

Optimisasi SCADA di MPS dengan FINS UDP vs Modbus TCP via Node-RED

Ruminto Subekti¹, Dini Hadiani², Akhtar Muhammad Ardaneshwara³

ruminto@polman-bandung.ac.id¹, dinibhs@polman-bandung.ac.id²,

akhtardaneshwara@gmail.com³

^{1,2,3} Politeknik Manufaktur Bandung

Informasi Artikel

Diterima : 15 Jul 2024

Direvisi : 13 Agu 2024

Disetujui : 29 Agu 2024

Kata Kunci

Integrasi PLC, *Modular Production System*, Node-RED, TCP, UDP

Abstrak

Meningkatnya populasi manusia dan urbanisasi pesat menyebabkan lonjakan kebutuhan produk manufaktur. Hal ini memerlukan sistem kontrol yang lebih kompleks dan terkoneksi digital. Otomasi industri, terutama penggunaan *Programmable Logic Controller* (PLC), menjadi kunci dalam meningkatkan efisiensi dan produktivitas. Namun, integrasi berbagai merek PLC dalam satu sistem menghadapi tantangan karena perbedaan sistem, kendali, dan protokol komunikasi. *Supervisory Control and Data Acquisition* (SCADA) diimplementasikan untuk mengatasi masalah ini dengan menghubungkan perangkat pengendali ke perangkat lunak pengawasan. Penelitian ini bertujuan mengembangkan SCADA berbasis Node-RED untuk memonitor dan mengontrol *Modular Production System* (MPS) dengan integrasi multi brand PLC. Integrasi pada multibrand diterapkan untuk mengimplementasikan dan menguji performa protokol FINS UDP dan Modbus TCP. Hasil pengujian menunjukkan FINS UDP memiliki *delay* lebih rendah (66,85 ms) dan *throughput* lebih tinggi (3706,2 bps) dibandingkan Modbus TCP, yang memiliki *delay* (348,05 ms) dan *throughput* (3353,25 bps).

Keywords

PLC integration, *Modular Production System*, Node-RED, TCP, UDP

Abstract

The increasing human population and rapid urbanization have led to a surge in the demand for manufactured products. This requires more complex and digitally connected control systems. Industrial automation, especially the use of a Programmable Logic Controller (PLC), is key to improve efficiency and productivity. However, the integration of different brands of PLCs in one system faces challenges due to differences in systems, controls, and communication protocols. Supervisory Control and Data Acquisition (SCADA) is implemented to overcome this problem by connecting the controlling device to the supervisory software. This research aims to develop Node-RED based SCADA to monitor and control Modular Production Systems (MPS) with multi brand PLC integration. Multibrand integration is applied to implement and test the performance of FINS UDP and Modbus TCP protocols. The test results show FINS UDP has a lower delay (66.85 ms) and higher throughput (3706.2 bps) than Modbus TCP, which has a delay of (348.05 ms) and a throughput of (3353.25 bps).

A. Pendahuluan

Meningkatnya populasi manusia di dunia dan urbanisasi pesat telah meningkatkan kebutuhan konsumsi dan permintaan produk manufaktur [1]. Penerapan otomasi dalam industri manufaktur menjadi penting untuk meningkatkan efisiensi, produktivitas, dan kualitas produksi [2]. Salah satu teknologi yang berperan penting adalah PLC yang digunakan di berbagai sektor industri, seperti manufaktur, farmasi, dan garmen [3], [4]. Meskipun PLC telah menjadi standar dalam otomasi industri, integrasi PLC dari berbagai merek dalam satu sistem menghadapi tantangan karena setiap PLC memiliki sistem, kendali, dan protokol komunikasi yang berbeda. Hal ini menimbulkan kendala seperti pencatatan data *offline* dan pemantauan kinerja yang memakan waktu [4].

Penelitian sebelumnya telah membahas solusi integrasi PLC dengan menghubungkan perangkat pengendali ke dalam perangkat lunak SCADA [5], [6], [7], [8]. SCADA adalah proses di mana data dikumpulkan dari perangkat fisik yang dipantau dan menjalankan perintah untuk mengontrol perangkat tersebut [9]. Implementasi SCADA dapat dilakukan untuk memonitor dan mengontrol MPS dengan konfigurasi multi brand PLC sebagai rekayasa jalur produksi industri. MPS dapat menyimulasikan keadaan jalur produksi dengan menggunakan komponen atau prinsip kerja berbeda pada setiap *station*-nya [10]. Salah satu contoh dari *station* MPS adalah *separating station* dan *storing station*. Akan tetapi, sistem SCADA yang terhubung menggunakan komunikasi digital tidak luput dari permasalahan, permasalahan yang pada umumnya terjadi ialah masalah *delay* antara perangkat dan pengolah data SCADA [11]. *Delay* akan berdampak signifikan pada kinerja SCADA, mengakibatkan perbedaan pembacaan data antara pengguna dan perangkat lapangan.

Penelitian "*Towards Connected and Autonomous Vehicle HIGHways*" membahas perbedaan performa UDP dan TCP, menunjukkan bahwa UDP memiliki *delay* 22,8% lebih rendah dan *throughput* 184% lebih tinggi dibandingkan TCP [12].

Penelitian ini akan mengembangkan implementasi Node-RED sebagai SCADA. Sistem SCADA akan ditampilkan pada halaman web berbasis *Local Area Network* (LAN), memungkinkan akses kontrol dan monitoring secara *real-time* dari berbagai perangkat. Keunggulan penelitian ini adalah penggunaan protokol UDP yang menunjukkan *delay* lebih rendah dibandingkan TCP, sehingga memberikan performa *real-time* yang lebih baik [12]. Tujuan penelitian ini adalah membuat sistem SCADA berbasis Node-RED untuk memonitoring dan mengontrol MPS dengan integrasi multi brand PLC menggunakan protokol FINS UDP dan Modbus TCP. Penelitian ini diharapkan dapat menjadi referensi sistem SCADA *open source* yang dapat diterapkan di bidang industri, pendidikan, atau masyarakat umum.

B. Metode Penelitian

Pada implementasi ini, digunakan metode pengembangan sistem model *waterfall* sebagai panduan utama dalam proses perancangan sistem. Metode *waterfall* terdiri dari beberapa tahapan, yaitu :

1. Requirement Analysis

Dalam tahap ini, dilakukan analisis dan pengumpulan data yang diperlukan untuk memenuhi persyaratan sistem.

1. Hardware

- MPS *Separating Station* dan *Storing Station*.
- PLC Siemens S7-300 pada *Separating Station* dan PLC Omron CP1L M30DR-A pada *Storing Station*.

2. Software

- TIA Portal V18 sebagai *software* pemrograman PLC Siemens dan CX-Programmer 9.7 sebagai *software* pemrograman PLC Omron.
- Node-RED, untuk mengolah data dari kedua PLC agar dapat dilakukan kontrol dan monitor pada Node-RED *Dashboard*.
- InfluxDB 2.x, sebagai database untuk menyimpan data dari Node-RED.

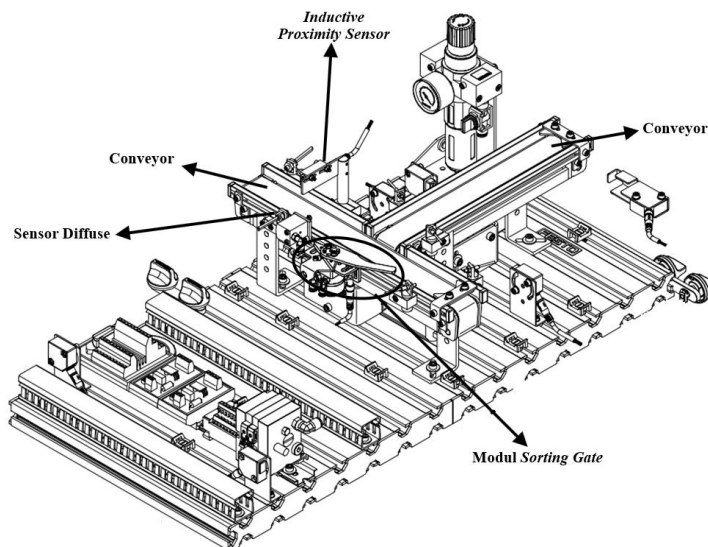
2. System Design

Perancangan desain ini bertujuan untuk memberikan gambaran komprehensif tentang apa yang perlu dilakukan.

1. Modular Production System

• Separating Station

Separating Station adalah salah satu *station* dari MPS yang berfungsi untuk mensortir benda kerja secara otomatis menggunakan sensor warna dan jenis material sebagai parameter penilaian.



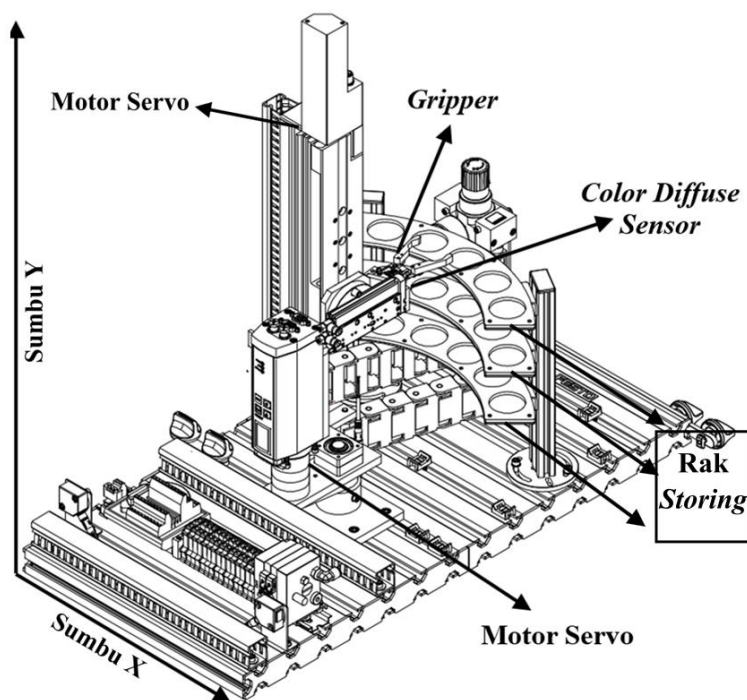
Gambar 1. *Separating Station* (sumber : [Festo.com](https://www.festo.com))

Benda kerja digerakkan oleh *conveyor* ke posisi *Inductive Proximity Sensor* untuk mendeteksi material metalik, sementara sensor diffuse mendeteksi warna benda kerja. Program PLC mengatur

konfigurasi *sorting*, dan benda kerja dipilah menggunakan modul *sorting gate* berbasis pneumatik yang mengubah posisi untuk mengarahkan benda kerja ke *station* berikutnya.

- **Storing Station**

Storing Station adalah *station* yang berfungsi untuk memposisikan benda kerja pada rak *storing* berdasarkan warna benda kerja. Proses dimulai dengan pendeteksian benda kerja di posisi awal, diikuti oleh identifikasi warna menggunakan *Color Diffuse Sensor*. Setelah jenis benda kerja teridentifikasi, benda kerja dipindahkan ke rak *storing* menggunakan dua motor servo yang bergerak pada sumbu X dan Y, dengan bantuan *gripper* pneumatik untuk menggenggam benda kerja.

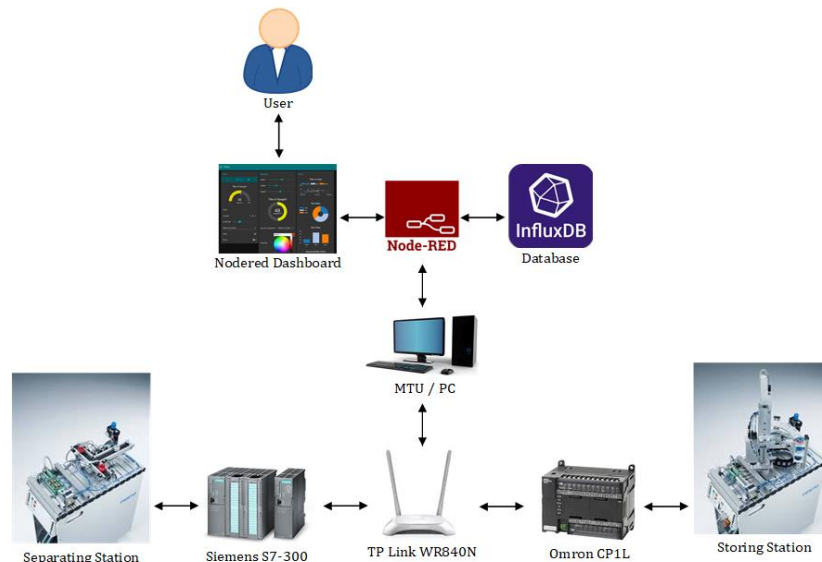


Gambar 2. *Storing Station* (sumber : Festo.com)

Kedua motor servo DC diatur oleh *driver* motor servo Festo yang menyimpan data posisi dan pergerakan. Motor servo menerima *input* dalam bentuk skema data bit, dengan 4 *input* yang merepresentasikan 16 posisi yang dapat disimpan. Lalu terdapat 2 *input* untuk menerima dan mengkonfirmasi data posisi motor servo agar dapat bergerak dengan tepat.

2. Gambaran Umum Sistem

Gambar 3. menampilkan secara umum mengenai implementasi sistem multi brand PLC pada Node-RED sebagai SCADA di MPS. Sistem terdiri dari kedua *station* sebagai *field device*, PLC sebagai *Remote Terminal Unit* (RTU), TP-Link sebagai *hub*. PC sebagai *Master Terminal Unit* (MTU) yang menjalankan Node-RED dan InfluxDB, dan User.



Gambar 3. Gambaran Umum Sistem

Station terdiri dari *separating station* yang menggunakan PLC Siemens S7-300, dan *storing station* yang menggunakan PLC Omron CP1L. Jaringan LAN menghubungkan kedua *station* dan PC menggunakan protokol komunikasi TCP pada Modbus dan UDP pada FINS. Fungsi PC adalah sebagai MTU yang bertanggung jawab untuk memantau dan mengendalikan *station* MPS juga menyimpan data pada database InfluxDB. Sistem pemantauan dan kontrol *station*, serta tampilan data terkait sensor dan aktuator, diimplementasikan melalui antarmuka grafis Node-RED.

3. Perancangan Sistem Pengujian

Dalam mengimplementasikan sistem, beberapa pengujian dilakukan pada sistem, yaitu:

- Pengujian Integrasi PLC dengan Node-RED
Dilakukan untuk menguji pengiriman data antara PLC dan Node-RED guna memastikan protokol komunikasi berfungsi optimal. Performa komunikasi diuji menggunakan Wireshark untuk analisis *delay* dan *throughput*.
- Pengujian Sistem Monitor dan Kontrol pada Node-RED Dashboard
Dilakukan dengan menguji *trigger* pada *input* dan validasi pada *output*. Hal ini bertujuan untuk memastikan fitur *input/output* monitor berfungsi dengan baik.

4. Teknis Analisis Data

• Delay

Delay adalah interval waktu yang diperlukan oleh paket data untuk melakukan perjalanan dari sumber ke tujuan [13]. Faktor-faktor seperti media fisik, jarak, dan waktu proses dapat mempengaruhi *delay* ini [14]. Perhitungan *delay* dapat hitung menggunakan rumus [13], [15] :

$$\text{Delay} = \text{Waktu Paket Pertama} - \text{Waktu Paket Terakhir} \quad (1)$$

Rata-rata *delay* dapat dihitung dengan menggunakan rumus [16]:

$$\text{Rata - rata delay} = \frac{\text{Total Data Delay}}{\text{Jumlah Data Pengujian}} \times 100\% \quad (2)$$

Parameter tingkat *delay* dinilai menggunakan standarisasi nilai *delay* berdasarkan TIPHON (*Telecommunications and Internet Protocol Harmonization Over Network*) [16].

Tabel 1. Standar TIPHON *Delay*

Kategori	Besar <i>Delay</i> (ms)	Indeks
Sangat Bagus	<150	4
Bagus	150 s/d 300	3
Sedang	300 s/d 450	2
Buruk	> 450	1

- *Throughput*

Kecepatan transfer data rata-rata dalam satuan *bits per second* (bps) dikenal sebagai *throughput* [17]. *Throughput* mengukur jumlah data yang dapat dikirim melalui jaringan dalam periode waktu tertentu.

$$\text{Throughput} = \frac{\text{Total Data yang berhasil terkirim}}{\text{Rentan Waktu Pengiriman Data}} \quad (3)$$

Parameter tingkat *throughput* dinilai menggunakan standarisasi nilai berdasarkan TIPHON [16].

Tabel 2. Standar TIPHON *Throughput*

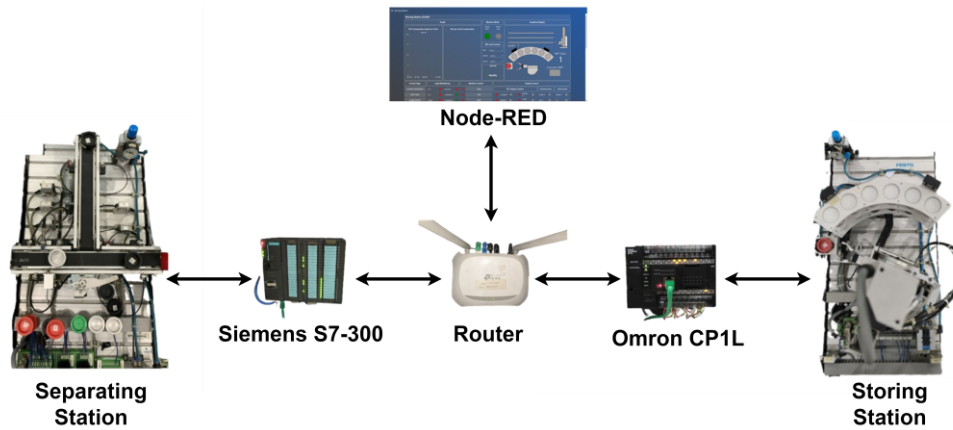
Kategori	<i>Throughput</i> (bps)	Indeks
Sangat Bagus	100	4
Bagus	75	3
Sedang	50	2
Buruk	< 25	1

C. Hasil dan Pembahasan

1. Implementasi Sistem

Dalam penelitian ini, semua perangkat dihubungkan menggunakan topologi jaringan bintang, melibatkan penghubungan perangkat utama seperti PLC Siemens S7-300 dan PLC Omron CP1L, serta *router* TP-Link TL-WR840N sebagai *hub*. Dalam mengimplementasikan sistem, konfigurasi IP pada setiap perangkat dilakukan agar seluruh perangkat dapat terhubung.

Setelah implementasi, fungsi-fungsi yang dikembangkan akan ditampilkan melalui antarmuka Node-RED sebagai SCADA.



Gambar 4. Implementasi Sistem menggunakan Topologi Jaringan Bintang

Implementasi perangkat lunak ditampilkan pada gambar 5, 6, 7, dan 8. Kontrol dan monitoring dapat dilakukan pada halaman SCADA dengan menggunakan *push button* dan *slider* sebagai *input* dan lampu digunakan sebagai indikator terhadap untuk menunjukkan kondisi aktual atau *output* mesin.



Gambar 5. Tampilan SCADA Separating Station 1

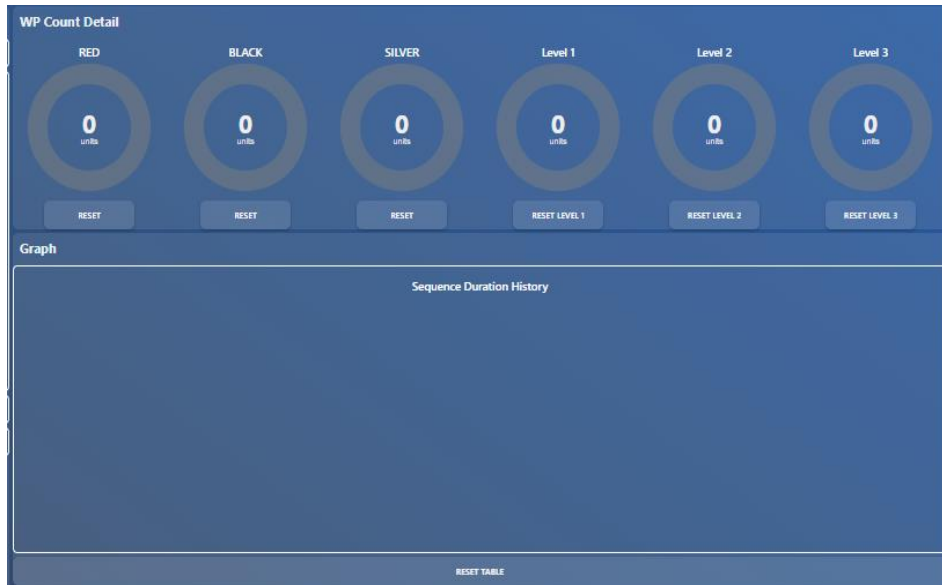


Gambar 6. Tampilan SCADA *Separating Station 2*

Alur benda kerja dapat di konfigurasi pada masing-masing halaman SCADA, pada gambar 5 *separating station* pengguna dapat memilih benda yang ingin di sortir pada menu *Sort WP*, lalu pada gambar 7 *storing station* memungkinkan pengguna memilih benda kerja yang ingin diletakkan pada rak level tertentu pada menu *WP Level Control*. Data grafik dan *gauge* pada gambar 6 dan 8 diakuisisi dari InfluxDB.



Gambar 7. Tampilan SCADA *Storing Station 1*



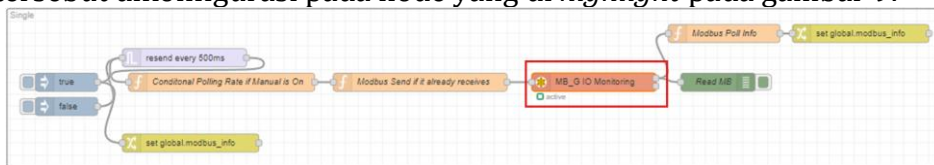
Gambar 8. Tampilan SCADA *Storing Station 2*

2. Hasil Pengujian Integrasi PLC dengan Node-RED

Pada pengujian ini, dilakukan verifikasi *delay* dan *throughput* antara Node-RED yang berjalan pada PC dan PLC Omron serta Siemens yang dihubungkan pada jaringan yang sama. Tujuan pengujian adalah memperoleh durasi pengiriman dan penerimaan data, serta mengetahui besar transfer data yang terjadi antar perangkat.

1. Hasil Pengujian *Delay* pada *Read 1* Alamat Data PLC

Pengujian pembacaan *read* pada 1 alamat data PLC bertujuan untuk mengetahui *delay* minimum pada masing-masing protokol komunikasi. Pengujian dilakukan dengan mengkonfigurasi Node-RED agar membaca hanya 1 alamat data pada masing-masing PLC. Hal tersebut dikonfigurasi pada node yang di *highlight* pada gambar 9.

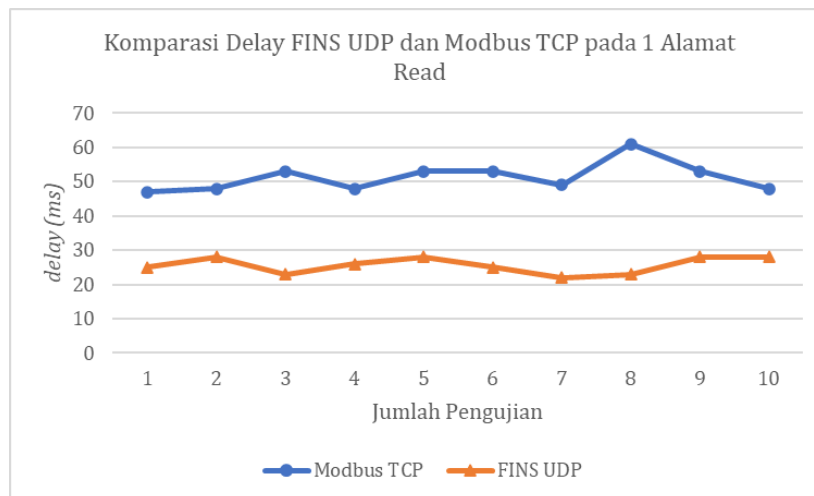


Gambar 9. Konfigurasi Pembacaan 1 Alamat Data pada Modbus

Tabel 3. Hasil Pengujian *Delay* pada *Read 1* Alamat Data PLC

Jumlah Pengujian	Modbus TCP (ms)	FINS UDP (ms)
1	47	25
2	48	28
3	53	23
4	48	26
5	53	28
6	53	25
7	49	22

8	61	23
9	53	28
10	48	28
Rata-rata	51,3	25,6

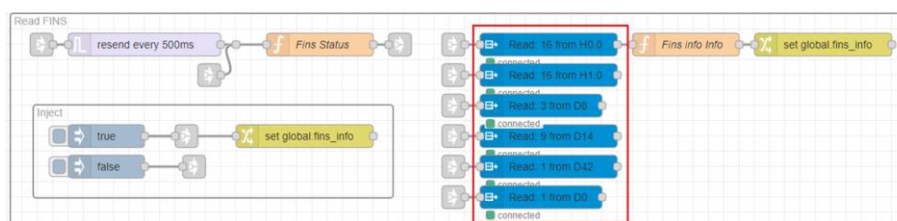


Gambar 10. Komparasi *Delay* Modbus TCP dan FINS UDP pada pembacaan 1 alamat data PLC

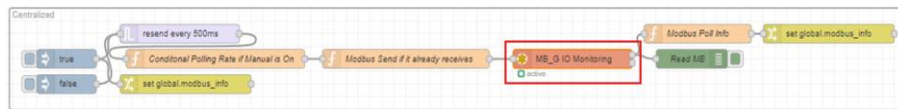
Pada pengujian *delay* pada pembacaan 1 alamat data PLC, terlihat bahwa pada kedua protokol komunikasi terdapat perbedaan rata-rata *delay* sebesar 26ms dimana *delay* pada FINS UDP jauh lebih baik dibandingkan Modbus TCP. Pada Tabel 3 FINS, memiliki *delay* rata-rata sebesar 25,6ms, sedangkan pada Modbus TCP memiliki *delay* rata-rata sebesar 51,3ms. Berdasarkan kedua nilai tersebut maka kedua protokol dapat dikategorikan “Sangat Bagus” menurut standar TIPHON [16] dalam membaca 1 alamat data PLC dengan nilai *delay* (<150 ms).

2. Hasil Pengujian *Delay* pada *Read* Seluruh Alamat Data PLC

Pengujian ini bertujuan untuk mengetahui besar *delay* pada kondisi nyata atau kondisi *delay* aktual ketika kedua protokol diberi beban untuk membaca seluruh data yang dibutuhkan dalam menampilkan kondisi *station*. Terdapat perbedaan mengenai jumlah bacaan data pada masing masing PLC, pada *separating station* dibutuhkan 40 alamat data PLC untuk melakukan monitoring sedangkan pada *storing station* dibutuhkan 46 alamat data PLC.



Gambar 11. Konfigurasi *Semi-Centralized* pada node FINS di nodeRED



Gambar 12. Konfigurasi *Centralized* pada node Modbus di nodeRED

Dalam melakukan riset ditemukan perbedaan performa dalam mengkonfigurasi node pada masing-masing protokol. Perbedaan tersebut terjadi ketika node dikonfigurasi secara *semi-centralized* seperti gambar 11 dan *centralized* gambar 12. Perbedaan konfigurasi tersebut terdapat pada berbedanya jumlah alamat data yang di *request* pada setiap node. *Semi-centralized* membutuhkan 6 node untuk *request* 46 alamat data PLC, sedangkan pada *centralized* membutuhkan 1 node untuk *request* 40 alamat data PLC.

Tabel 4. Hasil Pengujian *Delay* pada *Read* Seluruh Alamat Data PLC menggunakan Ethernet

Jumlah Pengujian	Modbus TCP - Semi-Centralized (ms)	Modbus TCP - Centralized (ms)	FINS UDP - Semi-Centralized (ms)	FINS UDP - Centralized (ms)
1	499	247	66	33
2	526	298	66	36
3	499	296	64	29
4	508	334	69	36
5	516	351	68	35
6	522	371	72	32
7	526	373	65	32
8	500	388	65	32
9	505	418	50	31
10	509	457	64	34
Rata-rata	511	353,3	64,9	33
Selisih	157,7		31,9	
Presentase Penurunan <i>Delay</i>	30,86%		49,15%	

Tabel 5. Hasil Pengujian *Delay* pada *Read* Seluruh Alamat Data PLC menggunakan WIFI

Jumlah Pengujian	Modbus TCP - Semi-Centralized (ms)	Modbus TCP - Centralized (ms)	FINS UDP - Semi-Centralized (ms)	FINS UDP - Centralized (ms)
1	1575	266	62	34
2	1579	283	65	34
3	1568	286	65	38
4	1597	316	63	32
5	1572	320	62	34

6	1568	391	83	30
7	1582	354	68	35
8	1572	391	77	34
9	1605	408	68	30
10	1605	413	75	35
Rata-rata	1582,3	342,8	68,8	33,6
Selisih	1239,5		35,2	
Presentase Penurunan <i>Delay</i>	78,34%		51,60%	

Pada tabel 4 dan 5 terlihat bahwa perbedaan konfigurasi node dapat mengakibatkan perbedaan *delay* yang cukup signifikan terutama bagi protokol Modbus TCP. Pada Tabel 4 dan Tabel 5 bahwa konfigurasi Modbus TCP dengan *Semi-Centralized* memiliki *delay* mencapai 511ms pada jaringan Ethernet dan 1582,3ms pada jaringan Wifi, hal yang sama dapat dilihat pada FINS UDP dengan konfigurasi *semi-centralized* memiliki *delay* yang lebih tinggi dibandingkan dengan *centralized*. Dengan konfigurasi *centralized*, persentase penurunan rata-rata *delay* berdasarkan data tersebut ialah sebesar 54,6% untuk implementasi terhadap Modbus dan 50,16% bagi implementasi FINS.

Tabel 6. Perbandingan Rata-Rata Pengujian *Delay Read* antara FINS dan Modbus

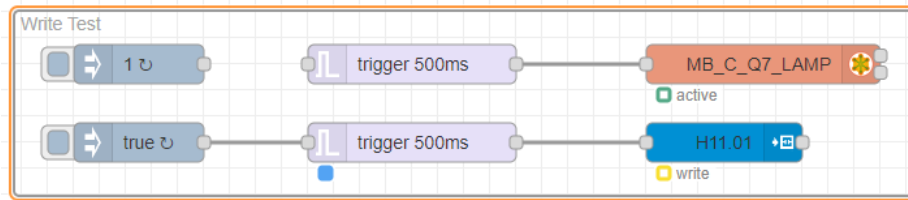
Kategori Jaringan	Rata-rata Modbus TCP - Centralized (ms)	Rata-rata FINS UDP - Semi-Centralized (ms)
Ethernet	353,3	64,9
Wifi	342,8	68,8
Rata-rata <i>Delay</i>	348,05	66,85

Pada sistem, kedua konfigurasi protokol komunikasi digunakan, pada *separating station* digunakan metode *centralized* sedangkan *storing station* digunakan metode *semi-centralized*, metode ini digunakan pada *storing station* dikarenakan program PLC memiliki keterbatasan dalam menyimpan data. Dengan konfigurasi seperti itu terlihat pada tabel 6 bahwa rata-rata *delay* FINS UDP sebesar 66,85ms berada dalam kategori “Sangat Bagus” menurut standar TIPHON [16] dikarenakan memiliki nilai (<150 ms) dan Modbus TCP memiliki rata-rata *delay* sebesar 348,05ms berada dalam kategori “Sedang” dikarenakan memiliki nilai (300 ms s/d 450 ms) berdasarkan standar yang sama.

3. Hasil Pengujian *Delay* pada *Write* 1 Alamat Data PLC

Pengujian *write* pada 1 alamat data PLC bertujuan untuk mengetahui *delay* minimum *write* pada masing-masing protokol komunikasi. Pengujian dilakukan dengan mengkonfigurasi Node-RED

agar mengirim hanya 1 alamat data pada masing-masing PLC. Pengujian *write* akan dilakukan pada jaringan WIFI dan Ethernet.



Gambar 13. Konfigurasi Pengujian *Write* pada Node FINS *Write* dan Modbus *Write* di Node-RED

Tabel 7. Hasil Pengujian *Delay* pada *Write* 1 Alamat Data PLC menggunakan Ethernet

Jumlah Uji	Modbus TCP (ms)	FINS UDP (ms)
1	500	27
2	502	28
3	521	27
4	522	22
5	512	23
6	523	29
7	525	25
8	506	23
9	508	28
10	503	23
Rata-rata	512,2	25,5

Hasil pengujian pada tabel 7 menunjukkan bahwa FINS UDP unggul dalam pengujian *write* 1 alamat data menggunakan Ethernet dengan rata-rata *delay* sebesar 25,5 ms, yang termasuk dalam kategori “Sangat Bagus” menurut standar TIPHON [16] dengan nilai *delay* (<150 ms). Sebaliknya, Modbus TCP menunjukkan rata-rata *delay* sebesar 512,2 ms, yang masuk ke dalam kategori “Buruk” menurut standar TIPHON dengan nilai *delay* (>450 ms).

Tabel 8 Hasil Pengujian *Delay* pada *Write* 1 Alamat Data PLC menggunakan WIFI

Jumlah Uji	Modbus TCP (ms)	FINS UDP (ms)
1	505	25
2	508	29
3	512	27
4	518	29
5	506	26

6	512	32
7	522	38
8	517	27
9	530	27
10	514	28
Rata-rata	514,4	28,8

Hasil pengujian pada tabel 8 menunjukkan bahwa FINS UDP unggul dalam pengujian *write* 1 alamat data menggunakan WiFi dengan rata-rata *delay* sebesar 28,8 ms, yang termasuk dalam kategori “Sangat Bagus” menurut standar TIPHON [16] dengan nilai *delay* (<150 ms). Di sisi lain, Modbus TCP menunjukkan rata-rata *delay* sebesar 514,4 ms, yang masuk ke dalam kategori “Buruk” menurut standar TIPHON dengan nilai *delay* (>450 ms).

Tabel 9. Perbandingan Rata-Rata Pengujian *Delay Write* antara FINS dan Modbus

Kategori Jaringan	Rata-rata Modbus TCP (ms)	Rata-rata FINS UDP (ms)
Ethernet	512,2	25,5
Wifi	514,4	28,8
Rata-rata <i>Delay</i>	513,3	27,15

Hasil pengujian pada tabel 9 menunjukkan bahwa FINS UDP unggul dalam pengujian *write* 1 alamat data dengan rata-rata *delay* sebesar 27,15 ms dengan kategori “Sangat Bagus” menurut standar TIPHON [16] dengan nilai *delay* (<150 ms), dan Modbus TCP dengan *delay write* sebesar 513,3ms masuk kedalam kategori “Buruk” menurut standar TIPHON dengan nilai (>450 ms).

4. Hasil Pengujian *Throughput* pada *Write* dan *Read* 1 Alamat Data PLC

Pengujian ini bertujuan untuk mengetahui besar data yang berhasil di transfer dari satu tempat ke tempat lain dalam jaringan selama periode waktu tertentu.

Tabel 10. Hasil Pengujian *Throughput* pada *Write* 1 Alamat Data PLC

Protokol Komunikasi	Besar Data (KB)	Jumlah Pengujian					Rata-rata
		1	2	3	4	5	
Modbus TCP (bps)	4	3280,5	3316	3298,5	3283	3402	3316
FINS UDP (bps)	4,1	3745	3635	3675	3631	3640	3665,2

Tabel 11. Hasil Pengujian *Throughput* pada *Read* 1 Alamat Data PLC

Protokol Komunikasi	Besar Data	Jumlah Pengujian	Rata-rata
---------------------	------------	------------------	-----------

	(KB)	1	2	3	4	5	
Modbus TCP (bps)	4,5	3398	3406,5	3355,5	3394,5	3398	3390,5
FINS UDP (bps)	4,5	3770,5	3756	3729,5	3756,5	3723,5	3747,2

Pada tabel 10 dan tabel 11 terlihat bahwa pengujian *throughput* menunjukkan bahwa FINS UDP memiliki nilai *throughput* yang lebih tinggi dibanding Modbus TCP dimana FINS UDP memiliki *throughput* sebesar 3665,2 bps pada *write* dan 3747,2bps pada *read* sedangkan Modbus TCP memiliki *throughput* 3316bps pada *write* dan 3390,5bps pada *read*. Pada pengujian *read* terlihat bahwa kedua komunikasi memiliki rata-rata besar data yang sama yaitu sebesar 4,5KB hal ini menunjukkan bahwa *throughput* Modbus TCP yang lebih rendah dibandingkan FINS UDP dipengaruhi *delay* pada Modbus TCP yang lebih besar dibandingkan FINS UDP baik pada *delay write* ataupun *delay read* berdasarkan tabel 9 dan tabel 3.

Tabel 12. Perbandingan Rata-Rata Pengujian *Throughput Write* dan *Read* antara FINS dan Modbus

Protokol Komunikasi	Rata-rata Besar Data (KB)	Rata-rata Pengujian		Rata-rata
		<i>Read</i>	<i>Write</i>	
Modbus TCP (bps)	4,25	3390,5	3316	3353,25
FINS UDP (bps)	4,3	3747,2	3665,2	3706,2

Pada tabel 12, analisis *throughput* menunjukkan bahwa FINS UDP memiliki performa lebih tinggi dibandingkan Modbus TCP. FINS UDP mencatat *throughput write* sebesar 3665,2 bps dan *read* sebesar 3747,2 bps, sedangkan Modbus TCP hanya mencapai 3316 bps pada *write* dan 3390,5 bps pada *read*. Rata-rata *throughput* keseluruhan juga mendukung hasil ini, dengan FINS UDP pada 3706,2 bps dan Modbus TCP pada 3353,25 bps. Meskipun kedua protokol memiliki rata-rata besar data yang hampir sama, perbedaan *Throughput* ini kemungkinan disebabkan oleh *delay* yang lebih rendah pada FINS UDP, menunjukkan efisiensi yang lebih tinggi dalam transmisi data dibandingkan Modbus TCP. Meskipun begitu, kedua protokol berada dalam kategori "Sangat Bagus" menurut standar TIPHON dengan nilai *throughput* (>100 bps).

D. Simpulan

Hasil pengujian menunjukkan bahwa protokol FINS UDP lebih unggul dibandingkan Modbus TCP dalam integrasi SCADA berbasis Node-RED pada MPS. Dari sisi *throughput*, pada tabel 12 FINS UDP memiliki rata-rata *throughput* sebesar 3.706,2 bps, lebih tinggi dibandingkan Modbus TCP yang memiliki rata-rata *throughput* sebesar 3.353,25 bps. Keduanya berada dalam kategori "Sangat Bagus" menurut standar TIPHON (>100 bps) pada

tabel 2. Dalam hal *delay*, FINS UDP menunjukkan performa yang jauh lebih baik dengan rata-rata *delay* yang dapat dilihat pada tabel 6 dan tabel 9 untuk operasi *read* sebesar 66,85 ms dan *write* sebesar 27,15 ms, yang masuk dalam kategori "Sangat Bagus" (<150 ms) pada standar TIPHON di tabel 1. Sebaliknya, Modbus TCP memiliki rata-rata *delay* yang dapat dilihat pada tabel 6 dan 9 dimana untuk operasi *read* sebesar 348,05 ms, yang masuk dalam kategori "Sedang" (300 ms - 450 ms), dan *delay* untuk operasi *write* sebesar 513,3 ms, yang masuk dalam kategori "Buruk". Oleh karena itu, FINS UDP lebih disarankan untuk aplikasi yang memerlukan respons cepat dan transmisi data efisien dalam sistem SCADA.

E. Referensi

- [1] R. Avtar, S. Tripathi, A. K. Aggarwal, and P. Kumar, "Population–Urbanization–Energy Nexus: A Review," *Resources*, vol. 8, no. 3, p. 136, Jul. 2019, doi: 10.3390/resources8030136.
- [2] S. Purboseno, Hermantoro, and Sunardi, "Peran Generasi Millenial Mendorong Percepatan Transformasi Digital Di Industri Perkebunan," *PRO*, vol. 1, no. 1, pp. 37–45, Jul. 2022, doi: 10.55180/pro.v1i1.240.
- [3] A. S. P. Ananda, I. M. Ii Munadhif, I. R. Isa, R. Y. A. Ryan, and R. I. Rini, "Integrasi Sistem Komunikasi Modbus TCP/IP pada PLC Siemens S7-1200, ESP32, dan HMI," *ELK*, vol. 10, no. 2, pp. 234–244, Jul. 2023, doi: 10.33795/elkolind.v10i2.3254.
- [4] M. Yusuf and A. Rohman, "Pengendalian Kecepatan Motor Induksi 3 Fasa menggunakan PLC Omron CP1E dengan Kontrol Proporsional," *JurnalEcotipe*, vol. 7, no. 2, pp. 92–98, Oct. 2020, doi: 10.33019/jurnalecotipe.v7i2.1465.
- [5] M. A. Sehr *et al.*, "Programmable Logic Controllers in the Context of Industry 4.0," *IEEE Trans. Ind. Inf.*, vol. 17, no. 5, pp. 3523–3533, May 2021, doi: 10.1109/TII.2020.3007764.
- [6] D. Ridwan, A. A. Masriwilaga, H. Kabir, and D. Komaludin, "Implementasi Sistem Supervisory Control and Data Acquisition untuk Proses Koagulasi pada Instalasi Pengolahan Air Berbasis Programmable Logic Controller dan Human Machine Interface," *TELEKONTRAN*, vol. 11, no. 1, 2023, doi: DOI:10.34010/telekontran.v11i1.10658.
- [7] Mr. Lavanuru Ashok and sri krishnadevaraya university. anantapur, "Liquid Level Monitoring and Flow based Liquid Distribution System using PLC and SCADA," *IJERT*, vol. V8, no. 12, p. IJERTV8IS120048, Dec. 2019, doi: 10.17577/IJERTV8IS120048.
- [8] H. Wicaksono, P. Santoso, I. Handoyo Putro, I. Surya Hutomo, and P. Alvina, "Towards Integration of Heterogeneous Controllers in an IOT-based Automation System," *E3S Web Conf.*, vol. 188, p. 00009, 2020, doi: 10.1051/e3sconf/202018800009.
- [9] A. Alcayde, I. Robalo, F. G. Montoya, and F. Manzano-Agugliaro, "SCADA System for Online Electrical Engineering Education," *Inventions*, vol. 7, no. 4, p. 115, Dec. 2022, doi: 10.3390/inventions7040115.

- [10] D. Pliatsios, P. Sarigiannidis, T. Lagkas, and A. G. Sarigiannidis, "A Survey on SCADA Systems: Secure Protocols, Incidents, Threats and Tactics," *IEEE Commun. Surv. Tutorials*, vol. 22, no. 3, pp. 1942–1976, 2020, doi: 10.1109/COMST.2020.2987688.
- [11] E. Joelianto, F. Ramdhani, and E. M. Budi, "Analisis Pengaruh Waktu Latensi Terhadap Akurasi Sistem SCADA Bacaan Metering Listrik Waktu Nyata Melalui Jaringan Internet," *JRE*, vol. 16, no. 3, Dec. 2020, doi: 10.17529/jre.v16i3.16465.
- [12] U. Z. A. Hamid and F. Al-Turjman, Eds., *Towards Connected and Autonomous Vehicle Highways: Technical, Security and Social Challenges*. in EAI/Springer Innovations in Communication and Computing. Cham: Springer International Publishing, 2021. doi: 10.1007/978-3-030-66042-0.
- [13] U. Achlison, Joseph Teguh Santoso, Khoirur Rozikin, and Fujiama Diapoldo, "Analisis Latensi Video Streaming Antara Jaringan Berbasis Local Area Network dan Web," *PIXEL*, vol. 15, no. 2, pp. 473–477, Jan. 2023, doi: 10.51903/pixel.v15i2.1037.
- [14] A. R. Maulana, H. Walidainy, and M. Irhamsyah, "Analisis Quality of Service (QoS) Jaringan Internet Pada Website e-Learning Universitas Syiah Kuala Berbasis Wireshark," *KITEKTRO: Jurnal Komputer, Teknologi Informasi, dan Elektro*, vol. 6, no. 2, 2021, doi: doi.org/10.24815/kitektro.v6i2.22284.
- [15] E. J. Pranata and Rizki Dewantara, "Analisis Dan Pengukuran Quality Of Service (Qos) Jaringan 4G (Operator Telkomsel, Xl, Dan Indosat)," *csecurity*, vol. 6, no. 2, pp. 69–75, Feb. 2024, doi: 10.14421/csecurity.2023.6.2.4246.
- [16] A. Alzi and H. Haeruddin, "Pengaruh Manajemen Bandwidth Terhadap QoS dengan Standar TIPHON Pada Alur Monitoring SNMP," *jitika*, vol. 17, no. 1, p. 9, Oct. 2022, doi: 10.32815/jitika.v17i1.883.
- [17] P. Risnaldy and I. Neforawati, "Analisa QOS (Quality of Service) Zeroshell pada Mekanisme Load Balancing dan Failover," *JURNAL MULTIMEDIA NETWORKING INFORMATICS*, vol. 6, no. 1, pp. 8–14, May 2020, doi: 10.32722/multinetics.v6i1.2750.