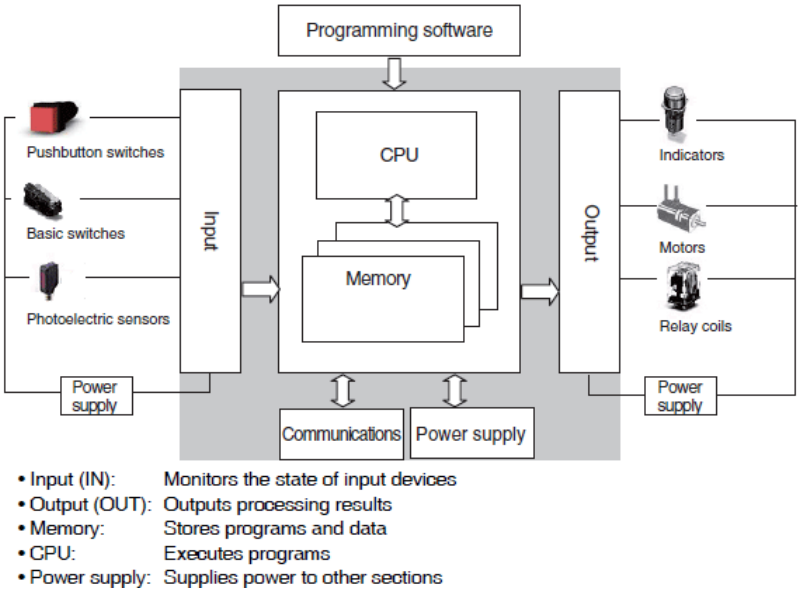
BAB II

TINJAUAN PUSTAKA

2.1 PLC (*Programmable Logic Controller*)

Konsep PLC, singkatan *dari Programmable Logic Controller*, adalah sebuah sistem yang memiliki beberapa karakteristik utama. Pertama, "*Programmable*" mengacu pada kemampuannya untuk menyimpan program dengan mudah yang bisa diubah-ubah sesuai kegunaannya. Kedua, "*Logic*" menunjukkan kemampuan dalam memproses data secara aritmatik seperti membandingkan, menjumlahkan, mengalikan, membagi, mengurangi, dan operasi. Terakhir, "*Controller*" menunjukkan kemampuannya dalam mengontrol dan mengatur proses sehingga menghasilkan output yang diinginkan. PLC menjadi otak dari sistem otomatisasi industri yang kompleks.(Frank Petruzella, 2017)



Gambar 2.1 Diagram Blok PLC
Sumber: omron.com

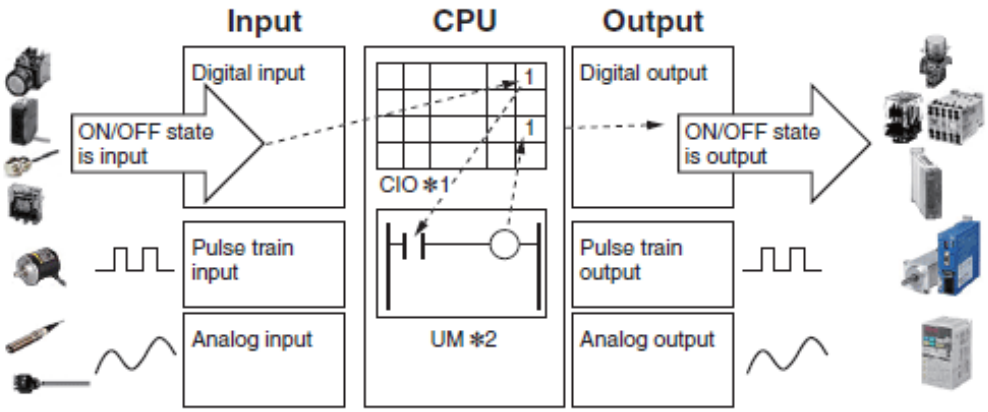
PLC memiliki beberapa komponen yang ada di dalamnya. Komponen tersebut dapat dilihat pada gambar 2.1,Fungsi dan bagian-bagian dari PLC adalah sebagai berikut:

1. *Central Processing Unit* (CPU)

CPU merupakan otak dari PLC yang biasanya terdiri dari Mikroproccesor berfungsi untuk mengimplementasikan logika dan mengendalikan komunikasi antar modul. CPU mengeksekusi program dengan mengambil data dari penyimpanan memori yang berisi program *ladder* dan status *I/O*, CPU dapat mengontrol semua aktivitas PLC yang dirancang agar pengguna dapat memberikan perintah yang diinginkan.

2. *Programming Memory* (PM)

PM digunakan untuk menyimpan program yang telah dibuat oleh. Program mencakup instruksi yang mengatur cara kerja PLC dalam menjalankan fungsi-fungsi kontrol. Seperti pada gambar 2.2, memori pada CPU meyimpan program dari *programming device* dan data yang dikirim dari modul input dan output.



Gambar 2.2 Cara Kerja PLC

Sumber: omron.com

3. *Power Supply*

Power Supply berfungsi untuk menyediakan daya listrik ke PLC dan perangkat yang terhubung ke PLC. Terdapat dua jenis daya yang dapat diterima berdasarkan jenis daya yang dapat diterima oleh PLC, yaitu PLC dengan daya AC dan PLC dengan daya DC. PLC dengan daya DC membutuhkan *Power Supply* DC yang terpisah untuk dapat digunakan, adapula PLC yang membutuhkan daya AC dengan terhubung langsung ke sumber listrik AC.

4. *Programming Device* (PD)

Programming Device adalah suatu perangkat yang digunakan untuk menulis, memodifikasi, dan memantau program yang ada pada memori PLC seperti yang terlihat pada gambar 2.2. Untuk dapat menggunakannya, pengguna diharuskan memiliki *software* yang sesuai dengan jenis PLC yang digunakan. Program yang digunakan umumnya menggunakan Logika *Ladder* dengan menggunakan simbol-simbol grafik yang dingiinkan.

5. Modul *Input*

Modul *input* adalah perangkat elektronik yang berfungsi untuk menerima sinyal dari perangkat input luar dan mengirimkan instruksi tersebut ke PLC. Cara kerja modul input bisa dilihat seperti pada gambar 2.2 dimana sinyal dari perangkat input luar dikirimkan ke memori PLC. Salah satu perangkat input luar tersebut dapat berupa sebuah *push button*, *push button* memberikan instruksi on/off ke modul input dan akan disimpan pada memori PLC.

6. Modul *Output*

Modul *Output* adalah perangkat elektronik dengan fungsi sebagai penghubung dari PLC ke perangkat Output luar. Cara kerjanya dapat dilihat pada gambar 2.2. Contoh dari perangkat tersebut adalah relay, PLC memberikan instruksi on/off pada modul output lalu instruksi tersebut diteruskan ke relay.

2.1.1 PLC Mitsubishi FX5U

Gambar 2.3. PLC Mitsubishi FX5U

Sumber : Dokumen Pribadi

Programmable logic controller (PLC) mitsubishi FX5U merupakan salah satu perangkat dalam seri MELSEC iQ-F buatan eitsubishi electric yang dirancang untuk memberikan solusi otomatisasi industri yang andal dan efisien(Mitsubishi Electric, 2016c). Secara umum, PLC ini digunakan untuk mengendalikan dan mengawasi berbagai proses industri dengan tingkat presisi dan kecepatan yang tinggi. Dibekali dengan berbagai fitur canggih, FX5U mampu mengontrol berbagai jenis peralatan dalam skala menengah hingga besar, menjadikannya pilihan yang tepat untuk aplikasi industri.

FX5U mendukung berbagai modul *input/output* (I/O) digital maupun analog, serta berbagai *port* komunikasi seperti Ethernet, dan RS-485(Mitsubishi

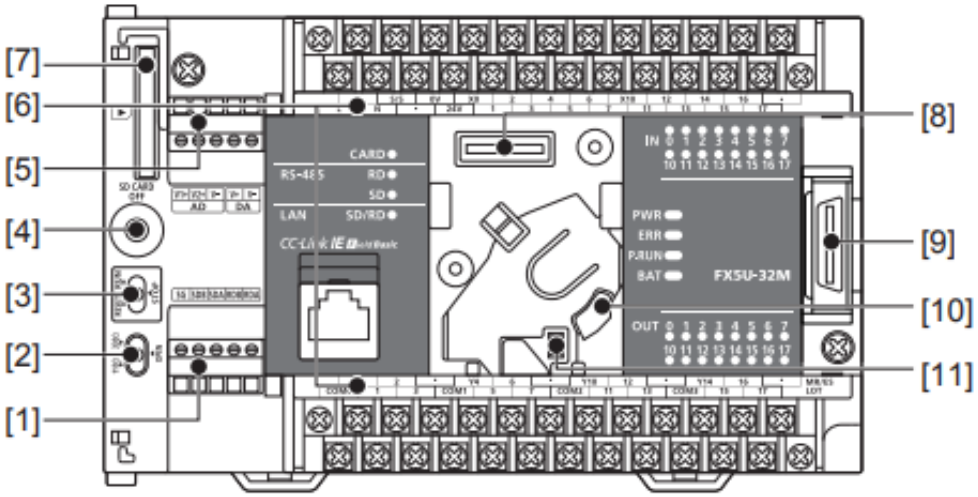
Electric, 2016). Ini memudahkan integrasi dengan perangkat lain dan memungkinkan kendali yang lebih fleksibel. Fitur lain yang menonjol adalah kontrol PID (*Proportional Integral Derivative*) dan *motion control*, yang membuat PLC ini sangat cocok untuk pengaplikasian pengendalian gerakan yang presisi. Untuk memberikan gambaran lebih rinci mengenai kemampuan alat ini, berikut adalah tabel spesifikasi teknis dari Mitsubishi FX5U yang bisa dilihat pada tabel 2.1

Tabel 2.1 Spesifikasi Teknis Mitsubishi FX5U

Kategori	Keterangan
Power Supply (V)	24V DC, 100-240V AC
Tipe CPU	FX5U
Kapasitas Memori Program	64K STEPS
Input Digital	32
Output Digital	32
Input Analog	2
Waktu Pemrosesan Instruksi (ns)	34
Komunikasi	Ethernet Port, RS-485
Slot Ekspansi	8 modul ekspansi
Port Memori Eksternal	Slot SD Card

Sumber: *mitsubishielectric.com*

Setelah memahami spesifikasi teknis, berikut adalah penguraian lebih lanjut tentang komponen-komponen fisik dari PLC Mitsubishi FX5U seperti pada gambar 2.4.



Gambar 2.4 Bagian-bagian pada PLC FX5U

Sumber: *mitsubishielectric.com*

1. *Built-in* RS-485 *communication terminal block*: Blok terminal untuk koneksi dengan perangkat yang kompatibel dengan RS-485.
2. *RS-485 terminal resistor selector switch*: Saklar untuk mengubah resistansi terminal pada komunikasi RS-485 bawaan.
3. *RUN/STOP/RESET switch*: Saklar untuk mengoperasikan modul CPU.
4. SD *memory card disable switch*: Saklar untuk menonaktifkan akses ke kartu memori SD saat kartu tersebut akan dilepas.
5. *Built-in analog I/O terminal block*: Blok terminal untuk menggunakan fungsi analog bawaan.
6. *Terminal names*: Nama-nama sinyal untuk suplai daya, terminal input, dan output.

7. SD *memory card slot*: Slot untuk memasukkan kartu memori SD.
8. *Expansion board connector*: Konektor untuk menghubungkan papan ekspansi.
9. *Extension connector*: Konektor untuk menghubungkan kabel ekstensi modul tambahan.
10. *Battery holder*: Tempat penyimpanan baterai opsional.
11. *Battery connector*: Konektor untuk menghubungkan baterai opsional.

2.1.2 GX Works 3

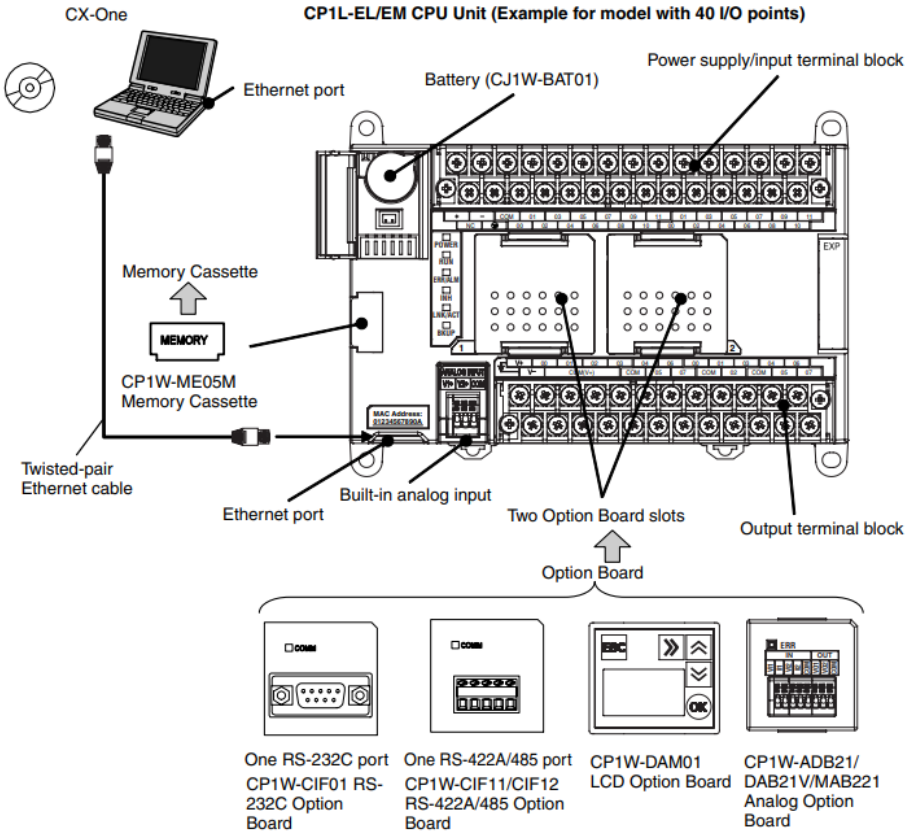
Untuk memprogram PLC FX5U, *Mitsubishi Electric* menyediakan software GX Works 3, yang dirancang dengan antarmuka intuitif dan mendukung berbagai bahasa pemrograman seperti Ladder Diagram (LD) dan Structured Text (ST). Software ini dilengkapi dengan fitur simulasi dan debugging yang memungkinkan pengguna untuk menguji program secara virtual, sehingga memudahkan identifikasi dan perbaikan kesalahan sebelum diunggah ke PLC. Selain itu, GX Works 3 menyederhanakan manajemen proyek dengan memungkinkan pengguna untuk mengatur dan mengelola elemen program secara efisien, serta menyediakan fitur untuk mengelola pengaturan komunikasi dan konfigurasi I/O. Dengan semua fitur ini, GX Works 3 menjadi alat yang sangat efisien untuk merancang, mengimplementasikan, dan mengelola sistem kontrol otomatisasi yang kompleks dan terintegrasi.

2.1.3 PLC Omron CP1L-E

Gambar 2.5. PLC Omron CP1L-E

Sumber: omron.com

PLC Omron CP1L-E merupakan salah satu model dari seri CP1 yang memiliki kemampuan dalam kontrol otomatisasi yang fleksibel untuk berbagai aplikasi industri. PLC ini dirancang untuk memberikan performa tinggi dengan ukuran yang kompak, membuatnya ideal untuk digunakan dalam ruang terbatas sambil tetap menawarkan fungsionalitas yang kaya. Dengan waktu pemrosesan yang cepat dan berbagai pilihan modul I/O, CP1L-E mampu menangani berbagai tugas kontrol, dari yang sederhana hingga yang kompleks(Omron Corporation, 2014). Berikut adalah uraian lebih lanjut tentang komponen-komponen fisik dari PLC Mitsubishi FX5U seperti yang dilihat pada gambar 2.6.



Gambar 2.6 Bagian-bagian pada PLC Omron CP1L-E

Sumber: *omron.com*

Kemampuan komunikasi yang mendukung berbagai protokol, termasuk Ethernet dan Modbus TCP/IP, memudahkan integrasi dengan perangkat lain dalam sistem otomasi, seperti sensor, aktuator, dan HMI (*Human-Machine Interface*). CP1L-E juga dilengkapi dengan berbagai fungsi pemrograman yang canggih, memungkinkan pengguna untuk menciptakan logika kontrol yang efisien sesuai dengan kebutuhan. Untuk memberikan gambaran lebih rinci mengenai kemampuan alat ini, berikut adalah tabel spesifikasi teknis dari Omron CP1l-E yang bisa dilihat pada tabel 2.2.

Tabel 2.2 Spesifikasi PLC Omron CP1L-E

Kategori	Keterangan
Power Supply (V)	24V DC
Memori Data	32 KB (dapat diperluas dengan modul tambahan)
Memori Program	60 KB
Input Digital Inputs	14
Output Digital	10
Input Analog	2
Waktu Pemrosesan	≤ 1 ms
Komunikasi	Ethernet, RS-485, RS-232
Fungsi Bawaan	PID Control, Timer, Counter, dan fungsi logika lainnya
Port Memori Eksternal	Slot SD Card

Sumber: *omron.com*

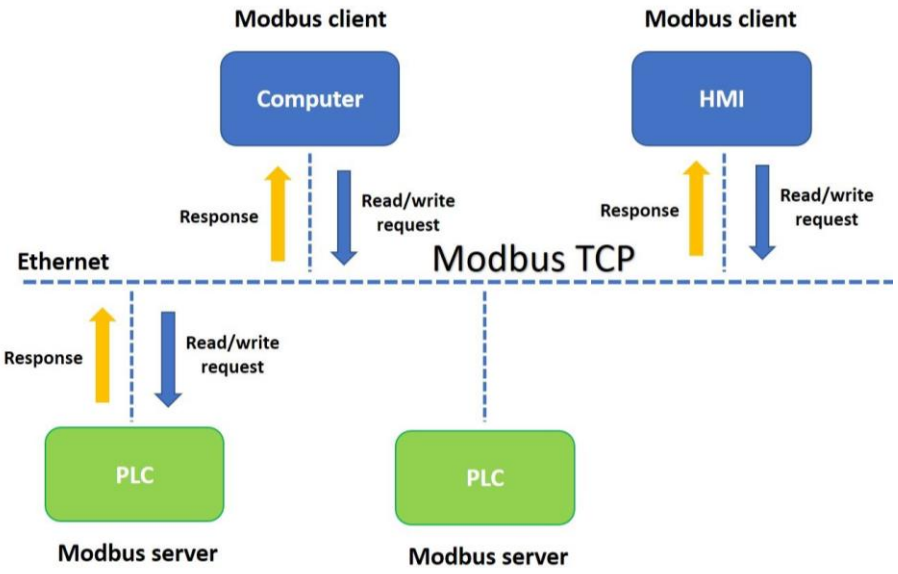
2.1.4 CX-Programmer

Untuk memprogram PLC CP1L-E, dapat menggunakan software CX-Programmer yang dikembangkan oleh Omron. CX-Programmer adalah software yang menyediakan interface grafis untuk memprogram pengendalian PLC(Cahya et al., 2022). Dalam CX-Programmer, pengguna dapat membuat instruksi program, mengatur konfigurasi *I/O*, memonitor program, serta mengimpor dan mengekspor data program. PLC Omron CP1L-E mendukung berbagai bahasa pemrograman yang memberikan fleksibilitas terhadap penggunanya untuk memilih bahasa pemrograman yang sesuai dengan kebutuhan diantaranya ada *ladder diagram* (LD), *Function Block Diagram* (FBD), *Stuctured Text* (ST), dan *Sequential Function Chart* (SFC).

2.2 Modbus TCP

Modbus TCP/IP merupakan salah satu varian dari protokol komunikasi Modbus yang dirancang untuk beroperasi di jaringan Ethernet(Modbusorg, 2006). Protokol ini memungkinkan pertukaran data antar perangkat industri seperti *Programmable Logic Controller* (PLC), sensor, aktuator, dan sistem *monitoring* berbasis komputer(S. Tamboli, 2015). Karena menggunakan protokol TCP/IP yang umum di internet, Modbus TCP/IP memungkinkan komunikasi jarak jauh melalui jaringan lokal (LAN) maupun jaringan luas (WAN) seperti internet.

Dalam protokol ini, setiap perangkat dapat berperan sebagai *client* atau peminta data dan sebagai *server* atau penyedia data. Seperti yang dilihat pada gambar 2.7, komunikasi dimulai oleh *client* yang mengirimkan permintaan, dan server merespon dengan data yang diminta.



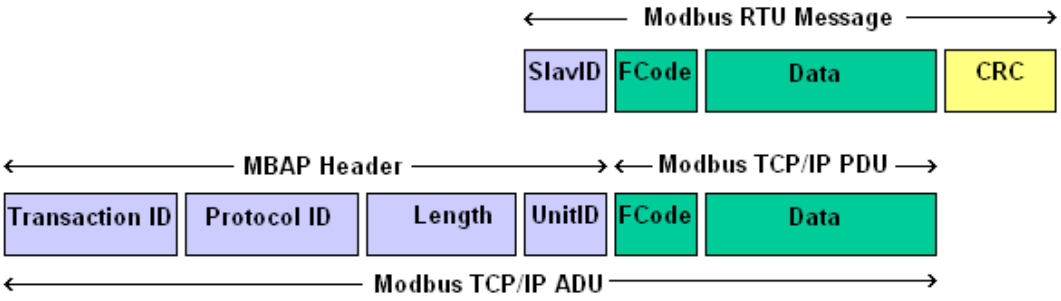
Gambar 2.7 Contoh Penggunaan Modbus TCP/IP

Sumber: *plcynergy.com*

Modbus TCP/IP beroperasi dalam model komunikasi berbasis client-server di atas jaringan *Ethernet* yang menggunakan alamat IP untuk mengidentifikasi perangkat dalam jaringan(Siddavatam et al., 2017). Seperti yang dilihat pada gambar 2.7, *client* dapat berupa PLC, PC, atau HMI yang mengirimkan permintaan ke *server*. *Server* kemudian merespon permintaan tersebut dengan data atau perintah yang diminta.

2.2.1 Struktur *Frame* Modbus TCP/IP

Modbus TCP/IP menggunakan format frame yang terdiri dari MBAP *Header* (*Modbus Application Protocol Header*) dan PDU (*Protocol Data Unit*)(Mitsubishi Electric, 2016b). MBAP *Header* terdiri dari 7 byte yang memberikan informasi dasar tentang pesan, seperti panjang data dan identifikasi perangkat.



Gambar 2.8 Struktur Frame Modbus TCP/IP

Sumber: *simplymodbus.ca*

Seperti yang ada pada gambar 2.8, MBAP Header terdiri dari 7 byte yang memberikan informasi penting seperti berikut:

1. Transaction Identifier (2 bytes): Ini adalah identifikasi transaksi yang digunakan untuk mengaitkan permintaan dan respons di client.

2. Protocol Identifier (2 bytes): Nilai ini diatur ke 0 untuk menunjukkan bahwa protokol yang digunakan adalah Modbus.
3. Length (2 bytes): Ini menunjukkan panjang frame Modbus, tidak termasuk MBAP header.
4. Unit Identifier (1 byte): Ini mengidentifikasi unit slave dalam jaringan Modbus.

Sedangkan PDU terdiri dari dua komponen utama yaitu *function code* dan data. *function code* menunjukkan jenis operasi yang akan dilakukan sepetri membaca register atau menulis *coil*. Sementara Data berisi informasi tambahan yang diperlukan, seperti alamat register atau nilai yang akan ditulis.

2.2.2 *Function Code* pada Modbus TCP/IP

Function code adalah bagian penting dalam PDU yang menunjukkan jenis operasi yang akan dilakukan oleh *client* terhadap *server*. Modbus memiliki serangkaian *function code* yang digunakan untuk operasi membaca, menulis, dan memantau data pada perangkat. Untuk penjelasan *function code* yang digunakan beserta penjelasannya, dapat dilihat pada tabel 2.3.

Tabel 2.3 *Function Code* Modbus TCP/IP

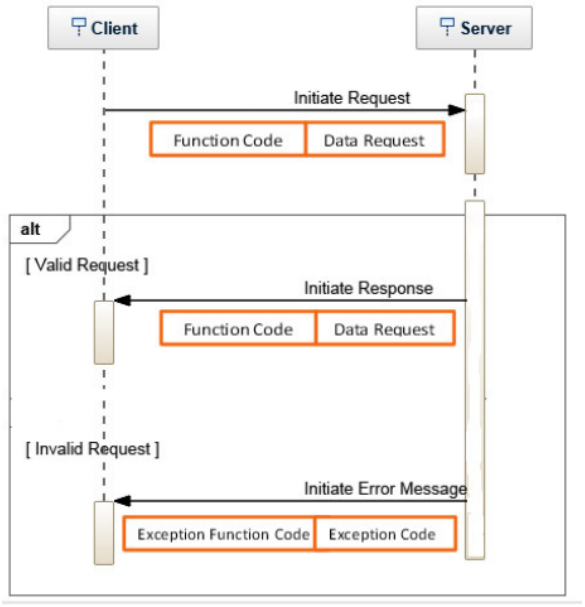
Code	Fungsi	Deskripsi
01	Read Coils	Membaca status satu atau lebih coil (output digital).
02	Read Discrete Inputs	Membaca status satu atau lebih input digital.
03	Read Holding Registers	Membaca nilai dari satu atau lebih register penyimpanan.
04	Read Input Registers	Membaca nilai dari satu atau lebih register input (input analog).

05	Write Single Coil	Menulis nilai *TRUE* atau *FALSE* ke satu coil.
06	Write Single Register	Menulis nilai ke satu register penyimpanan.
15	Write Multiple Coils	Menulis nilai *TRUE* atau *FALSE* ke beberapa coil sekaligus.
16	Write Multiple Registers	Menulis nilai ke beberapa register penyimpanan sekaligus.
23	Read/Write Multiple Registers	Melakukan pembacaan dan penulisan ke beberapa register dalam satu permintaan.

Sumber: Mitsubishi Electric

2.2.3 Proses Komunikasi Modbus TCP/IP

Komunikasi antara client dan server dalam Modbus TCP/IP mengikuti tahapan seperti pada gambar 2.9.



Gambar 2.9 Komunikasi Client dan Server

Sumber: (Irfan A, 2017)

Proses komunikasi diawali dengan *client* yang mengirimkan *request* kepada *server* dengan *function co*de dan alamat register atau *coil* yang akan diakses. Selanjutnya, Server mengirimkan *response* sesuai dengan data yang diminta atau dengan pesan *error* jika terjadi kesalahan karena Modbus TCP/IP dilengkapi dengan metode pengecekan *error* untuk memastikan integritas data.

2.2.4 Error Handling pada Modbus TCP/IP

Jika terjadi kesalahan selama proses komunikasi, server mengirimkan pesan error dengan exception code yang menggambarkan jenis kesalahan. Pesan error ini berguna untuk mendeteksi masalah pada komunikasi atau pada perangkat yang terhubung. Berikut adalah tabel 2.4 untuk pemahaman apa saja fungsi dari setiap code.

Tabel 2.4 Exception Code Modbus TCP/IP

Code	Fungsi	Deskripsi
01	Illegal Function	*Function code* yang dikirim tidak valid atau tidak didukung oleh server.
02	Illegal Data Address	Alamat register atau coil yang diminta oleh client tidak valid.
03	Illegal Data Value Read Holding Registers	Nilai data yang dikirimkan client berada di luar rentang yang diizinkan.
04	Server Device Failure	Kesalahan perangkat di sisi server menyebabkan operasi gagal.

Sumber: Mitsubishi Electric

2.3 Router

Router merupakan perangkat yang berfungsi untuk mengirim dan menerima data antara jaringan. Hal tersebut dilakukann dengan menetapkan alamat IP lokal

untuk masing-masing perangkat yang berada di dalam jaringan, menghubungkan berbagai perangkat yang berada di dalam jaringan dengan meneruskan paket data antar perangkat, dan memastikan bahwa paket data berakhir di tempat yang tepat, dan tidak tersesat di dalam jaringan.(Zou et al., 2022). Router yang digunakan pada penelitian ini adalah Router D-Link yang dapat pada gambar 2.10.

Gambar 2.10 Router D-link

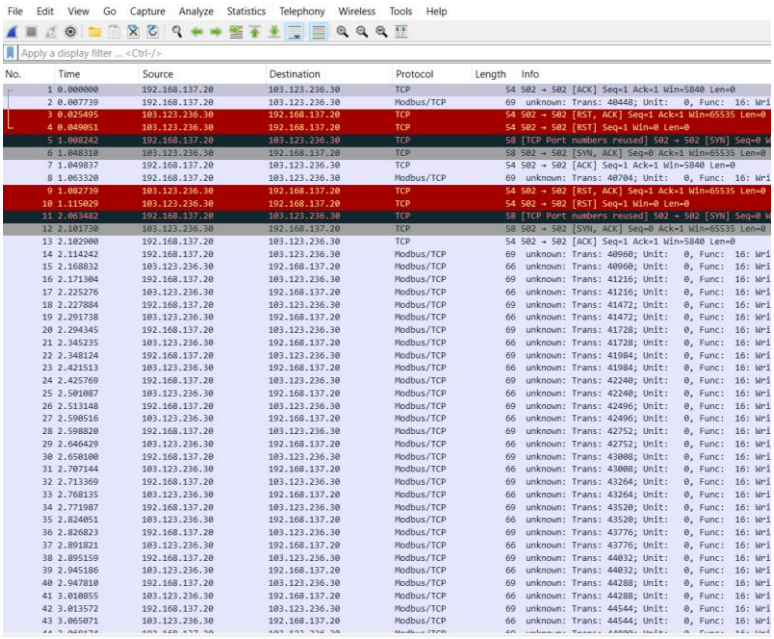
Sumber: Dlink.com

Router juga berfungsi sebagai firewall untuk melindungi jaringan dari serangan *cyber*, menghubungkan jaringan lokal ke internet, dan membantu meneruskan paket data secara efektif ke alamat IP yang diinginkan (Pratomo, 2023).

2.4 Wireshark

Wireshark adalah software analisis protokol jaringan *open source* yang populer digunakan untuk memantau dan menganalisis lalu lintas jaringan secara mendetail.(Xuan & Yongzhong, 2019) . Wireshark memungkinkan pengguna untuk

menangkap dan menampilkan data yang mengalir melalui jaringan komputer secara real-time, sehingga dapat membantu dalam mendiagnosis masalah jaringan, mengidentifikasi potensi gangguan keamanan, atau memahami bagaimana protokol jaringan beroperasi. Tampilan wireshark dapat dilihat pada gambar 2.11.



Gambar 2.11 Tampilan Wireshark

Sumber : (Xuan & Yongzhong, 2019)

Wireshark mendukung berbagai protokol jaringan seperti TCP/IP, UDP, HTTP, Modbus TCP/IP, dan banyak lainnya. Kelebihan itulah yang digunakan untuk berbagai jenis analisis jaringan, termasuk komunikasi Modbus TCP/IP antara PLC dan perangkat lain. Wireshark memiliki beberapa fitur lainnya seperti filter pada tampilan untuk memfokuskan analisis pada jenis paket tertentu, alanisis hasil data secara langsung, dan dekripsi untuk mempelajari paket terenkripsi dalam konteks keamanan jaringan.

2.5 Standar QoS TIPHON

TIPHON (*Telecommunications and Internet Protocol Harmonization Over Networks*) adalah standar yang dikembangkan oleh *European Telecommunications Standards Institute* (ETSI) untuk menetapkan kualitas layanan (*QoS*) dalam jaringan berbasis IP(Wulandari, 2016). Standar TIPHON digunakan sebagai acuan untuk mengukur parameter kualitas jaringan yaitu *delay*, *throughput*, *jitter*, dan *packet loss* dalam sistem komunikasi berbasis IP.

Standar TIPHON dipilih untuk pengujian ini karena menyediakan tolak ukur yang jelas untuk menilai kualitas komunikasi data melalui jaringan berbasis IP. Standar ini memungkinkan peneliti untuk Menilai Kinerja Sistem Dengan mengacu pada standar TIPHON, hasil pengujian dapat dibandingkan dengan kriteria yang diterima secara internasional untuk menilai seberapa baik performa sistem komunikasi PLC yang dirancang.. Pengujian dilaksanakan dengan metode *Quality of Service* (*Qos*). *Qos* merupakan metode pengujian kualitas jaringan, parameter yang diuji berdasarkan Standar TIPHON. Parameter yang diuji adalah Throughput, *Delay* (Latency), *Jitter*, dan *Packet loss*.

2.5.1 *Delay*

Delay adalah waktu yang dibutuhkan oleh paket data untuk berpindah dari pengirim ke penerima. Dalam komunikasi PLC, *delay* berpengaruh untuk menjaga sinkronisasi pada sistem otomatisasi. Pada penelitian ini *delay* yang dipakai merupakan *Round Trip Time* (RTT) *delay*. Karena sistem kerja Modbus TCP adalah *request* dan *response*, maka RTT *delay* dapat dipakai untuk menilai berapa cepat

respon total dari komunikasi. Nilai rata-rata *delay* ini dapat dicari dengan menggunakan persamaan 2.1.

$$Rata - rata\ Delay\ RTT\ (ms) = \frac{Total\ delay\ Round\ Trip\ Time}{Total\ siklus\ request-response} \quad \text{.....(2.1)}$$

Delay harus diminimalisir untuk memastikan komunikasi tetap stabil. Kategori *delay* dapat dilihat pada Tabel 2. 5.

Tabel 2. 5 Kategori *delay*

Kategori	***Delay* (ms)**	**Indeks**
Sangat bagus	< 150	4
Bagus	150 - 300	3
Sedang	>300 - 450	2
Jelek	> 450	1

Sumber: (Nisa et al., 2024)

2.5.2 *Throughput*

Throughput adalah jumlah data yang berhasil dikirimkan dari pengirim ke penerima dalam satu satuan waktu. Dalam penelitian ini, *throughput* yang tinggi memastikan bahwa sistem dapat mengirimkan data dalam jumlah yang cukup tanpa adanya *bottleneck*. *Throughput* menggunakan satuan bits per second (bps) atau bytes per second (Bps). Kategori dari *Throughput* dijelaskan pada Tabel 2. 6.

Tabel 2. 6 Kategori *Throughput*

Kategori	***Throughput* (kbps)**	**Indeks**
Excelent	>2100	4
Good	>1200-2100	3
Poor	>338-1200	2
Bad	0-338	1

Sumber: (Nisa et al., 2024)

Throughput yang tinggi mengindikasikan kinerja jaringan yang stabil. Pengujian throughput bisa menggunakaan rumus pada persamaan 2.2

$$\textit{Throughput (bps)} = \frac{Total\ data\ yang\ ditransfer}{Periode\ pengambilan\ data} \quad \text{.....(2.2)}$$

Dari rumus di atas, total data yang ditransfer merupakan ukuran data yang dikomunikasikan baik dari paket *request* maupun *response*.

2.5.3 *Packet loss*

Packet loss adalah persentase paket data yang hilang atau gagal mencapai tujuan selama transmisi. Dalam konteks komunikasi PLC, kehilangan paket dapat menyebabkan kesalahan dalam pengiriman data yang bisa mengakibatkan kegagalan sistem.

$$\textit{Packet loss}\ (\%) = \frac{Jumlah\ paket\ hilang}{Total\ Paket\ dikirim} \text{x}100\% \text{.....(2.3)}$$

Packet loss yang tinggi dapat menyebabkan penurunan kualitas komunikasi dan penurunan kinerja. Kategori *Packet loss* diperlihatkan pada Tabel 2. 7 berikut.

Tabel 2. 7 Kategori *Packet loss*

Kategori	***Packet loss (%)***	**Indeks**
Sangat bagus	<3%	4
Bagus	3-15%	3
Sedang	>15-25%	2
Jelek	>25%	1

Sumber: (Nisa et al., 2024)

2.5.4 *Jitter*

Jitter adalah variasi dalam waktu pengiriman paket data. Dalam konteks komunikasi antar PLC, *jitter* yang tinggi dapat menyebabkan ketidak konsistenan

dalam pengiriman data, yang berpotensi menyebabkan kesalahan dalam kontrol atau monitoring. Jika jumlah total siklus *request* dan *response* adalah N, maka *Jitter* dapat dihitung menggunakan persamaan 2.4.

$$Jitter\ (ms) = \frac{\sum_{n=2}^{N} |Delay\ RTT_n - Delay\ RTT_{n-1}|}{N-1} \quad \text{.....(2.4)}$$

Jitter banyaknya variasi *delay* pada komunikasi data. Kategori *Jitter* diperlihatkan pada Tabel 2. 8.

Tabel 2. 8 Kategori *Jitter*

Kategori	***Jitter* (ms)**	**Indeks**
Sangat bagus	0	4
Bagus	>0 - 75	3
Sedang	> 75 - 125	2
Jelek	>125 - 255	1

Sumber: (Nisa et al., 2024)

2.6 Penelitian Terkait

Terdapat beberapa penelitian sebelumya yang berkaitan dengan komunikasi antar PLC. Penjelasan penelitian-penelitian terdahulu yang berkaitan dengan komunikasi antar PLC menggunakan protokol Modbus dapat dilihat pada tabel 2.9.

Tabel 2.9 Penelitian Terkait

No.	**Judul Jurnal**	**Penulis, Tahun**	**Pembahasan Jurnal**
1	Rancang Bangun Jaringan Komunikasi Multi PLC dengan	(Iqra Gumilang et al., 2015)	Penelitian ini bertujuan untuk merancang jaringan komunikasi multi PLC dengan platform sistem SCADA-DCS terintegrasi, dengan

No.	Judul Jurnal	Penulis, Tahun	Pembahasan Jurnal
	Platform Sistem SCADA-DCS Terintegrasi		fokus pada kontrol yang optimal dan pamantauan sistem otomasi industri, dengan melalui pemilihan protokol komunikasi dan konfigurasi jaringan yang sesuai, sistem tersebut berhasil memonitor dan mengendalikan beberapa PLC , mengumpulkan data yang akurat pada waktu yang tepat, mempercepat waktu tanggap,dan memungkinkan pengendalian secara terpusat, sehingga memberikan manfaat yang signifikan dalam pengendalian sistem otomasi industri
2	Analisis Kinerja Sistem Komunikasi PLC dengan Raspberry Pi via Protokol Modbus dan Protokol FINS	Hafied Shobin A, 2021	Pada jurnal ini, dilakukan analisis kinerja sistem komunikasi antara PLC dan Raspberry Pi menggunakan protokol Modbus dan FINS berbasis IoT, dengan menemukan bahwa protokol Modbus memberikan kinerja yang baik dalam hal kecepatan dan keandalan, sementara protokol FINS memiliki keunggulan dalam

No.	Judul Jurnal	Penulis, Tahun	Pembahasan Jurnal
	Berbasis Internet of Things (IoT)		penggunaan bandwidth dan latensi, namun tantangan jaringan perlu diperhatikan.
3	Pengontrolan I/O VIA Komunikasi Modbus Master dan Modbus Slave PLC TM221 Berbasis SCADA	(Hamdani, 2020)	Jurnal ini membahas bagaimana mengimplementasikan komunikasi antara Modbus Master dan Modbus Slave untuk mengontrol I/O. Penulis menggunakan SCADA sebagai Modbus Master dan PLC TM221 sebagai Modbus Slave, SCADA memiliki fitur kuat yang unggul dalam memonitor, mengontrol, dan mengumpulkan dara dari PLC TM221. Hubungan komunikasi ini dapat memberikan manfaat dalam proses otomasi industri dengan pengendalian terpusat dan pemantauan yang efektif terhadap perangkat I/O
4	Integrasi Sistem Komunikasi	(Ananda et al., 2023)	Penelitian ini menunjukkan bahwa penggunaan protokol komunikasi

No.	Judul Jurnal	Penulis, Tahun	Pembahasan Jurnal
	Modbus TCP/IP pada PLC Siemens S7-1200, ESP32, dan HMI		Modbus TCP/IP dapat dijadikan sebagai penghubung antara PLC Siemens S7-1200, ESP32, dan HMI sebagai interface. Hasil penelitian ini menunjukkan bahwa integrasi sistem komunikasi Modbus TCP/IP pada PLC Siemens S7-1200, ESP32, dan HMI dapat meningkatkan fungsi kerja sistem otomasi. Dengan adanya koneksi jariangan dan protokol Modbus TCP/IP, pengguna dapat meningkatkan performa sistem otomasi dengan pengoptomalan operasi dan pemantauan perangkat secara terpusat.

Penelitian yang akan dilaksanakan dapat memperoleh manfaat tambahan dan pembandingan yang relevan dengan merujuk pada beberapa penelitian terkait. Berikut adalah daftar penelitian tersebut:

1. Iqra Gumilang et al., "Rancang Bangun Jaringan Komunikasi Multi PLC dengan Platform Sistem SCADA-DCS Terintegrasi,". Penelitian tersebut dapat membantu dalam merancang arsitektur komunikasi yang optimal dan

terintegrasi. Sistem pada penelitian tersebut menghubungkan komunikasi Multi PLC ke platform sistem SCADA, sedangkan pada penelitian ini menghubungkan komunikasi antar PLC saja

2. Hafied Shobin A, "Analisis Kinerja Sistem Komunikasi PLC dengan Raspberry Pi via Protokol Modbus dan Protokol FINS Berbasis Internet of Things (IoT). Penelitian ini dapat membantu dalam menganalisis dan memperbaiki kinerja sistem komunikasi dalam penelitian. Pada penelitian tersebut, protokol yang digunakan adalah Protokol Modbus RTU dan FINS, sedangkan pada penelitian yang akan dilaksanakan menggunakan Protokol Modbus TCP lewat internet.
3. Hamdani, "Pengontrolan I/O VIA Komunikasi Modbus Master dan Modbus Slave PLC TM221 Berbasis SCADA,". Penelitian ini dapat memberikan wawasan tentang implementasi komunikasi Modbus dalam pengontrolan I/O dan dapat memberikan ide untuk mengintegrasikan teknologi tersebut dalam penelitian. Penelitian tersebut memakai PLC TM221 yang dihubungkan dengan SCADA menggunakan protokol Modbus RTU dengan kabel serial. Sedangkan pada penelitian yang akan dilaksakan, sistem menggunakan PLC Omron CP1L-E sebagai Server dan PLC Mitsubishi FX5U sebagai Client dengan protokol Modbus TCP.
4. Ananda et al., "Integrasi Sistem Komunikasi Modbus TCP/IP pada PLC Siemens S7-1200, ESP32, dan HMI,". Penelitian ini dapat membantu dalam merencanakan dan mengimplementasikan integrasi komunikasi yang serupa dalam penelitian. Penelitian tersebut menghubungkan antara PLC Siemens

S7-1200,ESP32,dan HMI, sedangkan pada penelitian ini digunakan PLC Omron CP1L-E dan PLC Mitsubishi FX5U.

Merujuk pada penelitian terkait tersebut, dapat diperoleh manfaat tambahan dalam hal pemahaman tentang rancangan sistem, analisis kinerja, dan integrasi komunikasi. Selain itu juga dapat membandingkan pendekatan, metode, dan hasil dari penelitian terkait dengan penelitian yang akan dilakukan.