (s)
1	19.70
2	19.32
3	19.01
4	18.22
5	17.33
6	18.97
7	18.49
8	18.03
9	17.59
10	19.58

(Sumber: Dokumentasi Pribadi, 2026)

Tabel 4.27 menunjukkan hasil pengujian *Cycle Time* yang diperoleh secara otomatis menggunakan *gateway*. Pengujian dilakukan sebanyak sepuluh kali dengan

memanfaatkan waktu deteksi sensor pada *Distribution Station* dan *Sorting Station*, sehingga nilai *Cycle Time* dapat dihitung secara otomatis oleh sistem tanpa menggunakan *stopwatch*.



Gambar 4. 50 Payload Pengujian *Cycle Time* Pada Gateway
(Sumber : Dokumentasi Pribadi, 2026)

Berdasarkan hasil pengujian yang telah dilakukan pada Tabel 4.26 dan 4.27, nilai *Cycle Time* yang diperoleh dari sistem *gateway* menunjukkan hasil yang relatif mendekati pengukuran manual menggunakan *stopwatch*. Perbedaan nilai pengukuran pada beberapa percobaan dipengaruhi oleh waktu respons sensor, proses komunikasi data, serta kemungkinan keterlambatan respons operator saat melakukan pengukuran manual. Meskipun terdapat selisih nilai pada beberapa pengujian, sistem *gateway* telah mampu melakukan pencatatan *Cycle Time* secara otomatis dari objek terdeteksi pada *Distribution Station* hingga objek mencapai *Sorting station* pada proses distribusi *Modular Production System* (MPS).

Tabel 4. 28 Rata-rata Pengujian *Cycle Time*

Metode Pengujian	<i>Cycle Time</i> (s)
Stopwatch	19,34
Gateway	18,62

(Sumber: Dokumentasi Pribadi, 2026)

Berdasarkan hasil pengujian pada Tabel 4.28, rata-rata *Cycle Time* hasil pengukuran manual menggunakan *stopwatch* sebesar 19,34 detik, sedangkan hasil pengukuran menggunakan *gateway* sebesar 18,62 detik.

4.6.3 Skenario Pengujian dan Perhitungan OEE

Setelah sistem *Modular Production System* berhasil dibangun dan berfungsi dengan baik, langkah berikutnya adalah mengumpulkan data hasil distribusi dari mesin tersebut. Pengumpulan data dilakukan melalui beberapa skenario yang dirancang untuk mendapatkan nilai OEE yang bervariasi, dari rendah hingga Kelas Dunia.

Tabel 4. 29 Skenario Pengujian OEE

Pengujian Data ke -	Skenario	Keterangan
1	<i>User</i> menjalankan sistem <i>Modular Production System</i> dengan waktu <i>loading time</i> 60 menit dan <i>operating time</i> 53.91 menit, serta <i>downtime</i> selama 6.09 menit. Material yang diproses menghasilkan 85 produk baik dan 5 produk buruk, dengan total produk 90 unit. Nilai OEE pada pengujian ini mencerminkan efisiensi yang rendah karena jumlah <i>downtime</i> yang cukup besar dan performa distribusi yang masih rendah. Berdasarkan hasil perhitungan, nilai OEE yang diperoleh sebesar 43,92%.	OEE Rendah
2	Sistem kembali dijalankan dengan waktu <i>loading time</i> 60 menit dan <i>operating time</i> 56.02 menit, serta <i>downtime</i> selama 3.98 menit. Sistem menghasilkan 136 produk baik dan 7 produk buruk dari total produk 143 unit. Meskipun masih terdapat <i>downtime</i> , peningkatan jumlah produk baik dan total output distribusi menyebabkan nilai kualitas dan performa sistem meningkat dibandingkan pengujian sebelumnya. Berdasarkan hasil perhitungan, nilai OEE yang diperoleh sebesar 70,27%.	OEE Sedang
3	Sistem diuji kembali dengan waktu <i>loading time</i> 60 menit dan <i>operating time</i> 58.97 menit, serta <i>downtime</i> selama 1.04 menit. Pada pengujian ini, sistem berhasil	OEE Kelas Dunia

Pengujian Data ke -	Skenario	Keterangan
	menghasilkan 165 produk baik dan 2 produk buruk dari total 167 output. Jumlah produk baik yang tinggi serta nilai <i>downtime</i> yang kecil menyebabkan peningkatan nilai efektivitas sistem dibandingkan pengujian sebelumnya. Berdasarkan hasil perhitungan, nilai OEE mencapai 85,25%, yang menunjukkan performa sistem berada pada kategori Kelas dunia.	

(Sumber: Dokumentasi Pribadi, 2026)

Skenario pengujian ini dibuat berdasarkan hasil uji coba *Modular Production System Plant* yang telah dibuat, dan rinciannya dapat dilihat pada Tabel 4.29. Data yang terkumpul akan diproses secara berkala untuk menghitung OEE, dan hasilnya akan ditampilkan pada *dashboard* di Web-SCADA. Data dikumpulkan setiap satu jam, mewakili pengumpulan data setiap shift dalam industri.

Tabel 4. 30 Kategori *Overall Equipment Effectiveness (OEE)*

Nilai OEE	Kategori
40%-59%	Rendah
60%-84%	Sedang
85%-99%	Kelas Dunia
100%	Sempurna

(Sumber : (Solakhudin et al., 2024)

Dari skenario pengambilan data pada sistem *Modular Production System (MPS)*, diperoleh data operasional sistem yang digunakan untuk menghitung nilai OEE. Pengumpulan data dilakukan setiap satu jam, sehingga dapat dianalisis tingkat efektivitas proses distribusi dan *sorting* pada sistem. Data yang dicatat mencakup jumlah objek yang berhasil diproses (*good product*), jumlah objek yang tidak sesuai (*bad product*), waktu berhentinya sistem (*downtime*), durasi operasi sistem (*loading time*), serta waktu siklus proses (*cycle time*). Rincian parameter pengujian OEE tersebut ditampilkan pada Tabel 4.31 berikut.

Tabel 4. 31 Data untuk Perhitungan OEE

Data ke-	Data						
	<i>Loading Time</i> (Menit)	<i>Cycle Time</i> (Menit)	<i>Operating Time</i> (Menit)	<i>Downtime</i> (Menit)	<i>Good Product</i>	<i>Bad Product</i>	<i>Actual Output</i>
1	60	0.31	53.91	6.09	85	5	90
2	60	0.31	56.02	3.98	136	7	143
3	60	0.31	58.97	1.04	165	2	167

(Sumber: Dokumentasi Pribadi, 2026)

Untuk menghitung nilai OEE pada sistem *Modular Production System*, semua variabel yang diperlukan akan dimasukkan ke dalam sistem dan diproses. Data yang ditampilkan pada Tabel 4.31 merupakan hasil pengumpulan data dan akan dijadikan acuan dalam pengujian perhitungan OEE. Data tersebut akan dihitung baik secara manual maupun melalui sistem menggunakan rumus OEE. Dengan membandingkan hasil dari kedua metode tersebut, dapat diketahui persentase selisih (*error*) sebagai bahan evaluasi. Nilai error ini akan menjadi dasar untuk menilai seberapa baik akurasi perhitungan OEE yang dilakukan oleh sistem. Berikut adalah hasil perhitungan manual berdasarkan data operasional sistem yang telah dikumpulkan dan ditampilkan di bawah ini:

- Data ke- 1:

$$\begin{aligned}
 \text{Availability (\%)} &= \frac{53.91}{60} \times 100 \% \\
 &= 89.85 \%
 \end{aligned}$$

$$\begin{aligned}
 \text{Performance (\%)} &= \frac{0.31 \times 90}{53.91} \times 100 \% \\
 &= 51.75 \%
 \end{aligned}$$

$$\begin{aligned}
 \text{Quality (\%)} &= \frac{85}{90} \times 100 \% \\
 &= 94.44 \%
 \end{aligned}$$

Setelah hasil dari ketiga variabel utama OEE diperoleh, maka nilai akhir *Overall Equipment Effectiveness* berdasarkan data tersebut adalah:

$$\begin{aligned}
 \text{OEE (\%)} &= 89.85\% \times 51.75\% \times 94.44\% \\
 &= 43.92 \%
 \end{aligned}$$



Gambar 4. 51 Analisis OEE Pada OEE Page Data ke-1
(Sumber : Dokumentasi Pribadi, 2026)

Berikut Gambar 4.51 merupakan tampilan *dashboard Overall Equipment Effectiveness* (OEE) pada sistem Web-SCADA yang menampilkan hasil perhitungan *Availability*, *Performance*, *Quality*, dan OEE berdasarkan data operasional yang diperoleh secara langsung dari *Plant Modular Production System* (MPS). Pada pengujian ini, *gateway* multi-protokol berhasil mengakuisisi data dari PLC Omron CP2E-N30 DRA dan PLC Siemens S7-1200, kemudian mengirimkannya ke sistem Web-SCADA untuk dilakukan perhitungan OEE secara otomatis. Berdasarkan hasil pengujian, diperoleh nilai *Availability* sebesar 89,85%, *Performance* sebesar 51,75%, *Quality* sebesar 94,44%, sehingga menghasilkan nilai OEE sebesar 43,92%. Nilai OEE yang relatif rendah dipengaruhi oleh nilai *Performance* yang masih rendah serta downtime mesin selama 6,09 menit, sehingga efektivitas operasi sistem belum optimal.

- Data ke- 2:

$$\begin{aligned} \text{Availability (\%)} &= \frac{56.02}{60} \times 100 \% \\ &= 93.37 \% \end{aligned}$$

$$\begin{aligned} \text{Performance (\%)} &= \frac{0.31 \times 143}{56.02} \times 100 \% \\ &= 79.13 \% \end{aligned}$$

$$\begin{aligned} \text{Quality (\%)} &= \frac{136}{143} \times 100 \% \\ &= 95.1 \% \end{aligned}$$

Setelah hasil dari ketiga variabel utama OEE diperoleh, maka nilai akhir *Overall Equipment Effectiveness* berdasarkan data tersebut adalah:

$$\begin{aligned} \text{OEE (\%)} &= 93.37\% \times 79.13\% \times 95.1\% \\ &= 70.27\% \end{aligned}$$



Gambar 4. 52 Analisis OEE Pada OEE Page Data ke-2
(Sumber : Dokumentasi Pribadi, 2026)

Berikut Gambar 4.52 merupakan tampilan *dashboard Overall Equipment Effectiveness* (OEE) pada sistem Web-SCADA berdasarkan data operasional Plant *Modular Production System* (MPS) pada pengujian data ke-2. Hasil pengujian menunjukkan bahwa sistem berhasil melakukan akuisisi data, perhitungan metode OEE, serta menampilkan hasil analisis pada *dashboard* sesuai dengan kondisi operasional plant. Berdasarkan hasil perhitungan, diperoleh nilai *Availability* sebesar 93,37%, *Performance* sebesar 79,13%, dan *Quality* sebesar 95,10%, sehingga menghasilkan nilai OEE sebesar 70,27%. Peningkatan nilai OEE dibandingkan pengujian sebelumnya dipengaruhi oleh berkurangnya downtime menjadi 3,98 menit serta meningkatnya jumlah produk yang berhasil diproses, sehingga efektivitas sistem menjadi lebih baik.

- Data ke- 3:

$$\begin{aligned} \text{Availability (\%)} &= \frac{58.97}{60} \times 100 \% \\ &= 98.28 \% \end{aligned}$$

$$\begin{aligned} \text{Performance (\%)} &= \frac{0.31 \times 167}{58.97} \times 100 \% \\ &= 87.79 \% \end{aligned}$$

$$\begin{aligned} \text{Quality (\%)} &= \frac{165}{167} \times 100 \% \\ &= 98.8 \% \end{aligned}$$

Setelah hasil dari ketiga variabel utama OEE diperoleh, maka nilai akhir *Overall Equipment Effectiveness* berdasarkan data tersebut adalah:

$$\begin{aligned} \text{OEE (\%)} &= 98.28\% \times 87.79\% \times 98.8\% \\ &= 85.25\% \end{aligned}$$



Gambar 4. 53 Analisis OEE Pada OEE Page Data ke-3
(Sumber : Dokumentasi Pribadi, 2026)

Berikut Berikut Gambar 4.53 merupakan tampilan *dashboard Overall Equipment Effectiveness* (OEE) pada sistem Web-SCADA berdasarkan data operasional Plant *Modular Production System* (MPS) pada pengujian data ke-3. Hasil pengujian menunjukkan bahwa *gateway* multi-protokol berhasil mengirimkan data operasional ke sistem Web-SCADA sehingga nilai *Availability*, *Performance*, *Quality*, dan OEE dapat dihitung dan divisualisasikan secara otomatis. Berdasarkan hasil perhitungan, diperoleh nilai *Availability* sebesar 98,28%, *Performance* sebesar 87,79%, dan *Quality* sebesar 98,80%, sehingga menghasilkan nilai OEE sebesar 85,25%. Nilai tersebut merupakan hasil OEE tertinggi pada pengujian ini yang diperoleh karena downtime mesin menurun menjadi 1,04 menit, disertai peningkatan jumlah produk baik menjadi 165 unit dari total 167 unit yang diproses. Hasil tersebut menunjukkan bahwa sistem mampu merepresentasikan kondisi operasional plant dengan baik melalui visualisasi *dashboard* Web-SCADA.

Berdasarkan hasil perhitungan manual dan perhitungan yang dilakukan oleh sistem (Web-SCADA), keduanya dapat dibandingkan melalui tabel berikut. Perbandingan ini mencakup nilai akhir OEE serta masing-masing komponen penyusunnya, yaitu *Availability*, *Performance*, dan *Quality*.

Tabel 4. 32 Hasil Perbandingan OEE Manual dengan OEE WEB-SCADA

Data ke-	OEE Manual (%)	OEE Web-SCADA (%)	Error (%)
1	43.92	43.92	0
2	70.27	70.27	0
3	85.25	85.25	0

(Sumber: Dokumentasi Pribadi, 2026)

Berdasarkan hasil perbandingan antara perhitungan OEE secara manual dan perhitungan menggunakan sistem Web-SCADA pada Tabel 4.32, diperoleh hasil yang sama untuk ketiga data pengujian. Nilai OEE yang dihasilkan oleh kedua metode menunjukkan kesesuaian tanpa adanya selisih atau error (0%). Hasil ini menunjukkan bahwa sistem yang dibangun telah mampu menghitung nilai OEE dengan baik dan akurat. Selain itu, kesesuaian hasil tersebut membuktikan bahwa proses perhitungan OEE pada Web-SCADA telah berjalan sesuai dengan rancangan sistem. Dengan demikian, pengguna dapat memantau efektivitas kerja *Modular Production System* tanpa perlu melakukan perhitungan secara manual.

Tabel 4. 33 Hasil Perbandingan *Availability* Manual dengan Web-SCADA

Data ke-	Data (Menit)		<i>Availability</i> Manual (%)	<i>Availability</i> Web-SCADA (%)	<i>Error</i> (%)
	<i>Loading Time</i>	<i>Downtime</i>			
1	60	6.09	89.85	89.85	0
2	60	3.98	93.37	93.37	0
3	60	1.04	98.28	98.28	0

(Sumber: Dokumentasi Pribadi, 2026)

Hasil perbandingan pada Tabel 4.33 menunjukkan bahwa nilai *Availability* yang diperoleh dari perhitungan manual dan perhitungan menggunakan Web-SCADA memiliki hasil yang sama pada setiap pengujian. Tidak terdapat selisih antara kedua metode tersebut, sehingga nilai error yang diperoleh adalah 0%. Hasil ini menunjukkan bahwa sistem Web-SCADA telah mampu menghitung nilai *Availability* dengan akurat sesuai dengan perhitungan manual.

Tabel 4. 34 Hasil Perbandingan *Performance* Manual dengan Web-SCADA

Data ke-	OEE Manual (%)				<i>Performance</i> Manual (%)	<i>Performance</i> Web-SCADA (%)	Error (%)
	<i>Loading Time</i>	<i>Down time</i>	<i>Actual Output</i>	<i>Cycle Time</i>			
1	60	6.09	90	0.31	51.75	51.75	0
2	60	3.98	143	0.31	79.13	79.13	0
3	60	1.04	167	0.31	87.79	87.79	0

(Sumber: Dokumentasi Pribadi, 2026)

Berdasarkan data perbandingan pada Tabel 4.34, nilai *Performance* yang diperoleh dari perhitungan manual dan perhitungan menggunakan Web-SCADA menunjukkan hasil yang sama pada setiap pengujian. Tidak terdapat selisih antara kedua metode tersebut, sehingga persentase error yang diperoleh adalah 0%. Hasil ini menunjukkan bahwa sistem Web-SCADA telah mampu menghitung nilai *Performance* dengan akurat sesuai dengan perhitungan manual.

Tabel 4. 35 Hasil Perbandingan *Quality* Manual dengan Web-SCADA

Data ke-	Data (Menit)		<i>Quality</i> Manual (%)	<i>Quality</i> Web-SCADA (%)	Error (%)
	<i>Actual Output</i>	<i>Good Product</i>			
1	90	85	94.44	94.44	0
2	143	136	95.1	95.1	0
3	167	165	98.88	98.88	0

(Sumber: Dokumentasi Pribadi, 2026)

Berdasarkan hasil perbandingan pada Tabel 4.35, nilai *Quality* yang diperoleh dari perhitungan manual dan perhitungan menggunakan sistem Web-SCADA menunjukkan hasil yang sama. Tidak terdapat selisih antara kedua metode tersebut, sehingga nilai error yang diperoleh adalah 0%. Hasil ini menunjukkan bahwa sistem Web-SCADA telah mampu menghitung nilai *Quality* dengan akurat sesuai dengan perhitungan manual.

4.6.4 Pengujian OEE Aktual pada *Plant*

Pengujian OEE aktual dilakukan dengan menjalankan *plant distribution and sorting station* secara terus-menerus selama kurang lebih satu jam tanpa memberikan gangguan atau kondisi kerusakan secara sengaja. Pengujian ini digunakan untuk mengetahui nilai efektivitas *plant* berdasarkan kondisi operasi sebenarnya. Selama pengujian berlangsung, sistem Web-SCADA mencatat waktu operasi mesin, waktu henti, jumlah produk yang dihasilkan, jumlah *good product*, jumlah *bad product*, serta nilai *Availability*, *Performance*, *Quality*, dan OEE.

Data hasil pengujian OEE aktual diperoleh secara langsung dari sistem Web-SCADA setelah *plant* selesai beroperasi. Data yang digunakan dalam proses perhitungan meliputi *loading time*, *cycle time*, *operating time*, *downtime*, jumlah *good product*, jumlah *bad product*, dan *actual output*. Data hasil pengujian tersebut ditunjukkan pada Tabel 4.36.

Tabel 4. 36 Data Pengujian Aktual pada *Plant*

Data ke-	Data						
	<i>Loading Time</i> (Menit)	<i>Cycle Time</i> (Menit)	<i>Operating Time</i> (Menit)	<i>Downtime</i> (Menit)	<i>Good Product</i>	<i>Bad Product</i>	<i>Actual Output</i>
1	60	0.31	60	0	179	0	179

(Sumber: Dokumentasi Pribadi, 2026)

Berdasarkan Tabel 4.36, *plant* beroperasi selama 60 menit tanpa mengalami *downtime*. Selama waktu tersebut, *plant* menghasilkan sebanyak 179 benda kerja. Seluruh benda kerja tercatat sebagai *good product*, sedangkan jumlah *bad product* adalah 0. Data tersebut selanjutnya digunakan untuk menghitung nilai *Availability*, *Performance*, *Quality*, dan OEE secara manual.

Data Pengujian Aktual :

$$\begin{aligned}\text{Availability (\%)} &= \frac{60}{60} \times 100 \% \\ &= 100 \%\end{aligned}$$

$$\begin{aligned}\text{Performance (\%)} &= \frac{0.31 \times 179}{60} \times 100 \% \\ &= 92.47 \%\end{aligned}$$

$$\begin{aligned}\text{Quality (\%)} &= \frac{179}{179} \times 100 \% \\ &= 100 \%\end{aligned}$$

Setelah hasil dari ketiga variabel utama OEE diperoleh, maka nilai akhir *Overall Equipment Effectiveness* berdasarkan data tersebut adalah:

$$\begin{aligned} \text{OEE (\%)} &= 100\% \times 92.47\% \times 100\% \\ &= 92.47\% \end{aligned}$$



Gambar 4. 54 Analisis OEE Pada OEE *Page* Data Aktual
(Sumber : Dokumentasi Pribadi, 2026)

Gambar 4.54 menunjukkan hasil pengujian OEE aktual yang ditampilkan pada halaman OEE Web-SCADA setelah *plant* beroperasi selama kurang lebih satu jam. Sistem mencatat nilai *Availability* sebesar 100%, *Performance* sebesar 92,47%, *Quality* sebesar 100%, dan OEE sebesar 92,47%. Selain itu, selama pengujian diperoleh total produksi sebanyak 179 benda kerja yang seluruhnya tercatat sebagai *good product*, tanpa adanya *bad product* maupun *downtime*. Hasil perhitungan OEE secara otomatis pada Web-SCADA juga sesuai dengan hasil perhitungan manual, sehingga fungsi perhitungan pada sistem dapat dinyatakan berjalan dengan benar.

Nilai OEE sebesar 92,47% terutama dipengaruhi oleh komponen *Performance* yang belum mencapai 100%. Hal tersebut disebabkan oleh kendala mekanik pada bagian konveyor, yaitu benda kerja terkadang tersangkut pada *bracket* sehingga pergerakannya melambat dan waktu proses menjadi lebih panjang. Meskipun demikian, *plant* tetap dapat beroperasi tanpa waktu henti dan seluruh benda kerja berhasil diproses sebagai produk baik

BAB 5

KESIMPULAN DAN SARAN

Bab ini berisi uraian mengenai kesimpulan yang diperoleh berdasarkan hasil perancangan, implementasi, dan pengujian sistem yang telah dilakukan pada penelitian ini. Kesimpulan tersebut digunakan untuk menilai tingkat keberhasilan sistem dalam memenuhi tujuan penelitian yang telah ditetapkan. Selain itu, disampaikan pula beberapa saran yang diharapkan dapat menjadi masukan untuk pengembangan dan penyempurnaan sistem pada penelitian berikutnya.

5.1 Kesimpulan

Berdasarkan percobaan dan pengujian yang telah dilakukan pada Tugas Akhir yang berjudul “Rancang Bangun Sistem WEB-SCADA Pada *Modular Production System Menggunakan Multi Kontroller*” ini dapat disimpulkan sebagai berikut:

1. Sistem Web-SCADA pada *Modular Production System* (MPS) berhasil dirancang dan diimplementasikan untuk melakukan fungsi pemantauan dan pengendalian pada *Distribution station* dan *sorting station*. Sistem terdiri atas Web-SCADA lokal yang digunakan untuk *Monitoring* dan kontrol proses, serta Web-SCADA berbasis *cloud* yang digunakan untuk *Monitoring* data operasional, alarm, data logger, dan analisis OEE. Hasil implementasi menunjukkan bahwa data operasional *plant* dapat dipantau dan dikendalikan melalui antarmuka berbasis web sesuai dengan kondisi aktual sistem.
2. *Gateway* multi-protokol berbasis Node.js berhasil dibangun untuk menghubungkan PLC Omron CP2E-N30 DRA yang menggunakan protokol FINS UDP dan PLC Siemens S7-1200 yang menggunakan protokol S7 TCP/IP. Hasil pengujian menunjukkan bahwa kedua protokol dapat digunakan untuk pertukaran data antara PLC dan sistem Web-SCADA dengan baik. Pada pengujian *delay read*, protokol FINS UDP menghasilkan rata-rata *delay* sekitar 1,7–2,0 ms, sedangkan S7 TCP/IP menghasilkan rata-rata *delay* sekitar 12–13 ms. Berdasarkan standar TIPHON, kedua nilai tersebut termasuk dalam kategori sangat bagus karena berada di bawah 150 ms. Meskipun demikian, FINS UDP memberikan waktu respons yang lebih cepat dibandingkan S7 TCP/IP dalam proses pembacaan data PLC. Pada pengujian komunikasi *gateway* ke *cloud*, diperoleh rata-rata *latency* sebesar 315,2 ms. Berdasarkan standar TIPHON, nilai

rata-rata tersebut termasuk dalam kategori sedang. Seluruh pengujian komunikasi menunjukkan bahwa *gateway* mampu mengintegrasikan kedua PLC, mengirimkan data ke Web-SCADA lokal dan *cloud*, serta mendukung kebutuhan monitoring dan pengendalian pada sistem yang dikembangkan.

3. Data operasional dan nilai *Overall Equipment Effectiveness* (OEE) berhasil divisualisasikan pada sistem Web-SCADA dalam bentuk *dashboard monitoring* dan analisis. Sistem mampu melakukan perhitungan *Availability*, *Performance*, *Quality*, dan OEE secara otomatis berdasarkan data produksi yang diperoleh dari *gateway*. Nilai *cycle time* yang digunakan dalam perhitungan *Performance* diperoleh dari hasil pengujian waktu siklus produksi dengan rata-rata sebesar 0,31 menit per produk. Berdasarkan pengujian tiga skenario, diperoleh nilai OEE sebesar 43,92% pada data ke-1, 70,27% pada data ke-2, dan 85,25% pada data ke-3. Hasil perbandingan antara perhitungan manual dan perhitungan sistem menunjukkan nilai yang sama dengan *error* sebesar 0% pada seluruh data pengujian. Selain itu, pengujian menggunakan data aktual *plant* selama satu jam menghasilkan nilai *Availability* sebesar 100%, *Performance* sebesar 92,47%, *Quality* sebesar 100%, dan OEE sebesar 92,47%, dengan jumlah produksi sebanyak 179 produk baik tanpa produk cacat dan tanpa *downtime*. Nilai *Performance* yang belum mencapai 100% dipengaruhi oleh kondisi mekanis ketika benda kerja sesekali tersangkut pada bagian *bracket conveyor*, sehingga memperlambat proses produksi. Hasil tersebut membuktikan bahwa sistem Web-SCADA mampu melakukan perhitungan dan visualisasi OEE secara akurat berdasarkan data skenario maupun kondisi aktual *plant*, sehingga dapat digunakan sebagai sarana monitoring dan evaluasi kinerja mesin.

5.2 Saran

Meskipun sistem Web-SCADA yang dikembangkan telah mampu memenuhi tujuan penelitian, masih terdapat beberapa aspek yang dapat dikembangkan pada penelitian selanjutnya. Pengembangan tersebut meliputi peningkatan keamanan sistem, perluasan implementasi pada lingkungan yang lebih kompleks, serta pengembangan fitur pendukung untuk meningkatkan efektivitas proses *Monitoring* dan evaluasi kinerja mesin. Oleh karena itu, beberapa saran yang dapat diberikan adalah sebagai berikut.

1. Sistem dapat dikembangkan dengan menambahkan mekanisme keamanan yang lebih baik, seperti enkripsi data, manajemen hak akses pengguna, serta audit

keamanan secara berkala. Hal ini penting untuk meningkatkan keamanan komunikasi dan pertukaran data antara perangkat OT (*Operational Technology*) dan sistem IT, terutama ketika sistem diterapkan pada lingkungan industri yang terhubung dengan jaringan publik.

2. Penelitian selanjutnya dapat mengimplementasikan sistem pada *plant* atau lingkungan industri yang lebih kompleks dengan jumlah perangkat, variasi proses, dan volume data yang lebih besar. Pengujian pada skala yang lebih luas dapat digunakan untuk mengevaluasi performa, keandalan, dan stabilitas sistem dalam kondisi operasional yang lebih mendekati kebutuhan industri sebenarnya.
3. Sistem dapat dikembangkan lebih lanjut dengan menambahkan tampilan *dashboard* yang lebih informatif dan interaktif, serta fitur-fitur pendukung yang dapat meningkatkan kemudahan pengguna dalam melakukan *Monitoring* dan analisis. Pengembangan tersebut dapat berupa penambahan visualisasi data yang lebih beragam, ringkasan performa sistem, serta penyajian informasi yang lebih mudah dipahami oleh operator maupun manajemen.

(Halaman ini sengaja dikosongkan)

DAFTAR PUSTAKA

- Abizari, G. Al, Syafitra, F., & ... (2025). Modifikasi *Modular Production System* (Mps) Berbasis Programmable Logic Control (Plc) Siemens. *Prosiding Seminar*
<http://snitt.polman-babel.ac.id/index.php/snitt/article/view/648%0Ahttps://snitt.polman-babel.ac.id/index.php/snitt/article/download/648/410>
- ADAM RISKY MUHARAM. (2025). *INTEGRASI PLC MULTI VENDOR DALAM PENGENDALIAN REAL-TIME DENGAN HMI DAN VSD*. 6.
[https://repository.pnj.ac.id/id/eprint/28797/1/1 Bab dan 5 Bab.pdf](https://repository.pnj.ac.id/id/eprint/28797/1/1%20Bab%20dan%205%20Bab.pdf)
- Ardaneshwara, A. M. (2024a). *Implementasi SCADA pada Modular Production System (MPS) dengan Integrasi Multi Brand PLC menggunakan FINS UDP dan Modbus TCP pada Node-RED Implementasi SCADA pada Modular Production System (MPS) dengan Integrasi Multi Brand PLC menggunakan FINS UDP da.*
- Ardaneshwara, A. M. (2024b). Optimisasi SCADA di MPS dengan FINS UDP vs Modbus TCP via Node-RED. *Indonesian Journal of Computer Science*, 13(4), 284–301. <http://ijcs.net/ijcs/index.php/ijcs/article/view/4315/750>
- Asita, F. N., Hasin, M. K., Maulana, Z., & Putra, A. (2023). *Pengiriman Data Realtime PLC ke ERP Odoo dengan Dilengkapi Analisa OEE*. 15(x), 1–13.
- Auditiya Marizka, D., Lastiyan Ashila, A., Agustina Solihah, A., Adi Prasetyo, H., Riza, F., & Wicaksono, R. (2025). Rancang Bangun Sistem Kendali PID Motor DC Berbasis Blynk untuk meningkatkan Efisiensi dan Fleksibilitas. *Jurnal Inovasi Teknologi Terapan*, 3(2), 277–283. <https://doi.org/10.33504/jitt.v3i2.345>
- Aziz, A., & Aribowo, D. (2025). *Sistem Kendali Cross Transfer Pada Area Cooling Bed Di Pt . Krakatau Baja Konstruksi*. 9(2), 56–64.
- Dian, D., Prasti, D., & Akramunnisa, A. (2025). Rancang Bangun Sistem Informasi Pengelolaan Data Vaksinasi Berbasis Web pada Puskesmas Wara Barat Kota Palopo. *Jurnal Informasi Dan Teknologi*, 2(3), 1–8.
<https://doi.org/10.30605/jitu.v2i3.336>
- Fachrurrozi, M. Z., & Damayanti, E. (n.d.). *Perancangan Sistem Deteksi Kebakaran Berbasis Arduino Dan Plc Omron Cp1E N40 Menggunakan*. 13(3).
- Fitriani, L., & Alfari, M. T. (2025). *Rancang Bangun Aplikasi Monitoring Kebutuhan Masyarakat Untuk Relawan TIK Berbasis Web dan Mobile*. 1153–1164.

<https://doi.org/10.33364/algorithm/v.22-2.2574>

- Gazali, R., Fedianto, L., Permana, P., & Utomo, U. (2022). Perancangan Modul Latih Elektro Pneumatic Berbasis Plc. *Jeis: Jurnal Elektro Dan Informatika Swadharma*, 2(2), 49–54. <https://doi.org/10.56486/jeis.vol2no2.224>
- Idris, H. S., Saputra, A., & Mulyadi, H. (2023). 6415-Manuscript-22004-1-10-20231222 sistem scada berbasis iot. 7(2).
- Ilmiah, J., Tls, D., & Bukit, P. T. (2025). *JITS Penerapan Sensor proximity Untuk Sistem Monitoring Gate Weigh Bin*. 3(1), 196–202.
- Iswari, L., & Nasution. (2021). Penerapan React JS Pada Pengembangan *Frontend Automata*, 2(2), 193–200.
- Karyatanti, I. D. P. (2024). Pengembangan *Cloud SCADA 1.3* sebagai otomasi industri jarak jauh. *Cyclotron*, 7(01), 71–75. <https://doi.org/10.30651/cl.v7i01.20522>
- Kirana, A. T., & Widiasih, W. (2024). Analisis Perhitungan *Overall Equipment Effectiveness* (OEE) untuk Meningkatkan Efisiensi Mesin Haloong (Studi Kasus: PT Benteng Api Technic, Gresik). *Jurnal Tekstil: Jurnal Keilmuan Dan Aplikasi Bidang Tekstil Dan Manajemen Industri*, 7(1), 59–68. <https://doi.org/10.59432/jute.v7i1.89>
- Maulida, H., Yuhendri, M., Mirshad, E., & Yanto, D. T. (2025). *Servo motor control in a color-based sorting system using the Internet of Things (IoT)*. 02(02), 8–14.
- Muchtar, T., Tri, A., Utami, B., & Lutfi, D. *. (2024). Development of Integration Systems using PLC Siemens S71200 and PLC OMRON CP1L-L20DR-A Based on Ethernet and KepServerEx. *JEAT: Journal of Electrical and Automation*, 3(1), 16.
- MUHAMMAD NAUFAL HENDRI. (2025). *INTEGRASI SCADA SMARTICS MELALUI LOCALHOST PADA SORTING STATION BERBASIS MODBUS TCP/IP*. 25. <https://www.city.kawasaki.jp/500/page/0000174493.html>
- Polii, I. F. Y., Tumembow, M. N., & Kadili, A. (2025). Perancangan Trainer Kit Pneumatik Dua Silinder Berbasis PLC dan HMI di Laboratorium Mekatronika Politeknik Negeri Manado. *Jurnal Masina Nipake*, 5(1), 72–83. <http://p3m.polimdo.ac.id/jurnal/index.php/MASINA-NIPAKE/article/view/127%0Ahttp://p3m.polimdo.ac.id/jurnal/index.php/MASIN-A-NIPAKE/article/download/127/61>
- Putra, G. W. (2025). *Aplikasi berbasis IoT untuk pemantauan dan pemeliharaan pc mini secara real-time*. 18(2), 78–88.
- Putra, S. C., & Saripudin. (2025). Pengaruh Variasi Kecepatan Konveyor Terhadap

- Waktu Operasional Dan Pemilah Logam dan Non Logam. *Jurnal Informatika Dan Teknik Elektro Terapan*, 13(3S1).
- Samudra, A. I., Syawal, R. M., Satrio, T., Haryono, W., Suryadi, S., Widiyatmoko, B., & Setiono, A. (2025). Aplikasi Pemantauan dan Prediksi Kadar Kualitas Air Sungai Dengan Algoritma Random Forest Pada Pusat Riset Fotonika. *RIGGS: Journal of Artificial Intelligence and Digital Business*, 4(2), 5376–5387. <https://doi.org/10.31004/riggs.v4i2.1439>
- Siburian, K., & Arvianto, A. (2025). *Analisis Six Big Losses Untuk Meningkatkan Nilai Overall Equipment Effectiveness (Oee) Pada Mesin Palletizer Di Pt Solusi Bangun Indonesia Tbk Cilacap*. 1–7. https://ejournal3.undip.ac.id/index.php/ieoj/article/view/52508?utm_source
- Teknik, J., Kapal, K., Perkapalan, P., & Surabaya, N. (2025). *RANCANG BANGUN PANEL KONTROL MONITORING*.
- Wibowo, P. A. (2023). Analysis of *Overall Equipment Effectiveness* (OEE) as an Effort to Increase The Productivity Filling Line of Lithos Packaging Lubricant. *Simetrikal: Journal of Engineering and Technology*, 3(1), 1–8. <https://doi.org/10.32734/jet.v3i1.13435>

(Halaman ini sengaja dikosongkan)

LAMPIRAN BIODATA MAHASISWA

1. Nama : Muhammad Reza Alfiano
2. NRP : 0922040088
3. Progam Studi : D4 – Teknik Otomasi
4. Agama : Islam
5. Status : Belum Menikah
6. Alamat Asal : Jl. Biduri Bulan 2.4 No 22 KBD, Driyorejo
7. Nomor Telepon : 087857621497
8. Jenis Kelamin : Laki-Laki
9. Email : reza.alfiano@student.ppns.ac.id
10. Tempat, Tanggal Lahir : Surabaya, 03 Juli 2004
11. Nama Orang Tua / Wali : Risdianti
12. Alamat Orang Tua / Wali : Jl. Biduri Bulan 2.4 No 22 KBD, Driyorejo
13. Telepon Orang Tua / Wali : 081328977360
14. Riwayat Pendidikan :



PENDIDIKAN FORMAL			
Pendidikan	Tahun	Tempat Pendidikan	Jurusan
Diploma IV	2022 - 2026	Politeknik Perkapalan Negeri Surabaya	Teknik Otomasi
SMK	2019 - 2022	SMKN 1 Driyorejo	Teknik Elektronika Industri
SMP	2015 - 2018	SMP Al Fatah	-
SD	2009 - 2015	SDN Lidah Kulon 1	-

(Halaman ini sengaja dikosongkan)

Tugas Akhir_Bismillah Semhas.pdf

ORIGINALITY REPORT

13%

SIMILARITY INDEX

PRIMARY SOURCES

1	repository.ppnns.ac.id Internet	1141 words — 4%
2	repository.its.ac.id Internet	336 words — 1%
3	repository.untag-sby.ac.id Internet	89 words — < 1%
4	id.123dok.com Internet	73 words — < 1%
5	dspace.uui.ac.id Internet	71 words — < 1%
6	repository.ub.ac.id Internet	71 words — < 1%
7	Muhammad Rafi Solakhudin, Muhammad Khoirul Hasin, li Munadhif. "Implementation OEE in Integrating Siemens S7-1200 Data with Odoo ERP", Jurnal Ecotipe (Electronic, Control, Telecommunication, Information, and Power Engineering), 2024 Crossref	69 words — < 1%
8	123dok.com Internet	67 words — < 1%
9	repository.pnj.ac.id Internet	67 words — < 1%
10	text-id.123dok.com Internet	67 words — < 1%
11	erepository.uwks.ac.id Internet	61 words — < 1%