

Prototype Monitoring Data Pada Mesin

Wisma Febriamelia Permata, Hazwani Syalsabila, Ridwan, Indra Daulay

D4 Teknologi Rekayasa Elektronika, Jurusan Teknik Elektro, Program Studi Teknologi Rekayasa Elektronika,
Politeknik Negeri Batam, Indonesia

E-mail : wismafebryamelia@gmail.com, hazwanisyalsabila@gmail.com

Abstract—Pada era industri 4.0 saat ini sebagian besar pekerjaan telah digantikan dengan mesin. Hal tersebut dirancang untuk memudahkan pekerjaan manusia, salah satu contohnya ialah sistem otomasi dalam proses produksi. Sistem otomasi saat ini telah banyak mengotomatisasi berbagai proses industri dengan cara menggabungkan sistem mekanik, elektrik, dan program. Namun, ada kalanya sistem otomasi pada mesin mengalami kesalahan atau error sehingga memerlukan proses monitoring. Saat ini, proses monitoring pada mesin masih menggunakan metode konvensional, yaitu dengan pencatatan secara manual. Hal tersebut mengakibatkan proses monitoring membutuhkan waktu yang cukup lama sehingga dapat menunda proses produksi. Selain itu, terdapat kemungkinan dalam kesalahan pencatatan data akibat human error. Oleh karena itu, Prototype Monitoring Data Pada Mesin dirancang untuk dapat melakukan proses monitoring pada mesin. Pada sistem ini, CX-Programmer digunakan sebagai simulator PLC yang kemudian akan terhubung ke Visual Studio yang digunakan sebagai user interface dari monitoring data. Adapun data yang ditampilkan pada user interface di Visual Studio ialah jumlah barang masuk dan keluar, jumlah barang yang diproses, lokasi *error*, dan *cycle time*. Hasil analisa menunjukkan bahwa sistem monitoring yang dirancang dapat terkoneksi antara program PLC dengan Visual studio dapat membaca data dari CX Programmer sebagai simulator PLC. Adapun data yang ditampilkan pada *user interface* di Visual studio merupakan jumlah barang masuk, barang keluar, jumlah barang yang diproses, lokasi *error*, dan *cycle time*.

Keywords— Monitoring, CX Programmer, Visual studio

Abstract - In the current industrial era 4.0, most jobs have been replaced by machines. It is designed to facilitate human work, one example is the automation system in the production process. Today's automation systems have automated many industrial processes by combining mechanical, electrical, and program systems. However, there are times when the automation system on the machine experiences errors or errors that require a monitoring process. Currently, the monitoring process on the machine still uses conventional methods, namely by recording manually. This results in the monitoring process taking quite a long time so that it can delay the production process. In addition, there is a possibility of data recording errors due to human error. Therefore, the Prototype Monitoring Data on Machines is designed to be able to carry out the monitoring process on the machine. In this system, CX-Programmer is used as a PLC simulator which will then be connected to Visual Studio which is used as the user interface of monitoring data. The data displayed on the user interface in Visual Studio is the number of incoming and outgoing items, the number of items processed, the location of errors, and cycle time. The results of the analysis show that the monitoring system designed can be connected between the PLC program and Visual studio can read data from CX Programmer as a PLC simulator. The data displayed on the user interface in Visual studio is the number of incoming goods, outgoing goods, the number of goods processed, error location, and cycle time.

Keywords— Monitoring, CX Programmer, Visual studio

I. PENDAHULUAN

Revolusi industri 4.0 menitikberatkan pada otomatisasi dan mengkolaborasikannya dengan teknologi *cyber*. Ciri utama dari revolusi industri ini adalah penggabungan informasi dan teknologi komunikasi dalam bidang industri. Munculnya revolusi industri menyebabkan adanya perubahan dalam berbagai sektor. Jika semula membutuhkan pekerja yang cukup banyak, namun kini segala sesuatu bisa digantikan dengan penggunaan mesin teknologi[1]. Sebagian industri di Batam telah menggunakan sistem kendali otomasi. Sistem kendali otomasi yang dimaksud adalah sistem yang mampu menggerakkan suatu mekanik secara mandiri. PLC (*Programmable Logic Controller*) adalah salah satu yang sering digunakan dalam industri. PLC adalah alat yang digunakan untuk mengatur nyala (*ON*) atau mati (*OFF*) nya perangkat lain yang tersambung dengan perangkat tersebut dan dapat diubah-ubah (program)[2].

Pada sistem otomasi di sebuah mesin perlu dilakukannya monitoring. Monitoring adalah aktivitas yang dilakukan untuk mengetahui proses jalannya suatu program yang dirancang berjalan sesuai yang direncanakan dan dapat mengetahui hambatannya[3]. Penerapan monitoring di industri masih menggunakan sistem konvensional. Sistem konvensional dilakukan dengan cara pencatatan data secara manual dan sangat memungkinkan jika data yang dicatat tidak akurat. Hal ini akan berdampak pada informasi yang tidak valid.

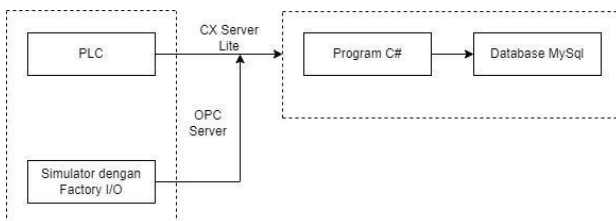
Dari permasalahan tersebut, maka dalam penelitian ini akan membuat *prototype* monitoring data pada mesin. Mesin yang dibuat dalam bentuk *prototype* berupa simulasi, dimana penelitian ini membuat program *ladder* PLC kemudian menampilkan simulasi tersebut pada Factory I/O. Factory I/O adalah simulasi pabrik 3D yang mempelajari tentang teknologi otomatisasi. Untuk simulasi menggunakan Factory I/O penelitian ini melakukan pengembangan dari jurnal Simulasi Alat Pengisi Barang Dan Pengepakan Barang Menggunakan Factory IO[4]. Untuk sistem monitoring dibuat dengan menggunakan *user interface* yang dibuat di Visual Studio untuk menampilkan data dari PLC. Bahasa pemrograman yang digunakan adalah C#. Komunikasi yang digunakan pada Visual studio untuk mengambil data ke PLC menggunakan CX server lite. Untuk sistem komunikasi tersebut penelitian ini melakukan pengembangan dari Tugas Akhir Perancangan Komunikasi Antar PLC dan Visual Studio Menggunakan CX Server lite Untuk Automation Tool's Scrib[5]. Dari referensi yang diperoleh maka penulis melakukan pengembangan tersebut dengan membuat sistem simulator mesin di Factory I/O dan membuat sistem monitoring data. Perbedaan monitoring ini terdapat pada data yang ditampilkan sistem monitoring berupa data *input*, *output*, *cycle time*, dan *error location*. Data tersebut akan disimpan ke dalam database MySQL untuk bisa dilakukan monitoring secara berulang.

II. METODE PENELITIAN

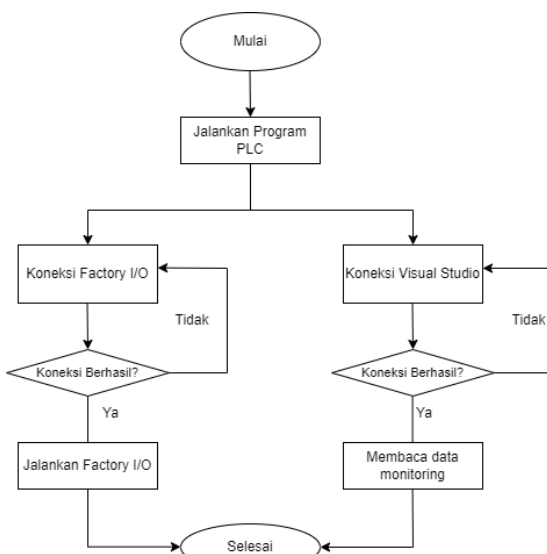
A. Desain Perencanaan Keseluruhan Sistem

Perencanaan sistem ini dimulai dari membuat program PLC dengan Cx Programmer dan membuat desain di Factory I/O. Program PLC yang dibuat untuk menjalankan simulator mesin dengan aplikasi Factory I/O. Program tersebut untuk mengetahui *input*, *output*, *cycle time*, dan *error location*. Untuk menghubungkan antara program PLC dan Factory I/O perlu protokol komunikasi OPC server. OPC Server adalah *protocol* komunikasi yang disediakan oleh Omron untuk menghubungkan software CX programmer dengan *software* yang lain[6]. dengan menambahkan *file* cdm program PLC yang telah dibuat ke Factory I/O, dengan memastikan *file* cdm tersebut sudah disimpan pada OPC server. Setelah dihubungkan maka simulator yang dibuat di Factory I/O akan berjalan sesuai dengan program yang dibuat di Cx Programmer.

Setelah perancangan *prototype* mesin, maka membuat sistem monitoring nya dengan membuat program di Visual studio dengan Bahasa pemrograman *c#*. Cara untuk *c#* membaca data dari PLC adalah dengan menggunakan protokol komunikasi Cx Server Lite[7]. Dengan menambahkan *library* Cx Server Lite yang sudah di *download* ke Visual studio maka *c#* sudah dapat mengakses data di PLC jika pemrograman yang dibuat sudah sesuai[8]. Ketika data sudah terbaca oleh *c#* maka akan ditampilkan di tampilan *form* yang dibuat dan disimpan di Database MySql untuk dapat memonitoring data yang sebelumnya. Pada penelitian ini semua *software* yang digunakan terdapat pada satu PC yang sama. Berikut gambaran perancangan sistem terdapat pada Gambar 1 dan Gambar 2:



Gambar 1 Diagram Blok Sistem



Gambar 2 Flowchart Sistem Kerja

B. Desain Perancangan Simulator Mesin dan Program PLC

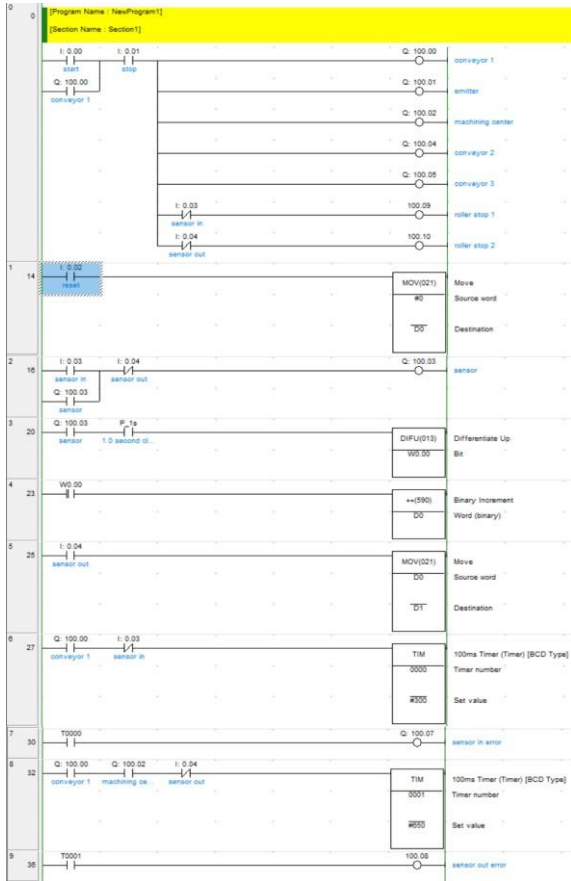
Pada perancangan simulasi mesin menggunakan 2 Sensor *diffuse* untuk mendeteksi barang dan menghitung input diawal dan output diakhir *conveyor*. Untuk pembuatan skema sistem menggunakan Factory I/O, dimana didalamnya terdapat *library* yang tersedia sebagai macam alat dan komponen yang akan digunakan untuk simulasi[9].

Pada skema perancangan sistem yang dirancang pertama adalah sistem *conveyor* yang digunakan, dengan menambahkan Sensor *diffuse* di awal *conveyor* dan di akhir *conveyor*. Ketika Sensor *diffuse* ingin menghitung barang maka barang akan berhenti, untuk membuat barang tersebut berhenti dengan menambahkan *Roller stop* di kedua Sensor *diffuse*. Selanjutnya menambahkan panel operator pada simulasi untuk membantu kinerja simulasi ini. Gambar 3 merupakan rancangan simulasi mesin yang dibuat.



Gambar 3 Rangkaian Sistem *Prototype* Mesin

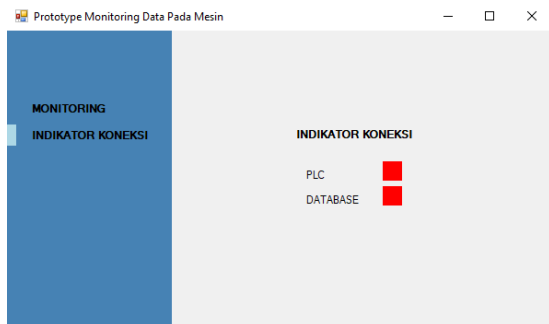
Pada perancangan program PLC Ketika *Push button start* ditekan maka *conveyor* akan berjalan. Pada program PLC yang dibuat ketika *conveyor* berjalan Sensor *diffuse input* belum aktif, harus di aktifkan pada program PLC maka Sensor *diffuse* akan menghitung barang yang melewati Sensor tersebut. Fungsi *Differential Up* untuk menghitung *Cycle Time* dari barang terdeteksi Sensor *diffuse input* sampai Sensor *diffuse output*. Pada bagian Reset di program PLC berfungsi untuk menghentikan barang berjalan dan sensor yang digunakan otomatis tidak akan mendeteksi barang. Berikut pada Gambar 4 terdapat program PLC yang telah dibuat pada CX Programmer



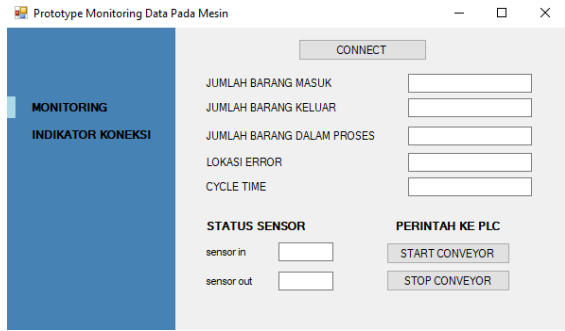
Gambar 4 Program PLC

C. Desain Perancangan Program Monitoring dan Penyimpanan data

Perancangan program monitoring pada penelitian ini menggunakan aplikasi Visual Studio dengan Bahasa pemrograman C#. Pada tampilan *form* terdapat dua tampilan dalam satu *form*, yaitu tampilan cek koneksi dan tampilan monitoring. Fungsi pada tampilan cek koneksi sebagai indikator apakah koneksi antara C#, PLC, dan Database terkoneksi. Pada tampilan monitoring menampilkan jumlah barang yang masuk, barang keluar, Lokasi *error*, dan *cycle time*. Selain menampilkan data terdapat status sensor jika status sensor “True” maka sensor dalam keadaan hidup, jika “False” maka sensor dalam keadaan tidak hidup. Untuk Button yang ada pada *form* monitoring digunakan untuk menghidupkan program PLC yang dibuat. Gambar 5 dan Gambar 6 merupakan tampilan *form* yang telah dibuat untuk sistem monitoring penelitian ini.

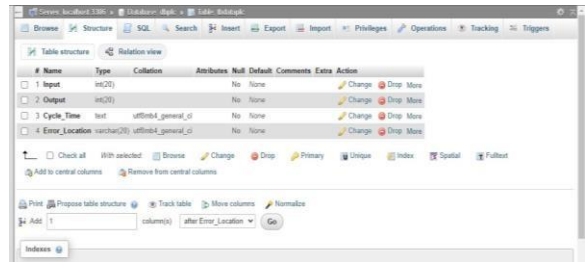


Gambar 5 Tampilan Cek Koneksi



Gambar 6 Tampilan monitoring

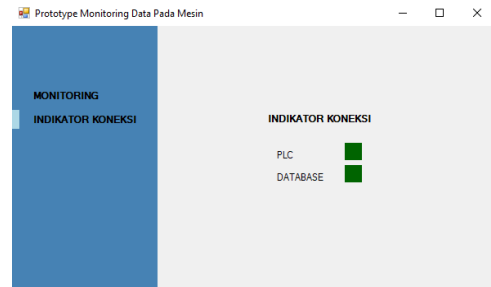
Pada perancangan penyimpanan data penelitian ini menggunakan Database MySQL. Data akan tersimpan di Database dengan menambahkan *library* My Sql ke Visual studio[10], dan menambahkan *coding* program untuk menyimpannya. Ketika data ditampilkan pada *form* monitoring maka otomatis akan langsung tersimpan di Database yang telah dibuat. Fungsi Database pada penelitian ini untuk memonitoring data yang diperoleh dari mesin setiap mesin dinyalakan. Untuk tipe data yang digunakan pada Database ini nilai *input* dan *output* tipe data int, untuk *error location* menggunakan tipe data *varchar*, dan untuk *cycle time* menggunakan tipe data *date time*[11]. Berikut Gambar 7 merupakan desain Database yang sudah dibuat.



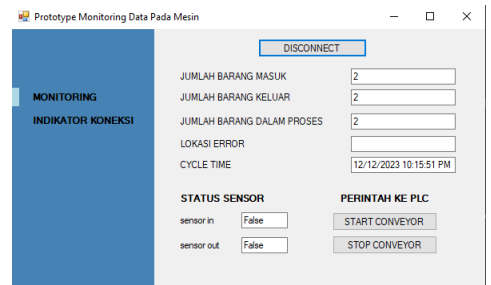
Gambar 7 Desain Database

III. HASIL DAN PEMBAHASAN

A. Hasil



Gambar 8 Tampilan Cek koneksi ketika terhubung

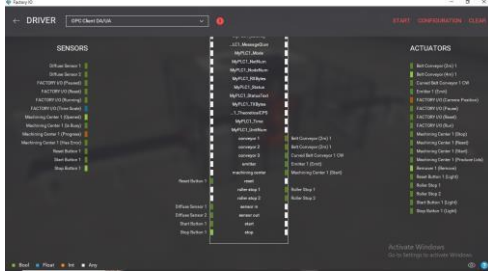


Gambar 9 Tampilan form yang sedang monitoring data PLC

Input	Output	Error_Location	Cycle_Time
0	0	none	
1	0	none	12/11/2023 11:54:50 AM
1	0	none	12/11/2023 11:54:50 AM
2	0	sensor out	12/11/2023 11:55:00 AM
2	0	sensor out	12/11/2023 11:55:00 AM
2	0	sensor out	12/11/2023 11:55:00 AM
2	0	sensor out	12/11/2023 11:55:00 AM
2	1	none	12/11/2023 11:55:00 AM
2	2	none	12/11/2023 11:55:00 AM

Gambar 10 Data yang tersimpan di Database

B. Pengujian OPC terhadap Factory I/O



Gambar 11 Pengujian OPC terhadap Factory I/O

Setelah OPC Server berhasil terkoneksi dengan Factory IO, tampilan *driver* akan muncul. Pada gambar diatas, dapat terlihat Factory I/O membaca *point* yang tersedia pada file *.cdm, yaitu tombol *start*, tombol *stop*, tombol *reset*, Sensor *input*, dan Sensor *output* pada bagian *input*. Kemudian pada bagian *output* terdapat *point conveyor*.

Point tersebut kemudian dipasangkan dengan komponen dari Factory I/O. Indikator berwarna hijau pada masing-masing *point* di dalam *driver* mengindikasikan bahwa komponen terhubung atau terkoneksi dengan baik ke *driver*.

C. Pengujian Koneksi PLC terhadap C# dan Database

Tabel 1 Pengujian koneksi PLC terhadap C#

No	Message Box Menampilkan "Berhasil Terkoneksi Ke PLC dan Database "	Indikator Warna PLC	Indikator Warna Database
1	OK	Green	Green
2	OK	Green	Green
3	OK	Green	Green
4	OK	Green	Green
5	OK	Green	Green
6	OK	Green	Green
7	OK	Green	Green
8	OK	Green	Green
9	OK	Green	Green
10	OK	Green	Green

Pada tabel 1 ditampilkan hasil pengujian koneksi dari PLC terhadap C# dan Database, jika koneksi antara PLC dengan C# dan Database berhasil maka akan menampilkan *message box* "Berhasil Terkoneksi Ke PLC dan Database" dan lampu indikator pada tampilan *form* akan hidup

berwarna hijau. Apabila gagal terkoneksi maka akan menampilkan *message box* "Koneksi Gagal" dan lampu indikator akan tetap berwarna merah.

D. Pengujian Perbandingan Waktu PLC terhadap Database

Tabel 2 Pengujian perbandingan waktu PLC terhadap Database

No	Perhitungan Waktu PLC (detik)	Perhitungan Waktu Database (detik)
1	33 detik	40 detik
2	32 detik	40 detik
3	33 detik	40 detik
4	33 detik	42 detik
5	32 detik	39 detik
6	34 detik	41 detik
7	33 detik	39 detik
8	33 detik	40 detik
9	32 detik	40 detik
10	33 detik	40 detik

Pada tabel 2 menampilkan data perhitungan waktu PLC terhadap Database yang telah dibuat dengan 10 percobaan. Perhitungan waktu pada PLC dan Database tidak sama karena *update rate* dibuat 6 detik, jadi ada *delay* sekitar 7 detik. Pemilihan *update rate* 6 dikarenakan terdapat *delay* antara Factory I/O ke CX Programmer. *Delay* tersebut mengakibatkan data terbaca lebih dari sekali jika menggunakan *update rate* 1 detik. Setelah mengukur lama *delay* yang terjadi antara Factory IO dan CX-Programmer, barulah *update rate* 6 detik dipilih untuk dapat menyesuaikan dengan waktu *delay* tersebut.

E. Pengujian Error location Pada Sensor output dalam kondisi dipaksa mati

Tabel 3 Pengujian indikasi error

No	Button Start Hidup	Proses Berjalan (detik)	Timer Berjalan (detik)	Indikasi Error
1	On	50 detik	65 detik	Sensor Out
2	On	50 detik	65 detik	Sensor Out
3	On	50 detik	65 detik	Sensor Out
4	On	50 detik	65 detik	Sensor Out
5	On	50 detik	65 detik	Sensor Out
6	On	50 detik	65 detik	Sensor Out
7	On	50 detik	65 detik	Sensor Out
8	On	50 detik	65 detik	Sensor Out
9	On	50 detik	65 detik	Sensor Out
10	On	50 detik	65 detik	Sensor Out

Pada tabel 3 ini menampilkan data pengujian indikasi error pada Sensor out terjadi ketika *Button start* hidup dan proses berjalan. Selama proses berjalan, *timer* akan

menghitung waktu hingga 65 detik. Jika *timer* telah mencapai 65 detik dan jumlah barang keluar tidak bertambah, maka sistem akan menyatakan *error* pada sensor out.

KESIMPULAN

Berdasarkan hasil dan analisa yang telah dilakukan dapat disimpulkan bahwa sistem monitoring ini dapat terkoneksi antara program PLC dengan Visual studio dan dapat membaca data dari CX Programmer tanpa menggunakan PLC fisik dengan menggunakan komunikasi CX server lite yang disediakan oleh pihak Omron. Adapun data yang di tampilkan saat monitoring pada penelitian ini adalah data jumlah barang yang masuk, jumlah data yang keluar, *cycle time*, dan *error location*.

REFERENCES

- [1] N. Purba, M. Yahya, and Nurbaiti, "Revolusi Industri 4.0 : Peran Teknologi Dalam Eksistensi Penguasaan Bisnis Dan Implementasinya," *J. Perilaku Dan Strateg. Bisnis*, vol. 9, no. 2, pp. 91–98, 2021.
- [2] A. Susanto, "Modul Programmable Logic Controller (Plc) Berbasis Arduino Severino," *J. Edukasi Elektro*, vol. 1, no. 2, 2017, doi: 10.21831/jee.v1i2.17413.
- [3] A. Khumaidi, "Sistem Monitoring dan Kontrol Berbasis Internet of Things untuk Penghematan Listrik pada Food and Beverage," *J. Ilm. Merpati (Menara Penelit. Akad. Teknol. Informasi)*, vol. 8, no. 3, p. 168, 2020, doi: 10.24843/jim.2020.v08.i03.p02.
- [4] G. R. Hidayat and I. H. Kurniawan, "Simulasi Alat Pengisi Barang Dan Pengepakan Barang Menggunakan Factory IO," *J. Ris. Rekayasa Elektro*, vol. 3, no. 1, 2021, doi: 10.30595/jrre.v3i1.9666.
- [5] F. A. Pohan, I. Arifin, D. Teknik, E. Otomasi, and F. Vokasi, "PERANCANGAN KOMUNIKASI ANTAR PLC DAN VISUAL STUDIO MENGGUNAKAN CX – SERVER LITE UNTUK AUTOMATION TOOL ' S CRIB," 2018.
- [6] G. Started, "CX-Server OPC User Manual Getting Started," pp. 1–42.
- [7] F. A. Pohan, A. Santoso, I. Arifin, A. A. Zurqi, and M. A. Hady, "Design of PLC–Visual Studio Communication using CX–Server Lite for Automation Tool's Crib," *IPTEK J. Proc. Ser.*, vol. 0, no. 3, p. 40, 2019, doi: 10.12962/j23546026.y2019i3.5838.
- [8] O. Guide and U. C. Lite, "CX-Server Lite User Manual Guide to using CX-Server Lite in Microsoft . Net," pp. 1–54.
- [9] Romero, "Automatización de una planta de almacenaje y distribución de mercancías usando Factory I/O y Codesys," *Simulasi Alat Pengisi Barang Dan Pengepakan Barang Menggunakan Fact. IO*, 2018, [Online]. Available: <https://idus.us.es/handle/11441/84172>
- [10] G. A. Laksana, P. Santoso, and F. Pasila, "Aplikasi untuk memonitor PLC pada mesin filling dan capping," *J. Tek. Elektro*, vol. 10, no. 2, pp. 48–53, 2017, doi: 10.9744/jte.10.2.48-53.
- [11] Risawandi, "Mudah Menguasai PHP & MySQL Dalam 24 Jam," *Unimal Press*, p. 72, 2019.