



InComTech: Jurnal Telekomunikasi dan Komputer
vol.16, no.2, Agustus 2026, 171-185
<http://publikasi.mercubuana.ac.id/index.php/Incomtech>
P-ISSN: 2085-4811 E-ISSN: 2579-6089

Desain dan Evaluasi Performa Sistem Komunikasi UDP Untuk Interkoneksi PLC pada Aplikasi Framework Open Source

Siti Trilestari^{1*}, Rivira Yuana², Masbah RT Siregar³, Muhammad Nur Al-Majid⁴

^{1,2,3}*Teknik Elektro, Institut Sains dan Teknologi Nasional
Jl. Moch. Kahfi II No.30, Jakarta Selatan 12630, Indonesia*

⁴*Folks Automation*

Jl. Samudera Residence Depok No.C-29, Kota Depok 16412, Indonesia

*Email Penulis Koresponden: sititrilestari56@gmail.com

Abstrak

Perkembangan Industri 4.0 mendorong meningkatnya kebutuhan sistem otomasi industri yang mampu melakukan pertukaran data secara cepat, andal, dan interoperabel antarperangkat. Namun, komunikasi antar PLC dari berbagai vendor masih didominasi oleh protokol proprietary maupun protokol berbasis koneksi yang menimbulkan *overhead* komunikasi dan peningkatan *delay*, sehingga kurang optimal untuk aplikasi *real-time*. Maka, diperlukan sistem komunikasi yang ringan, fleksibel, dan mampu mendukung interkoneksi PLC pada lingkungan multi-vendor. Tujuan riset melakukan perancangan dan evaluasi *framework* komunikasi berbasis UDP menggunakan Node.js yang mendukung komunikasi PLC secara *real-time* pada sistem otomasi industri. *Novelty* riset ini terletak dipengembangan *framework* komunikasi berbasis UDP menggunakan Node.js serta evaluasi kinerja komprehensif melalui perbandingan TCP dan OPC UA berdasarkan parameter QoS, yaitu *delay*, *throughput*, dan *packet loss*. Metode riset adalah eksperimen yang membandingkan kinerja ketiga protokol pada skenario interkoneksi PLC multi-vendor. Hasil riset menunjukkan bahwa *framework* berbasis UDP menghasilkan rerata *delay* terendah yakni 19,8ms, dibandingkan TCP yakni 79,5ms dan OPC UA yakni 33,4ms. Meskipun OPC UA menghasilkan *throughput* tinggi, namun disertai *latency*. *Packet loss* UDP yakni 0,066% masih dibawah ambang batas 1% berdasarkan standar TIPHON, sehingga berkategori sangat baik untuk sebuah aplikasi otomasi industri *real-time*. Hasil riset menunjukkan *framework* berbasis UDP mampu memberikan keseimbangan efektif antara latensi rendah, efisiensi komunikasi, interoperabilitas, dan keandalan, sehingga layak diterapkan sebagai solusi interkoneksi PLC pada sistem otomasi industri *real-time*.

Kata Kunci:

*Komunikasi PLC;
Middleware Open Source;
Quality of Service;
Sistem Otomasi Industri;*

Riwayat Artikel:

Diserahkan 9 Februari 2026
Direvisi 12 Juli 2026
Diterima 30 Juli 2026

DOI:

10.22441/incomtech.v16i2.38054

This is an open access article under the [CC BY-NC](https://creativecommons.org/licenses/by-nc/4.0/) license



1. PENDAHULUAN

Sejalan dengan konsep Industri 4.0, sistem otomasi industri menuntut integrasi antar sistem melalui antarmuka yang terstandarisasi dan interoperabel guna menjamin komunikasi yang andal dan efisien [1]. Perkembangan pesat teknologi otomasi serta integrasi sistem tertanam dalam kendali industri telah meningkatkan kompleksitas sistem otomasi modern [2]. Kondisi ini mendorong kebutuhan akan sistem yang mampu melakukan kontrol dan pemantauan secara *real-time* untuk menjamin kontinuitas proses dan peningkatan produktivitas [3]. Sistem otomasi industri saat ini secara luas mengandalkan PLC sebagai pengendali utama proses industri [4]. Komponen penting dari sistem otomatisasi industri telah diidentifikasi sebagai *Programmable Logic Controller* (PLC), yang dapat digunakan untuk memantau dan mengelola suatu proses [5]. PLC saat ini tidak digunakan secara terpisah tetapi diintegrasikan bersama menggunakan kombinasi produk sehingga memungkinkan integrasi sistem dan kontrol terpusat dalam sistem otomatisasi terdistribusi [6]. Namun demikian, integrasi PLC dari berbagai vendor dalam satu sistem masih menjadi tantangan utama karena perbedaan arsitektur, sistem kendali, serta protokol komunikasi yang umumnya bersifat proprietary [7] [8].

Kondisi tersebut menimbulkan permasalahan interoperabilitas dan ketergantungan terhadap vendor tertentu. Komunikasi antar PLC pada sistem otomasi modern umumnya memanfaatkan jaringan industri berbasis Ethernet karena menawarkan kecepatan tinggi dan kemudahan integrasi dengan teknologi informasi [9]. Akan tetapi, penggunaan perangkat dan pengendali yang heterogen, baik dari sisi jenis PLC maupun protokol komunikasi, meningkatkan kompleksitas integrasi dan pertukaran data antar komponen sistem [10]. Oleh karena itu, diperlukan protokol komunikasi industri yang mampu menjamin interoperabilitas, transparansi, serta kinerja *real-time* yang andal [11]. Pada praktiknya, sebagian besar sistem otomasi industri masih menggunakan protokol komunikasi berbasis *Transmission Control Protocol/Internet Protocol* (TCP/IP), yakni Modbus TCP/IP dan *Open Platform Communications Unified Architecture* (OPC UA), maupun protokol *proprietary vendor*. Meskipun protokol tersebut menyediakan mekanisme keandalan yang baik, proses *handshake*, kontrol kemacetan, dan retransmisi data sering menimbulkan *overhead* komunikasi dan peningkatan *latency*, sehingga kurang optimal untuk aplikasi *real-time* [7]. Pada sistem interkoneksi PLC, yang dimana data yang dikirim bukan file besar melainkan paket-paket kecil yang dikirimkan secara terus-menerus. Sebagai contoh data yang dikirimkan yaitu status sensor, status input/output, nilai register, alarm dan perintah ON/OFF actuator.

Apabila sistem menggunakan jenis protokol TCP, maka setiap paket harus disertai proses *handshake*, *acknowledgement* (ACK) sebagaimana proses retransmisi pada TCP. Sehingga sebagian bandwidth akan digunakan untuk lalu lintas kontrol protokol bukan untuk data proses. Hal ini mengakibatkan bandwidth efektif untuk data PLC menjadi lebih kecil, jumlah paket yang dapat dikirim setiap detik berkurang dan waktu respons sistem meningkat ketika banyak perangkat PLC berkomunikasi secara bersamaan. Sebaliknya, *User Datagram Protocol* (UDP) mengirim paket secara langsung tanpa paket kontrol tambahan, sehingga proporsi *bandwidth* yang digunakan untuk payload data aplikasi menjadi lebih besar. Oleh karena itu, penggunaan bandwidth yang efisien dan *overhead* komunikasi yang rendah menjadi lebih krusial dibandingkan mekanisme reliabilitas TCP. Dengan *overhead* yang lebih kecil, *middleware* dapat memproses lebih banyak paket dalam

waktu yang sama sehingga latensi komunikasi menjadi lebih rendah dan respons sistem tetap terjaga.

UDP merupakan protokol *transport* yang tidak menyediakan mekanisme keamanan bawaan seperti autentikasi, enkripsi, maupun verifikasi integritas data. Oleh karena itu, implementasi sistem komunikasi berbasis UDP pada lingkungan industri memerlukan mekanisme keamanan tambahan untuk mengurangi potensi ancaman siber. Beberapa mekanisme yang dapat diterapkan meliputi segmentasi jaringan menggunakan *Virtual Local Area Network* (VLAN), penggunaan *firewall* industri, penerapan *Virtual Private Network* (VPN), maupun implementasi protokol keamanan (Ipsec) atau *Datagram Transport Layer Security* (DTLS). Selain itu, autentikasi pada tingkat aplikasi dan pembatasan akses terhadap middleware Node.js juga diperlukan untuk mencegah akses tidak sah terhadap sistem kontrol industri. Sehingga, riset ini melakukan observasi dan evaluasi terhadap pemanfaatan mekanisme komunikasi UDP sebagai respons terhadap tantangan sistem otomasi industri yang menuntut komunikasi berlatensi rendah, ringan, dan fleksibel dalam lingkungan PLC *multi-vendor*. Meskipun OPC UA telah banyak diadopsi sebagai solusi standar untuk interoperabilitas industri, beberapa penelitian menunjukkan adanya keterbatasan terkait *overhead* pemrosesan dan manajemen sesi, terutama pada aplikasi *real-time* dengan jumlah klien yang besar [12].

Berdasarkan permasalahan dan temuan penelitian tersebut, hipotesis penelitian ini menyatakan bahwa penerapan kerangka kerja komunikasi berbasis UDP pada sistem otomasi industri mampu memberikan kinerja *latency* yang lebih rendah dibandingkan protokol berbasis koneksi, dengan tingkat *packet loss* yang masih dapat diterima, sehingga layak digunakan untuk mendukung komunikasi PLC secara *real-time*. Beberapa studi menunjukkan bahwa UDP memiliki keunggulan dari sisi *latency* dan efisiensi transmisi data dibandingkan TCP, khususnya untuk aplikasi yang membutuhkan respons waktu nyata [13]. UDP menerapkan mekanisme komunikasi tanpa koneksi dan pengiriman data secara *best-effort*, sehingga mampu mengurangi waktu tunda pengiriman [14]. Karakteristik tersebut menjadikan UDP potensial untuk diterapkan pada sistem otomasi industri *real-time*, selama tingkat kehilangan paket masih berada dalam batas yang dapat diterima. Sehingga hasil pengujian membuktikan hipotesis penelitian ini, yaitu bahwa penggunaan kerangka kerja komunikasi berbasis UDP dengan *middleware* Node.js mampu menghasilkan *latency* yang lebih rendah dibandingkan protokol berbasis koneksi, dengan tingkat *packet loss* yang masih berada dalam batas yang dapat diterima untuk aplikasi otomasi industri *real-time*.

Selanjutnya penelitian yang dilakukan oleh Cibrario Bertolotti et al, menyatakan bahwa implementasi *Modbus over UDP* mampu mengurangi waktu tunda *round-trip* (RT) secara signifikan dibandingkan pendekatan berbasis TCP pada skenario *Industrial Internet of Things* (IIoT) *real-time*, sehingga menunjukkan potensi UDP sebagai protokol transport untuk aplikasi industri *soft real-time*, selama tingkat kehilangan paket masih berada dalam batas yang dapat diterima [15] [16]. Meskipun demikian, dalam praktik industri, pemanfaatan UDP masih didominasi untuk keperluan pertukaran data sederhana yang tidak memerlukan mekanisme kendali transmisi. Akibatnya, penggunaan UDP sebagai protokol utama untuk komunikasi yang terstruktur dan *real-time* dalam sistem otomasi industri masih relatif terbatas. Kondisi ini mengindikasikan bahwa penelitian yang secara khusus mengkaji potensi UDP dalam mendukung komunikasi industri yang andal, terukur,

dan terintegrasi, terutama berdasarkan parameter *Quality of Service* (QoS), masih belum banyak dilakukan. Oleh karena itu, diperlukan kajian lanjutan untuk mengevaluasi kelayakan dan kinerja UDP sebagai alternatif protokol komunikasi pada sistem otomasi industri modern [17]. Berdasarkan permasalahan dan temuan penelitian tersebut, hipotesis penelitian ini menyatakan bahwa penerapan kerangka kerja komunikasi berbasis UDP pada sistem otomasi industri mampu memberikan kinerja latensi yang lebih rendah dibandingkan protokol berbasis koneksi, dengan tingkat packet loss yang masih dapat diterima, sehingga layak digunakan untuk mendukung komunikasi PLC secara *real-time*.

Untuk menjawab tantangan tersebut, riset ini mengusulkan solusi alternatif yakni sistem komunikasi interkoneksi antar PLC yang memanfaatkan teknologi *open-source* [18]. Pendekatan ini memerlukan *middleware open-source* yang ringan dan efisien untuk mendukung komunikasi *real-time*. Node.js dipilih sebagai *middleware* karena merupakan platform *JavaScript* berbasis *event-driven* dan *non-blocking I/O* yang mampu menangani komunikasi jaringan secara asinkron dengan *latency* rendah [19] [20]. Smerpitak et al. menyatakan *framework* berbasis Node-RED *open source* pada sisi *edge* yang mendukung interoperabilitas *netral vendor* antara PLC *heterogen* dan perangkat IoT serta menunjukkan kelayakan dan responsivitas *real-time*. Namun, riset tersebut fokus pada interoperabilitas fungsional dan belum menyajikan evaluasi kinerja komunikasi yang komprehensif, khususnya perbandingan parameter QoS antara protokol UDP, TCP, dan OPC UA untuk interkoneksi PLC [1]. Berdasarkan celah penelitian sebelumnya, maka dapat ditarik hipotesis bahwa penerapan *middleware open-source* berbasis Node.js dengan protokol komunikasi UDP mampu memberikan kinerja QoS yang lebih baik, khususnya dalam hal *latency* yang lebih rendah, dibandingkan dengan protokol TCP dan OPC UA pada sistem interkoneksi PLC *multi-vendor*.

Penelitian ini bertujuan untuk merancang dan mengevaluasi kerangka kerja komunikasi *open-source* berbasis UDP menggunakan *middleware* Node.js untuk interkoneksi PLC dalam sistem otomasi industri. Evaluasi kinerja dilakukan melalui pengukuran parameter QoS, meliputi *delay*, *throughput*, dan *packet loss*. Maka kontribusi riset ini melakukan pendekatan baru yakni perancangan sistem komunikasi data interkoneksi PLC yang ringan, *netral vendor*, serta tetap mempertahankan interoperabilitas dan kemudahan integrasi.

Sistematika penulisan artikel ini disusun sebagai berikut. Bagian 1 menyajikan pendahuluan terkait sistem komunikasi data pada sistem otomasi industri, termasuk pembahasan protokol komunikasi yang umum digunakan serta keterbatasannya dalam mendukung komunikasi *real-time* dan interoperabilitas antar PLC. Pada bagian ini juga diuraikan celah penelitian serta kontribusi yang diusulkan. Bagian 2 menjelaskan metodologi penelitian, meliputi perancangan *framework* komunikasi berbasis UDP menggunakan Node.js sebagai *middleware*, serta skenario pengujian yang diterapkan. Bagian 3 menyajikan hasil pengujian dan analisis kinerja sistem berdasarkan parameter QoS, yaitu *delay*, *throughput*, dan *packet loss*, serta perbandingan performa antara UDP, TCP, dan OPC UA, serta membahas hasil penelitian dan implikasinya terhadap penerapan komunikasi *real-time* pada sistem otomasi industri. Terakhir, Bagian 4 menyimpulkan hasil penelitian dan memberikan arahan untuk pengembangan penelitian selanjutnya.

2. METODE

Untuk mengetahui kualitas komunikasi data yang digunakan, diperlukan suatu metode evaluasi performa jaringan yang terukur dan terstandarisasi. Salah satu pendekatan yang umum digunakan adalah pengukuran QoS, yang meliputi parameter *delay*, *throughput*, dan *packet loss*.

Pemilihan ketiga parameter tersebut mengacu pada rekomendasi standar *Telecommunications and Internet Protocol Harmonization Over Networks* (TIPHON ETSI EG 202 057) dan ITU-T Y.1541 yang menjadikan *delay*, *throughput*, dan *packet loss* sebagai indikator utama kualitas layanan jaringan. Ketiga parameter dipilih karena mampu merepresentasikan tiga aspek penting komunikasi PLC, yaitu kecepatan respons, kapasitas transmisi data, dan keandalan pengiriman paket. Mengingat penelitian ini berfokus pada evaluasi performa komunikasi antar PLC menggunakan *middleware* berbasis UDP, ketiga parameter tersebut dinilai telah memadai untuk mengevaluasi kelayakan system. Parameter-parameter tersebut memberikan gambaran nyata mengenai kemampuan jaringan dalam mendukung kebutuhan sistem industri. Berdasarkan hasil pengukuran parameter tersebut, TIPHON mengelompokkan nilai QoS ke dalam empat kategori kualitas yang ditentukan oleh nilai indeks tertentu. Tabel 3.8 berikut menyajikan kategori nilai standar QoS berdasarkan klasifikasi TIPHON [21]. Berdasarkan hasil pengukuran parameter tersebut, TIPHON mengelompokkan nilai QoS ke dalam empat kategori kualitas yang ditentukan oleh nilai indeks tertentu. Tabel 1 berikut menyajikan kategori nilai standar QoS berdasarkan klasifikasi TIPHON.

Tabel 1. Standarisasi Nilai QoS Berdasarkan TIPHON

No	Nilai	Persentase	Indeks
1	3,8 - 4	95 - 100	Sangat Baik
2	3 - 3,79	75 - 94,75	Baik
3	2 - 2,99	50 - 74,75	Nilai Sedang
4	1 - 1,99	25 - 49,75	Buruk

Evaluasi QoS merupakan tahapan penting untuk memastikan bahwa infrastruktur jaringan mampu memenuhi kebutuhan pengguna secara optimal. QoS dievaluasi melalui pengukuran sejumlah parameter teknis, antara lain *latency*, *packet loss*, dan *throughput*, yang secara langsung memengaruhi kualitas pengalaman pengguna dalam memanfaatkan layanan jaringan. Parameter Pengujian Evaluasi kinerja sistem diukur berdasarkan parameter QoS yang mencakup tiga indikator utama:

1. *Delay (Latency)*: Waktu respons sistem secara *end-to-end*, serta indikator krusial bagi kinerja sistem *real-time* maupun *soft real-time* [22]–[24]. Kategori delay mengacu pada standar yang ditunjukkan pada Tabel 2.

Tabel 2. Kategori Delay Berdasarkan QoS

No	Kategori Delay	Persentase	Indeks
1	Sangat Baik	<150 ms	4
2	Baik	150 ms s.d 300 ms	3
3	Nilai Sedang	300 ms s.d 400 ms	2
4	Buruk	>450 ms	1

2. *Throughput*: Kapasitas rata-rata pengiriman data yang dapat ditangani oleh sistem dalam satuan waktu tertentu [25]–[27]. Kategori throughput berdasarkan standar TIPHON ditunjukkan pada Tabel 3.

Tabel 3. Kategori Throughput Berdasarkan QoS

No	Kategori Throughput	Persentase	Indeks
1	Sangat Baik	100	4
2	Baik	75	3
3	Nilai Sedang	50	2
4	Buruk	<25	1

3. *Packet Loss*: Persentase paket data yang hilang selama proses transmisi, yang digunakan untuk mengukur tingkat keandalan (*reliability*) sistem [28]–[30]. Kategori packet loss berdasarkan standar TIPHON ditunjukkan pada Tabel 4.

Tabel 4. Kategori Packet Loss Berdasarkan QoS

No	Kategori Latensi	Persentase	Indeks
1	Sangat Baik	<1%	4
2	Baik	1-3%	3
3	Nilai Sedang	3-15%	2
4	Buruk	>15%	1

Pengumpulan dan analisis pengumpulan data (*data logging*) dilakukan pada *middleware* Node.js di pusat. Alasan di balik metode pilihan ini adalah karena memberikan analisis ujung ke ujung yang konsisten dan akurat karena merupakan salah satu landasan perantara dalam komunikasi. Data yang diterima dibandingkan untuk memberikan kecukupan protokol UDP sebagai sumber daya komunikasi yang ringan dan *open-source* serta mengidentifikasi manfaat alat tersebut dalam efisiensi presentasi sistem otomatisasi industri baru.

2.1 Mekanisme Komunikasi UDP dan Arsitektur Sistem

Penelitian ini menggunakan UDP sebagai protokol transport untuk mendukung komunikasi tanpa koneksi dengan latensi rendah. Data dikirim dalam bentuk *payload* tingkat aplikasi yang dikemas dalam *header* UDP, sehingga memungkinkan transmisi yang efisien tanpa *overhead* pembentukan koneksi [17]. *Header* UDP terdiri dari beberapa *field* utama yaitu *source port*, *destination port*, *length* dan *checksum*. Identifikasi *header* UDP dilakukan untuk memastikan bahwa paket data yang dikirim dari PLC dapat diterima dan diproses dengan benar oleh *web server*. Proses ini dilakukan dengan memanfaatkan perangkat lunak *Wireshark* untuk menangkap dan menganalisis paket UDP yang melintas di jaringan.

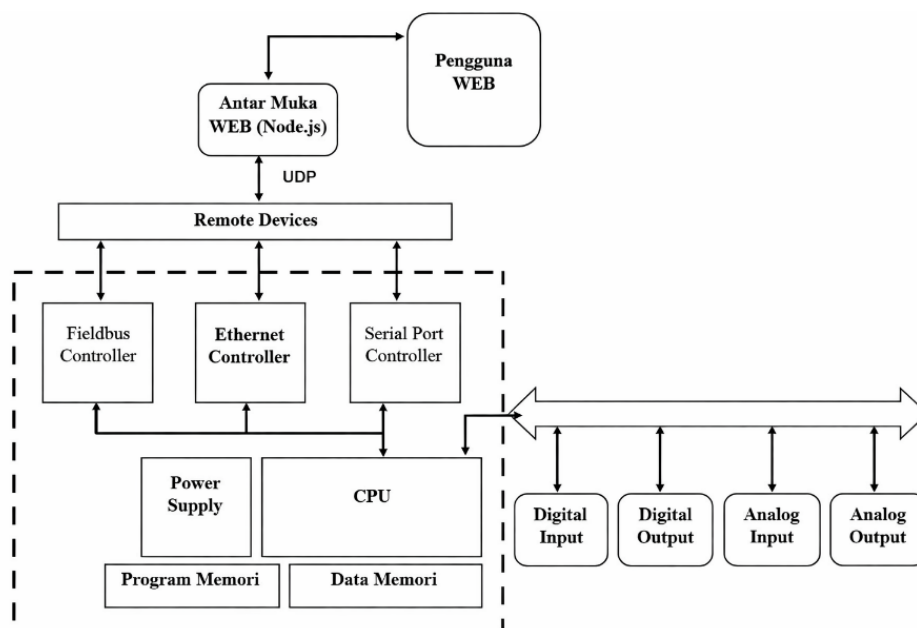
Sistem yang dikembangkan menghubungkan perangkat PLC dengan aplikasi berbasis web melalui jaringan lokal menggunakan protokol UDP. *Framework* komunikasi dirancang berbasis *open-source* dengan memanfaatkan Node.js sebagai *middleware* yang berfungsi sebagai *server* dan *client*, serta menyediakan antarmuka web untuk keperluan *monitoring* dan pengendalian sistem. *Framework* ini menggunakan bahasa JavaScript dengan arsitektur modular. Secara fungsional, sistem menangani pengiriman dan penerimaan paket UDP, pemetaan data PLC ke

format yang dapat diproses aplikasi *web*, serta pengelolaan komunikasi antara *back end* dan *front end*. Diagram blok sistem ditampilkan pada Gambar 1.

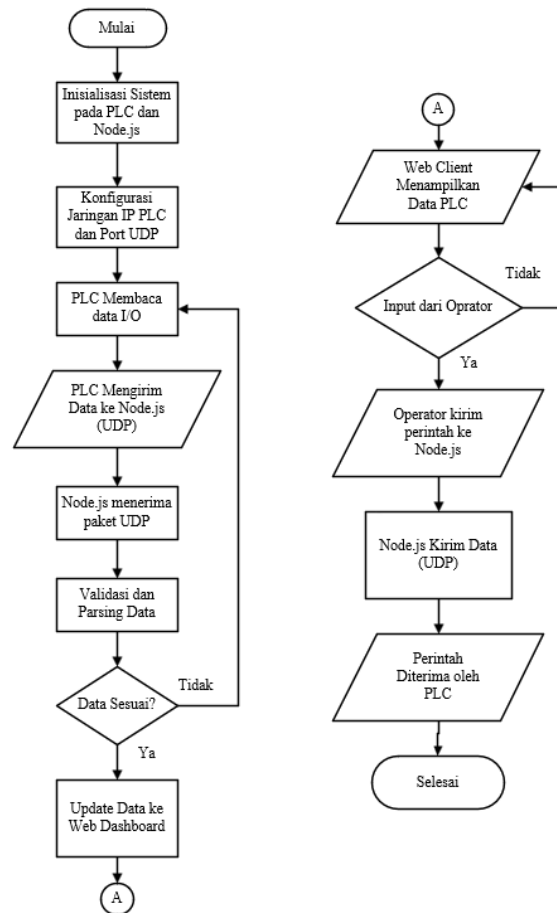
Gambar 2 menunjukkan *flowchart* perancangan sistem yang berfungsi untuk memvisualisasikan alur proses dan mempermudah pemahaman kerja sistem. Perancangan sistem ini terdiri atas dua komponen utama, yaitu *hardware* dan *software*. Pada sisi perangkat keras, sistem menggunakan PLC *Folks Automation* tipe FC100-B IoT yang berperan sebagai pengendali masukan dan keluaran sesuai fungsinya masing-masing. Setiap PLC dikonfigurasi dengan alamat IP berbeda agar dapat dikenali dalam jaringan. Pada sisi perangkat lunak, sistem dirancang untuk melakukan *monitoring* dan pengendalian *status input* dan *output* PLC melalui aplikasi berbasis *web* menggunakan *framework open-source*. Node.js digunakan sebagai server utama yang berfungsi membuka jalur komunikasi berbasis UDP sekaligus menjalankan *web server*. PLC mengirimkan *data status*, seperti nilai *sensor* dan kondisi *register*, ke *server* melalui UDP, kemudian *server* memproses dan memvalidasi data tersebut sebelum menyajikannya pada antarmuka *web*. Pengguna mengakses *web client* melalui *browser* untuk memantau kondisi PLC secara *real-time*. Selain itu, perintah kendali yang dikirimkan oleh operator melalui *web client* akan diteruskan oleh *server* dalam bentuk paket UDP menuju PLC.

2.2. Perancangan Perangkat Lunak

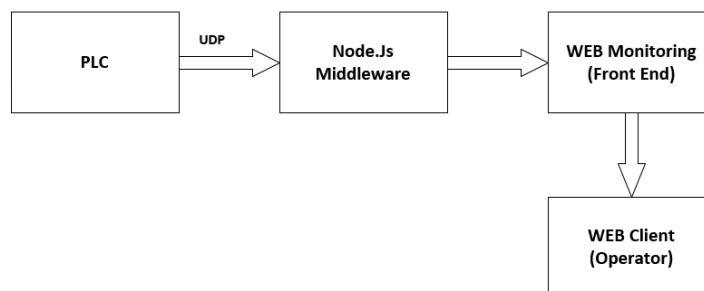
Perancangan *software* difokuskan untuk membangun sebuah *open-source* UDP *communication framework* yang mampu berinterkoneksi dengan PLC. Perancangan dimulai dari konfigurasi komunikasi pada sisi PLC hingga pengembangan *web-based communication* server sebagai penghubung dan *interface system*. Pada tahap perancangan sistem *software*, digunakan dua aplikasi utama yaitu *Codesys* dan *Visual Studio Code* yang dimana untuk aplikasi *codesys* berperan sebagai pembuatan *Network Variable List* (NVL) dan pemrograman pada PLC, kemudian *visual studio code* digunakan sebagai pemrograman *web*.



Gambar 1. Blok Diagram Sistem



Gambar 2. Flowchart Sistem



Gambar 3. Blok Diagram Perancangan Perangkat Lunak

Codesys digunakan sebagai *integrated development environment* (IDE) dalam perancangan perangkat lunak PLC pada penelitian ini. Pemilihan *Codesys* didasarkan pada dukungannya terhadap standar IEC 61131-3 serta kemampuannya dalam mengonfigurasi komunikasi jaringan berbasis *Ethernet* dan UDP. Pada tahap perancangan, *Codesys* digunakan untuk pembuatan struktur program PLC dan pembuatan NVL.

Gambar 3 merupakan diagram blok perancangan perangkat lunak. *Back end* berbasis node.js dirancang sebagai *middleware* yang menjembatani komunikasi

antara PLC dan antarmuka pengguna berbasis *web*. Node.js dipilih karena mendukung arsitektur *event-driven* dan *non-blocking I/O*, sehingga sesuai untuk menangani komunikasi UDP dan pertukaran data secara simultan. Pada perancangan ini digunakan aplikasi Visual Studio Code untuk membuat struktur program *back end* maupun *front end*.

3. HASIL DAN PEMBAHASAN

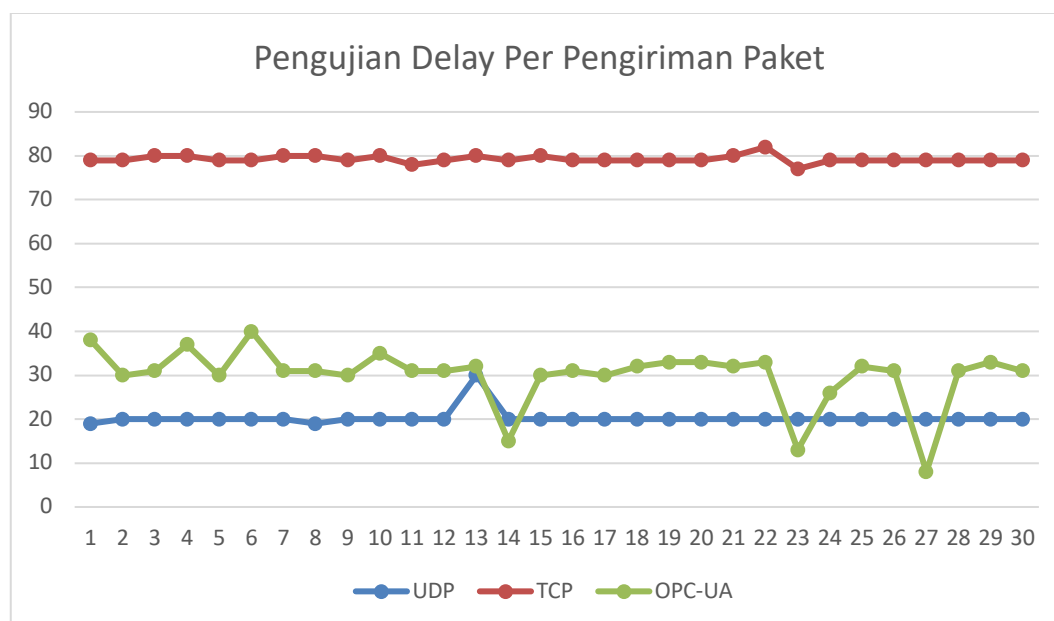
Pengujian sistem dilakukan untuk mengevaluasi kinerja komunikasi data antara PLC dan *web server* berbasis Node.js menggunakan protokol UDP, TCP, dan OPC UA. Evaluasi difokuskan pada parameter QoS, yaitu *delay*, *packet loss*, dan *throughput*, sebagai indikator utama kualitas komunikasi pada sistem otomasi industri. Pengujian dilakukan dengan mengirimkan data proses secara periodik dari PLC ke *server* untuk setiap protokol. Selama pengujian, lalu lintas paket dianalisis menggunakan perangkat lunak *Wireshark* guna memperoleh informasi waktu transmisi, jumlah paket yang diterima, dan volume data yang berhasil dikirimkan.

3.1 Hasil Pengujian Protokol Komunikasi

Parameter *delay*, *packet loss* dan *throughput* diuji berdasarkan rata-rata hasil pengamatan menggunakan aplikasi *wireshark*.

1. Delay

Pengujian *delay* melibatkan proses melakukan transmisi tiga puluh kali ($n = 30$) dari setiap protokol komunikasi yang sedang diuji. Nilai *delay* adalah perbedaan waktu ketika paket data ditransmisikan oleh PLC dan diterima oleh *middleware* Node.js. Gambar 4 menunjukkan grafik hasil pengujian transmisi tiga puluh kali. Persamaan (1) digunakan untuk menghitung penundaan rata-rata dan hasil pengukuran penundaan dari setiap protokol ditabulasi dalam Tabel 1.



Gambar 4. Hasil Pengujian Delay

$$\text{Delay Rata - rata} = \frac{\sum_{i=1}^n \text{Delay } i}{n} \quad (1)$$

Tabel 1. Hasil Pengujian Delay

No	Protokol	Delay (ms)	Indeks	Keterangan
1	UDP	20,26	4	Sangat Bagus
2	TCP	79,24	4	Sangat Bagus
3	OPC-UA	30,03	4	Sangat Bagus

Nilai *delay* yang diperoleh selanjutnya dirata-ratakan untuk menghasilkan nilai *delay* rata-rata selama proses pengujian, sebagaimana disajikan pada Tabel 1. Nilai *delay* yang rendah menunjukkan bahwa sistem komunikasi mampu memberikan respons waktu yang cepat, sehingga mendukung kebutuhan sistem otomasi industri yang bersifat real-time. Selain nilai rata-rata, dilakukan perhitungan standar deviasi delay untuk mengetahui tingkat kestabilan waktu tunda pada masing-masing protokol komunikasi. Standar deviasi dihitung menggunakan Persamaan (2) dengan pendekatan standar deviasi populasi, karena seluruh data hasil pengujian digunakan dalam proses analisis. Perhitungan standar deviasi ini bertujuan untuk mengukur tingkat penyimpangan nilai delay terhadap nilai rata-ratanya, sehingga dapat menggambarkan tingkat kestabilan atau variasi delay yang terjadi pada sistem komunikasi yang diuji.

$$\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^n (d_i - d)^2} \quad (2)$$

Dimana:

σ = standar deviasi delay

d_i = nilai delay pengujian ke- i

d = nilai delay rata-rata

n = jumlah pengujian

Berdasarkan Persamaan (2), diperoleh nilai standar deviasi delay yakni:

1. Standar Deviasi Delay Protokol UDP

$$\sigma_{UDP} = \sqrt{\sum_{i=1}^n \frac{(d_i - d)^2}{n - 1}} = \sqrt{\sum_{i=1}^n \frac{(d_i - 20,26)^2}{30 - 1}} = 1.86 \text{ ms}$$

2. Standar Deviasi Delay Protokol TCP

$$\sigma_{TCP} = \sqrt{\sum_{i=1}^n \frac{(d_i - d)^2}{n - 1}} = \sqrt{\sum_{i=1}^n \frac{(d_i - 79,24)^2}{30 - 1}} = 0.83 \text{ ms}$$

3. Standar Deviasi Delay Protokol OPC-UA

$$\sigma_{OPC\ UA} = \sqrt{\sum_{i=1}^n \frac{(di - d)^2}{n - 1}} = \sqrt{\sum_{i=1}^n \frac{(di - 30,03)^2}{30 - 1}} = 7,38\ ms$$

Berdasarkan hasil perhitungan standar deviasi delay yang telah dilakukan, nilai standar deviasi untuk tiap protokol komunikasi disajikan dalam Tabel 2.

Tabel 2. Hasil Perhitungan Standar Deviasi Delay

Protokol	Delay Rata-rata (ms)	Standar Deviasi (ms)
UDP	20,26	1,86
TCP	79,24	0,83
OPC UA	30,03	7,38

Berdasarkan hasil pengujian dan perhitungan, protokol UDP menghasilkan rata-rata delay paling rendah sebesar 20,27 ms dengan standar deviasi 1,86 ms, yang menunjukkan bahwa UDP mampu memberikan waktu respons yang cepat dengan tingkat variasi delay yang relatif kecil. Kondisi ini mengindikasikan bahwa UDP memiliki keseimbangan yang baik antara kecepatan transmisi dan kestabilan komunikasi, sehingga sesuai untuk aplikasi komunikasi PLC yang membutuhkan respons cepat (*soft real-time*). Di sisi lain, TCP memiliki rata-rata delay paling tinggi, yaitu 79,27 ms, namun menghasilkan standar deviasi paling kecil sebesar 0,83 ms. Nilai tersebut menunjukkan bahwa meskipun waktu pengiriman paket lebih lama akibat mekanisme *connection-oriented*, *acknowledgment*, dan *retransmission*, TCP mampu mempertahankan waktu transmisi yang sangat konsisten selama proses pengujian. Sementara itu, OPC-UA memperoleh rata-rata delay sebesar 29,37 ms dengan standar deviasi 7,38 ms, yang merupakan nilai terbesar di antara ketiga protokol. Hal ini menunjukkan bahwa OPC-UA memiliki variasi delay yang lebih tinggi, yang dipengaruhi oleh mekanisme komunikasi yang lebih kompleks, seperti manajemen sesi, pemodelan informasi, dan layanan keamanan.

2. Packet Loss

Pengujian *packet loss* terjadi selama proses pengiriman 1.500 paket ke sistem untuk menguji stabilitas sistem pada kondisi pengiriman paket yang meningkat dengan beban transmisi yang meningkat. Parameter kehilangan paket dapat didefinisikan sebagai persentase paket data yang diperoleh dan tidak diperoleh oleh sistem selama pengujian berlangsung. Perhitungan nilai kehilangan paket dilakukan dengan bantuan persamaan (2) dan hasil pengukuran kehilangan paket dari setiap protokol komunikasi ditunjukkan pada Tabel 2.

Tabel 2. Hasil Pengujian Packet Loss

No	Protokol	Packet Loss (%)	Indeks	Keterangan
1	UDP	0,066	4	Sangat Bagus
2	TCP	0	4	Sangat Bagus
3	OPC-UA	0	4	Sangat Bagus

Analisis *packet loss* mengacu pada parameter *Quality of Service* (QoS) berdasarkan standar TIPHON, yang mengkategorikan *packet loss* kurang dari 1% sebagai Sangat Baik. Hasil pengujian menunjukkan bahwa protokol UDP memiliki *packet loss* sebesar 0,066%, sedangkan TCP dan OPC-UA sebesar 0%, sehingga seluruh protokol memenuhi kategori Sangat Baik. TCP dan OPC-UA mencapai nilai tersebut karena didukung mekanisme pengendalian kesalahan, sedangkan UDP tetap menunjukkan performa yang baik meskipun tidak memiliki mekanisme *retransmission*. Walaupun *packet loss* UDP sangat kecil, kehilangan paket masih berpotensi memengaruhi aplikasi kendali industri apabila data yang hilang merupakan instruksi kritis, seperti *emergency stop* atau *interlock* keselamatan. Namun, karena pengujian dilakukan pada jaringan LAN yang stabil, tingkat kehilangan paket tersebut tidak memberikan pengaruh signifikan terhadap kinerja sistem. Dengan demikian, sistem komunikasi yang dikembangkan memenuhi standar QoS berdasarkan TIPHON dan layak digunakan untuk interkoneksi PLC pada framework *open-source*.

3. Throughput

Pengujian throughput melibatkan estimasi laju efektif data yang dikomunikasikan yang berhasil dikirimkan oleh protokol komunikasi tertentu dalam interval pengujian. Parameter *throughput* adalah rasio waktu data yang diterima sistem dengan sukses dan waktu transmisi data. Nilai *throughput* dapat diperoleh dengan menggunakan persamaan (3) tetapi hasil dari masing-masing protokol ditunjukkan pada Tabel 3.

$$\text{Throughput} = \frac{\text{Paket Data diterima}}{\text{Lama Pengujian}} \quad (3)$$

Tabel 3. Hasil Pengujian Throughput

No	Protokol	Throughput (bps)	Indeks	Keterangan
1	UDP	11100,090	4	Sangat Bagus
2	TCP	9460,428	4	Sangat Bagus
3	OPC-UA	452830,188	4	Sangat Bagus

Berdasarkan hasil *throughput* serta mengacu standar TIPHON dan parameter QoS. Berdasarkan standar TIPHON, *throughput* lebih dari 100bps dikategorikan sebagai sangat baik. Hasil pengujian menunjukkan bahwa tingkat *throughput* protokol UDP sebesar 11.100,090bps. Hal ini disebabkan oleh karakteristik UDP yang bersifat *connectionless* dan tidak memiliki mekanisme pengendalian koneksi maupun pengiriman ulang paket, sehingga *overhead* komunikasi menjadi lebih kecil dan *bandwidth* jaringan dapat dimanfaatkan secara lebih optimal. Sementara itu, protokol TCP menghasilkan nilai *throughput* paling rendah, yaitu sebesar 9.460,428bps. Nilai ini dipengaruhi oleh mekanisme *connection-oriented* yang diterapkan TCP, seperti pembentukan koneksi, pengurutan paket, serta pengiriman ulang data jika terjadi kesalahan. Kemudian protokol OPC-UA menghasilkan nilai *throughput* paling tinggi dibandingkan dengan UDP dan TCP. Tingginya nilai *throughput* ini dipengaruhi oleh ukuran paket data OPC-UA yang relatif besar serta mekanisme pertukaran data berbasis layanan, sehingga dalam satu kali pengiriman dapat memuat data dalam jumlah yang lebih banyak.

3.2 Evaluasi Performa Protokol Komunikasi

Evaluasi performa sistem komunikasi dilakukan untuk menilai efektivitas protokol UDP dalam mendukung interkoneksi PLC pada aplikasi *framework open source* yang dirancang. Parameter evaluasi mengacu pada QoS dan standar TIPHON, yang meliputi *delay*, *packet loss*, dan *throughput*. Hasil evaluasi juga dibandingkan dengan protokol TCP dan OPC-UA sebagai pembanding performa. Tabel 4 merupakan hasil pengujian keseluruhan.

Tabel 4. Hasil Pengujian Keseluruhan

No	Protokol	Delay	Throughput	Packet Loss	Kelayakan <i>Soft Real-Time</i>
1	UDP	Rendah	Rendah	Sedang	Sangat Layak
2	TCP	Tinggi	Sangat Rendah	Rendah	Kurang Optimal
3	OPC-UA	Sedang	Sangat Rendah	Tinggi	Layak (Non-RT)

Hasil pengujian menunjukkan bahwa protokol UDP menghasilkan nilai *delay* rata-rata terendah sebesar 19,8ms, dibandingkan dengan TCP sebesar 79,5ms dan OPC UA sebesar 33,4ms. Berdasarkan klasifikasi TIPHON, seluruh nilai *delay* termasuk dalam kategori sangat baik (<150ms), namun UDP memberikan respons paling cepat sehingga paling sesuai untuk aplikasi interkoneksi PLC yang menuntut waktu tanggap rendah. Pengujian *packet loss* menunjukkan bahwa UDP mengalami kehilangan paket sebesar 0,066%, sedangkan TCP dan OPC UA tidak mengalami kehilangan paket. Meskipun UDP tidak menerapkan mekanisme pengiriman ulang, nilai *packet loss* tersebut masih berada dalam kategori sangat baik (<1%), sehingga keandalan sistem tetap terjaga. Dari sisi *throughput*, OPC UA menghasilkan nilai tertinggi sebesar 452.830,188bps, diikuti oleh UDP sebesar 11.100,090bps dan TCP sebesar 9.460,428bps. Tingginya *throughput* OPC UA dipengaruhi oleh karakteristik komunikasi berbasis layanan dan ukuran paket, sementara UDP menunjukkan efisiensi *bandwidth* yang lebih baik dibandingkan TCP akibat *overhead* protokol yang lebih rendah. Secara keseluruhan, evaluasi performa menunjukkan bahwa UDP layak digunakan sebagai media komunikasi interkoneksi PLC pada *framework open-source*. Dengan *delay* paling rendah, *throughput* yang efisien, serta *packet loss* dalam batas toleransi standar TIPHON, UDP menawarkan solusi komunikasi yang cepat dan ringan untuk aplikasi industri *real-time*.

4. KESIMPULAN

Berdasarkan hasil pengujian, *framework* komunikasi berbasis UDP yang dikembangkan menggunakan Node.js sebagai *middleware open-source* berhasil mendukung komunikasi antara PLC dan sistem *monitoring* berbasis *web* dalam sistem otomasi industri. Evaluasi kinerja berdasarkan parameter QoS menunjukkan nilai *delay* yang rendah, *packet loss* dalam batas toleransi, serta *throughput* yang stabil, sehingga memenuhi kategori sangat baik menurut standar TIPHON. Hasil ini membuktikan bahwa Node.js efektif digunakan sebagai *middleware* komunikasi PLC secara *real-time* serta memberikan fleksibilitas dan kemudahan integrasi untuk pengembangan sistem otomasi industri berbasis *open-source*. Namun, penelitian ini memiliki keterbatasan karena pengujian masih dilakukan pada skala laboratorium. Oleh karena itu, penelitian selanjutnya disarankan untuk memperluas lingkup pengujian pada skala jaringan yang lebih besar dengan variabel trafik yang lebih

kompleks, serta menambahkan evaluasi terkait masa pakai (*durability*) aplikasi dalam jangka panjang dan evaluasi parameter *jitter*. Dengan demikian, penelitian ini diharapkan dapat menjadi acuan awal bagi pengembangan sistem komunikasi PLC yang lebih fleksibel dan sesuai dengan kebutuhan aplikasi industri tertentu.

REFERENSI

- [1] K. Smerpitak, "Toward Interoperable SCADA : Node-RED-Based Edge Integration of Heterogeneous PLCs and IoT Devices," vol. 18, no. 7, 2025, doi: 10.22266/ijies2025.0831.57.
- [2] B. Ling and C. Chu, "Automatic PLC Control Logic Generation Method Based on SysML System Design Model," 2025.
- [3] N. A. Nurzеха, M. Dwiyaniti, and N. Nadhiroh, "Implementasi SCADA pada Sistem PLC Multivendor Berbasis Modbus TCP / IP Abstrak Pendahuluan Metode Penelitian," vol. 11, pp. 1–8, 2025.
- [4] M. Schreyer and M. M. Tseng, "DESIGN FRAMEWORK OF PLC-BASED CONTROL FOR RECONFIGURABLE MANUFACTURING SYSTEMS".
- [5] E. V. Nascimento and R. V. Aroca, "An Applied Approach for Integrating Legacy PLC-Based Systems into Industry 4 . 0 Environments using Low-Code Platforms," 2026.
- [6] C. K. Ryu, M. C. Lee, I. H. Hong, J. H. Park, J. D. Lee, and S. Y. Choi, "Heterogeneous PLC-Based Distributed Controller with Embedded Logic-Monitoring Blackbox for Real-Time Failover," pp. 1–37, 2025.
- [7] R. Subekti, D. Hadiani, and A. M. Ardaneshwara, "The Indonesian Journal of Computer Science," vol. 13, no. 4, pp. 6669–6685, 2024.
- [8] R. Zheng, S. Zheng, C. Liu, L. Yue, and H. Wu, "A Software-Defined Gateway Architecture with Graphical Protocol Modeling for Industrial Control Systems," pp. 1–28, 2025.
- [9] Y. Mardiana and J. Sahputra, "Analisa Performansi Protokol TCP, UDP dan SCTP Pada Lalu Lintas Multimedia," *J. Media Infotama*, vol. 13, no. 2, pp. 73–84, 2017, doi: 10.37676/jmi.v13i2.455.
- [10] T. A. F. W. Ignatius Deradjad Pranowo, Y. B. Theo Bagastama, "Communication between PLC different vendors using OPC server improved with application device," *Telkomnika*, vol. 18, pp. 1491–1498, 2020.
- [11] G. Gaoyang, C. Zhiyi, F. Qamar, A. Hafizah, and M. Aman, "A Comparative Study of TCP and UDP Performance Using NS-3 Simulation Kajian Perbandingan Prestasi TCP dan UDP Menggunakan Simulasi NS-3," vol. 14, no. 1, pp. 1–19, 2025.
- [12] C. Platforms, M. R. Opc, and K. A. Alaghbari, "Middleware," pp. 1–39, 2022.
- [13] D. Aribowo, A. Marellaningtyas, M. Azamah, M. Rivaldan, and A. Pramayuda, "ANALISIS SINYAL GEDUNG C FKIP UNTIRTA MENGGUNAKAN," vol. 13, no. 2.
- [14] Nabilah Humairah, "Penjelasan tentang Layanan pada Protokol TCP dan UDP," *Penjelasan tentang Layanan pada Protok. TCP dan UDP*, 2017.
- [15] K. Yao, "Industrial Sustainable Development : The Development Trend of Programmable Logic Controller Technology software and digital services ; and 4 . Autonomous systems and the provision of customer services , high customization , responsiven chains , interactive products , and the return of human labor to the upcoming Industry 5 . 0 and from IoT formed by industrial in by big data utilization , it is evident that the process factories have introduced Programmable Logic Controllers (PLCs) in Figure that the PLC serves other traditional integration development the Internet of the sustainable sen- the cu Delphi method the identi fi cation of possib [10]. Given the in fi nite possibilities and limited resources in Fuzzy Analytic Hierarchy Process further priority trends [11 , 12]. The application of technology roadmaps , with t the Fuzzy developmen characteristics , present," 2024.
- [16] S. Zhao *et al.*, "Enhancing Bidirectional Modbus TCP ↔ RTU Gateway Performance : A UDP Mechanism and Markov Chain Approach," pp. 1–27, 2025.
- [17] I. C. Bertolotti, "Rethinking Modbus-UDP for Real-Time IIoT Systems," 2025.
- [18] G. Ramadan, N. F. Efendi, L. P. Yuansyah, and Y. Prastyo, "CLOUD MANUFACTURING ROBOT SPOT WELDING 5 D03 , PLC MITSUBISHI DAN SERVER MENGGUNAKAN PROTOKOL ETHERNET NODE-RED," vol. 3, no. 3, pp. 694–700, 2025.

- [19] A. Ramadhani, N. Iriadi, and R. Hidayat, "Implementasi Teknologi Rest Api Dengan Node Js Untuk Aplikasi Rekomendasi Destinasi Wisata," *Indones. J. Comput. Sci.*, vol. 4, no. 1, pp. 22–29, 2025, doi: 10.31294/t3z8vz27.
- [20] D. Lake, I. For, and I. Workflows, "ISSN No : 2348-9510 International Journal of Core Engineering & Management ISSN No : 2348-9510," no. 10, pp. 216–229, 2020.
- [21] H. R. Oktaseli and A. A. Slameto, "Evaluation of Wireless LAN Quality of Service (QoS) in Primary Education Using TIPHON Standards," vol. 9, no. 2, 2025.
- [22] L. M. Silalahi and A. Kurniawan, "Analisis Keamanan Jaringan Menggunakan Intrusion Prevention System (Ips) Dengan Metode Traffic Behavior," *Electr. J. Rekayasa dan Teknol. Elektro*, vol. 17, no. 1, pp. 71–76, 2023.
- [23] L. M. Silalahi *et al.*, "Application of MPLS Tunnel Service L2TP-VPN Optimization Concept with Traffic Engineering Method for Looping-Protection Service Analysis," *Int. J. Electron. Telecommun.*, vol. 69, no. 1, pp. 115–120, Jul. 2023, doi: 10.24425/ijet.2023.144339.
- [24] I. U. V. Simanjuntak, Heryanto, A. D. Rochendi, and L. M. Silalahi, "Simulation and Analysis of Link Failover Using Routing Border Gateway Protocol (BGP) Multi- Protocol Label Switching (MPLS) Networks," in *2023 International Conference on Radar, Antenna, Microwave, Electronics, and Telecommunications (ICRAMET)*, Nov. 2023, pp. 341–346.
- [25] Y. S. Putri and L. Medriavin Silalahi, "Analysis Performance Long Term Evolution Network on Route of Subway Tunnel Mass Rapid Transit (MRT) Bundaran HI - Senayan," in *2020 International Conference on ICT for Smart Society (ICISS)*, Nov. 2020, pp. 1–6.
- [26] I. U. V. Simanjuntak, J. Haidi, H. Heryanto, and L. M. Silalahi, "Simulation and Performance Analysis of Network Backup Systems Using Hot Standby Router Protocol (HSRP) Method on Real-Time Networks," *Int. J. Electron. Telecommun.*, vol. 69, no. 4, pp. 763–768, Nov. 2023, doi: 10.24425/ijet.2023.147698.
- [27] S. B. dan Kristiani Nahampun dan Freddy Silaban dan Lukman Silalahi dan Fajar R, "Optimalisasi Private Cloud Storage Berbasis Devstack Guna Meningkatkan Performansi Network Function Virtual," *TELKA - J. Telekomun. Elektron. Komputasi dan Kontrol*, vol. 6, no. 1, pp. 1–9, 2020, doi: 10.15575/telka.v6n1.1-9.
- [28] T. Amaliya and L. M. Silalahi, "Analisa Performansi Jaringan LTE (Long Term Evolution) Pada LRT (Light Rail Transit) Rute Velodrom - Pegangsaan Dua," *InComTech J. Telekomun. dan Komput.*, vol. 14, no. 2, pp. 87–97, 2024, doi: <https://dx.doi.org/10.22441/incomtech.v14i2.19024>.
- [29] L. M. Silalahi, P. S. Juwita, and S. Budiyanto, "Analisa Quality of Service Jaringan LTE-A Menggunakan Metode CA-Inter-Band di Wilayah Kiaracandong," *InComTech J. Telekomun. dan Komput.*, vol. 13, no. 3, pp. 209–218, 2023, doi: <https://dx.doi.org/10.22441/incomtech.v13i3.19026>.
- [30] L. M. Silalahi and D. Gunawan, "Routing Architecture: Pioneering Traffic Engineering for Multiprotocol Label Switching and Beyond," in *2025 IEEE 15th Symposium on Computer Applications & Industrial Electronics (ISCAIE)*, IEEE, May 2025, pp. 373–378. doi: 10.1109/ISCAIE64985.2025.11080920.