

SCADA : Water Cooled Chiller System

Laporan ini disusun untuk memenuhi mata kuliah SCADA yang diampu oleh :

Erik Haritman, S.Pd. MT.

Muhammad Adli Rizqullah, S.Pd., MT.



Disusun oleh:

Fahmi Jabbar (1801389)

ELEKTRONIKA INDUSTRI 2018

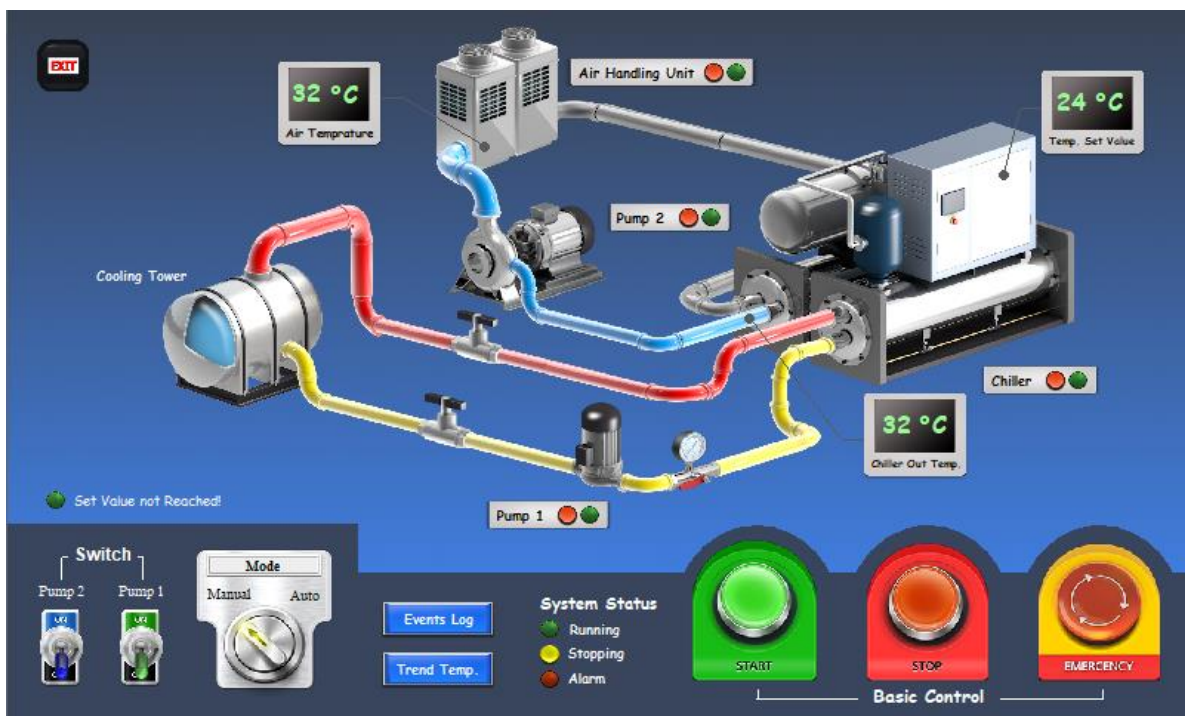
DEPARTEMEN PENDIDIKAN TEKNIK ELEKTRO

FAKULTAS PENDIDIKAN TEKNOLOGI DAN KEJURUAN

UNIVERSITAS PENDIDIKAN INDONESIA

2020

Plant Water Cooled Chiller System



Deskripsi System :

Water Cooled Chiller system merupakan sebuah system pendingin yang biasa digunakan di gedung-gedung bertingkat. System ini menggunakan sebuah mesin pendingin (chiller) sebagai komponen utamanya dalam menghasilkan dingin kemudian ada 2 buah pompa air, 1 buah cooling tower, dan 1 buah Air Handling Unit. Cara kerja chiller secara singkat terbagi kedalam dua buah bagian, diantaranya :

1. Bagian Kondenser : Bagian ini merupakan bagian dimana freon akan dikompresi kedalam tekanan tinggi. Mengapa perlu dikompresi ke tekanan tinggi? Akan dijelaskan pada bagian kedua. Namun dibagian ini, saat freon dikompres ke tekanan tinggi hal ini menyebabkan pipa yang menaungi freon menjadi panas. Akibatnya diperlukan sebuah pendingin alami, terdapat beberapa cara untuk melakukan pendinginan pada pipa tersebut yakni dengan udara ataupun air. Karena di plant ini menggunakan air, maka disebutlah water cooled chiller system. Air tersebut nantinya akan merendam pipa panas tadi, sehingga panas di pipa akan di transferkan ke airnya. Air yang masuk di bagian kondenser kemudian disirkulasikan menggunakan sebuah pompa menuju cooling tower, agar air yang panas barusan menjadi kembali kepada suhu normal.

Cooling tower sendiri, merupakan salah satu metode untuk mendinginkan air secara alami, yakni dengan cara membuat pipa dengan lubang lubang kecil sehingga air tersebut terpisah

menjadi molekul molekul berukuran kecil seperti tetesan air. Apabila sudah berukuran kecil kemudian akan ada sebuah blower/kipas yang meniupkan udara dari dalam tower keluar, sehingga air tersebut akan kehilangan energi panasnya dan temperaturnya akan turun kembali. Setelah itu air yang ada di cooling tower ini akan disirkulasikan kembali kedalam bagian kondenser untuk mendinginkan pipa yang panas tadi. Hal ini berlangsung seterusnya seperti itu hingga system dimatikan.

2. Bagian evaporator : Bagian ini merupakan bagian dimana freon akan diubah secara drastis tekanannya dari tinggi ke rendah. Perubahan tekanan yang drastis inilah yang mengakibatkan pipa yang menaungi freon dibagian evaporator menjadi dingin. Inilah mengapa diperlukan perubahan tekanan dari rendah ke tinggi, dan tinggi ke rendah pada freon agar dapat menghasilkan dingin. Pipa yang dingin tadi sudah terendam dalam sebuah air (yang berbeda dengan bagian kondensor). Air tersebut kemudian akan menyerap dingin yang ada pada pipa freon, sehingga airnya pun menjadi dingin. Air dingin ini kemudian disirkulasikan menuju Air Handling Unit menggunakan sebuah pompa.

Pada Air Handling Unit (AHU), merupakan sebuah pipa yang berkelok kelok (kemungkinan menyerupai bentuk persegi, dsb...) ditujukan agar luas penampang dari pipa yang mengandung air dingin tadi menjadi besar. Saat air dingin tadi melewati pipa pada AHU, pipa tersebut kemudian di tiupkan angin menggunakan blower/kipas. Udara yang ditiupkan melewati pipa di AHU (yang mengandung air dingin) itulah yang menjadikan udara disekitar menjadi terasa lebih dingin bahkan lama kelamaan akan mendinginkan seluruh ruangnya.

Program pada Ladder PLC :

1. Untuk bagian kontrol terdapat 3 buah tombol basic control, yakni :

- Start : Untuk menyalakan system
- Stop : Untuk mematikan system
- Emergency : Untuk mematikan system secara keseluruhan (tidak bisa di start lagi)

2. Apabila tombol start ditekan, kemudian operator diharuskan memilih mode pengoperasian diantaranya terdapat mode manual dan mode otomatis. Default switchnya terdapat pada mode manual, pada mode ini pengaturan penggunaan pump 1 dan pump 2 akan dinyalakan secara manual. Sedangkan pada mode otomatis, tidak diperlukan pengaturan pensaklaran pump 1 dan pump 2. Selain itu, diasumsikan apabila tombol ini ditekan, Cooling tower bekerja otomatis.

3. Pada case mode manual, apabila pump 1 dinyalakan. Maka air dari cooling tower akan disirkulasikan ke chiller bagian kondenser untuk mentransfer panas dari pipa freon yang dikompresi. Hal ini akan terjadi secara terus menerus hingga switch pump 1 dimatikan (pada

mode manual). Selain itu diasumsikan juga apabila menswitch pump 1 ke on, maka Chiller akan menyala bersamaan dengan pump 1 nya.

4. Pada case mode manual, apabila pump 2 dinyalakan **sebelum** pump 1 dinyalakan. Maka pada HMI akan ada sebuah pesan warning karena pump 2 menyala sebelum pump 1. Hal ini dinyatakan warning karena air pada bak evaporator masih belum dingin (di awal system berjalan) sehingga perlu dinyalakan terlebih dahulu pump 1 nya. Apabila pump 2 dinyalakan setelah pump 1 menyala, yang terjadi adalah pump 2 akan menyala kemudian mensirkulasikan air ke AHU dan kembali lagi ke bak evaporator. Diasumsikan saat switch pump 2 dinyalakan, maka AHU juga ikut bekerja bersamaan dengan pump 2 nya.

5. Pada case mode otomatis, pump 1 dan pump 2 akan bekerja secara bersamaan tanpa harus dinyalakan saklar pump 1 dan pump 2 nya terlebih dahulu. Diasumsikan begitu pula untuk Cooling tower, AHU dan Chillernya bekerja bersamaan.

6. Apabila tombol stop ditekan saat keadaan apapun, maka semua mesin akan ada pada kondisi off (system berhenti).

7. Apabila tombol emergency ditekan, saat keadaan apapun, maka semua mesin akan ada pada kondisi off (system berhenti) kemudian tombol start dan stop tidak akan berfungsi untuk menyalakan atau mematikan system. Hal ini diibaratkan dengan diputusnya pasokan listrik untuk menyalakan system sehingga tidak ada tombol yang dapat bekerja, kecuali tombol emergencynya untuk mengembalikan pasokan listriknya kembali.

8. Berikut ini merupakan detail dari manipulasi data/memory dari system chiller yang dibuat :

- Apabila tombol start ditekan, maka yang terjadi adalah akan di MOVE nya sebuah nilai pada setiap register tertentu untuk mendefinisikan nilai awal Set temperature, Chiller Temperature, Actual Temperature, dan Water level.

- Saat system berjalan juga, terdapat sebuah rung untuk mengecek terjadinya warning event dan danger event. Dalam hal ini warning event merupakan terjadinya switch on pada pump 2 sebelum pump 1 dan untuk danger event ketika suhu actual temperature $\geq 40^{\circ}\text{C}$.

- Saat system berjalan juga, terdapat sebuah rung untuk mengecek terjadinya good event, dimana suhu Set Temperature dan suhu Actual Temperature nilainya sudah sama.

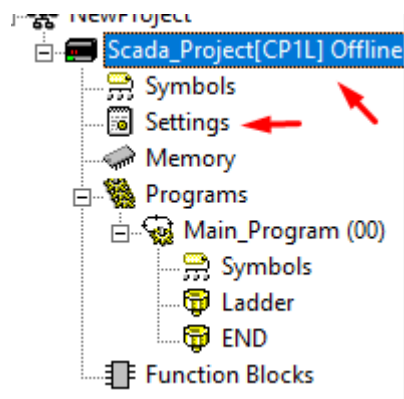
- Ketika pump 1 dinyalakan, terdapat sebuah rung untuk memanipulasi naiknya temperature pada chiller ($+1^{\circ}\text{C}$). Naiknya temperature chiller ini menggunakan timer selama #600 BCD (kurang lebih sekitar 60 detik). Selain itu ada juga rung yang digunakan untuk memanipulasi naik turunnya air pada Cooling tower yang mengindikasikan bahwa air bersirkulasi.

- Ketika pump 2 dinyalakan, terdapat sebuah rung untuk memanipulasi naiknya temperature pada ruangan (actual temperature +1°C). Naiknya temperature pada ruangan ini menggunakan timer selama #600 BCD (kurang lebih sekitar 60 detik).
- Ketika pump 1 dinyalakan, (berbarengan dengan rung sebelumnya juga) terdapat sebuah rung untuk memanipulasi turunnya temperature pada chiller (-1°C). Turunnya temperature chiller ini menggunakan timer selama #100 BCD (kurang lebih sekitar 10 detik) dan konstan selama pump 1 tetap menyala.
- Ketika pump 2 dinyalakan, (berbarengan dengan rung sebelumnya juga) terdapat sebuah rung untuk memanipulasi turunnya temperature pada ruangan (actual temperature -1°C). Turunnya temperature pada ruangan ini menggunakan timer selama waktu yang berbeda beda bergantung pada suhu chiller. Ketika suhu chiller $\geq 30^{\circ}\text{C}$, maka timer untuk menurunkan suhu ruangan akan berlangsung selama #700 BCD (kurang lebih sekitar 70 detik). Ketika suhu chiller $< 30^{\circ}\text{C}$, maka timer untuk menurunkan suhu ruangan akan berlangsung selama #200 BCD (kurang lebih sekitar 20 detik). Ketika suhu chiller $< 28^{\circ}\text{C}$, maka timer untuk menurunkan suhu ruangan akan berlangsung selama #150 BCD (kurang lebih sekitar 10,5 detik). Ketika suhu chiller $< 25^{\circ}\text{C}$, maka timer untuk menurunkan suhu ruangan akan berlangsung selama #100 BCD (kurang lebih sekitar 10 detik). Hal ini akan terjadi selama pump 2 menyala.

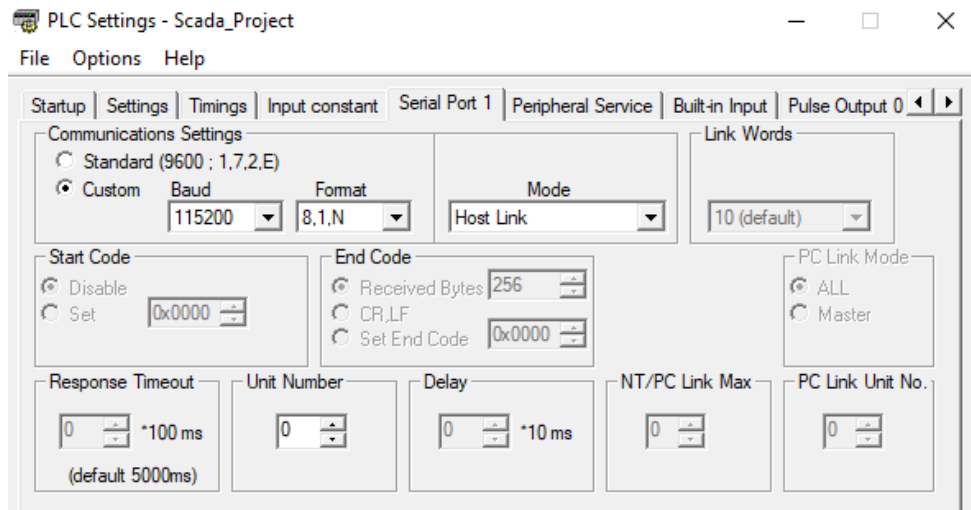
Konfigurasi Projek SCADA :

1. CX-Programmer

- Pertama, buka software CX-Programmer, biasanya terletak pada kumpulan software CX-One.

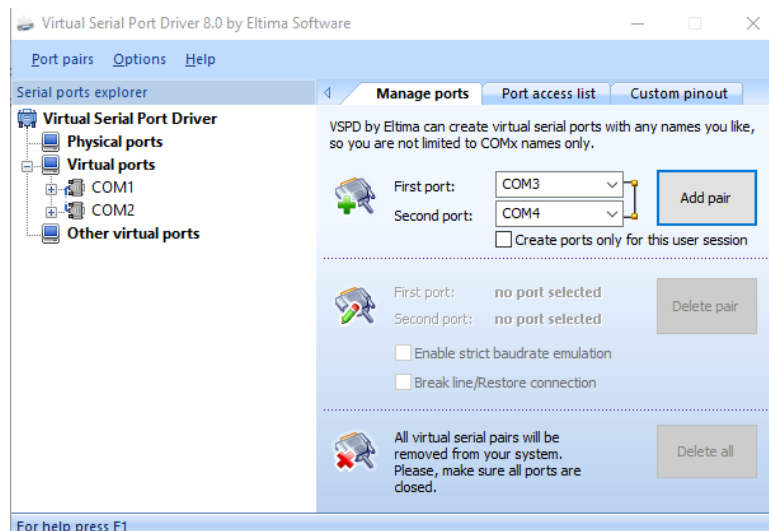


- Siapkan terlebih dahulu program ladder diagram dari system yang dibuat atau bila belum, bisa membuat projek dari awal. Kemudian ganti tipe PLC dengan mengklik nama device/projek (panah paling atas) ke **CP1L** > klik **Settings** > Ganti CPU Typenya menjadi **L**.
- Kemudian klik **Settings** (panah kedua) > klik tab **Serial Port** > Ubah konfigurasinya menjadi seperti berikut ini :



- Jangan tutup software CX-Programmer Terlebih Dahulu

2. Virtual Serial Port Driver by Eltima

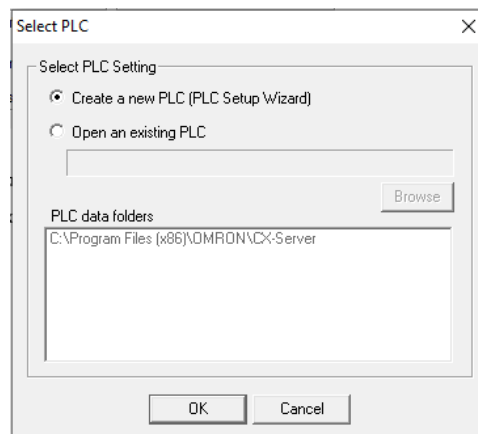


Untuk mensimulasikan hubungan antara software PLC dan software HMI biasanya dibutuhkan hubungan antara device fisik (asli). Namun kali ini dapat dilakukan sepenuhnya secara simulasi dengan memanfaatkan software Virtual Serial Port Driver. Dengan software ini, klik Add Pair pada COM berapapun, disini dicontohkan pada COM1 dan COM2 sehingga pada virtual port terdapat COM1 dan COM2. Bila sudah membuat pair dari COM, langkah selanjutnya adalah mengingat nomor dari COMnya untuk digunakan pada langkah selanjutnya.

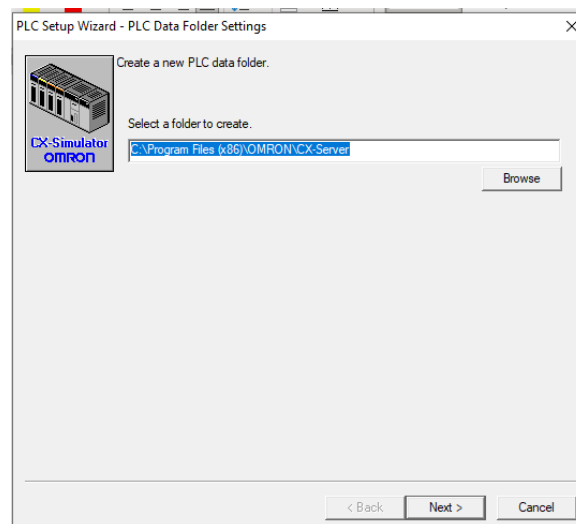
Untuk memverifikasi apakah virtual portnya sudah dibuat atau belum, silahkan klik kanan di start menu lalu pilih **Device Manager**, kemudian nanti akan ada sebuah paired port yang terlist di device managernya. Bila sudah silahkan tutup saja software ini.

3. CX-Simulator

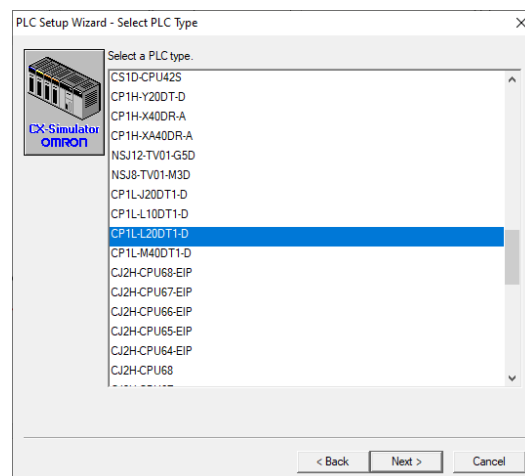
- Pertama, buka software CX-Simulator, biasanya terletak pada kumpulan software CX-One.



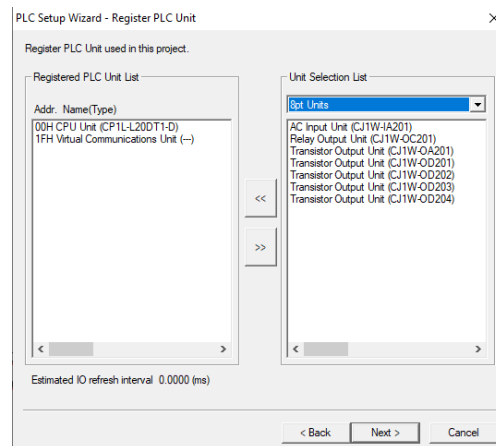
- Akan terdapat popup window seperti diatas, kemudian klik **Create a new PLC (PLC Setup Wizard)**.



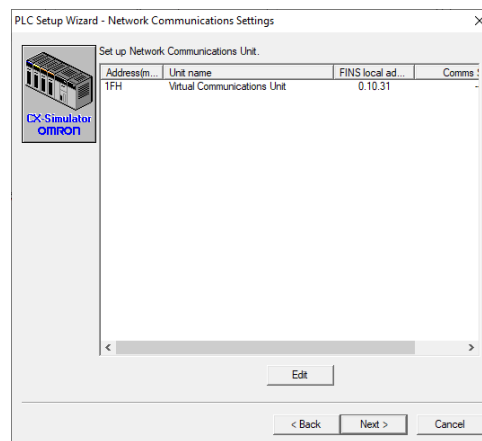
- Pada bagian ini, silahkan pilih folder untuk menyimpan data PLC (bebas) , bila sudah silahkan klik **Next**.



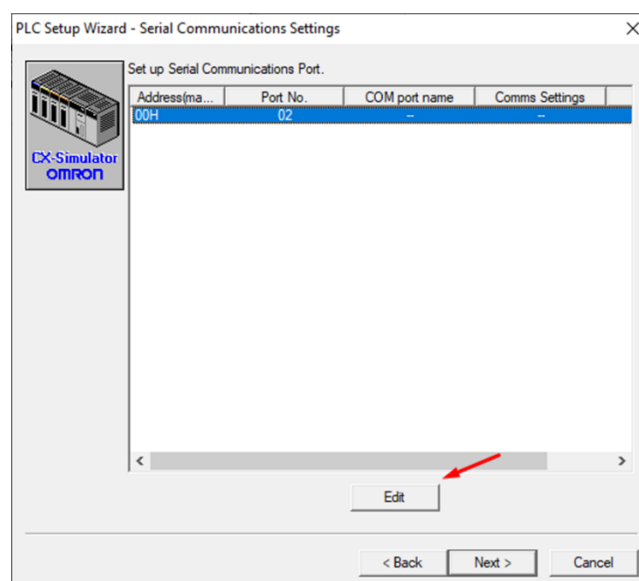
- Kemudian pilih tipe PLC **CP1L-L20DT1-D**. Bila sebelumnya di CX Programmer menggunakan tipe PLC yang berbeda maka silahkan disesuaikan. Lalu klik **Next**.



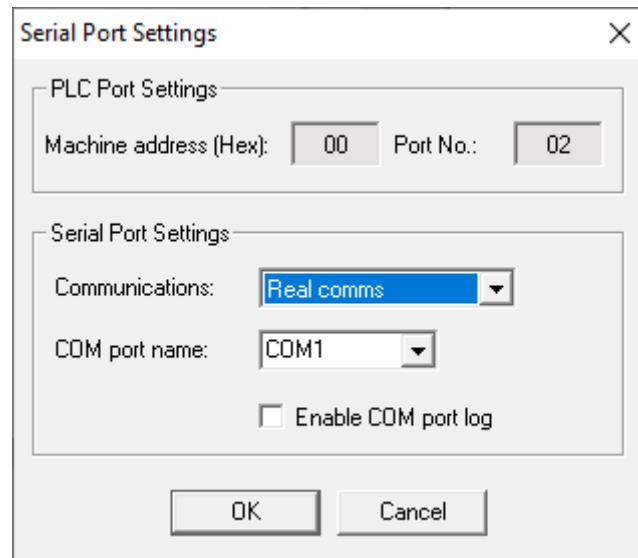
- Untuk register PLC Unit gunakan default, klik **Next**.



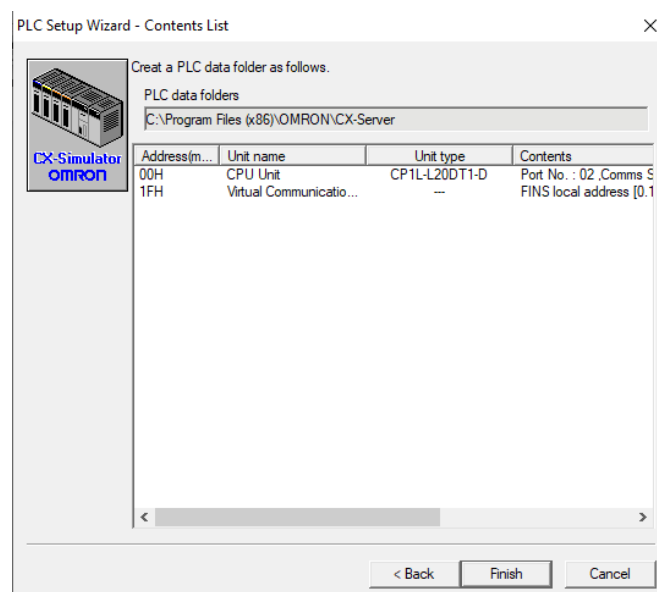
- Untuk Set Up Network Communication Unit gunakan default, klik **Next**.



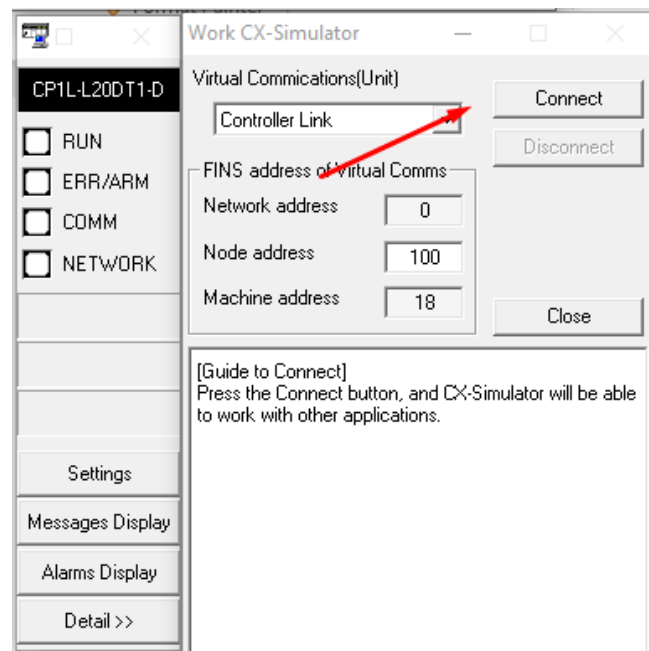
- Untuk Set Up Communications Port, klik address yang paling atas kemudian klik **Edit**.



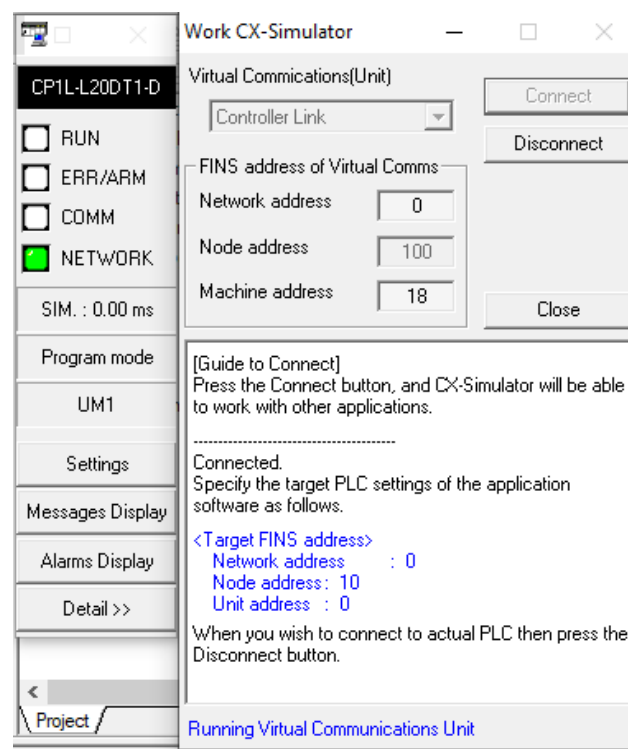
- Kemudian akan ada popup window Serial Port Settings. Pada menu dropdown communications, pilih **Real Comms** dan untuk COM port name pilih salah satu pair dari COM yang barusan dibuat di software Virtual Serial Port Driver. Disini dicontohkan COM1. Kemudian klik **OK**. Bila ada popup window kembali mengenai peringatan, klik **OK** saja.



- Kemudian akan ada tampilan seperti ini, bila sudah sesuai silahkan klik **Finish**. Bila ada peringatan mengenai existing folder (karena sebelumnya pernah membuat) silahkan klik **Yes** saja.
- Setelah itu akan muncul window terpisah seperti dibawah ini. Lalu klik **Connect**.

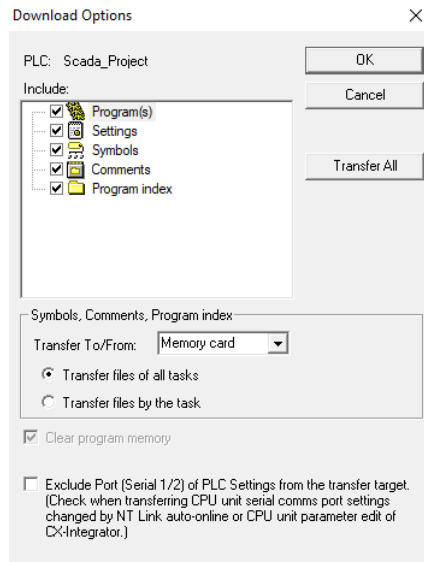


- Bila berhasil, maka akan lampu pada network akan berkedip, dan terdapat keterangan pada window Work CX-Simulator



- Kemudian beralih ke program CX-Programmer, Lalu klik **Work Online Simulator**, kemudian klik menu **PLC > Transfer > To PLC**.

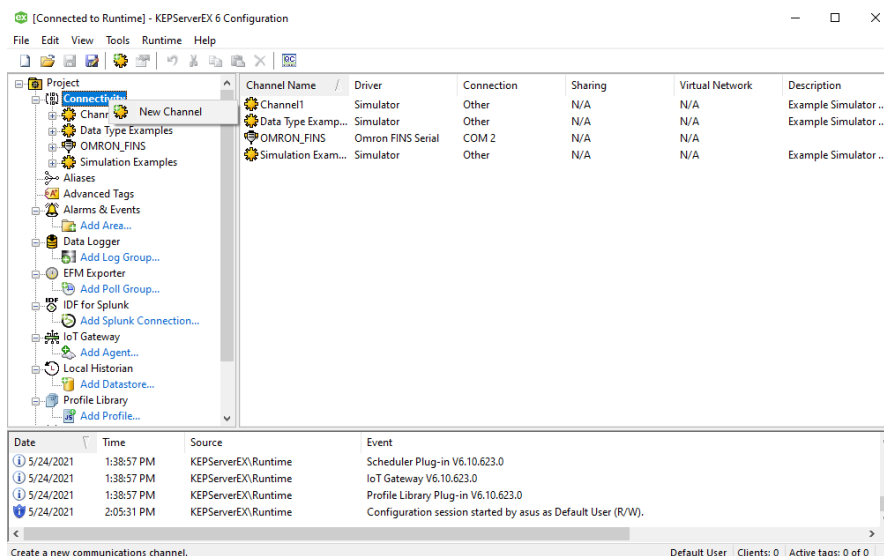
Kemudian pastikan semua ini dicentang, kemudian klik **OK**



- Bila ada popup untuk beralih ke program mode, silahkan klik **OK**. Selanjutnya akan ada popup untuk beralih ke monitor mode lalu klik **OK**.
- Jangan tutup software ini untuk melanjutkan ke step selanjutnya.

4. Pada KEPServerEX 6

- Software KEPServerEX ini merupakan sebuah software OPC Server dan OPC Client. Namun terlebih dahulu kita setup servernya. Pertama buka configuration pada KEPServerEX (di tray system icon > cari icon KEPServerEX > klik kanan, lalu pilih **Configuration**).
- Kemudian klik di bagian menu **Edit > Configuration > Property Groups OPC UA > ubah value Allow anonymous Login menjadi Yes**.



- Kemudian Pada bagian Connectivity, klik kanan dan pilih **New Channel**. Buat sebuah Channel dengan konfigurasi seperti berikut :

Type of Channel : **Omron FINS Serial**

Name : **Test_Channel** (Silahkan disesuaikan nama channelnya)

Physical Medium : **COM Port**

Physical Port Number : **2** (Pilih sesuai dengan pair COM yang sudah dibuat sebelumnya, tetapi yang tidak digunakan di CX-Simulator)

Baud Rate : **115200**

Data Bits : **8**

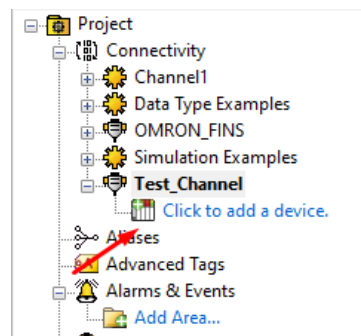
Parity : **None**

Stop Bits : **1**

Flow Control : **None**

Setelah itu silahkan pilih **Next** saja sampai selesai.

- Langkah selanjutnya klik Add Device pada **Test_Channel**



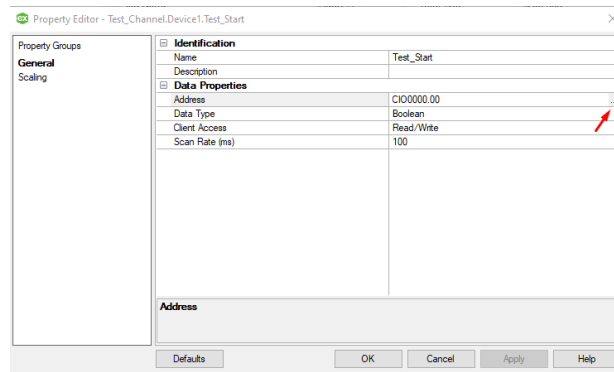
- Buat konfigurasi device seperti berikut ini :

Name : **Device1** (Silahkan disesuaikan)

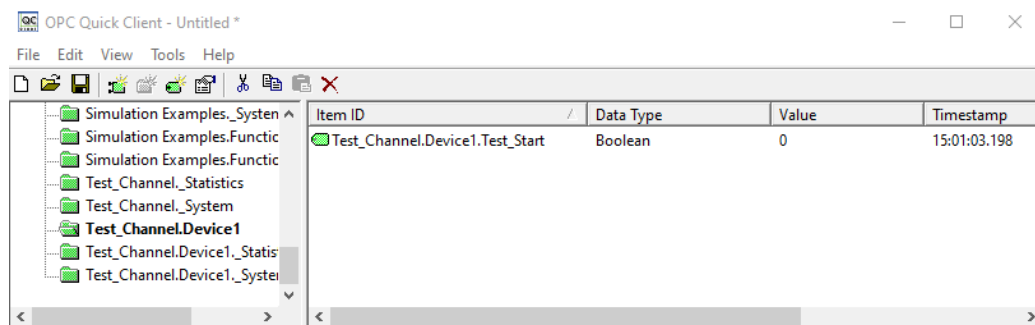
Model : **CS1** (Apabila memilih CP1L di CX-Programmer, maka model ini cocok untuk addressing CP1L)

ID : **0.0.10** (Lihat kembali konfigurasi pada tab CX-Simulator susunan penulisan alamatnya adalah *UnitAddress.NetworkAddress.NodeAddress*)

Untuk selanjutnya pilih **Next** saja hingga selesai.



- Kemudian Add Static Tag, sesuai dengan address yang ingin dibaca dari CX-Programmer. Diatas ini merupakan salah satu contoh cara mengkonfigurasi penambahan static tagnya. Apabila ada kebingungan dengan masalah penulisan address silahkan klik titik 3 (panah merah) untuk hint alamatnya.
- Untuk menguji apakah OPC Server sudah membaca data dari CX-Programmer (Software PLC) maka kita dapat melakukan uji pembacaannya menggunakan OPC Quick Client milik KEPServerEX 6. Caranya adalah dengan mengklik menu **Tools** pada KEPServerEX > Pilih OPC Quick Client. Kemudian navigasikan ke folder **Nama_Channel>Nama_Device**. Didalamnya akan ada sebuah tag yang barusan dibuat. Cobalah untuk merubah state dari tombol tersebut di CX-Programmer.



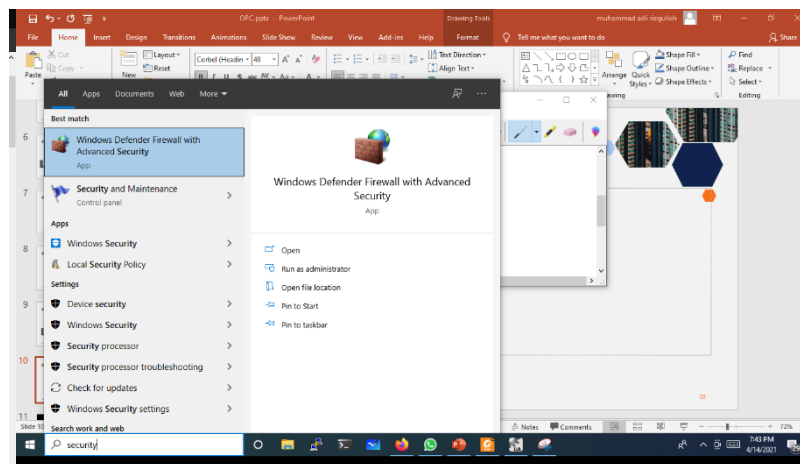
Apabila valuenya berubah sesuai dengan perubahan di CX-Programmer Maka integrasi dari CX-Programmer ke OPC Server sudah selesai. Pada proyek Water Cooled Chiller System ini, berikut adalah list tag yang diimport

Tag Name	Address	Data Type	Scan Rate	Scaling
Button_Start	CIO0000.00	Boolean	500	None
Button_Stop	CIO0000.01	Boolean	500	None
Button_Emergency	CIO0000.02	Boolean	500	None
Button_Mode	CIO0000.03	Boolean	500	None
Button_Pump1	CIO0000.04	Boolean	500	None
Button_Pump2	CIO0000.05	Boolean	500	None
IR_System	CIO0010.00	Boolean	500	None
IR_EventDanger1	CIO0010.01	Boolean	500	None
IR_EventWarning1	CIO0010.02	Boolean	500	None
Out_Chiller	CIO0100.00	Boolean	500	None
Out_AHU	CIO0100.01	Boolean	500	None
Out_Pump1	CIO0100.02	Boolean	500	None
Out_Pump2	CIO0100.03	Boolean	500	None
Out_IndicatorSet	CIO0100.04	Boolean	500	None
Out_IndicatorAlarm	CIO0100.05	Boolean	500	None
Memory_SetTemperature	D00000	Word	500	None
Memory_ActualTemperature	D00001	Word	500	None
Memory_ChillerTemperature	D00002	Word	500	None
Memory_WaterLevel	D00003	Word	500	None

5. Membuka Port 49320 pada Windows Defender Firewall.

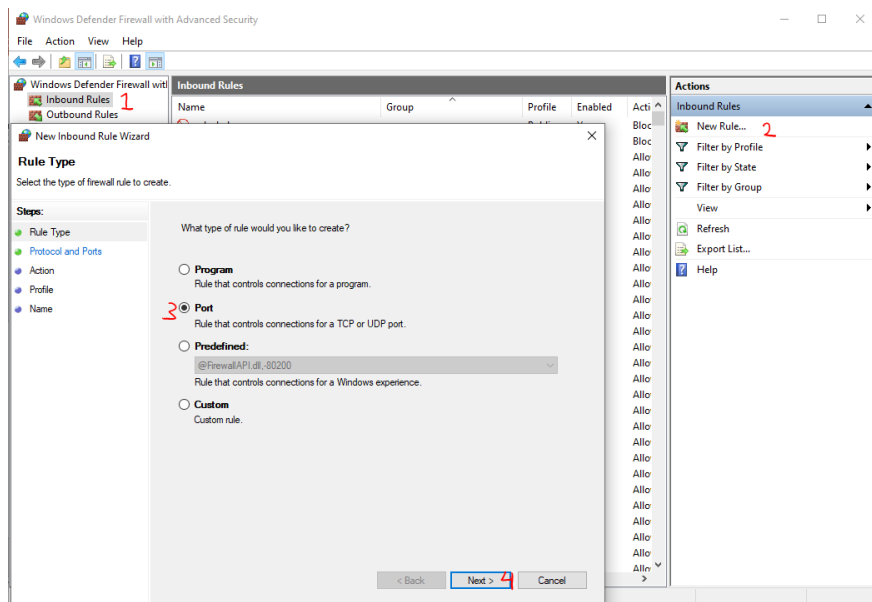
Fungsi membuka Port 49320 ini adalah untuk memberikan akses pada OPC UA Client untuk mengakses OPC Servernya. OPC Client yang digunakan pada EasyBuilder Pro adalah OPC Client bertipe UA, sehingga port ini harus dibuka terlebih dahulu.

- Pertama ketik di start menu **“Windows Defender Firewall with Advanced Security”**



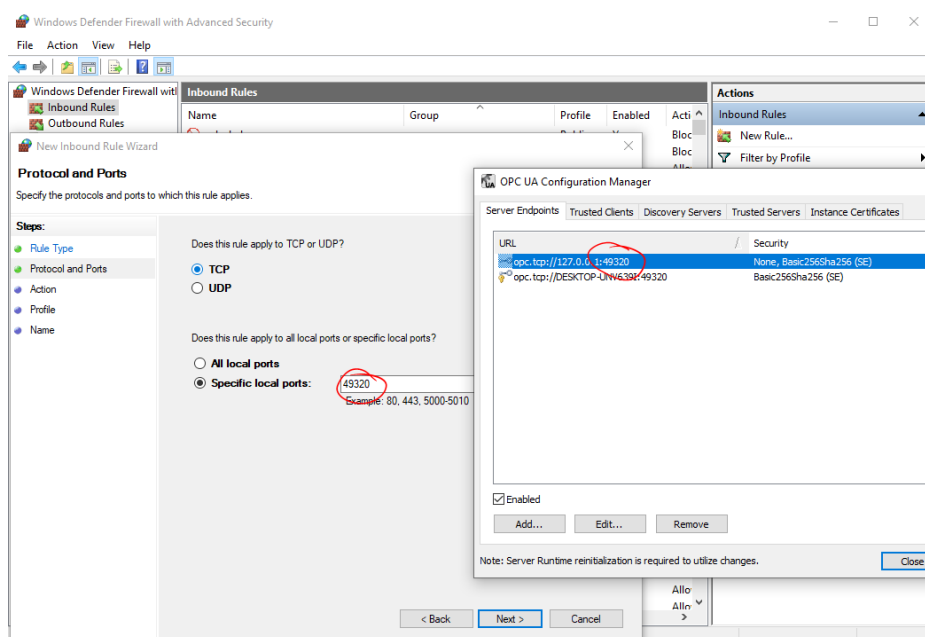
- Kemudian klik **Inbound Rules > New Rule...** > Kemudian konfigurasi Inbound Rules seperti berikut ini :

Rule Type : **Port**

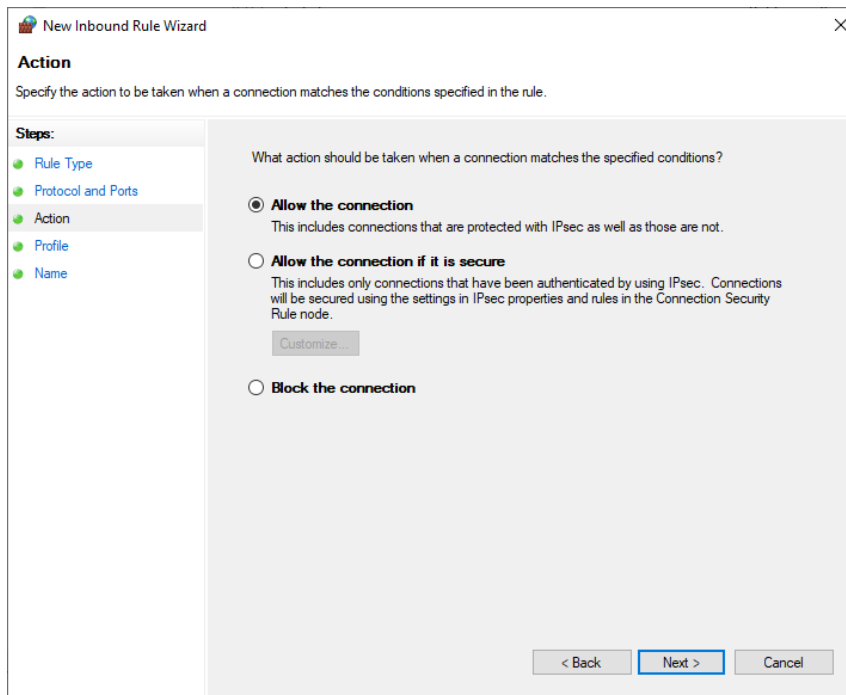


Does the rule apply to TCP or UDP : **TCP**

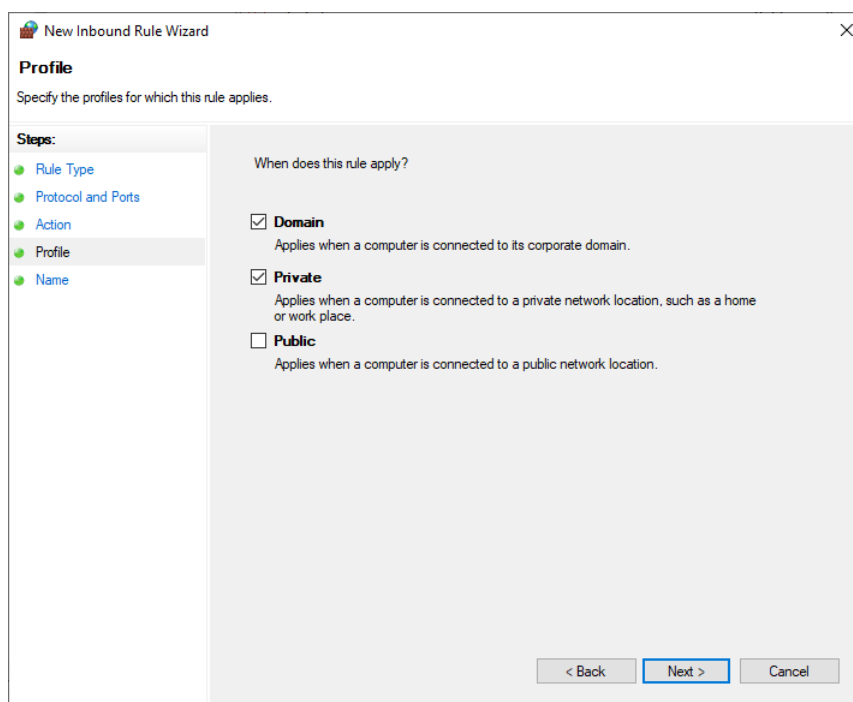
Specific Local Ports : **49320**



Tab Action : **Allow the Connection**



Tab Profile Centang **Domain** dan **Private** saja.

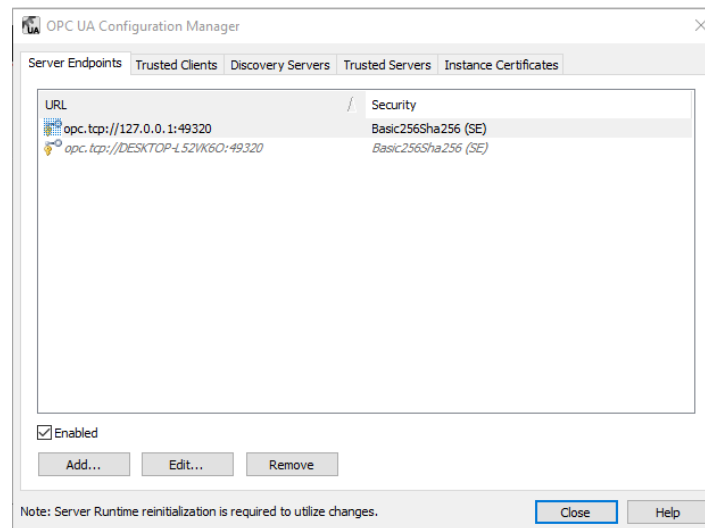


Tab Name : **OPC UA Interface** (Silahkan disesuaikan)

Untuk menguji apakah sudah dapat mengakses data pada OPC Server menggunakan OPC UA Client, langkah selanjutnya adalah menggunakan software Matrikon OPC UA Explorer. Bagian ini dapat di skip apabila diyakini sudah dapat diakses data dari OPC Servernya.

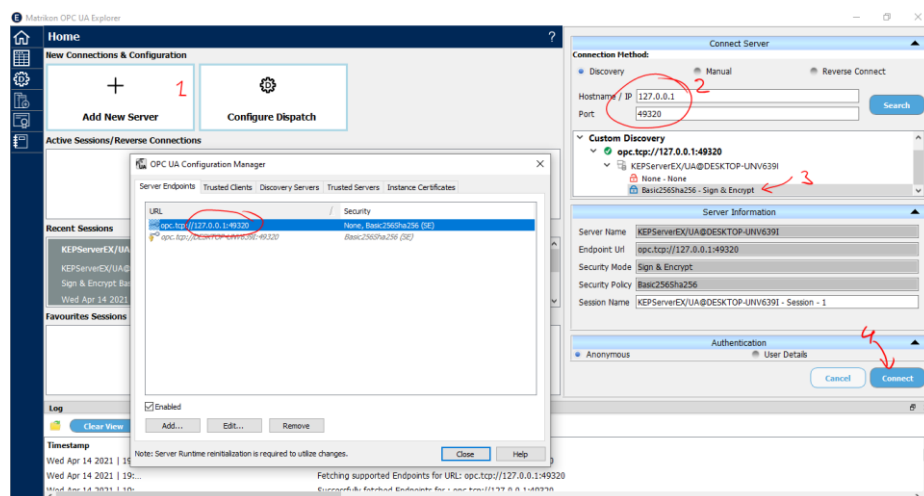
5. Pada Matrikon OPC UA Explorer (Optional)

- Pastikan pada KEPServerEX 6, konfigurasi pada OPC UA bagian **Allow Anonymous Login** sudah diberi value **Yes** agar tidak diperlukan kredensial dalam mengakses datanya.
- Pertama persiapkan software Matrikon OPC UA Explorer, kemudian buka software tersebut.
- Beralih pada system tray icon, klik kanan pada logo KEPServerEX 6 kemudian pilih OPC UA Configuration.



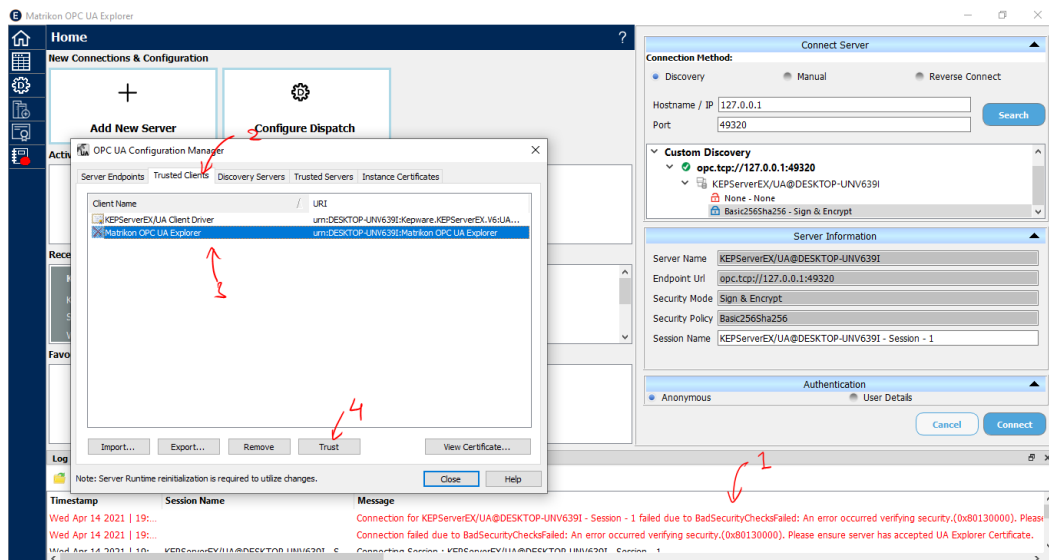
Inilah salah satu alasan mengapa port yang dibuka adalah port 49320, karena pada server endpoint OPCnya secara default adalah port 49320. Sekarang beralih kembali ke software Matrikon OPC UA Explorer.

- Klik **Add New Server**, kemudian isikan **Hostname/IP** dan **Port** sesuai dengan yang ada pada OPC UA Configuration Manager dari KEPServer EX 6. Setelah itu klik **Connect**.

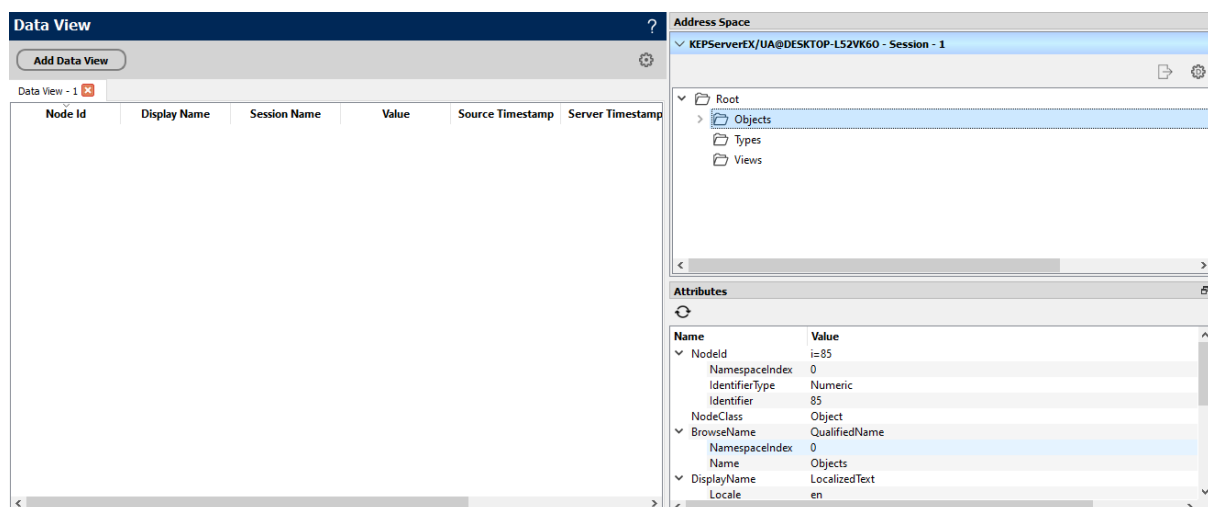


- Nantinya pada bagian **Custom Discovery** akan terdapat Server Endpoint yang sesuai dengan yang ada di KEPServerEX 6. Expand saja seluruh isinya, kemudian klik file yang paling akhir dengan tulisan “... **Sign & Encrypt**” Setelah itu klik Connect.

- Biasanya saat pertama kali connect, akan terjadi error **BadSecurityChecksFailed**. Beralih ke window OPC UA Configuration Manager, kemudian pilih tab **Trusted Client**. Akan ada client baru yang memiliki symbol silang seperti dibawah ini. Klik nama clients tersebut lalu pilih **Trust**. Lakukan kembali step selanjutnya, seharusnya sudah bisa masuk ke Data View dari OPC UA Explorer secara baik.



- Setelah itu, pada sebelah kanan akan terdapat **Address Space**, expand folder (klik 2x) berikut ini : **Root > Objects > NamaChannel > NamaDevice** > kemudian drag tag yang ingin dibaca kedalam table **Data View**.

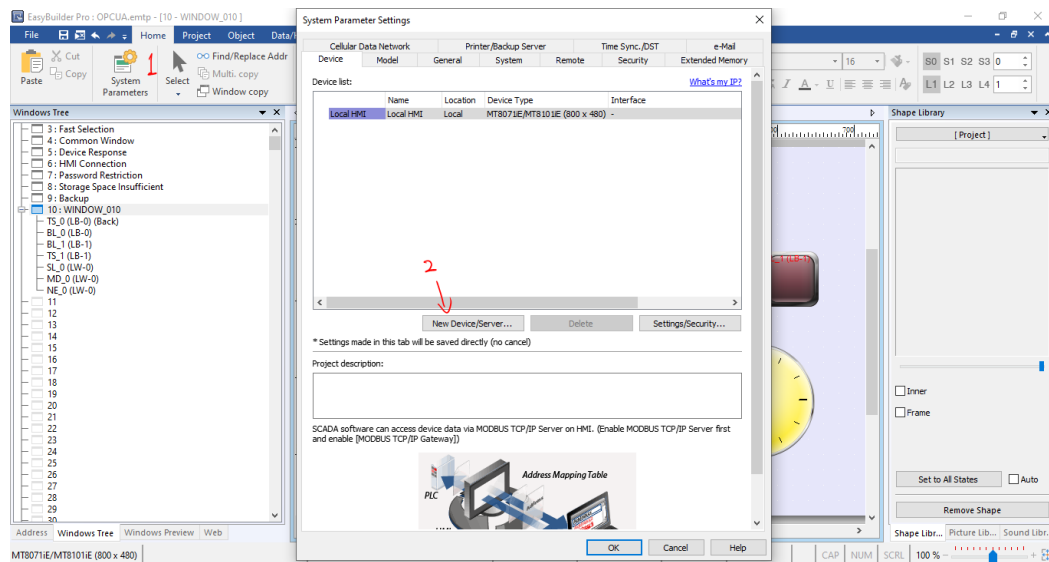


- Jalankan kembali **Work Online Simulator** pada CX-Programmer, pastikan hasil pembacaan pada OPC UA Explorer sudah memiliki status **Good: Success** Seperti berikut ini.

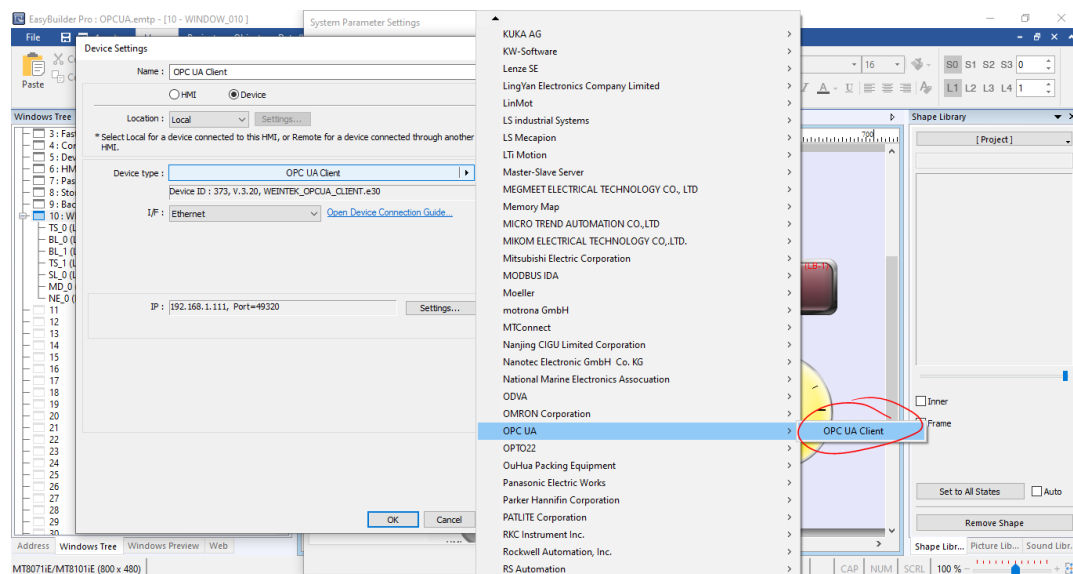
Data View - 1						
Node Id	Display Name	Session Name	Value	Source Timestamp	Server Timestamp	Status Code
1	ns=2;s=Test_Start	KEPServerEX/...	false	2021-05-25 ...	2021-05-25 ...	Good: Success.(0x0)

6. Pada EasyBuilder Pro

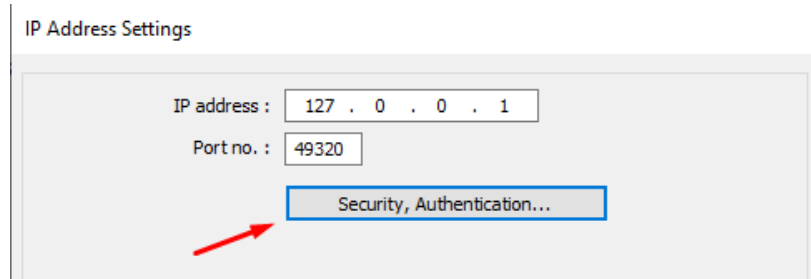
- Pertama, siapkan terlebih dahulu design plant dari system yang akan dibuat. Diprojek ini tipe HMI yang digunakan adalah HMI MT8070iE/MT8100iE (800 × 480).
- Setelah itu pada tab **Home** klik **System Parameters**. Kemudian klik New Device/Server.



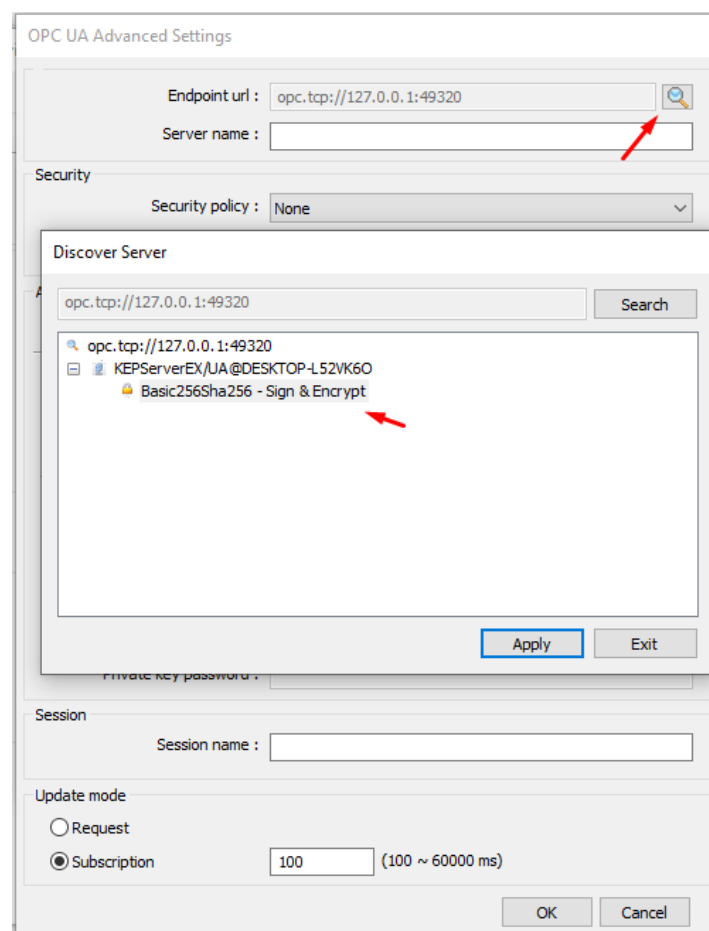
- Kemudian pilih Device Type **OPC UA > OPC UA Client**.



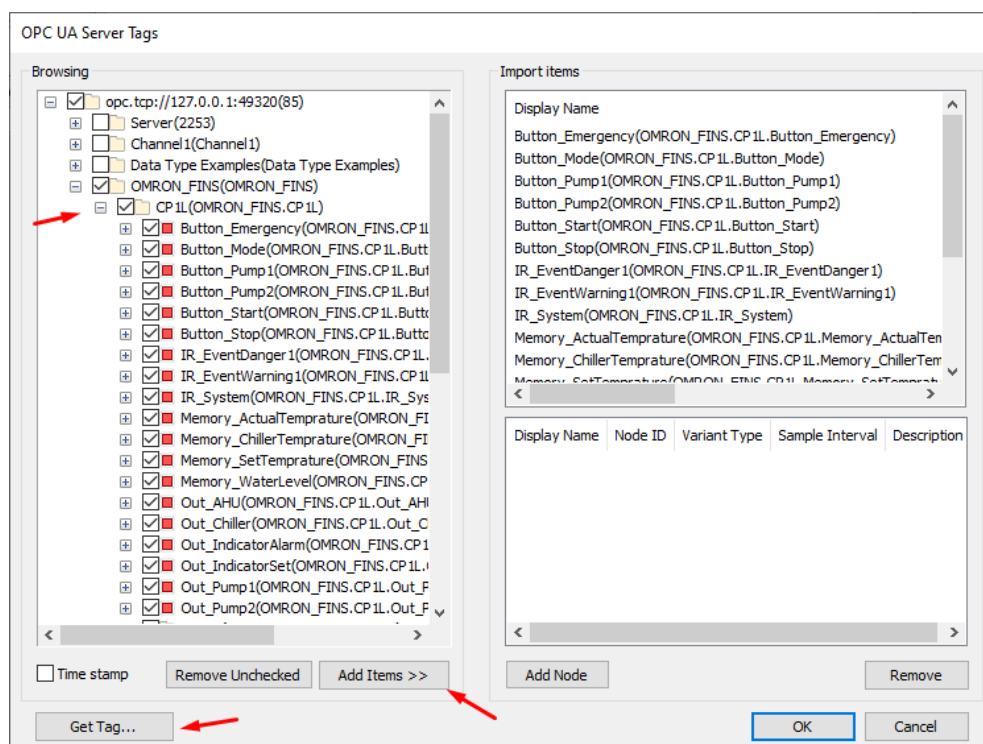
- Setelah itu pada bagian IP klik **Settings**. Tuliskan IP dan Port yang sesuai dengan yang ada di OPC UA Configuration Manager. Yakni 127.0.0.1 dan 49320 kemudian klik **Security, Authentication...**



- Pada window Security & Authentication, klik icon search lalu pilih **Basic256Sha256 – Sign & Encrypt**. Bila belum ada expand terlebih dahulu pilihan servernya dengan mengklik tanda +. Kemudian Klik **OK**. Hingga kembali ke window System Parameter Settings.



- Setelah ada device baru, klik device baru yang tadi sudah dibuat, kemudian pilih **Tag Manager** > bila ada popup window mengenai pemilihan string type klik **OK** saja untuk pilihan default.
- Bila terjadi Connection Failed Kembali ke windown **OPC UA Configuration** tab **Trusted Clients**, Trust Client baru yang ada. Lalu coba kembali untuk mengklik Tag Manager.
- Bila berhasil akan ada window seperti berikut. Bila tagnya belum tersedia klik Get Tag. Kemudian telusuri folder **opc.tcp://127.0.0.1:49320** > **Nama Channel** > **Nama Device** > Lalu centang Tag yang ingin diimport ke EasyBuilder Pro. Lalu Klik Add Items. Bila ada pop up untuk mengecek status dari tag, klik OK saja.



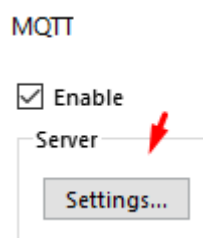
Setelah itu assign kan semua tag tadi ke setiap bit lamp, set bit, dll... yang ada pada design HMI EasyBuilder Pro. Kemudian untuk mengujinya lakukan Online Simulation pada tab **Project** > klik **Online Simulation**. Cobalah untuk merubah-rubah state dari Button yang ada di HMI ataupun yang ada di CX-Programmer. Bila hasilnya berubah secara sinkron, maka koneksi dari Program HMI ke Program PLC melalui OPC Server sudah berhasil. Silahkan *observe* juga value yang ada di Matrikon OPC UA Explorer ataupun OPC Quick Client dari KEPServerEX, value tersebut akan ikut berubah bersamaan dengan dirubahnya state suatu tag/address dari Software HMI/PLC.

7. Penggunaan MQTT di EasyBuilder Pro.

Penggunaan MQTT ini ditujukan untuk meng-“ekspos” data dari device agar kita dapat memiliki kendali pada suatu tag/address di platform lain. Misalkan website/aplikasi android atau iOS/aplikasi deskop dan yang lainnya.

- Pertama klik tab IIoT/Energy pada EasyBuilder Pro. Lalu klik **MQTT**. Catatan, perhatikan kembali tipe HMI yang digunakan pada project. Terkadang ada tipe HMI yang belum support MQTT. Namun di project ini dengan tipe HMI MT8070iE/MT8100iE (800 × 480) MQTT sudah di support.

- Defaultnya MQTT di project belum di enable, silahkan enable dengan cara mencentang pilihan enable. Kemudian klik **Settings** pada bagian Server.



- Isikan Settingnya sebagai berikut :

Tab General :

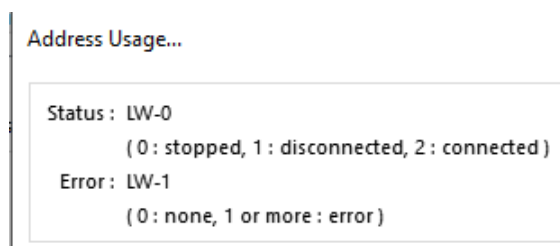
- Domain name : **test.mosquitto.org** (centang **use domain name** terlebih dahulu)

- Port : **8883**

- Client ID : Buat se-random mungkin. Dapat memanfaatkan juga wildcard yang tersedia seperti %0 atau %2, dll..

- Biarkan field lainnya default.

Tab Address : Silahkan pilih Address yang sesuai, diprojek ini dipilih address **LW-0** Sehingga status MQTT dapat dilihat di Address Usage seperti berikut :



Tab TLS/SSL : Centang **Enable**, biarkan field lainnya default.

Tab System Topic : biarkan default.

Kemudian klik **OK**.

Jika sudah, kemudian saatnya untuk men-setup MQTT Topicnya. Pada sebuah topic akan ada 2 aksi yang dapat dilakukan oleh user yakni Publish dan Subscribe (Pub/Sub). Untuk publish

berfungsi untuk mempush data dari HMI ke Topic MQTT. Sedangkan untuk subscribe berfungsi untuk membaca data dari Topic MQTT dan dituliskan ke HMI. Pertama lakukan setup untuk Topic Publisher. Klik tab **Topic Publisher** pada MQTT Topic lalu klik **New**. Lalu isikan konfigurasinya seperti berikut :

Nickname : Silahkan disesuaikan

Topic : Silahkan disesuaikan juga, usahakan buat topic dengan konfigurasi 3 atau lebih level path. Misalkan path1/path2/path3.

Sisanya, silahkan ikuti pengaturan tab general dibawah.

The screenshot shows the 'MQTT Topic Publisher Object's Properties' dialog box with the 'General' tab selected. The 'Nickname' field is set to 'Device Status'. The 'Topic' field is set to 'scada-fahm-08/status/device'. The 'Sending mode' is set to 'Address (Auto.)'. The 'Value-trigger-based' checkbox is checked, and the 'Time-based' checkbox is unchecked. The 'Compression type' is set to 'None'. The 'QoS' is set to '0'. The 'Retain message' checkbox is checked. The 'Content format' is set to 'JSON (Simple)'. The 'Include timestamp' and 'Use top-level key "id" for all addresses' checkboxes are unchecked.

Pada tab Address silahkan tambahkan Address apa saja yang ingin di ekspos datanya ke Topic MQTT. Berikut ini adalah Address yang di publish pada project Water Cooled Chiller System.

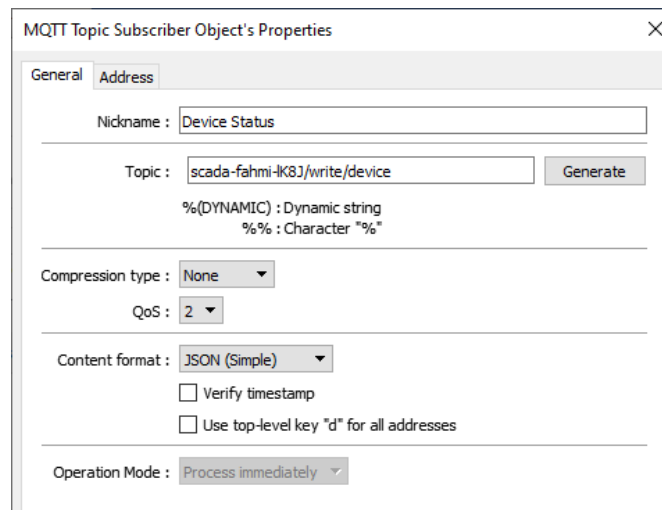
Name	Device name	Address	Address format	Ac
buttonEmergency	OPC UA Client	Button_Emergen...	BOOL	1
buttonMode	OPC UA Client	Button_Mode(O...	BOOL	1
buttonPump1	OPC UA Client	Button_Pump1(...	BOOL	1
buttonPump2	OPC UA Client	Button_Pump2(...	BOOL	1
buttonStart	OPC UA Client	Button_Start(O...	BOOL	1
buttonStop	OPC UA Client	Button_Stop(OM...	BOOL	1
flagDanger	OPC UA Client	IR_EventDanger...	BOOL	1
flagWarning	OPC UA Client	IR_EventWarnin...	BOOL	1
flagSystem	OPC UA Client	IR_System(OMR...	BOOL	1
outAHU	OPC UA Client	Out_AHU(OMRO...	BOOL	1
outChiller	OPC UA Client	Out_Chiller(OMR...	BOOL	1
outAlarm	OPC UA Client	Out_IndicatorAl...	BOOL	1
outSet	OPC UA Client	Out_IndicatorSe...	BOOL	1
outPump1	OPC UA Client	Out_Pump1(OM...	BOOL	1
outPump2	OPC UA Client	Out_Pump2(OM...	BOOL	1
memoryActualTe...	OPC UA Client	Memory_ActualT...	UINT	1
memoryChillerTemp	OPC UA Client	Memory_ChillerT...	UINT	1
memorySetTemp	OPC UA Client	Memory_SetTem...	UINT	1

- Bila sudah selesai lanjut ke setup **Topic Subscriber** lalu klik **New**.
- Pada bagian Topic Subscriber kurang lebih konfigurasinya adalah sebagai berikut:

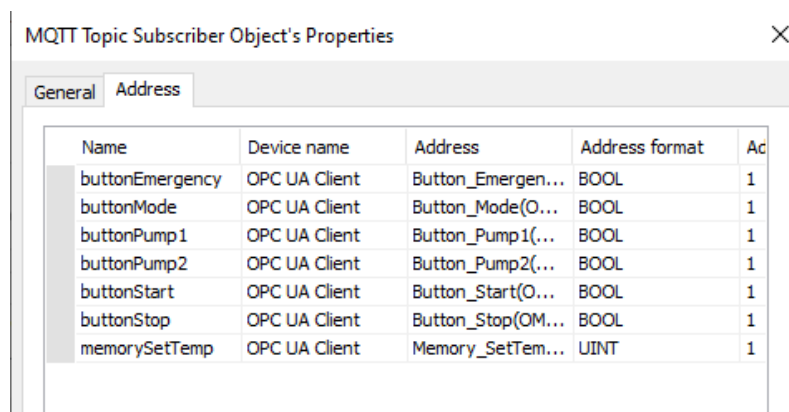
Nickname : Silahkan disesuaikan

Topic : Silahkan disesuaikan juga, usahakan buat topic dengan konfigurasi 3 atau lebih level path. Misalkan path1/path2/path3. Boleh disamakan dengan nama Topic yang di Publish.

Sisanya silahkan ikuti konfigurasi dibawah untuk Tab General.



Untuk Tab Address disarankan hanya beberapa address yang diperlukan saja. Pasalnya Topic Subscribe berfungsi untuk mengetahui perubahan value yang dilakukan dari platform lain. Sehingga bila semua address dimasukan disini, maka dapat terjadi Lag pada HMI yang digunakan karena terlalu banyak proses yang dijalankan. Berikut ini adalah Address yang di subscribe pada project Water Cooled Chiller System.



Name	Device name	Address	Address format	Action
buttonEmergency	OPC UA Client	Button_Emergen...	BOOL	1
buttonMode	OPC UA Client	Button_Mode(O...	BOOL	1
buttonPump1	OPC UA Client	Button_Pump1(...	BOOL	1
buttonPump2	OPC UA Client	Button_Pump2(...	BOOL	1
buttonStart	OPC UA Client	Button_Start(O...	BOOL	1
buttonStop	OPC UA Client	Button_Stop(OM...	BOOL	1
memorySetTemp	OPC UA Client	Memory_SetTem...	UINT	1

Bila sudah klik OK. Lalu coba jalankan melalui **Online Simulator** pada tab **Project**. Untuk menguji apakah data tersebut sudah terpublish dan bisa di subscribe, kita dapat menggunakan layanan MQTT Websocket Client pada URL berikut ini :

<http://www.hivemq.com/demos/websocket-client/>

- Buka URL diatas, kemudian isikan konfigurasi Connectionnya sebagai berikut ini :

Host : **test.mosquitto.org**

Port : **8081**

ClientID : Dibuat serandom mungkin.

Kemudian klik Connect.

- Kemudian pada bagian Subscriptions, isikan dengan nama topik yang ada pada Topic Publisher di EasyBuilder. Bila sudah, ketika project sudah dijalankan dengan Online Simulator maka di tab Messages di Web Client, akan muncul pesan mengenai state device di HMInya seperti berikut ini :



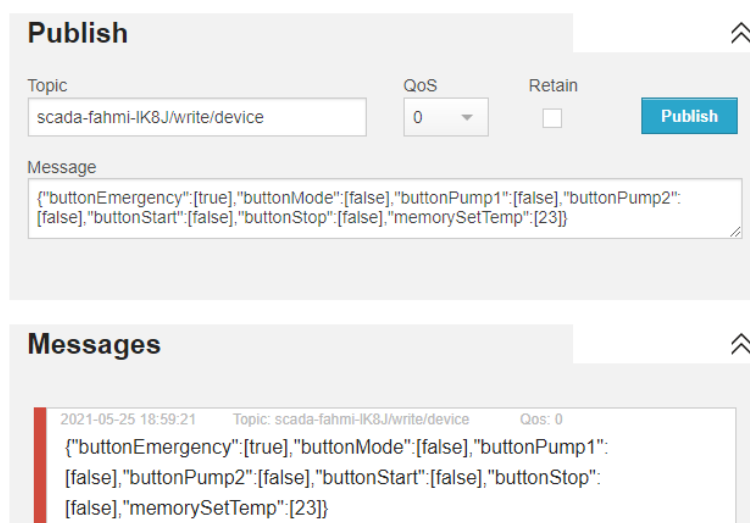
- Sedangkan untuk menguji bagian publish gunakan konfigurasi/format seperti berikut :

Topic : Nama topic yang di subscribe di HMI

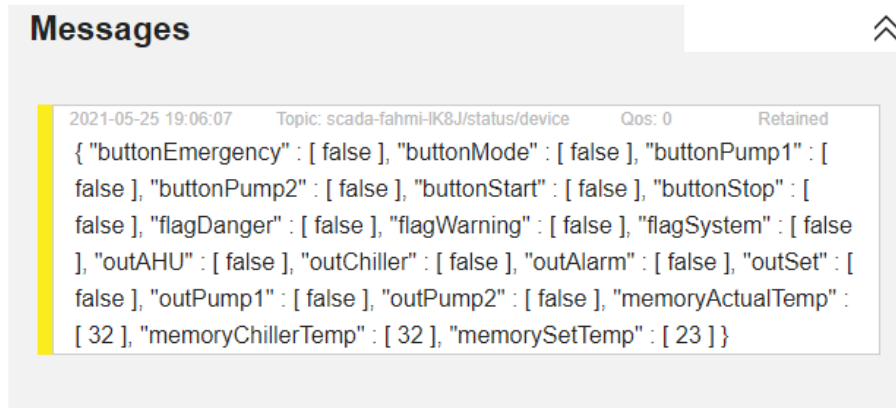
QOS : 0 (atau yang 1 atau 2, menyesuaikan)

Message : { "Nama":[state/value], "Nama":[state/value] } (Refer kembali ke tab address di MQTT Subscriber. Pastikan jumlah device yang ditulis di message sama. Valuenya disesuaikan dengan tipe data addressnya.

Bila berhasil mempublish maka tampilannya akan seperti ini:



Juga value atau state HMI akan berubah :



Dengan demikian, peng-“eksposan” data device dari HMI ke Topic MQTT telah selesai dilaksanakan.

8. SCADA Berbasis Web.

Dalam membuat SCADA berbasis web, akan dimanfaatkan sebuah library bernama Paho JavaScript - MQTT Client Library. Untuk menggunakannya cukup sederhana (selama memiliki pengetahuan dasar HTML, CSS dan Javascript ES6).

- Pertama, Buat file HTML dengan mengincludekan library PAHO MQTT Client Library di bawah body

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-
scale=1.0" />
    <title>Water Cooled Chiller - UAS SCADA</title>
    <meta content="width=device-width, initial-scale=1, maximum-
scale=1, user-scalable=no" name="viewport" />
    <meta name="description" content="Web based control system for
Water Cooled Chiller" />
  </head>

  <body>
  </body>
  <!-- Paho MQTT Client -->
  <script src="https://cdnjs.cloudflare.com/ajax/libs/paho-
mqtt/1.0.1/mqttws31.min.js" type="text/javascript"></script>
</html>
```

- Kemudian buka file HTML tersebut di [Local Webserver](#), Kemudian buka Chrome DevTools dengan menekan klik kanan > Inspect atau (CTRL + SHIFT + I).

- Berdasarkan [dokumentasinya](#) Berikut penjelasan untuk melakukan koneksi pada broker MQTT melalui Websocket Client sesuai dengan settingan project Water Cooled Chiller System.

```
// Create a client instance: Broker, Port, Random Client ID
client = new Paho.MQTT.Client("test.mosquitto.org", 8081, clientId);

// set callback handlers
client.onConnectionLost = function(responseObject) {
    console.log("Connection Lost: " + responseObject.errorMessage);
    // Do Something else, give warning, etc ...
}

client.onMessageArrived = function(message) {
    console.log("Message Arrived: " + message.payloadString);
    // Do Something else, Manipulate DOM, give warning, store to database,
    etc ...
}

// Called when the connection is made
function onConnect() {
    console.log("Conncted");
    // Do Something else, give warning, etc ...
}

// Connect the client, providing an onConnect callback
client.connect({
    onSuccess: onConnect,
    useSSL: true
});
```

Karena SCADA berbasis Web ini di deploy ke firebase hosting yang domainnya sudah memiliki SSL (https://) maka konektivitas dari MQTT melalui Websocket perlu menggunakan SSL agar tidak terjadi Mixed-Content/Insecure Request.

Sisanya dari SCADA berbasis Web ini adalah mendesign web lalu memanipulasi elemen pada web agar dapat tampil sesuai dengan data yang diterima/dikirim.

9. Store data MQTT ke Firebase Realtime Database.

Pada potongan kode diatas, terdapat sebuah keterangan ketika callback *onMessageArrived* kita dapat melakukan store data ke realtime Database Layanan milik Firebase. Caranya dapat diikuti seperti berikut sesuai dengan dokumentasi :

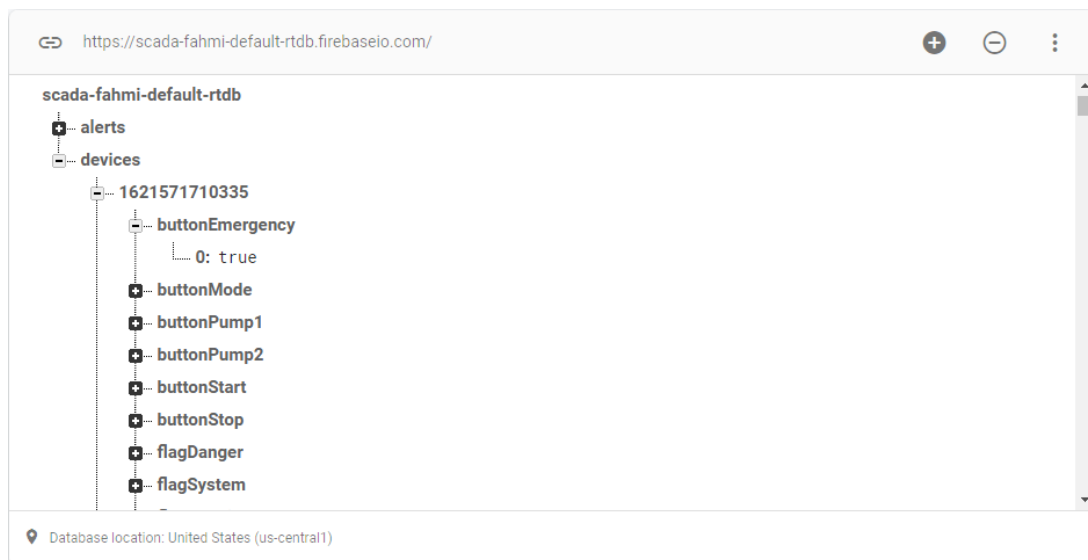
- Login ke akun Google Kemudian buka [Firebase Console](#).
- Buat project di Firebase Console dengan panduan seperti pada link [berikut](#). Jangan lupa include juga library javascript milik firebase ke proyek web yang dibuat.

- Konfigurasi Realtime Database (pada project yang dibuat) seperti pada panduan di link [berikut](#) ini.

Bila sudah, berdasarkan dokumentasi penggunaan library Firebase Realtime Database berikut adalah cara untuk menyimpan datanya yang disisipkan pada callback *onMessageArrived* :

```
client.onMessageArrived = (message) => {  
  datas = JSON.parse(message.payloadString);  
  firebase  
    .database()  
    .ref(`devices/${new Date().getTime()}`)  
    .once("value", (snapshot) => {  
      if (!snapshot.exists()) {  
        firebase.database().ref(`devices/${new  
Date().getTime()}`).set(datas);  
      }  
    });  
}
```

Sehingga bila dimonitor pada Realtime Database di Console Firebase akan seperti berikut ketika datangnya data :



Data tersebut kemudian dapat ditampilkan ke web atau di ekspor ke file Excel, PDF, dan yang lainnya memanfaatkan penggunaan library javascript.

Referensi/Pranala Luar

Inspirasi Design Plant : <https://www.maplesystems.com/product/modelname/cmt3072x>

Tutorial Konfigurasi CX-Programmer, OPC Server dan EasyBuilder : Penjelasan dari Mata kuliah SCADA oleh Bapak Muhammad Adli Rizqullah, S.Pd., MT.

Dokumentasi PAHO MQTT Javascript Library :

<https://www.eclipse.org/paho/files/jsdoc/Paho.MQTT.Client.html>

Dokumentasi Setup Firebase di Web : <https://firebase.google.com/docs/web/setup>

Dokumentasi Firebase Realtime Database :

<https://firebase.google.com/docs/database/web/start>

Link Download KEPServerEX 6 : <https://www.kepware.com/en-us/products/kepserverex/>

Link Download Eltima Software : <https://www.virtual-serial-port.org/>

Link Download Matrikon OPC Explorer :

<https://www.matrikonopc.com/downloads/176/software/index.aspx>

Link Download Matrikon OPC UA Explorer :

<https://www.matrikonopc.com/opc-ua/products/opc-ua-explorer.aspx>

Link Download EasyBuilder Pro :

<https://www.weintek.com/globalw/Download/Download.aspx>

Link Download CX-One : <https://industrial.omron.eu/en/products/cx-one>