

# Perbandingan Performa OpenPLC dengan PLC Komersial

Enas Duhri Kusuma<sup>1\*</sup>, Nifwan Arbi Nugroho<sup>1</sup>, Addin Suwastono<sup>1</sup>, Sujoko Sumaryono<sup>1</sup>

<sup>1</sup>Departemen Teknik Elektro dan Teknologi Informasi, Universitas Gadjah Mada, Yogyakarta, Indonesia

\*Email: enas@ugm.ac.id

**Abstract**—PLC konvensional biasanya mahal dan kurang fleksibel. Produsen PLC umumnya mendesain PLC mereka sedemikian rupa sehingga hanya dapat diprogram oleh perangkat lunak yang dibuat oleh mereka sendiri. Hal ini membuat fasilitas pendidikan seperti perguruan tinggi kesulitan dalam melatih operator PLC. Salah satu cara alternatif untuk melatih operator PLC adalah dengan menggunakan perangkat OpenPLC. OpenPLC adalah perangkat PLC *open source* yang relatif murah dan dirancang untuk bekerja seperti PLC konvensional dari tinjauan *hardware* maupun *software*. Karya tulis ini akan membahas perbandingan kinerja antara PLC konvensional dengan sistem OpenPLC. Beberapa aspek kinerja utama yang akan dibahas dalam makalah ini meliputi ukuran memori, waktu *scan*, dan ekspresi logika. Tulisan ini juga akan membahas tentang beberapa fitur yang hanya tersedia di salah satu jenis PLC. Pada bagian akhir kelebihan dan kekurangan dari PLC dan OpenPLC akan dibahas beserta tujuan penggunaannya masing-masing

**Kata Kunci:** Industrial, PLC, OpenPLC, Perbandingan Performa

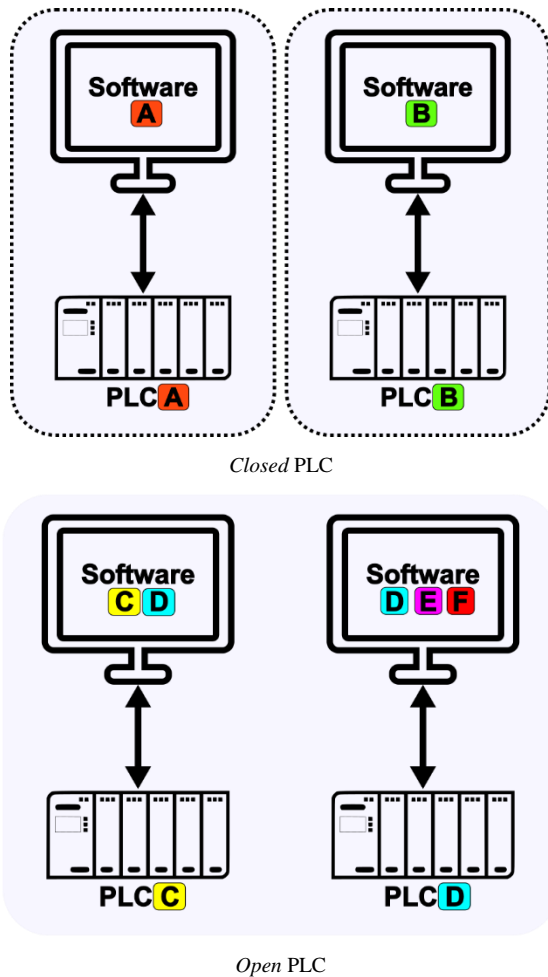
**Abstract**—Conventional PLCs are usually expensive and less flexible. PLC manufacturers generally design their PLCs that they can only be programmed by software they have distributed themselves. This makes it difficult for educational facilities such as universities to train PLC operators. One alternative way to train PLC operators is to use the OpenPLC device. OpenPLC is a relatively inexpensive open source PLC device that is designed to work like a conventional PLC in terms of both hardware and software. This paper will discuss the performance comparison between a conventional PLC and the OpenPLC system. Several points that will be described in this paper include memory size, scan time, and logic expressions. This paper will also discuss several features available in one of the PLC variants. At the end of the section, the advantages and disadvantages of PLC and OpenPLC will be discussed along with their utilizations.

**Keywords**—PLC, OpenPLC, Industrial, Performance comparison

## I. PENDAHULUAN

PADA awal perkembangan industri, pelaku industri biasanya menggunakan kontaktor atau relay untuk membuat proses berjalan secara otomatis [1]. Pada masa tersebut, relay dapat diandalkan dan dapat mengotomatisasi proses sederhana dengan mudah. Namun, relay menjadi kurang dapat diandalkan seiring dengan meningkatnya kerumitan proses industri. PLC diperkenalkan pada tahun 1960 oleh Modicon untuk menggantikan metode konvensional yang menggunakan relay untuk menghasilkan operasi logika. PLC lebih kecil dan lebih ringkas dibandingkan relayautom [2], [3]. Karena PLC tidak memiliki bagian yang bergerak, PLC dapat bekerja lebih cepat jika dibandingkan dengan relay mekanis. Beberapa PLC masih mempunyai beberapa relay namun hanya sebagai perantara keluarannya.

Bagi industri modern, PLC mempunyai peranan penting untuk memastikan suatu mesin dapat bekerja dengan baik dan benar. Proses penggunaan PLC meliputi pembacaan sensor dan pemberian keluaran, biasanya berupa aktuator. Beberapa mesin dilengkapi dengan PLC-nya sendiri sebagai bagian dalam paket pembelian. Hal ini dapat menimbulkan masalah ketika beberapa mesin dan PLC-nya hanya dapat diprogram menggunakan perangkat lunak tertentu yang tidak kompatibel dengan PLC lain. Sebagian besar PLC dan perangkat lunak pemrogramannya memiliki harga yang mahal [4]. Membeli berbagai macam perangkat lunak untuk memprogram beberapa jenis PLC merupakan hal yang tidak efisien dari segi biaya. Berdasarkan keterbukaan software pemrogramannya, PLC yang dapat dikonfigurasi dengan beberapa software disebut open PLC, sedangkan PLC yang hanya dapat diprogram menggunakan satu software tertentu disebut closed PLC [5]. Perbedaan antara sistem open PLC dan sistem closed PLC diilustrasikan pada Gambar 1.



Gambar. 1. Perbandingan antara closed PLC dan open PLC [5]

OpenPLC dibuat oleh Thiago Rodrigues Alves sebagai sistem PLC open source [6]. Perangkat lunak OpenPLC dapat berjalan di berbagai sistem seperti Windows, Mac, Linux, Raspberry Pi, dan bahkan mikrokontroler [7], [8]. OpenPLC dapat meniru cara kerja PLC konvensional dalam hal pemrograman. OpenPLC dapat diprogram menggunakan diagram tangga standar dan bahasa pemrograman tingkat tinggi seperti C [9]. Sistem OpenPLC menawarkan fleksibilitas operasi yang lebih tinggi dibandingkan PLC konvensional. Kinerja OpenPLC bergantung pada perangkat keras yang menjalankannya.

Kinerja antara PLC konvensional dan OpenPLC akan dibandingkan dalam tinjauan literatur ini. Poin-poin kinerja yang akan dibahas dalam artikel ini meliputi ukuran memori, waktu scan, dan ekspresi logika PLC dan OpenPLC. Karya tulis ini bertujuan untuk memberikan pembaca perbandingan kinerja yang komprehensif antara OpenPLC dan PLC konvensional.

## II. METODE

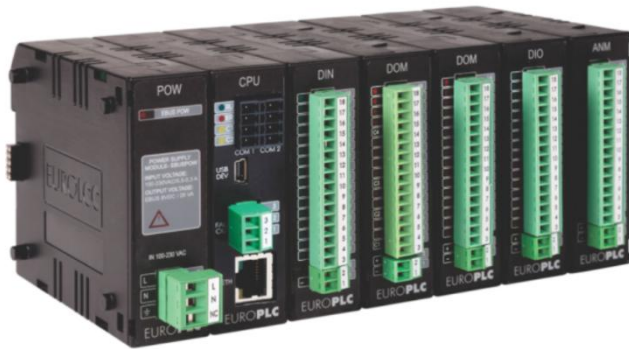
Artikel ini membahas perbandingan antara OpenPLC dan PLC konvensional, berdasarkan spesifikasi yang dikeluarkan perusahaan dan spesifikasi pengendali yang dipasang pada OpenPLC. Sementara itu, pengukuran langsung ke perangkat keras belum dilakukan. Artikel ini juga menyajikan beberapa teknik pengujian perangkat keras OpenPLC yang pernah dilakukan.

Perangkat keras PLC pada umumnya terdiri dari CPU, memori, modul input, modul keluaran, antarmuka komunikasi, dan catu daya. Semua input dan keluaran PLC terisolasi dari sirkuit kontrol untuk memastikan pemisahan antara bagian daya rendah dan daya tinggi. Terminal keluaran PLC dapat dibuat dari relay mekanis atau transistor. Mode keluaran dapat berupa sinking atau sourcing. Dalam mode sinking, arus akan mengalir ke terminal keluaran saat diaktifkan. Sedangkan pada mode sourcing arus akan mengalir keluar dari terminal keluaran ketika diaktifkan. Dalam sistem digital, istilah tersebut bisa dikaitkan mode active high dan active low. Mode sinking dan sourcing PLC juga berlaku untuk input. Mode I/O pada PLC harus dipilih dengan tepat berdasarkan tujuan penggunaannya agar PLC berfungsi dengan benar. Dari segi struktur fisik, PLC tersedia dalam berbagai bentuk. Tipe pertama adalah PLC kotak tunggal atau compact.

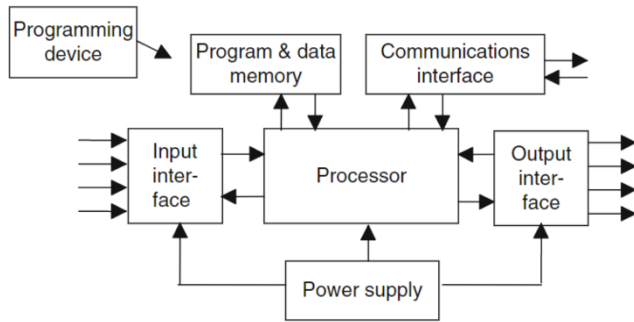
PLC ini merupakan perangkat keras yang dilengkapi dengan komponen lengkap. PLC ini memiliki bagian kontrol, daya, komunikasi, input, dan keluaran dalam satu kotak. Meskipun dimaksudkan untuk dioperasikan sebagai peralatan siap pakai, PLC satu kotak masih dapat ditingkatkan dengan modul pelengkap. Tipe PLC yang kedua adalah PLC modular. PLC tipe ini tersusun atas rak untuk memasang semua komponen PLC seperti CPU, catu daya, komunikasi, dan modul I/O. Jumlah dan jenis modul yang dapat dipasang pada rak PLC berbeda-beda tergantung modelnya.



Gambar. 2. PLC Compact



Gambar. 3. PLC Modular



Gambar. 4. Arsitektur PLC secara umum [10]

Modul I/O pada PLC bisa dioperasikan untuk sinyal analog atau digital. Input yang paling sederhana seperti tombol, saklar limit, atau keluaran relay dapat dibaca oleh PLC menggunakan modul input digital. Input yang lebih kompleks yang melibatkan variasi tegangan atau arus dapat dibaca oleh PLC menggunakan modul input analog. Dalam lingkungan industri, sinyal analog dari sensor dapat berupa tegangan atau arus. Umumnya sinyal dikirim dalam bentuk arus listrik di antara 4 sampai 20 mA. Hal ini dilakukan untuk menghindari drop tegangan di sepanjang jalur transmisi sinyal. Keluaran digital dapat digunakan untuk mengendalikan aktuator sederhana seperti katup solenoid atau lampu. Keluaran analog dapat digunakan untuk mengontrol aktuator yang memiliki kontrol analog seperti motor dan servo.

Semua PLC yang tersedia di pasar memenuhi standar IEC 61131-3. Standar tersebut mendefinisikan setiap aturan yang harus dimiliki oleh setiap PLC termasuk aspek logika, kelistrikan, dan mekanik dari sebuah PLC [11]. Bahasa pemrograman standar PLC sesuai dengan IEC 61131-3 yaitu Ladder Diagram (LD), Function Block Diagram (FBD), Sequential Function Chart (SFC), Structured Text (ST), dan Instruction List (IL). Bahasa pemrograman yang paling umum digunakan dalam PLC adalah bahasa grafis yaitu LD, FBD, dan SFC. Bahasa grafis dalam PLC dirancang sedemikian rupa sehingga operator dengan sedikit pengalaman pemrograman masih dapat memahami program PLC. Diagram tangga pada

PLC meniru logika relay yang digunakan di industri sebelum PLC diperkenalkan. Meskipun semua PLC menggunakan bahasa pemrograman standar yang sama, hampir tidak mungkin untuk memigrasikan satu program dari suatu PLC ke PLC lain dari vendor yang berbeda tanpa melalui perantara

PLC dapat diprogram menggunakan komputer yang telah ter-install dengan perangkat lunak PLC yang sesuai. Program yang dibuat di komputer dapat diunduh ke PLC secara langsung menggunakan USB atau melalui jaringan [12], [13]. Perangkat lunak PLC juga dapat digunakan untuk mensimulasikan logika PLC sebelum mengimplementasikan program ke dalam PLC sebenarnya. Setelah program dijalankan di PLC, operator dapat memantau status PLC termasuk input, keluaran, dan data variabel. Semua variabel termasuk input dan keluaran disebut sebagai “coil”. Kumpulan imajiner yang bukan merupakan masukan atau keluaran dan tidak mempunyai wujud fisik disebut kumpulan memori (memory coil) [14]. Untuk data selain nilai biner seperti bilangan bulat, kumpulan memori khusus yang disebut kumpulan memori kata (word) digunakan.

PLC Komersial memiliki prosesor untuk menangani semua tugas komputasi. Agar dapat berfungsi dengan baik, sistem operasi ditambahkan pada PLC. Sistem operasi PLC memastikan bahwa suatu program dapat diimplementasikan ke dalam I/O fisik. Semua program yang diunduh ke PLC akan berjalan di atas sistem operasi. Ada berbagai sistem operasi PLC yang tersedia. OS yang dipasang di PLC didasarkan pada preferensi produsen. Beberapa OS yang digunakan oleh vendor PLC ditunjukkan pada Tabel 1 [22].

TABEL I  
SISTEM OPERASI BEBERAPA PLC KOMERSIAL

Jenama PLC	Sistem Operasi
Allen-Bradley PLC5	Microware OS-9
Allen-Bradley ControlLogix	VxWorks
Emerson DeltaV	VxWorks
Schneider Modicon Quantum	VxWorks
Yokogawa FA-M3	Linux
Wago 750	Linux
PLC reference platform	QNX Neutrino
Siemens SIMATIC WinAC RTX	Microsoft Windows

OpenPLC adalah PLC open source pertama yang memiliki standar. OpenPLC dikembangkan berdasarkan standar IEC 61131-3. OpenPLC dapat digunakan sebagai pengendali, peralatan otomatis rumah, dan bahan penelitian. Sama seperti sistem PLC biasa, sistem OpenPLC juga terdiri dari perangkat keras dan perangkat lunak. Perangkat lunak OpenPLC berfungsi seperti perangkat lunak PLC biasa namun perangkat keras OpenPLC tidak sama dengan PLC biasa. OpenPLC tidak menyediakan perangkat keras terpisah sebagai produknya.

Walaupun demikian, OpenPLC menyediakan runtime yang dapat di-install di berbagai platform [15]. Beberapa platform yang secara resmi dapat mendukung runtime OpenPLC adalah:

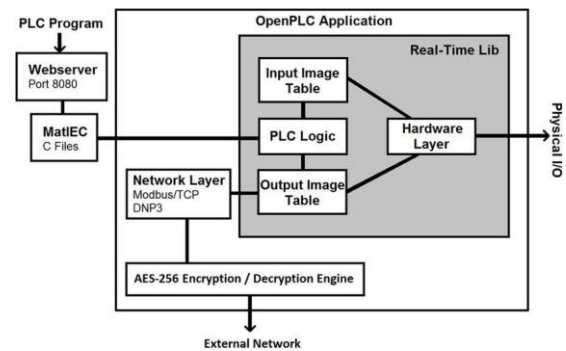
1. Arduino Uno / Nano / Leonardo / Micro
2. Arduino Mega / Due
3. Arduino Nano Every / IoT / BLE
4. Arduino RB2040 Connect
5. Arduino Mkr / Zero / WiFi
6. Arduino Pro
7. Controllino Maxi / Automation / Mega / Mini
8. Productivity Open P1AM
9. ESP8266 (nodemcu)
10. ESP32
11. Raspberry Pi 2 / 3 / 4
12. PiXtend
13. UniPi Industrial Platform
14. Neuron PLC
15. FreeWave Zumlink
16. FreeWave ZumIQ
17. Windows
18. Linux

Runtime OpenPLC juga dapat di-install di semua jenis mikrokontroler yang kompatibel dengan compiler Arduino. Cip lain yang tidak kompatibel dengan Arduino mungkin masih dapat menjalankan OpenPLC tetapi akan sangat sulit karena pengguna harus mengganti Arduino CLI dan banyak komponen lainnya [16].

Karena mikrokontroler tidak mempunyai sistem operasi. Program PLC dan runtime OpenPLC akan dimasukkan ke mikrokontroler secara bersamaan [17]. Karena program di mikrokontroler tidak berjalan di sistem operasi, maka siklus pemindaian program PLC bisa lebih cepat dibandingkan program sejenis yang dijalankan di atas sistem operasi seperti di Raspberry Pi.

Runtime OpenPLC juga dapat di-install di semua jenis mikrokontroler yang kompatibel dengan compiler Arduino. Cip lain yang tidak kompatibel dengan Arduino mungkin masih dapat menjalankan OpenPLC tetapi akan sangat sulit karena pengguna harus mengganti Arduino CLI dan banyak komponen lainnya [16].

Karena mikrokontroler tidak mempunyai sistem operasi. Program PLC dan runtime OpenPLC akan dimasukkan ke mikrokontroler secara bersamaan [17]. Karena program di mikrokontroler tidak berjalan di sistem operasi, maka siklus pemindaian program PLC bisa lebih cepat dibandingkan program sejenis yang dijalankan di atas sistem operasi seperti di Raspberry Pi.



Gambar. 4. Arsitektur OpenPLC Neo [19]

Karena OpenPLC mematuhi standar IEC 61131-3, OpenPLC dapat diprogram menggunakan bahasa pemrograman yang sama dengan yang digunakan pada PLC biasa. Bahasa pemrograman yang dapat digunakan pada OpenPLC meliputi namun tidak terbatas pada [5]:

1. *Ladder Diagram* (LD)
2. *Function Block Diagram* (FBD)
3. *Instruction List* (IL)
4. *Structured Text* (ST)
5. *Sequential Function Chart* (SFC)

Semua pengalamatan di OpenPLC sama dengan PLC biasa, semua alamat memori dimulai dengan simbol persen “%” diikuti dengan prefix lokasi, prefix ukuran, dan satu atau lebih angka. Prefix standar ditunjukkan pada Tabel 2. Alamat fisik perangkat keras OpenPLC telah ditentukan oleh perangkat lunak OpenPLC. Dokumentasi lengkap pemetaan alamat memori dan alamat fisik dapat dilihat pada rilis resmi OpenPLC [18]. Penting untuk diketahui bahwa bit alamat memori paling signifikan (paling kanan) hanya berkisar antara 0 hingga 7. Jika alamat berisi dua angka atau lebih, bit alamat memori paling signifikan (paling kiri) berkisar antara 0 hingga 1023.

TABEL II  
PENGALAMATAN MEMORI OPENPLC [20]

Tipe Prefix	Prefix	Deskripsi
Prefix lokasi	I	Input
	Q	Keluaran
	M	Memori
Prefix ukuran	X	Biner (1 bit)
	B	Byte (8 bits)
	W	Word (16 bits)
	D	Double word (32 bit)
	L	Long word (64 bit)

Komunikasi OpenPLC Modbus bekerja mirip dengan PLC biasa. Perangkat yang dijalankan oleh runtime OpenPLC dapat dioperasikan sebagai slave Modbus. Runtime OpenPLC memungkinkan perangkat OpenPLC membuka komunikasi Modbus melalui saluran TCP atau serial. Slave Modbus OpenPLC dapat dikendalikan oleh master Modbus yang biasanya dijalankan pada komputer utama untuk melakukan pengendalian dan pemantauan. Sistem OpenPLC Modbus dapat

mengirimkan instruksi menggunakan kode fungsi Modbus sebagai berikut:

1. Baca koil output (0x01)
2. Tulis koil output (0x05)
3. Tulis beberapa koil output (0x0F)
4. Baca beberapa kontaktor (0x02)
5. Baca register input (0x04)
6. Baca holding register (0x03)
7. Tulis holding register (0x06)
8. Tulis beberapa holding register (0x10)

Ukuran data dan alamat Modbus dapat berbeda jika dijalankan pada perangkat yang berbeda. Beberapa spesifikasi seperti sistem CPU, memori, dan RAM akan mempengaruhi data Modbus dan ukuran alamat. Sebagai contoh, ukuran data terbesar untuk runtime Windows/Linux adalah 64 bit sedangkan di Arduino Uno ukuran data maksimum adalah 16 bit. Perbandingan antara OpenPLC dan PLC komersial yang dapat disajikan pada makalah ini adalah sebatas pada studi literatur. Karena itu, poin-poin perbandingan pada tulisan ini hanya dibatasi pada teknik pemrograman pada masing-masing jenis PLC dan protokol komunikasi yang dipakai. Adapun hal-hal yang terkait dengan pengujian lapangan dan keandalan, tidak dapat diujikan sekedar dengan studi literatur saja.

### III. HASIL DAN PEMBAHASAN

Perbandingan perangkat keras OpenPLC pertama dilakukan oleh pembuatnya sendiri, Thiago Rodrigues Alves dan tim [19]. Pada penelitian tersebut mereka membuat prototipe perangkat keras yang mirip dengan konstruksi PLC modular. Prototipe terdiri dari bagian-bagian berikut:

- a. Rak modul
- b. Catu daya
- c. CPU
- d. Input
- e. Keluaran

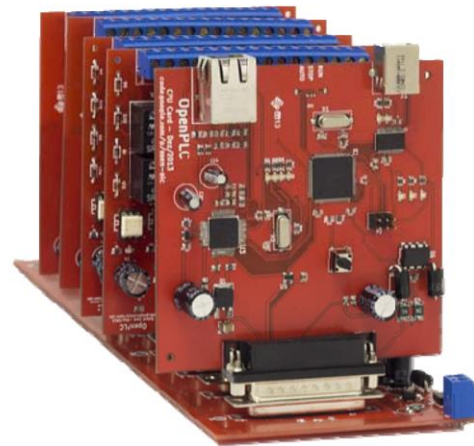
CPU prototipe dibuat dengan basis mikrokontroler Atmega2560, serupa dengan yang digunakan pada Arduino Mega. Semua modul terhubung ke rak atau papan bus dan berkomunikasi menggunakan bus RS-485. Rak didesain sedemikian rupa sehingga setiap modul akan mengetahui posisi fisiknya di dalam rak. Modul input dan keluaran menggunakan Atmega328P untuk menghasilkan keluaran dan menerima input serta untuk berkomunikasi dengan modul lain di rak bus.

Sebuah studi [19] membandingkan perangkat keras prototipe OpenPLC dengan PLC Siemens S7-200 komersial. Pengujian yang dilakukan pada penelitian ini hanya membandingkan ekspresi logika masing-masing PLC. Pengujian ini melibatkan 13 input dan 10 keluaran. Bug yang sama ditemukan pada kedua

PLC namun setelah program diperbaiki, kedua PLC dapat beroperasi dengan baik.

Studi kedua [20] yang dilakukan oleh Alves dan Morris menguji lebih banyak fitur OpenPLC dibandingkan penelitian sebelumnya. Pada penelitian ini mereka melakukan empat pengujian untuk mengevaluasi waktu siklus scan, eksekusi ladder logic, konektivitas SCADA, dan respons terhadap serangan injeksi Modbus. Perangkat keras yang dibandingkan dalam penelitian ini adalah OpenPLC Neo, Schneider M221, Siemens S7-1214C, Omron CP1L-L20DR-D, dan Allen-Bradly Micro Logix 1400. OpenPLC Neo merupakan perangkat keras yang dikembangkan oleh komunitas OpenPLC yang bentuknya mirip dengan PLC biasa. Arsitektur OpenPLC Neo dapat dilihat pada Gambar 5 [19]. Sementara itu, secara fisik, purwarupa hardware OpenPLC baru dibuat dalam bentuk slot yang terhubung ke mainboard, tanpa enclosure seperti ditunjukkan pada Gambar 6. Perangkat OpenPLC Neo masih belum tersedia di pasaran. Penelitian ini mengklaim bahwa arsitektur OpenPLC Neo lebih baik daripada PLC lainnya karena prosesor performa tinggi Allwinner H3.

Tes waktu scan pertama dilakukan menggunakan program inversi I/O sederhana. Keluaran terminal dibalik pada setiap pemindaian. OpenPLC Neo mengungguli setiap PLC lainnya. Waktu scan tercepat dicapai oleh Open PLC Neo di angka 1,00 ms sedangkan waktu scan terlama dicapai oleh Omron di angka 11,06 ms.



Gambar 5. Prototipe Pertama OpenPLC [22]

Tes logika [19] yang dilakukan melibatkan empat program LD. Program yang diuji pada semua PLC adalah bistable, timer, counter, dan logika aritmatika. Semua PLC diuji oleh mikrokontroler yang menghasilkan vektor masukan dan merekam keluarannya. Hasil pengujian menunjukkan bahwa beberapa PLC dapat menghasilkan keluaran yang berbeda dengan PLC lainnya. Studi tersebut mengatakan bahwa beberapa PLC dapat menghasilkan keluaran yang berbeda karena perbedaan standar yang diterapkan pada setiap PLC.

Uji konektivitas SCADA menunjukkan bahwa OpenPLC mencapai kinerja serupa dengan PLC konvensional. Hanya

konfigurasi kecil di setiap PLC yang mengakibatkan perbedaan kecil antara perilaku PLC. Serangan injeksi Modbus menunjukkan bahwa hanya PLC Siemens yang tahan terhadap serangan tersebut. Namun karena OpenPLC adalah perangkat lunak open source, peneliti dapat melakukan konfigurasi ulang perangkat lunak OpenPLC agar mampu meniru perilaku PLC Siemens. Hasil perubahan tersebut menunjukkan bahwa OpenPLC menunjukkan perilaku yang sama seperti PLC Siemens, OpenPLC Neo menjadi kebal terhadap serangan injeksi Modbus.

Studi terbaru yang secara eksplisit membandingkan kinerja OpenPLC dengan PLC juga pernah disusun [21]. Penelitian ini membandingkan kinerja kecepatan OpenPLC dan PLC biasa. Perangkat keras yang digunakan untuk mengimplementasikan OpenPLC pada penelitian ini adalah Arduino Uno, Arduino Mega, dan ESP8266. Representasi PLC reguler diwakili oleh Wago PFC200 yang diprogram dengan CODESYS. Pada penelitian ini terdapat empat program yang dibuat untuk menguji seluruh perangkat keras PLC. Hasil penelitian ini menunjukkan bahwa semua MCU memiliki waktu scan yang lebih cepat dibandingkan WAGO PCF200. Penelitian ini menunjukkan bahwa waktu pemindaian seluruh MCU meningkat seiring dengan meningkatnya beban komputasi. Walaupun waktu pemindaian seluruh MCU meningkat, waktu tersebut masih lebih cepat dibandingkan PCF200. PCF200 memindai lebih lambat namun lebih stabil ketika beban komputasi meningkat. Satu-satunya faktor yang dapat mempengaruhi waktu pemindaian PCF200 secara signifikan adalah beban bus CANopen-nya. Penelitian ini mengatakan bahwa PCF200 memiliki kinerja yang lambat namun lebih stabil karena programnya berjalan di atas sistem operasi.

Melalui metode studi literatur yang ditempuh pada penyusunan makalah ini, didapat beberapa cara pengujian yang dapat diterapkan untuk menguji suatu PLC, mulai dari level pengaksesan I/O, eksekusi program, sampai respon terhadap serangan pada jaringan Modbus. Beberapa poin yang didapat barulah berupa penjelasan mengenai hal-hal yang dapat diujikan pada suatu PLC dan bukan berupa hasil pengujian langsung, karena metode yang dipakai baru pada studi literatur. Walaupun begitu, poin-poin sasaran pengujian ini sangat diperlukan untuk dijadikan peta jalan sebagai modal pengujian lapangan.

#### IV. KESIMPULAN

Artikel ini membahas perbandingan OpenPLC dan PLC komersial berdasarkan spesifikasi produk dan beberapa literatur yang membahas keduanya. OpenPLC dapat diimplementasikan ke arsitektur perangkat keras yang beragam. Oleh karena itu, kinerja OpenPLC sangat bergantung pada spesifikasi perangkat kerasnya. OpenPLC yang diimplementasikan pada mikrokontroler sederhana dapat berjalan lebih cepat namun rentan terhadap beban komputasi. OpenPLC yang berjalan pada sistem operasi cenderung memiliki waktu scan yang lebih

lambat namun lebih tahan terhadap beban komputasi. Perangkat lunak OpenPLC dapat disesuaikan untuk memenuhi kebutuhan pengguna. OpenPLC dirancang dengan tujuan penggunaannya dapat mengakses memori di dalamnya, tanpa ada batasan tertentu yang ada pada PLC konvensional. Perancangan OpenPLC mengikuti standar IEC 61131—3 sehingga dapat mewakili.

#### REFERENSI

- [1] M. Wang, "Application of PLC Technology in Electrical Engineering and Automation Control," 2021, hlm. 131–135. doi: 10.1007/978-3-030-51556-0\_20.
- [2] B. Ahmad, G. Goswami, dan M. F. Khan, "Implementation of Automatic Excitation Initiator Circuit using Microcontroller based PLC for Standalone Induction Generator," *Water And Energy International*, vol. 65, no. 9, hlm. 29–36, 2022.
- [3] G. Mahalakshmi, S. Sangeetha, Maladhi, dan T. Bhavatharini, "Automation of industrial drives using PLC and SCADA," 2022, hlm. 040009. doi: 10.1063/5.0108098.
- [4] C.-C. Andrei, G. Tudor, M. Arhip-Calin, G. Fierascu, dan C. Urcan, "Raspberry Pi, an Alternative Low-Cost PLC," dalam *2020 International Symposium on Fundamentals of Electrical Engineering (ISFEE)*, IEEE, Nov 2020, hlm. 1–6. doi: 10.1109/ISFEE51261.2020.9756175.
- [5] S. Fujita, K. Hata, A. Mochizuki, K. Sawada, S. Shin, dan S. Hosokawa, "OpenPLC based control system testbed for PLC whitelisting system," *Artif Life Robot*, vol. 26, no. 1, hlm. 149–154, Feb 2021, doi: 10.1007/s10015-020-00635-1.
- [6] J. A. Fortoul-Diaz, L. A. Carrillo-Martinez, A. Centeno-Tellez, F. Cortes-Santacruz, I. Olmos-Pineda, dan R. R. Flores-Quintero, "A Smart actory Architecture Based on Industry 4.0 Technologies: Open-Source Software Implementation," *IEEE Access*, vol. 11, hlm. 101727–101749, 2023, doi: 10.1109/ACCESS.2023.3316116.
- [7] P. F. S. de Melo dan E. P. Godoy, "Controller Interface for Industry 4.0 based on RAMI 4.0 and OPC UA," dalam *2019 II Workshop on Metrology for Industry 4.0 and IoT (MetroInd4.0&IoT)*, IEEE, Jun 2019, hlm. 229–234. doi: 10.1109/METRO4.2019.8792837.
- [8] H. Jack dan S. C. Rowe, "Teaching Industrial Control with Open-Source Software," dalam *2023 ASEE Annual Conference & Exposition*, 2023.
- [9] P. O. dan Z. B., "Prototyping of control units for systems with industrial controllers," *System technologies*, vol. 2, no. 151, hlm. 50–61, Apr 2024, doi: 10.34185/1562-9945-2-151-2024-05.
- [10] S. Stankovski, G. Ostojic, M. Saponjic, M. Stanojevic, dan M. Babic, "Using micro/mini PLC/PAC in the Edge Computing Architecture," dalam *2020 19th International Symposium INFOTEH-JAHORINA (INFOTEH)*, IEEE, Mar 2020, hlm. 1–4. doi: 10.1109/INFOTEH48170.2020.9066309.
- [11] V. V. Potekhin, A. P. Alekseev, E. V. Kuklin, A. E. Misnik, dan D. Khitrova Ya, "Programming of open distributed industrial systems based on the international standard IEC 61499," *Computing, Telecommunications and Control*, vol. 17, no. 1, hlm. 10–19, 2024.
- [12] A. Ghaleb, S. Zhioua, dan A. Almulhem, "On PLC network security," *International Journal of Critical Infrastructure Protection*, vol. 22, hlm. 62–69, Sep 2018, doi: 10.1016/j.ijcip.2018.05.004.
- [13] K. Rajkumar, K. Thejaswini, dan P. Yuvaswari, "Automation of Sustainable Industrial Machine using PLC," *J Phys Conf Ser*, vol. 1979, no. 1, hlm. 012049, Agu 2021, doi: 10.1088/1742-6596/1979/1/012049.
- [14] Md. A. Kabir, K. R. Fahim, T. Hasan, Md. A. Ali, dan Md. A. Hossain, "Automatic Transfer Switch Using Programmable Logic Controller," dalam *2021 International Conference on Science & Contemporary Technologies (ICSCST)*, IEEE, Agu 2021, hlm. 1–6. doi: 10.1109/ICSCST53883.2021.9642528.

- [15] P. Maynard, K. McLaughlin, dan S. Sezer, "An Open Framework for Deploying Experimental SCADA Testbed Networks," 2018. doi: 10.14236/ewic/ICS2018.11.
- [16] R. EUSEBIO-GRANDE, M. N. IBARRA-BONILLA, J. AMARO-BALANZAR, dan F. SÁNCHEZ-TEXIS, "Graphical user interface for a PLC programming implemented in an ARM Cortex-M4 microcontroller.," *Journal Industrial Engineering/Revista de Ingenieria Industrial*, vol. 6, no. 16, 2022.
- [17] H. P. Guntaka, R. Kushalkar, N. Venkat, A. Chipkar, V. Easwaran, dan K. Moudgalya, "Modular Hardware, Open-Source Software and Training Material for PLC Training," dalam *2022 10th International Conference on Control, Mechatronics and Automation (ICCMA)*, IEEE, Nov 2022, hlm. 237–242. doi: 10.1109/ICCMA56665.2022.10011589.
- [18] N. Zubair, A. Ayub, H. Yoo, dan I. Ahmed, "PEM: Remote forensic acquisition of PLC memory in industrial control systems," *Forensic Science International: Digital Investigation*, vol. 40, hlm. 301336, Apr 2022, doi: 10.1016/j.fsidi.2022.301336.
- [19] T. Alves dan T. Morris, "OpenPLC: An IEC 61,131–3 compliant open source industrial controller for cyber security research," *Comput Secur*, vol. 78, hlm. 364–379, Sep 2018, doi: 10.1016/j.cose.2018.07.007.
- [20] T. Alves, T. Morris, dan S.-M. Yoo, "Securing SCADA Applications Using OpenPLC With End-To-End Encryption," dalam *Proceedings of the 3rd Annual Industrial Control System Security Workshop*, New York, NY, USA: ACM, Des 2017, hlm. 1–6. doi: 10.1145/3174776.3174777.
- [21] S. Niauronis, "OPENPLC Hardware Speed Performance Comparison," *PROFESSIONAL STUDIES: Theory And Practice*, vol. 27, no. 1, hlm. 65–71, 2023.
- [22] J. Hajda, R. Jakuszewski, dan S. Ogonowski, "Security Challenges in Industry 4.0 PLC Systems," *Applied Sciences*, vol. 11, no. 21, hlm. 9785, Okt 2021, doi: 10.3390/app11219785.