

## **TUGAS AKHIR**

### **PERANCANGAN SIMULATOR TELEKONTROL MENGUNAKAN MODUL *SCHNEIDER AUTOMATION* SERVER SEBAGAI KONTROL *MULTI MODBUS DEVICE***

*Diajukan untuk memenuhi tugas-tugas dan syarat – syarat  
untuk memperoleh gelar sarjana pada program studi teknik elektro fakultas teknik  
universitas muhammadiyah sumatera utara*

Disusun oleh:

**Muhammad Mirza**

**1407220102**



# **UMSU**

Unggul | Cerdas | Terpercaya

**PROGRAM STUDI TEKNIK ELEKTRO  
FAKULTAS TEKNIK  
UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA  
MEDAN  
2018**

**LEMBAR PENGESAHAN**

**TUGAS AKHIR**


**PERANCANGAN SIMULATOR TELEKONTROL MENGGUNAKAN  
MODUL *SCHNEIDER AUTOMATION SERVER* SEBAGAI KONTROL  
*MULTI MODBUS DEVICE***

*Diajukan Untuk Melengkapi Syarat-Syarat  
Memperoleh Gelar Sarjana Teknik (S.T)*

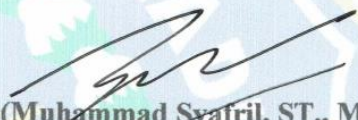
**Telah Diuji dan Disidangkan Pada Tanggal :  
( 22 SEPTEMBER 2018 )**

Oleh :  
**Muhammad Mirza**  
1407220064

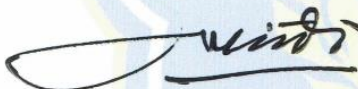
Pembimbing I

  
(Rohana, ST., MT)

Pembimbing II

  
(Muhammad Syafril, ST., MT)

Penguji I

  
(Ir. Yusniati MT)

Penguji II

  
(Ir. Zul Arsil Siregar)

Diketahui dan Disahkan  
Ketua Prodi Teknik Elektro

  
(Faisal Irsan Pasaribu ST., MT)

**PROGRAM STUDI TEKNIK ELEKTRO  
FAKULTAS TEKNIK  
UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA  
MEDAN  
2018**

## PERNYATAAN ORISINALITAS SKRIPSI

Saya yang bertanda tangan dibawah ini :

Nama : Muhammad Mirza  
Tempat/tgl. Lahir : Medan, 08 Juni 1996  
NPM : 1407220102  
Bidang Keahlian : Instrumentasi Dan Pengendalian  
Program Studi : Teknik Elektro  
Fakultas : Teknik



Menyatakan dengan sesungguhnya dan sejujurnya, bahwa laporan Tugas Akhir yang berjudul :

### **PERANCANGAN SIMULATOR TELEKONTROL MENGGUNAKAN MODUL *SCHNEIDER AUTOMATION SERVER* SEBAGAI KONTROL *MULTI MODBUS DEVICE***

Dengan sebenar-benarnya bahwa sepanjang pengetahuan saya dan berdasarkan hasil penelusuran berbagai karya ilmiah, gagasan dan masalah ilmiah yang diteliti dan diulas di dalam Naskah Skripsi ini adalah asli dari pemikiran saya. Tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu Perguruan Tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis dikutip dalam naskah ini dan disebutkan dalam sumber kutipan dan daftar pustaka.

Apabila ternyata di dalam naskah Skripsi ini dapat dibuktikan terdapat unsur-unsur jiplakan, saya bersedia Skripsi ini dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, pasal 25 ayat 2 dan pasal 70).

Medan, 19 September 2018



Saya yang menyatakan,

  
MUHAMMAD MIRZA

## ABSTRAK

*MODBUS Protocol merupakan protokol komunikasi yang berperan penting terhadap alat – alat kendali elektronis. Penelitian ini bertujuan untuk menganalisa reliabilitas/kehandalan Automation Server dan kekuatan sinyal nirkabel dalam mengawasi dan mengendalikan perangkat slave (Power meter, Module I/O, Thermostat, PLC) serta menganalisa penyesuaian pembacaan Modbus point masing – masing register type dari masing – masing perangkat slave. Simulasi yang dilakukan dengan mengatur nilai baudrate dan transfer rate serta jumlah Modbus Point pada masing – masing perangkat slave yang berbeda– berbeda setiap port-nya pada Automation Server dan simulasi dilakukan dengan mengatur jarak nirkabel antara smart device dan TP-Link serta simulasi dilakukan dengan pemrograman dan grafis yang sederhana untuk melakukan pengawasan dan pengendalian. Hasil penelitian ini menunjukkan bahwa (1) semakin jauh jarak antara smart device dan TP-Link MR3420, sensitivitas kekuatan sinyal (dbm) semakin kecil. Kekuatan sinyal mengalami penurunan 94,7% pada jarak 150 meter. Semakin jauh jarak antara smart device dan TP-Link MR3420, kualitas sinyal semakin kecil. Kualitas sinyal mengalami penurunan 48,5% pada jarak 150 meter (2) semakin besar nilai baud rate dan transfer rate maka receive timeout semakin kecil, dan untuk penyesuaian pembacaan nilai input/output baik digital maupun analog pada perangkat slave adalah dengan mengatur tipe register pada masing – masing Modbus Point (3) pemrograman dan pembuatan grafis serta kombinasi dengan beberapa perangkat slave Modbus berhasil dilakukan dengan perangkat lunak internal dan teknik Binding. Indikator keberhasilan program dan grafis dinilai melalui penyesuaian pembacaan dan pengendalian Input/Output baik digital maupun analog pada perangkat-perangkat slave Modbus*

*Kata kunci: MODBUS Protocol, Receive Timeout, Kekuatan sinyal nirkabel*



## KATA PENGANTAR



Assalamualaikum warahmatullahi wabarakatuh, Tidak ada kata lain untuk menyampaikan rasa terima kasih yang sebesar-besarnya kepada Allah SWT kecuali ucapan puji syukur atas segala nikmat dengan curahan kasih sayang-Nya atas selesainya penulisan skripsi ini dengan baik dengan judul "PERANCANGAN SIMULATOR TELEKONTROL MENGGUNAKAN MODUL *SCHNEIDER AUTOMATION SERVER* SEBAGAI KONTROL *MULTI MODBUS DEVICE*".

Penulisan skripsi ini dimaksudkan guna melengkapi sebagian persyaratan meraih gelar sarjana di Universitas Muhammadiyah Sumatera Utara Fakultas Teknik Program Studi Teknik Elektro.

Di dalam menyusun Penelitian ini penulis tidak dapat melupakan jasa orang-orang yang telah ikut berperan serta sehingga Penelitian ini dapat selesai. Oleh karena itu, pada kesempatan ini penulis ingin mengucapkan rasa terima kasih yang sebesar-besarnya kepada :

1. Kedua orang tua saya yakni Bapak Alm. Chairil Anwar dan Ibu Jauhariyah yang selalu mendoakan saya dan mendidik saya dengan baik.
2. Bapak Drs. Agussani, M.AP. selaku Rektor Universitas Muhammadiyah Sumatera Utara.
3. Bapak Munawar Alfansury Siregar, ST., MT. selaku Dekan Fakultas Teknik Elektro Universitas Muhammadiyah Sumatera Utara.
4. Bapak Faisal Irsan Pasaribu, ST., MT. selaku Ketua Program Studi Teknik Elektro Universitas Muhammadiyah Sumatera Utara.

5. Bapak Partaonan Harahap, ST., MT. selaku Sekretaris Program Studi Teknik Elektro Universitas Muhammadiyah Sumatera Utara.
6. Ibu Rohana ST., MT. selaku Dosen Pembimbing I yang telah memberikan bimbingan dan pengarahan serta meluangkan waktu kepada saya dalam menyelesaikan Penelitian.
7. Bapak Muhammad Safril, ST., MT. selaku Dosen Pembimbing II yang telah memberikan bimbingan dan pengarahan serta meluangkan waktu kepada saya dalam menyelesaikan Penelitian.
8. Bapak Ir. Zulfadli Pelawi M.T selaku direktur CV. MEDCON-E yang berperan penting dalam peminjaman dan pengarahan penggunaan alat untuk mendukung dalam melakukan penelitian ini.
9. Dan juga kepada setiap orang yang tidak dapat disebutkan satu persatu terlebih kepada orang yang penulis sayangi, atas dukungan dan doanya Penelitian ini dapat selesai.

Penulis menyadari bahwa Penelitian ini masih jauh dari sempurna, sehingga masih banyak hal yang perlu dikaji lebih lanjut untuk pengembangan penelitian di bidang ini. Akhirnya penulis berharap semoga Penelitian ini memperkaya khasanah ilmu pengetahuan di bidang Sistem Kontrol Peralatan Listrik.

Wassalamua'alaikum Wr. Wb.

Medan, September 2018

Penulis

Muhammad Mirza

## DAFTAR ISI

<b>ABSTRAK .....</b>	<b>i</b>
<b>KATA PENGANTAR.....</b>	<b>ii</b>
<b>DAFTAR ISI.....</b>	<b>iv</b>
<b>DAFTAR GAMBAR.....</b>	<b>ix</b>
<b>DAFTAR TABEL .....</b>	<b>xiv</b>
<b>DAFTAR LAMPIRAN .....</b>	<b>xvii</b>
<b>BAB I PENDAHULUAN.....</b>	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	2
1.3 Tujuan Penelitian .....	3
1.4 Batasan Masalah.....	3
1.5 Manfaat Penulisan.....	5
1.5.1 Bagi Mahasiswa .....	5
1.5.2 Bagi Universitas Muhammadiyah Sumatera Utara.....	6
1.5.3 Bagi Masyarakat Umum .....	6
1.6 Sistematika Penulisan.....	7
<b>BAB II TINJAUAN PUSTAKA.....</b>	<b>9</b>
2.1 Tinjauan Pustaka Relevan .....	9
2.2 Pengenalan <i>Modbus</i> .....	12
2.2.1 Prinsip Dasar <i>Modbus</i> .....	15
2.2.1.1 Mode Transmisi Serial.....	17

2.2.1.1.1	Mode RTU ( <i>Remote Terminal Unit</i> ).....	17
2.2.1.1.2	Mode ASCII ( <i>American Standard Code Information Interchange</i> ).....	18
2.2.1.2	<i>Modbus Message Framing</i> .....	18
2.2.1.2.1	ASCII Framing .....	18
2.2.1.2.2	RTU Framing .....	19
2.2.1.3	<i>Address Field</i> .....	21
2.2.1.4	<i>Function Field</i> .....	22
2.2.1.5	Fungsi Dasar <i>Modbus</i> .....	22
2.2.1.6	<i>Data Field</i> .....	23
2.2.1.7	<i>Exception Response</i> .....	23
2.2.2	<i>Modbus TCP</i> .....	25
2.2.2.1	TCP/IP .....	25
2.2.2.2	<i>Modbus RTU over TCP</i> .....	25
2.2.2.3	ADU & PDU .....	26
2.2.2.4	MBAP Header .....	26
2.2.3	WLAN ( <i>Wireless Local Area Network</i> ) .....	28
2.2.3.1	Standarisasi WLAN .....	28
2.3	Perangkat Keras .....	30
2.3.1	<i>Schneider Automation Server</i> .....	30
2.3.2	<i>Power Analyzer (Power Meter) Circutor CVM-NRG96</i> .....	33
2.3.2.1	Bagian – Bagian Tombol Fungsi Pada CVM-NRG96.....	35
2.3.2.2	Bagian – Bagian Terminal Pada CVM-NRG96 .....	37
2.3.2.3	Pengawatan Pada Terminal CVM –NRG96 .....	38
2.3.2.4	<i>Modbus Protocol</i> Pada CVM-NRG96 .....	39



2.3.3	<i>Thermostat Schneider TC-300</i> .....	42
2.3.3.1	Bagian – Bagian <i>Thermostat TC-300</i> .....	43
2.3.3.2	Pengawatan Pada <i>Thermostat TC-300</i> .....	44
2.3.3.3	<i>Modbus Protocol</i> Pada <i>Thermostat TC-300</i> .....	44
2.3.3.4	<i>Modbus Register</i> Pada <i>Thermostat TC-300</i> .....	45
2.3.4	<i>WELLPRO Module I/O WP9038ADAM</i> .....	46
2.3.5	PLC Schneider TM221ME16R.....	47
2.3.5.1	Bagian – Bagian PLC Modicon TM221ME16R .....	48
2.3.5.2	Metode Pemrogram PLC .....	50
2.3.6	TP-Link MR3420 <i>Wireless</i> dan <i>Router</i> .....	50
2.3.6.1	Bagian – Bagian TP-Link MR3420 .....	51
2.3.7	<i>Current Transformer (CT)</i> .....	52
2.4	Perangkat Lunak.....	54
2.4.1	<i>Lizard Wifi Scanner</i> .....	54
2.4.2	<i>Netgear Wifi Analytics</i> .....	56
2.4.3	<i>Struxureware Building Operation Workstation</i> .....	56
2.4.4	<i>SoMachine Basic</i> .....	59
2.4.4.1	Bahasa Pemrograman <i>Instruction List</i> .....	61
2.4.4.2	Bahasa Pemrograman <i>Ladder Diagram</i> .....	61
2.4.5	<i>Wellpro Debugging Software</i> .....	62
2.4.6	<i>Struxureware Building Operation Tech Tool</i> .....	63
2.4.7	<i>Struxureware Building Operation Device Administrator</i> .....	65
<b>BAB III</b>	<b>PERANCANGAN SISTEM</b> .....	<b>66</b>
3.1	Pendahuluan .....	66

3.2 Perancangan Sistem .....	67
3.2.1 Perancangan <i>Hardware</i> .....	67
3.2.1.1 Diagram Alir Pengaturan PC dan <i>TP-Link</i> .....	68
3.2.1.2 Diagram Alir Konfigurasi PLC M221 .....	71
3.2.1.3 Diagram Alir Konfigurasi <i>Wellpro Module I/O</i> .....	75
3.2.1.4 Konfigurasi <i>Thermostat TC-300</i> .....	77
3.2.1.5 Diagram Alir Konfigurasi <i>Power Meter CVM-NRG96</i> .....	79
3.2.1.6 Diagram Alir Konfigurasi <i>Automation Server</i> .....	85
3.2.2 Perancangan Perangkat Lunak ( <i>Software</i> ) .....	89
3.2.2.1 Pembuatan <i>Interface MODBUS Master Network Port A</i> .....	91
3.2.2.2 Pembuatan <i>Interface MODBUS Master Network Port B</i> .....	107
3.2.2.3 Pembuatan <i>Interface MODBUS TCP Network</i> .....	112
3.2.2.4 Pemrograman dan Pembuatan Grafis .....	116
3.2.2.5 Proses <i>Binding</i> Pada <i>Input/Output</i> Komponen Program dan Komponen Grafis .....	121
 <b>BAB IV ANALISA DAN PENGUJIAN .....</b>	<b>126</b>
4.1 Pengujian Kekuatan Sinyal WLAN ( <i>Wireless Local Area Network</i> ) .....	126
4.2 Pengujian Konfigurasi Kecepatan <i>Automation Server</i> dan <i>Register Type</i> .....	129
4.2.1 Analisa Kecepatan <i>Receive Timeout</i> Pada Port COM-A .....	130
4.2.2 Analisa Kecepatan <i>Receive Timeout</i> Pada Port COM-B .....	131
4.2.3 Analisa Kecepatan <i>Receive Timeout</i> Pada Port Ethernet .....	132
4.2.4 Pengujian <i>Register Type</i> Pada Masing – Masing Perangkat Modbus .....	135
4.3 Pengujian Program dan Grafis .....	139
4.3.1 Pengujian Program .....	139

4.3.1.1 Pengujian Blok Diagram Fungsi dan Fungsi Dasar Modbus Pada Program <i>Over Current</i> (Arus Lebih) .....	139
4.3.1.2 Pengujian Blok Diagram Fungsi dan Fungsi Dasar Modbus Pada Program <i>Starting Generator</i> .....	141
4.3.1.3 Pengujian Blok Diagram Fungsi dan Fungsi Dasar Modbus Pada Program <i>Temperature Alarm</i> .....	144
4.3.1.4 Pengujian Blok Diagram Fungsi dan Fungsi Dasar Modbus Pada Program <i>Button Booster Water Pump dan Compressor Chiller</i> .....	146
4.3.2 Pengujian Grafis.....	149
4.3.2.1 Pengujian Grafis Terhadap Perangkat <i>Power Meter Analyzer CVM-NRG96</i> .....	150
4.3.2.2 Pengujian Grafis Terhadap Perangkat <i>Thermostat TC-300</i> .....	151
4.3.2.3 Pengujian Grafis Terhadap Perangkat PLC M221 dan Wellpro <i>Module Input/Output</i> .....	152
4.3.2.4 Pengujian Program dan Grafis Menggunakan Ponsel Pintar ( <i>Smartphone</i> ) .....	153
<b>BAB V PENUTUP .....</b>	<b>159</b>
5.1 Kesimpulan .....	159
5.2 Saran.....	160
<b>DAFTAR PUSTAKA</b>	
<b>LAMPIRAN</b>	

## DAFTAR GAMBAR

Gambar 2.1	Arsitektur Komunikasi Modbus Secara Umum.....	13
Gambar 2.2	Arsitek Komunikasi Antara <i>Master</i> dengan <i>Slave</i> .....	16
Gambar 2.3	Interval $T_{3,5}$ dalam <i>frame</i> .....	19
Gambar 2.4	Interval $T_{1,5}$ antar karakter dalam <i>frame</i> .....	20
Gambar 2.5	Paket yang akan dikirimkan melalui internet/MBAP <i>Header</i> .....	27
Gambar 2.6	Tabel Spesifikasi WiFi Standar IEEE.....	29
Gambar 2.7	Tabel <i>Channel</i> WiFi pada standar 802.11b/g .....	29
Gambar 2.8	<i>Automation Server</i> .....	30
Gambar 2.9	<i>Power Analyzer Circutor CVM-NRG96</i> .....	34
Gambar 2.10	Tampak Depan CVM NRG96 .....	35
Gambar 2.11	Bagian – Bagian Terminal Pada CVM NRG96.....	37
Gambar 2.12	Pengawatan pada terminal CVM-NRG96 .....	38
Gambar 2.13	Thermostat TC300 .....	42
Gambar 2.14	Tampak Depan TC-300 .....	43
Gambar 2.15	Pengawatan pada terminal Thermostat TC-300.....	44
Gambar 2.16	WP9038ADAM .....	46
Gambar 2.17	PLC Schneider TM221ME16R .....	48
Gambar 2.18	Bagian - Bagian PLC Schneider TM221ME16R .....	48
Gambar 2.19	TP-Link MR3420 <i>Wireless and Router</i> (tampak depan) .....	51
Gambar 2.20	TP-Link MR3420 (tampak belakang).....	51
Gambar 2.21	<i>Current Transformer (CT)</i> .....	53
Gambar 2.22	Ilustrasi Arus Mengalir Pada CT 300A/5A .....	53



Gambar 2.23 Tampilan Kerja Pada <i>Software Lizard Wifi Scanner</i> .....	55
Gambar 2.24 Tampilan Kerja Pada Perangkat Lunak <i>Netgear Wifi Analytics</i> .....	56
Gambar 2.25 Tampilan Beranda Pada Perangkat Lunak <i>Workstation</i> .....	57
Gambar 2.26 Tampilan Ruang Kerja Pada Perangkat Lunak <i>Workstation</i> .....	59
Gambar 2.27 Tampilan Beranda Perangkat Lunak <i>Somachine Basic</i> .....	60
Gambar 2.28 Bahasa pemrograman <i>Intruction List</i> pada <i>Somachine Basic</i> .....	61
Gambar 2.29 Bahasa pemrograman <i>Ladder Diagram</i> pada <i>Somachine Basic</i> .....	62
Gambar 2.30 Tampilan Halaman Pemindai Pada <i>Wellpro Debugging</i> .....	62
Gambar 2.31 Ilustrasi Piranti Lunak <i>Tech Tool</i> Pada <i>Smart Device</i> .....	63
Gambar 2.32 Tampilan <i>Dashboard</i> Pada <i>SBO Tech Tool</i> .....	64
Gambar 2.33 Tampilan Awal <i>SBO Device Administrator</i> .....	65
Gambar 3.1 Blok Diagram Sistem .....	66
Gambar 3.2 Diagram Alir Pengaturan PC dan <i>TP-Link</i> .....	68
Gambar 3.3 Pengaturan <i>IP Address</i> pada PC/Laptop .....	69
Gambar 3.4 Pengaturan <i>TP-Link</i> sebagai pemancar Wifi .....	70
Gambar 3.5 Diagram Alir Konfigurasi PLC M221 .....	71
Gambar 3.6 Diagram Pengawatan Relay <i>Output</i> PLC M221 .....	72
Gambar 3.7 Pengaturan <i>IP Address</i> PLC di <i>Somachine Basic</i> .....	73
Gambar 3.8 Program PLC M221 .....	74
Gambar 3.9 Proses mengunduh program kedalam PLC .....	74
Gambar 3.10 Diagram Alir Konfigurasi <i>Wellpro Module I/O</i> .....	75
Gambar 3.11 Diagram Pengawatan Keluaran <i>Wellpro Module I/O</i> .....	76
Gambar 3.12 Tampilan konfigurasi parameter <i>Wellpro Module I/O</i> .....	77
Gambar 3.13 Diagram Alir Konfigurasi <i>Power Meter Analyzer CVM-NRG96</i> .....	79

Gambar 3.14 Diagram pengawatan <i>Power Analyzer</i> .....	80
Gambar 3.15 Diagram Alir Konfigurasi <i>Automation Server</i> .....	85
Gambar 3.16 Koneksi Personal Komputer dengan <i>Automation Server</i> .....	86
Gambar 3.17 Tampilan <i>Device Administrator</i> Pindai <i>Automation Server</i> .....	86
Gambar 3.18 Tampilan Untuk <i>Upgrade Device</i> Pada <i>Automation Server</i> .....	87
Gambar 3.19 Tampilan Opsi <i>New Database</i> Pada <i>Automation Server</i> .....	87
Gambar 3.20 Tampilan Proses <i>Upgrade Device</i> Pada <i>Automation Server</i> .....	87
Gambar 3.21 Tampilan <i>Opsi Network Setting</i> .....	88
Gambar 3.22 Tampilan Konfigurasi <i>Ethernet 1</i> Pada <i>Automation Server</i> .....	88
Gambar 3.23 Jendela Awal <i>SBO Workstation</i> .....	90
Gambar 3.24 Tampilan Konfigurasi Password Baru .....	91
Gambar 3.25 <i>Configuration Setting MODBUS Master Network Port A</i> .....	93
Gambar 3.26 Pengaturan pengenalan alamat perangkat <i>Thermostat TC300</i> .....	94
Gambar 3.27 Konfigurasi <i>MODBUS Point Room Temperature</i> .....	95
Gambar 3.28 Konfigurasi Modbus Point Fan Mode .....	96
Gambar 3.29 Konfigurasi <i>Modbus Point Eco Mode</i> .....	97
Gambar 3.30 Konfigurasi <i>MODBUS Point Set Temperature</i> .....	98
Gambar 3.31 Tampilan Nilai – Nilai <i>MODBUS Point Thermostat TC-300</i> .....	98
Gambar 3.32 Pengaturan pengenalan alamat perangkat <i>Power Meter Circutor</i> <i>CVM-NRG96</i> .....	99
Gambar 3.33 Konfigurasi <i>MODBUS Point Voltage (V L1)</i> .....	101
Gambar 3.34 Konfigurasi <i>MODBUS Point Current (A L1)</i> .....	102
Gambar 3.35 Konfigurasi <i>MODBUS Point Active Power (kW L1)</i> .....	103
Gambar 3.36 Konfigurasi <i>MODBUS Point Reactive Power (kVar L1)</i> .....	104
Gambar 3.37 Konfigurasi <i>MODBUS Point Power Factor (PF L1)</i> .....	105

Gambar 3.38 Konfigurasi <i>MODBUS Point Frequency</i> (Hz).....	106
Gambar 3.39 Tampilan Nilai – Nilai <i>MODBUS Point Power Meter Circutor CVM-NRG96</i> .....	106
Gambar 3.40 <i>Configuration Setting MODBUS Master Network Port B</i> .....	108
Gambar 3.41 Pengaturan Pengenalan Alamat <i>Wellpro Module I/O</i> .....	109
Gambar 3.42 Konfigurasi <i>MODBUS Point Digital Input 1</i> .....	110
Gambar 3.43 Konfigurasi <i>MODBUS Point Digital Output 1</i> .....	111
Gambar 3.44 Tampilan Nilai – Nilai <i>MODBUS Point Wellpro Module I/O</i> .....	112
Gambar 3.45 <i>Configuration Setting MODBUS TCP Network Port Ethernet</i> .....	113
Gambar 3.46 Pengaturan pengenalan alamat PLC M221.....	114
Gambar 3.47 Konfigurasi <i>MODBUS Point Digital Output 1 (Q0.0)</i> .....	115
Gambar 3.48 Tampilan Nilai – Nilai <i>MODBUS Point PLC M221</i> .....	115
Gambar 3.49 Program Kontrol <i>Multi MODBUS Device</i> .....	118
Gambar 3.50 Grafis Kontrol <i>Multi MODBUS Device</i> .....	121
Gambar 3.51 Tampilan <i>Binding</i> Komponen Program.....	123
Gambar 3.52 Tampilan <i>Binding</i> Komponen Grafis.....	125
Gambar 4.1 Grafik kekuatan sinyal dan kualitas sinyal antara laptop dan <i>TP-Link MR3420</i> .....	127
Gambar 4.2 Grafik kekuatan sinyal dan kualitas sinyal antara <i>Smartphone</i> dan <i>TP-Link MR3420</i> .....	128
Gambar 4.3 Grafik Perbandingan <i>receive timeout</i> port komunikasi <i>MODBUS COM-A</i> dan <i>MODBUS COM-B</i> .....	132
Gambar 4.4 Grafik Perbandingan <i>receive timeout</i> pada <i>Port Ethernet</i> masing – masing <i>transfer rate</i> .....	134
Gambar 4.5 Grafik Perbandingan <i>receive timeout</i> pada masing – masing port komunikasi <i>MODBUS</i> .....	135
Gambar 4.6 Program sebelum masukan nilai arus dibawah 5 Ampere .....	140
Gambar 4.7 Program setelah masukan nilai arus diatas 5 Ampere .....	141

Gambar 4.8 Program ketika nilai tegangan diatas 150 Volt dan <i>Switch Generator</i> dalam keadaan <i>OFF</i> .....	141
Gambar 4.9 Program ketika nilai tegangan diatas 150 Volt dan <i>Switch Generator</i> dalam keadaan <i>ON</i> .....	142
Gambar 4.10 Program ketika nilai tegangan dibawah 150 Volt dan <i>Switch Generator</i> dalam keadaan <i>OFF</i> .....	142
Gambar 4.11 Program ketika nilai tegangan dibawah 150 Volt dan <i>Switch Generator</i> dalam keadaan <i>ON</i> .....	143
Gambar 4.12 Program ketika nilai suhu berada diatas 18°C dan dibawah 25°C ..	144
Gambar 4.13 Program ketika nilai suhu berada diatas 25°C .....	145
Gambar 4.14 Program ketika nilai suhu berada dibawah 18°C .....	145
Gambar 4.15 Program <i>Button Booster Water Pump dan Compressor Chiller</i> pada saat semua <i>button</i> dalam keadaan <i>OFF</i> .....	147
Gambar 4.16 Program <i>Button Booster Water Pump dan Compressor Chiller</i> pada saat semua <i>button</i> dalam keadaan <i>ON</i> .....	147
Gambar 4.17 Tampilan Pembacaan Nilai – Nilai <i>Modbus Point Power Meter CVM-NRG96</i> Pada Grafis .....	149
Gambar 4.18 Tampilan Pembacaan dan Pengendalian Nilai – Nilai <i>Modbus Point Thermostat TC-300</i> Pada Grafis .....	151
Gambar 4.19 Tampilan Pembacaan dan Pengendalian Nilai – Nilai <i>Modbus Point PLC M221 dan Wellpro Modul I/O</i> Pada Grafis .....	152
Gambar 4.20 Tampilan <i>Login ke Automation Server</i> melalui perangkat lunak <i>SBO Tech Tool</i> pada ponsel pintar .....	155
Gambar 4.21 Tampilan beranda <i>tab system</i> pada perangkat lunak <i>SBO Tech Tool</i> setelah <i>login ke Automation Server</i> .....	156
Gambar 4.22 Tampilan grafis pengawasan dan pengendalian perangkat – perangkat <i>Modbus</i> melalui perangkat lunak <i>SBO Tech Tool</i> .....	156



## DAFTAR TABEL

Tabel 2.1	<i>ASCII Framing</i> .....	18
Tabel 2.2	<i>RTU Framing</i> .....	19
Tabel 2.3	Kode Fungsi <i>MODBUS</i> .....	23
Tabel 2.4	<i>Exception code</i> dalam <i>exception response</i> .....	24
Tabel 2.5	Pembacaan Parameter Listrik Pada CVM-NRG96 .....	35
Tabel 2.6	<i>Function Button</i> pada CVM-NRG96 .....	36
Tabel 2.7	Format untuk setiap <i>byte</i> dalam <i>mode RTU</i> .....	39
Tabel 2.8	Kode fungsi <i>MODBUS</i> pada CVM-NRG96 .....	40
Tabel 2.9	<i>MODBUS variable</i> pada CVM-NRG96.....	40
Tabel 2.10	<i>MODBUS variable (register)</i> pada <i>Thermostat TC-300</i> .....	45
Tabel 3.1	Daftar Alamat IP untuk perangkat yang didukung oleh <i>MODBUS TCP/IP</i> .....	67
Tabel 3.2	Daftar parameter untuk perangkat yang didukung <i>MODBUS RTU</i> ...	67
Tabel 3.3	Konfigurasi <i>Input/Output</i> Pada Pemrograman PLC M221 .....	73
Tabel 3.4	Parameter – parameter yang dapat diubah pada <i>Thermostat TC-300</i> .	78
Tabel 3.5	Pengalamatan <i>Input/Output</i> Pada Program Antarmuka .....	116
Tabel 3.6	Diagram Blok dan <i>Function</i> Pada Program .....	117
Tabel 3.7	Pengalamatan <i>bindname</i> pada komponen – komponen grafis .....	120
Tabel 3.8	Pengalamatan <i>Binding</i> komponen program dengan perangkat <i>MODBUS</i> dan komponen grafis.....	122
Tabel 3.9	Pengalamatan <i>Binding</i> komponen grafis dengan perangkat <i>MODBUS</i> dan komponen program .....	124
Tabel 4.1	Hasil kekuatan sinyal WLAN .....	126
Tabel 4.2	Perbandingan <i>receive timeout</i> port komunikasi <i>MODBUS COM-A</i> dan <i>MODBUS COM-B</i> .....	131

Tabel 4.3	Perbandingan <i>receive timeout</i> pada port Ethernet masing – masing <i>transfer rate</i> .....	134
Tabel 4.4	Perbandingan <i>receive timeout</i> pada masing – masing port komunikasi MODBUS .....	134
Tabel 4.5	Pengujian <i>register type</i> pada perangkat <i>Power meter Circutor CVM-NRG96</i> .....	135
Tabel 4.6	Pengujian <i>register type</i> pada perangkat <i>Thermostat TC-300</i> .....	137
Tabel 4.7	Pengujian <i>register type</i> pada perangkat <i>Wellpro Module I/O</i> .....	138
Tabel 4.8	Pengujian <i>register type</i> pada perangkat PLC M221.....	138
Tabel 4.9	Hasil pengujian respon fungsi dasar <i>Modbus program Over Current</i> .....	140
Tabel 4.10	Hasil pengujian Diagram Blok Fungsi Program <i>Over Current</i> .....	141
Tabel 4.11	Hasil pengujian respon fungsi dasar <i>Modbus program Starter Generator</i> .....	143
Tabel 4.12	Hasil pengujian Diagram Blok Fungsi Program <i>Starter Generator</i> .	144
Tabel 4.13	Hasil pengujian respon fungsi dasar Modbus program <i>Temperature Alarm</i> .....	146
Tabel 4.14	Hasil pengujian Diagram Blok Fungsi Program <i>Temperature Alarm</i> .....	146
Tabel 4.15	Hasil pengujian respon fungsi dasar Modbus program <i>Button Booster Water Pump dan Compressor Chiller</i> .....	148
Tabel 4.16	Hasil pengujian Diagram Blok Fungsi Program <i>Button Booster Water Pump dan Compressor Chiller</i> .....	148
Tabel 4.17	Hasil pengujian Fungsi Dasar <i>Modbus Point</i> pada <i>Power meter</i> untuk dibaca dan dikendalikan melalui Grafis.....	150
Tabel 4.18	Hasil pengujian Grafis terhadap nilai – nilai <i>Modbus Point Powermeter CVM-NRG96</i> .....	150
Tabel 4.19	Hasil pengujian Fungsi Dasar <i>Modbus Point</i> pada <i>Thermostat</i> untuk dikendalikan melalui Grafis .....	151
Tabel 4.20	Hasil pengujian Grafis terhadap nilai – nilai <i>Modbus Point Thermostat</i> .....	152

Tabel 4.21 Hasil pengujian Fungsi Dasar <i>Modbus Point</i> pada PLC M221 dan <i>Wellpro Modul I/O</i> untuk dibaca dan dikendalikan melalui Grafis ..	153
Tabel 4.22 Hasil pengujian Grafis terhadap nilai – nilai <i>Modbus Point</i> PLC M221 dan <i>Wellpro Module I/O</i> .....	153
Tabel 4.23 Hasil pengujian grafis dalam melakukan pengawasan dan pengendalian melalui ponsel pintar .....	157

## DAFTAR LAMPIRAN

Lampiran 1	Modul Simulasi Telekontrol <i>Multi Modbus Device</i> Keseluruhan .....
Lampiran 2	Pengawatan Keseluruhan Modul Telekontrol Modbus Device .....
Lampiran 3	<i>Modbus Register Power Meter Analyzer Circutor CVM-NRG96</i> .....
Lampiran 4	<i>Modbus Register Thermostat TC-300</i> .....
Lampiran 5	<i>Modbus Register Wellpro WP9038ADAM</i> .....
Lampiran 6	<i>Modbus Register PLC Modicon M221</i> .....
Lampiran 7	Laporan Skripsi Dalam Bentuk Jurnal.....

# BAB I

## PENDAHULUAN

### I.1 Latar Belakang

Di era serba otomatisasi dan *monitoring* saat ini, perangkat kendali dan perangkat *monitoring* merupakan sebuah komponen utama yang paling banyak digunakan dalam dunia industri dan gedung komersial. Perangkat kendali otomatis dan perangkat *monitoring* cenderung digunakan pada industri dan gedung komersial karena memiliki keuntungan antara lain pengendalian yang sederhana, dapat menghemat biaya operasional, serta mudah mengawasi peralatan elektronik yang sangat prioritas untuk dilindungi.

Dalam penggunaan peralatan kendali dan peralatan *monitoring* terkadang menimbulkan biaya besar yaitu pada hal operasional pengkabelan yang menghubungkan antara perangkat – perangkat *monitoring* dan perangkat – perangkat kendali ke komputer operator. Pada kondisi yang lain dibutuhkan beberapa operator yang akan mengawasi dan mengendalikan di ruangan yang berbeda – beda atau dilantai yang berbeda – beda.

Pengawasan dan pengendalian pada perangkat kendali dan perangkat *monitoring* tersebut dapat dilakukan dengan dua metode, yaitu menggunakan sistem pengkabelan (*wired*) dan menggunakan sistem nirkabel (*wireless*). Untuk melakukan pengawasan dan pengendalian secara nirkabel dibutuhkan sebuah alat yang biasa disebut ‘*Router*’, dimana akan dihubungkan dengan ‘*Automation Server*’ yang akan mengawasi dan mengendalikan perangkat *slave Modbus* melalui sebuah protokol komunikasi yaitu ‘*MODBUS Protocol*’.

Oleh karena itu banyak dari kalangan personal individu maupun perusahaan melakukan usaha dalam bentuk analisis dan penelitian untuk pengembangan *Automation Server* dalam mengawasi dan mengendalikan perangkat *monitoring* dan perangkat kendali secara efektif dan efisien pada kondisi yang selalu berubah-ubah.

Melihat dari kondisi sistem pengkabelan yang tidak efektif dan kurang handal, penelitian ini ditujukan untuk mensimulasikan sebuah alat berupa modul *Automation Server* yang dibuat oleh pabrik *Schneider Electric* untuk melakukan pengawasan dan pengendalian multi perangkat *Modbus* secara nirkabel melalui router TP-Link MR3420.

Berdasarkan uraian diatas, penulisan ini akan fokus mensimulasikan pengawasan dan pengendalian secara nirkabel dengan perangkat – perangkat *monitoring* dan perangkat – perangkat kendali yang memiliki dukungan *MODBUS Protocol*.

## **I.2 Rumusan Masalah**

Dari uraian yang berada pada latar belakang diatas, maka didapatkan rumusan masalah yaitu,

1. Bagaimana mengetahui nilai kekuatan dan kualitas sinyal (*signal strength*) nirkabel antara *Smart Device* dan *TP-Link MR3420* dalam mempermudah pengawasan dan pengendalian?
2. Bagaimana mengatur kecepatan *Automation Server* sebagai *Master* dalam merespon multi perangkat *MODBUS* dan mengatur *Register Type* perangkat *slave* dalam proses *monitoring* (pengawasan) dan *controlling* (pengendalian) untuk dapat dibaca serta mencegah terjadinya *crash timeout error respon*?

3. Bagaimana melakukan pemrograman dan pembuatan grafis serta mengombinasikannya dengan beberapa perangkat *slave MODBUS* untuk melakukan pengawasan dan pengendalian?

### I.3 Tujuan Penelitian

Berdasarkan rumusan masalah diatas, tujuan dari penelitian ini adalah sebagai berikut:

1. Menganalisa nilai kekuatan dan kualitas sinyal (*signal strength*) nirkabel antara *Smart Device* dan *TP-Link MR3420* dalam mempermudah pengawasan dan pengendalian.
2. Menganalisa kecepatan *Automation Server* sebagai *Master* dalam merespon multi perangkat *MODBUS* dan menganalisa *Register Type* perangkat *slave* dalam proses *monitoring* (pengawasan) dan *controlling* (pengendalian) untuk dapat dibaca serta mencegah terjadinya *crash timeout error respon*.
3. Menganalisa pemrograman dan pembuatan grafis serta mengkombinasikannya dengan beberapa perangkat *slave MODBUS* untuk melakukan pengawasan dan pengendalian.

### I.4 Batasan Masalah

Agar masalah yang akan dibahas tidak meluas dan tujuan dari penulisan ini tidak mengalami kekeliruan dari pemahaman serta pembahasan, maka penulis mencoba memberikan batasan masalah antara lain.

1. Simulasi yang akan dibuat hanya menggunakan satu metode yakni sistem nirkabel dan tidak membandingkan dengan metode lain. Dikarenakan metode hanya digunakan untuk mempermudah simulasi.

2. Analisa yang dilakukan untuk mengukur nilai kekuatan sinyal yang dipengaruhi jarak antara *Smart Device* dan *TP-Link MR3420* hanya menggunakan perangkat lunak *Lizard WiFi Scanner* untuk *Personal Computer* dan *Netgear Wifi Analytics* untuk *Smart Phone*.
3. Analisa yang dilakukan pada Modul *Automation Server* hanya pada konfigurasi nilai *Receive Timeout* dalam menerima respon dari perangkat – perangkat *MODBUS*.
4. Protokol komunikasi yang digunakan menggunakan *MODBUS RTU* dan *MODBUS TCP/IP*.
5. Tidak membahas *MODBUS Protocol* secara rinci mulai dari pertama dikembangkan sampai ke pengembangan yang ada pada saat ini.
6. Modul *server* yang digunakan adalah *Automation Server Premium* pabrikan *Schneider Electric*.
7. Modul *slave Power Meter* yang digunakan adalah satu unit *Circutor CVM- NRG96*.
8. Modul *slave Thermostat* yang digunakan adalah satu unit *Schneider Thermostat TC-300*.
9. Modul *slave modul input/output* yang digunakan adalah satu unit *Wellpro WP9038ADAM*.
10. Modul *slave PLC* yang digunakan adalah satu unit *Schneider PLC TM221ME16R*.
11. Trafo *CT* yang digunakan adalah satu unit *CT 50A/5A* pabrikan *CIC* dan tidak menganalisa secara rinci mengenai perhitungan *CT* dalam simulasi ini.



12. Simulasi hanya dilakukan secara *Local Area Network* (LAN) dan tidak untuk jaringan luas atau *Wide Area Network* (WAN).
13. Simulasi multi *MODBUS Device* hanya dilakukan untuk simulasi proteksi jaringan listrik, pengendalian dan pengawasan komponen – komponen listrik yang didukung oleh *Protocol* komunikasi *MODBUS*.
14. Pada perancangan simulasi ini, alat indikator keluaran/*output* simulasi hanya menggunakan *pilot lamp*.
15. Pada simulasi ini, pengaruh radius pemancaran dan penerimaan sinyal tidak dibahas secara rinci dan akurat.

### **I.5 Manfaat Penulisan**

Berikut penulis menguraikan manfaat penulisan ini diantaranya bagi mahasiswa, bagi lembaga pendidikan, dan masyarakat umum.

#### **1.5.1 Bagi Mahasiswa**

Adapun manfaat penulisan ini bagi mahasiswa yaitu :

- a. Memberi gambaran secara nyata kepada mahasiswa tentang *MODBUS Protocol*.
- b. Mengembangkan dan meningkatkan kajian yang terkait analisa pada *MODBUS Protocol*.
- c. Dapat mengetahui konsep pengawasan dan pengendalian perangkat yang didukung *MODBUS Protocol*.

### 1.5.2 Bagi Universitas Muhammadiyah Sumatera Utara (UMSU)

Adapun manfaat penulisan ini bagi Universitas Muhammadiyah Sumatera Utara (UMSU) antara lain sebagai berikut:

- a. Dapat mengetahui konsep – konsep *MODBUS Protocol* dan menjadi bagian salah satu mata kuliah di lingkungan Universitas Muhammadiyah Sumatera Utara.
- b. Menjadi tolak ukur dalam penilaian peningkatan universitas melalui perkembangan pola pikir dan analisa kemampuan mahasiswa di lingkungan Universitas Muhammadiyah Sumatera Utara.
- c. Dapat memahami dan menelaah simulasi ini untuk meningkatkan Universitas Muhammadiyah Sumatera Utara dalam bidang pembangunan beserta manajemennya yaitu *Smart Building Management System*.

### 1.5.3 Bagi Masyarakat Umum

Adapun manfaat penulisan ini bagi masyarakat umum yaitu :

- a. Memberikan gambaran dan pengetahuan umum tentang *MODBUS Protocol*.
- b. Mengetahui penggunaan dan pengembangan *MODBUS Protocol* sampai saat ini.
- c. Membantu masyarakat umum dalam kegiatan operasional manajemen pembangunan (*Building Management*).
- d. Dapat membuat sistem *monitoring* dan sistem pengendalian yang lebih efektif, hemat, dan *reliable* (handal).
- e. Dapat mengetahui metode yang tepat untuk diaplikasikan di lapangan kerja dalam sebuah sistem manajerial yang pintar agar lebih efektif.

## 1.6 Sistematika Penulisan

Penulisan ini membahas pokok permasalahan secara cermat dan sistematis. Untuk itu penulisan disusun sedemikian rupa dengan materi pembahasan yang saling berhubungan dan sistematis sebagai berikut:

### **BAB I     Pendahuluan**

Pada bab ini berisi pendahuluan yang mencakup latar belakang, perumusan masalah, batasan masalah, tujuan penelitian, manfaat penulisan dan sistematika penulisan.

### **BAB II    Landasan Teori**

Bab ini membahas tinjauan pustaka relevan dan dasar – dasar teori mengenai *system Protocol MODBUS* dan komponen – komponen yang digunakan dalam melengkapi rancangan simulasi telekontrol yang akan dibuat.

### **BAB III   Perancangan Sistem**

Bab ini menjelaskan perancangan sistem konfigurasi masing – masing perangkat *MODBUS* baik pada perangkat keras maupun perangkat lunak, dan pemrograman dan pembuatan grafis dari sistem simulasi telekontrol multi *MODBUS device* menggunakan modul *Schneider Automation Server*.

### **BAB IV   Analisa Dan Pengujian Sistem**

Bab ini berisi analisa dan pengujian konfigurasi, pemrograman dan grafis dari sistem perancangan simulasi telekontrol multi *MODBUS device* menggunakan modul *Schneider Automation Server*.

**BAB V Penutup**

Bab ini memuat tentang kesimpulan dari hasil simulasi yang telah dilakukan dan saran yang bisa diberikan kepada pihak – pihak yang terkait.



## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Tinjauan Pustaka Relevan

Agus Tiyono dkk (2007) melakukan penelitian ‘Sistem Telekontrol Scada Dengan Fungsi Dasar Modbus Menggunakan Mikrokontroler AT89S51 Dan Komunikasi Serial RS485’. Pada penelitian ini dibuat sebuah model sistem SCADA dengan menerapkan salah satu fungsinya, yaitu sebagai pengendali jarak jauh (*telecontrolling*). Dalam sistem telekontrol SCADA ini digunakan mikrokontroler AT89S51 sebagai pembentuk komponen *Master* dan *Slave*. Komunikasi antara *Master* dan *Slave* menggunakan fungsi dasar protokol *Modbus* dan komunikasi serial RS485. Hasil pengujian menunjukkan bahwa sistem telekontrol SCADA *Modbus* yang dibuat mendukung fungsi protokol *Modbus* 05, 06, 15 dan 16 sebagai fungsi *telecontrolling*, serta fungsi 01, 02, 03 dan 04 sebagai *monitoring*. Fungsi telekontrol 05, 06, 15, dan 16 dapat berjalan dengan baik untuk mode pengalamatan *broadcast* dan *unicast*. Pengiriman *query* dengan *Slave ID* yang tidak didukung sistem SCADA *Modbus* akan menghasilkan *time-out error*, sedangkan untuk kode fungsi yang tidak didukung sistem akan menghasilkan *exception response*. Komunikasi antara *Master* dan *Slave* dapat berjalan dengan baik menggunakan panjang kabel 6 m dan 100 m, serta dengan *delay* waktu pengiriman antar karakter pesan tidak lebih besar dari 1656,90  $\mu$ s.

Ferdina I. Gumilang dkk (2014) melakukan penelitian ‘Rancang Bangun Jaringan Komunikasi Multi PLC dengan Platform Sistem SCADA-DCS Terintegrasi’. Sistem PLC dibuat dengan *system master slave* dan ditampilkan

pada HMI yang dikoneksikan dengan koneksi Ethernet. PLC yang digunakan sebagai *master* adalah Siemens S7-1200, sedangkan PLC *slave* menggunakan PLC Twido Schneider dan PLC Omron CP1H. HMI untuk menampilkan data menggunakan *software* Vijeo Citect. Pada PLC CP1H menggunakan komunikasi serial yang diubah menjadi Ethernet TCP/IP menggunakan *device* WIZ110SR. PLC *Master* mengirim/menerima data terhadap PLC Twido menggunakan protokol *Modbus* TCP, sedangkan terhadap PLC Omron menggunakan protokol *Profinet* yang diubah menjadi data Serial. Ujicoba komunikasi dilakukan dengan membuat suatu sistem pengendalian. Masing-masing PLC *Slave* mengendalikan satu jenis pengendalian dan terintegrasi dengan PLC *Master*. PLC *Master* dapat berelaborasi dengan PLC *Slave*, yaitu dengan mengirim data parameter *system* ke PLC *Slave* dan membaca data pembacaan sistem tersebut lalu ditampilkan pada HMI. Status koneksi tiap-tiap PLC *slave* dapat diketahui pada tampilan HMI dan juga mampu melakukan *reconnect* jika koneksi terputus antara *slave* dan *master*. Dengan demikian, komunikasi multi PLC berbeda merk berhasil dilakukan dengan menggunakan koneksi Ethernet melalui protokol *Modbus* TCP dan *Profinet*.

Andi Adriansyah dan Rizally Priatmadja (2015) melakukan penelitian 'Rancang Bangun Protocol Modbus Pada KWH Meter Elektronik TIPE ION 8600 Untuk Memonitor Besaran Energi Listrik Trafo Dengan Menggunakan Aplikasi Citect Scada'. Penelitian ini pada perancangan sistem dibagi menjadi tiga bagian yaitu sistem *server*, sistem *client*, dan sistem komunikasi. Sistem *server* dirancang menggunakan aplikasi Citect Scada v.6 yang diinstal kedalam PC/Laptop yang mampu terkoneksi dengan jaringan intranet PLN. Sistem *Client* dirancang dengan



memanfaatkan peralatan primer berupa CT (*Current Transformer*) dan CVT (*Capasitive Voltage Transformer*) serta kWh Meter sebagai *device* untuk mengkonversi sinyal analog menjadi digital sehingga mampu dibaca oleh *server*. Untuk sistem komunikasi menggunakan fiber optik dan radio frekuensi yang telah tersedia di jaringan PLN. Dari hasil percobaan menunjukkan bahwa sistem *monitoring* yang dibuat telah berfungsi sesuai dengan perancangan. *Server* mampu menampilkan data pembacaan energi secara *realtime* yang diambil dari kWh Meter melalui *protocol modbus* dengan akurasi 0.3-0.5 %.

Triyanto Pangaribowo dan Hibnu Yulianda (2016) melakukan penelitian ‘Sistem Monitoring Suhu Melalui Sistem Komunikasi Programmable Logic Controller To Personal Computer’. Pada penelitian ini memanfaatkan PLC (*Programable Logic Controller*) untuk monitoring suhu yang diterapkan pada *Building Auto System*. PLC merupakan suatu *controller* yang umum digunakan pada dunia industri. PLC digunakan untuk *monitoring* sistem kerja panel dalam satu gedung dalam jarak yang jauh dengan memanfaatkan komunikasi PLC sebagai *indicator* status ON/OFF serta pembacaan suhu. Sebagai *interface* antara PLC dan user, maka digunakan HMI (*Human Machine Interface*). Pemanfaatan PLC sebagai *monitoring* dilakukan pada sebuah gedung yang memiliki sistem kontrol dalam jumlah yang banyak serta memiliki jarak yang jauh. Pada penelitian ini dilakukan pengujian sistem *monitoring* menggunakan koneksi PLC To PC, PLC To PLC dengan PC. Untuk koneksi menggunakan *protocol Modbus* serial RS485. Aplikasi yang dirancang untuk sistem *monitoring* ini lebih efektif, karena dapat menghemat waktu dan memudahkan dalam mengetahui status suatu kontrol apakah bekerja atau tidak. Berdasarkan hasil Analisa dan pengukuran respon

waktu sistem terhadap perubahan suhu untuk setiap kenaikan rata-rata  $2,4^{\circ}\text{C}$  pada sistem koneksi PC to PLC rata-rata 2,6 detik, dan Pada sistem *monitoring* PLC to PLC yang ditampilkan pada layar PC memiliki respon waktu terhadap perubahan suhu untuk setiap kenaikan rata-rata  $2,3^{\circ}\text{C}$  rata-rata 2,67 detik. Sistem *monitoring* mampu bekerja pada jarak 100 meter

## 2.2 Pengenalan *MODBUS*

*MODBUS* adalah protokol komunikasi *serial* yang diterbitkan oleh Modicon pada 1979 untuk diaplikasikan pada PLC. Kemudian protokol ini telah menjadi standar protokol komunikasi di industri, dan sekarang *MODBUS* merupakan protokol komunikasi dua-arah yang paling umum digunakan sebagai media penghubung dengan perangkat industri atau media elektronik lainnya dengan computer (*Schneider Electric*, 2015).

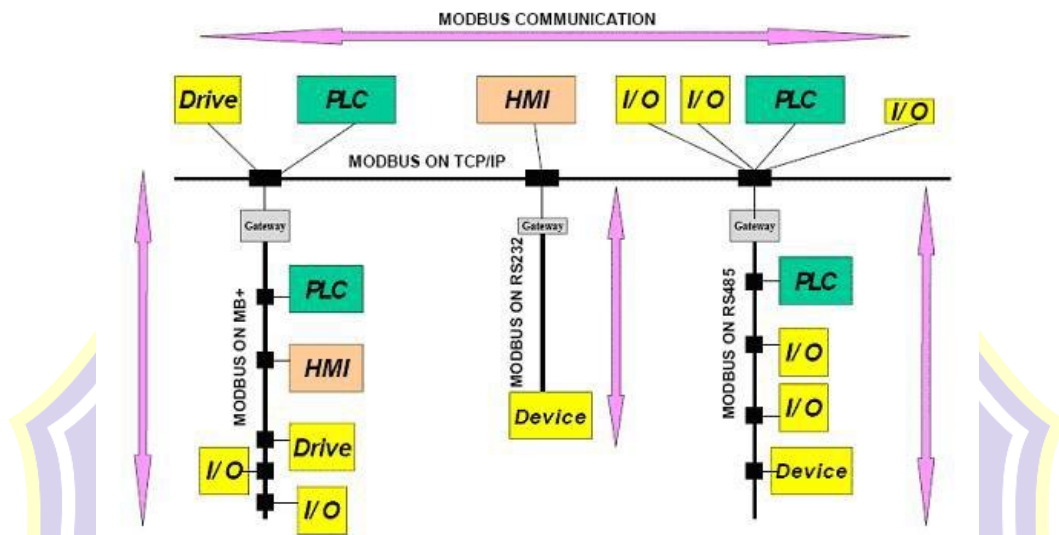
Menurut Anjali S. Ashtekar, Bhagsen J.Parvat, dan Chandrakant B. Kadu (2013) alasan utama penggunaan *MODBUS* secara ekstensif sebagai protokol komunikasi adalah:

1. *MODBUS* diterbitkan sebagai *open protocol* dan *bebas royalti*.
2. *MODBUS* relatif mudah untuk digabungkan dengan jaringan industri.
3. *MODBUS* melakukan *transfer data* “*raw bits*” atau “*words*” tanpa membatasi jenis vendor atau jenis merk/brand pabrikan perangkat industri yang digunakan.

Menurut *Modbus Organization* (2012:2) *MODBUS* adalah protokol pesan lapisan aplikasi untuk komunikasi client / *Server* antara perangkat yang terhubung pada berbagai jenis bus atau jaringan. Saat ini diimplementasikan menggunakan:



1. TCP / IP over Ethernet.
2. Transmisi serial *asynchronous* atas berbagai media (kawat: EIA / TIA-232-E, EIA-422, EIA / TIA-485-A; serat, radio, dll)
3. *MODBUS PLUS*, sebuah jaringan token kecepatan tinggi.



**Gambar 2.1** Arsitektur Komunikasi *MODBUS* Secara Umum. *Modbus Organization* (2013).

Menurut Yinglan Fang\*, Xianfeng Han and Bing Han (2013) *MODBUS* adalah tipe baru teknologi bus lokal dengan kemampuan adaptasi yang terbuka, cerdas, tinggi, dan karakteristik lainnya. Ini juga memiliki instalasi dan pemeliharaan yang baik, bobot inisiatif integrasi sistem yang tinggi, jangkauan yang akurat dan dapat diandalkan dan titik kuat lainnya. Selain itu, kesepakatan ini telah secara ketat merumuskan struktur pesan yang dapat mengidentifikasi jaringan bus dari *Master* dan *Slave*, terlepas dari jaringan apa yang akan dikomunikasikan. Selama rangkaian sinyal dikonfigurasi sesuai dengan struktur data yang ditetapkan oleh pengaturannya. Sistem yang berbeda dapat saling

berhubungan, sehingga dapat secara fleksibel memilih medium transportasi yang mendasari di bus lokal.

*MODBUS* memiliki karakteristik siklus permintaan respons, perangkat *Master* dan *Slave* dapat berkomunikasi secara terpisah, namun juga menggunakan siaran untuk berkomunikasi ke semua perangkat *Slave*. Jika berkomunikasi sendiri, perangkat *Slave* mengembalikan pesan sebagai tanggapan. Protokol *MODBUS* menetapkan format kueri perangkat utama: alamat (kode dan kode fungsi) perangkat dan semua data yang akan dikirim dan domain deteksi kesalahan. Pesan respons perangkat *Slave* juga terdiri dari protokol *MODBUS*. Ini termasuk mengkonfirmasi domain tindakan dan data yang akan dikembalikan dan domain deteksi kesalahan. Jika pesan terjadi kesalahan saat penerimaan, atau perangkat *Slave* tidak menjalankan perintahnya, perangkat *Slave* akan membuat pesan kesalahan dan mengirimkannya sebagai tanggapan. Bila perangkat yang berbeda berkomunikasi dalam jaringan *MODBUS*, setiap pengontrol memiliki alamat perangkat sesuai dengan persyaratan protokol. Perangkat proses komunikasi mengidentifikasi pesan yang dikirim oleh alamat, dan memutuskan tindakan apa yang dihasilkan oleh unit peralatan yang ditentukan. Jika perlu respon, pengendali akan menghasilkan umpan balik dan menggunakan unit protokol *MODBUS* yang dikirim ke perangkat yang ditentukan. Di jaringan lain, termasuk pesan protokol *MODBUS* yang dikonversi ke struktur *frame* atau paket di jaringan ini. Konversi ini juga memperluas solusi sesuai dengan jaringan spesifik yang memecahkan alamat simpul, jalur *routing*, dan metode deteksi kesalahan.

### 2.2.1 Prinsip Dasar MODBUS

Perangkat *MODBUS* berkomunikasi menggunakan teknik *Master-Slave*, dimana hanya satu perangkat (*Master*) yang dapat melakukan transaksi atau melakukan perintah atau permintaan yang disebut “*queries*”. Perangkat lain (*Slave*) merespon dengan menyediakan data yang diminta untuk *Master*, atau dengan melakukan aksi sesuai dengan yang diminta dalam *query*. *Master* dapat mengirimkan *request* ke satu *Slave* secara individu, atau dapat mengirim pesan broadcast ke semua *Slave*. *Slave* memberikan respon untuk pertanyaan yang ditujukan kepada mereka secara individu. Sedangkan *query broadcast* dari *Master* tidak akan diberikan respon oleh *Slave*.

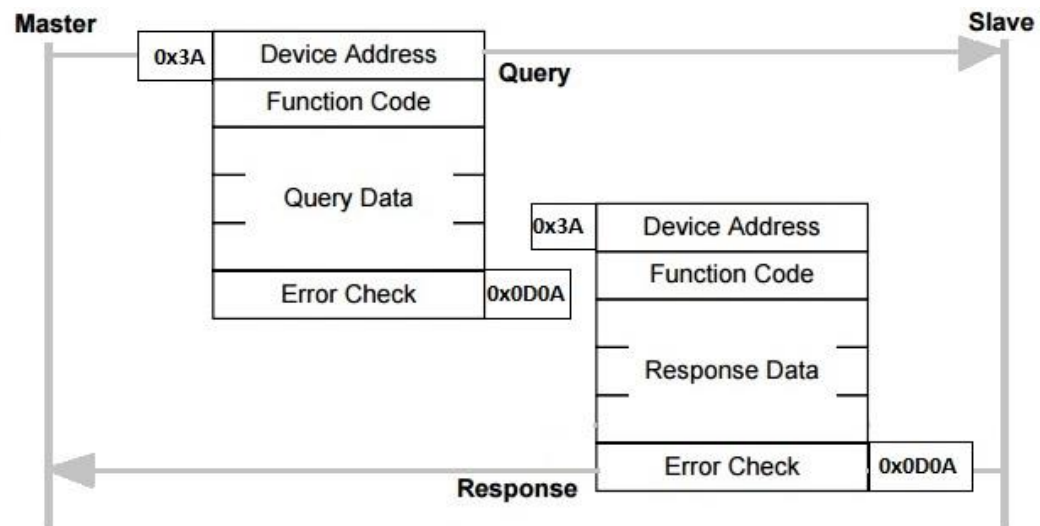
Protokol *MODBUS* mempunyai format tertentu untuk setiap *query* dari *Master* yang berisi alamat dari perangkat (*Slave*) yang dituju, kode fungsi yang mendefinisikan aksi yang diminta, data yang akan dikirim, dan pemeriksaan kesalahan. Pesan respon *Slave* juga mempunyai format tertentu dalam protokol *MODBUS*. Format ini berisi tentang konfirmasi tindakan yang dilakukan, data yang akan dikirim, dan bidang pemeriksaan kesalahan (M-System, tanpa tahun:4).

Menurut *Modbus Organization* (2002:7) bahwa pada saat mengirimkan *query* ke *Slave*, *Master* menggunakan 2 *mode* pengalamatan, yaitu:

*Unicast mode* : *Master* mengirimkan *query* kepada satu *Slave*. Setelah menerima dan memproses *query*, *Slave* akan memberikan jawaban berupa respon kepada *Master*.

*Broadcast mode*: *Master* mengirimkan perintah (*query*) kepada semua *Slave*. Pada *mode* pengalamatan ini *Slave* tidak mengirimkan respon kepada *Master*.

Menurut Modicon Inc. (1996:5) Protokol *MODBUS* membentuk sebuah format pesan untuk *query Master* dan respon *Slave*. Proses *Query-Response* pada protokol *MODBUS* mempunyai format seperti gambar dibawah ini.



**Gambar 2.2** Arsitek Komunikasi Antara *Master* dengan *Slave*

Format untuk *query Master*, *device address* merupakan alamat *Slave* yang akan diambil datanya, *Function Code* merupakan kode fungsi yang mendefinisikan aksi yang diminta, *Query data* merupakan blok data informasi dan *Error Check* merupakan pemeriksaan kesalahan (cek data dari kesalahan komunikasi), Respon *Slave* pada protokol *MODBUS* juga mempunyai format yang sama, *Function Code* berisi konfirmasi tindakan yang dilakukan, *Response Data* merupakan data yang akan dikembalikan, dan *error check* sebagai bidang pemeriksaan kesalahan.

Sistem komunikasi pada jaringan standar *MODBUS* mempunyai dua *mode* transmisi: ASCII (*American Standard Code for Informasi Interchange*) atau RTU (*Remote Terminal Unit*). Pada satu jaringan *MODBUS*, *mode* transmisi untuk semua perangkat/*device* yang terhubung harus sama.

Dalam *mode* ASCII, setiap *byte* 8-bit dalam pesan yang dikirim sebagai dua karakter ASCII. Dalam modus RTU, setiap *byte* 8-bit dalam pesan berisi dua buah 4-bit karakter heksadesimal. *MODBUS* RTU, dengan kepadatan yang lebih besar karakternya, memungkinkan *throughput data* yang lebih baik dari ASCII untuk *baud rate* yang sama. Algoritma yang digunakan dalam memeriksa kesalahan, tergantung pada metode transmisi yang digunakan; LRC (*Longitudinal Redundancy Check*) dalam *mode* ASCII; CRC (*Redundancy Check Siklis*) dalam *mode* RTU.

### **2.2.1.1 Mode Transmisi Serial**

Menurut *Modbus Organization* (2002:12) Dalam jaringan *MODBUS* terdapat 2 *mode* transmisi *serial*, yaitu *mode* RTU dan *mode* ASCII. Setiap peralatan dalam sebuah jaringan *MODBUS* harus mempunyai *mode* dan parameter *serial* yang sama. Pengaturan *default MODBUS* adalah RTU, sedangkan *mode* ASCII adalah pilihan.

#### **2.2.1.1.1 Mode RTU (Remote Terminal Unit)**

Format masing-masing *byte* (11 bit) dalam *mode* RTU adalah:

- a) *Coding system*: 8 bit biner, heksadesimal 0-9,A-F.
- b) *Bits per byte*: 1 start bit.
- c) 8 data bits, *Least Significant Bit* (LSB) dikirim pertama.
- d) 1 bit untuk *even/odd parity*, no bit untuk *no parity*.
- e) 1 *stop bit* jika menggunakan *parity*, 2 bits untuk *no parity*.
- f) *Error check field*: *Cyclical Redundancy Check* (CRC).



### 2.2.1.1.2 Mode ASCII (*American Standard Code Information Interchange*)

Format masing-masing *byte* (10 bit) dalam *mode* ASCII adalah:

- a) *Coding system*: Heksadesimal, karakter ASCII 0-9, A-F.
- b) *Bits per byte*: 1 start bit.
- c) 7 data bits, *Least Significant Bit* (LSB) dikirim pertama.
- d) 1 bit untuk *even/odd parity*, *no bit* untuk *no parity*.
- e) 1 *stop bit* jika menggunakan *parity*, 2 bits untuk *no parity*.
- f) *Error check field*: *Longitudinal Redundancy Check* (LRC).

### 2.2.1.2 MODBUS Message Framing

Modicon Inc. (1996:8-9) *MODBUS Message Framing* terbagi dua yaitu:

1. ASCII Framing, dan
2. RTU Framing.

#### 2.2.1.2.1 ASCII Framing

Pada *mode* ASCII, pesan dimulai dengan sebuah karakter “colon” (:) dalam ASCII 3A hex, dan diakhiri dengan sebuah pasangan “*carriage return – line feed*” (CRLF) dalam ASCII 0D dan 0A hex.

*Frame* pesan pada *mode* transmisi ASCII ditunjukkan pada Tabel 2.1

Tabel 2.1 ASCII Framing.

Start	Address	Function	Data	LRC Check	END
1 CHAR :	2 CHAR	2 CHAR	n CHAR	2 CHAR	2 CHAR CRLF

### 2.2.1.2.2 RTU Framing

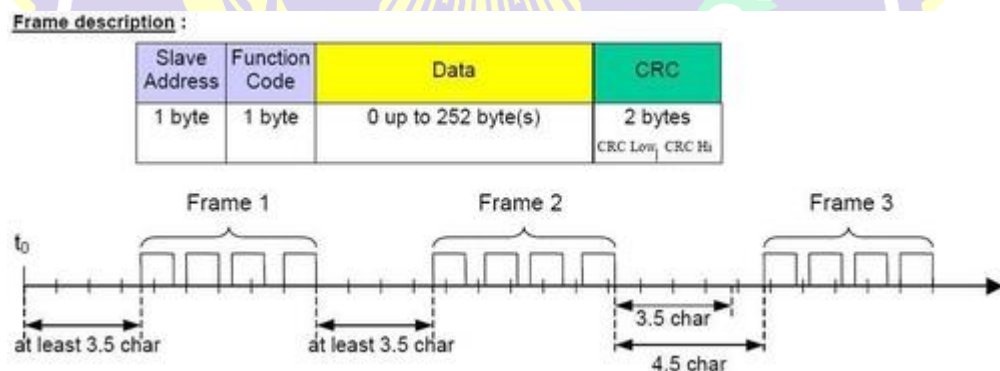
Pada *mode* RTU, *frame* pesan dipisahkan oleh *silent interval* paling sedikit waktu 3,5 karakter. Interval waktu ini disebut  $T_{3,5}$ . Seluruh karakter dalam *frame* pesan harus ditransmisikan secara bersambung. Interval antar karakter dalam *frame* pesan tidak boleh lebih besar dari waktu 1,5 karakter ( $T_{1,5}$ ). Jika interval antar karakter lebih besar dari  $T_{1,5}$ , maka *frame* pesan tersebut dinyatakan tidak lengkap dan akan diabaikan.

*Frame* pesan pada *mode* transmisi RTU ditunjukkan pada Tabel 2.2.

Tabel 2.2 RTU Framing

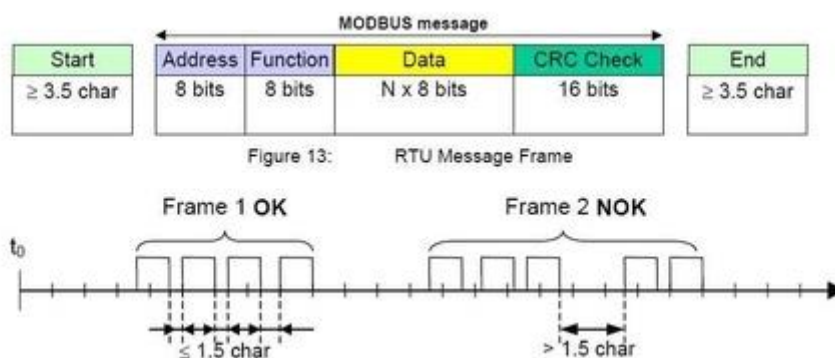
Start	Address	Function	Data	CRC Check	END
$T_{3,5} \geq 3,5$ CHAR	8 bit	8 bit	$n \times 8$ bit	16 bit	$T_{3,5} \geq 3,5$ CHAR

Interval  $T_{3,5}$  dalam *frame* pesan mode transmisi RTU ditunjukkan pada gambar dibawah ini.



Gambar 2.3 Interval  $T_{3,5}$  dalam *frame*.

Interval  $T_{1,5}$  antar karakter dalam *frame* pesan mode transmisi RTU ditunjukkan pada gambar dibawah ini.



**Gambar 2.4** Interval  $T_{1,5}$  antar karakter dalam *frame*.

Menurut Agus Tiyono (2007) pada *state* “mengirim *query*” setelah karakter terakhir dikirim, Master akan menginisialisasi dan mulai menghitung  $T_{3,5}$ . Untuk *mode broadcast*, setelah  $T_{3,5}$  selesai Master akan mulai menghitung waktu tunda *broadcast* sampai waktu tersebut selesai. Pada *mode unicast*, setelah  $T_{3,5}$  selesai Master akan mulai menghitung *time-out response*. Besarnya waktu tunda *broadcast* adalah 100 ms ( $2 \times 50$  ms).

*State* “menunggu respon dari *Slave*” merupakan keadaan menunggu pesan jawaban dari *Slave*. Jika Master tidak menerima respon sampai waktu *time-out response* selesai, maka akan terjadi *time-out error*. Apabila Master menerima respon, maka Master akan memeriksa karakter pertama dari respon tersebut. Jika karakter tersebut bukan *Slave ID* tujuan, maka Master akan menunggu kembali sampai didapatkan *Slave ID* tujuan atau terjadi *time-out error*.

Setelah Master mendapatkan *Slave ID* tujuan, Master akan menghentikan *time-out response* serta menginisialisasi dan mulai menghitung  $T_{1,5}$  dan  $T_{3,5}$ , kemudian menuju ke *state* “*penerimaan respon*”. Setiap terjadi penerimaan karakter, Master akan menginisialisasi dan menghitung ulang  $T_{1,5}$  dan  $T_{3,5}$ . Setelah  $T_{1,5}$  selesai Master akan mengecek respon (CRC) dan menunggu waktu



$T_{3,5}$  selesai. Jika sebelum waktu  $T_{3,5}$  selesai terdapat karakter yang diterima, maka *Master* akan memberi tanda (*flag*) *frame error*, ditandai dengan *flag* RI = 1.

Apabila  $T_{3,5}$  selesai dan terdeteksi *frame error* atau nilai CRC salah, maka *Master* tidak akan mengirimkan pesan ke HMI. Namun jika  $T_{3,5}$  selesai dan cek CRC benar, maka *Master* akan mengirimkan respon yang diterima dari *Slave* menuju HMI. Setelah proses ini, selanjutnya *Master* kembali menunggu perintah dari HMI.

Besarnya waktu  $T_{3,5}$  dan *Receive Timeout* dapat dihitung melalui persamaan (2.1) dan (2.2) sebagai berikut:

$$T_{3,5} = 3,5 \times \frac{\text{Jumlah bit tiap karakter}}{\text{Baud Rate}} \dots\dots\dots(2.1)$$

$$\text{Receive Timeout} = T_{3,5} \times \text{Jumlah Karakter} \dots\dots\dots(2.2)$$

Besarnya waktu  $T_{3,5}$  untuk jumlah bit bernilai 11 dan *baudrate* 9600 serta jumlah karakter 8 dihitung menggunakan persamaan (2.1) dan (2.2) sebagai berikut:

$$T_{3,5} = 3,5 \times \frac{\text{Jumlah bit tiap karakter}}{\text{Baud Rate}} \dots\dots\dots(2.1)$$

$$T_{3,5} = 3,5 \times \frac{11 \text{ bit}}{9600 \text{ bps}}$$

$$T_{3,5} = 0,00401 \text{ s} = 4010 \mu\text{s}$$

$$\text{Receive Timeout} = T_{3,5} \times \text{Jumlah Karakter} \dots\dots\dots(2.2)$$

$$\text{Receive Timeout} = 0,00401 \times 8 = 32,08 \text{ ms}$$

### 2.2.1.3 Address Field

Menurut M-System, (tanpa tahun:5) Masing-masing *Slave* harus mempunyai alamat yang berbeda dalam *range* 1 – 247 untuk pengalamatan individual. Alamat 0 digunakan untuk pengalamatan *broadcast*.

#### 2.2.1.4 *Function Field*

Menurut M-System, (tanpa tahun:5) *Function field* pada *frame* pesan berisi nomer kode fungsi (*function code*). Kode fungsi yang valid mempunyai *range* 1 – 255, dimana kode 1 – 127 untuk fungsi normal, sedangkan 128 – 255 untuk fungsi *exception response*. *Function code* berfungsi untuk memberitahu *Slave* tentang perintah yang harus dikerjakan dan sebagai indikasi respon normal atau jenis *error* yang terjadi (*exception response*). Pada sistem komunikasi *MODBUS*, jumlah *function code* yang didukung bervariasi tergantung kontroler dan peralatan *Slave* yang digunakan.

#### 2.2.1.5 Fungsi Dasar *MODBUS*

Menurut M-System, (tanpa tahun:7) Fungsi dasar *MODBUS* merupakan fungsi – fungsi yang tertanam pada *MODBUS* Protocol untuk melakukan pembacaan dan memberi perintah pada suatu alamat *database*, fungsi dasar *MODBUS* yaitu sebagai berikut:

1. *Coil* = fungsi *input* secara *biner* untuk menyatakan *input* 1 adalah ON dan 0 adalah off, dapat diatur untuk fungsi untuk membaca, mengatur juga menetapkan fungsi status (*force*) dengan *output* digital baik untuk satu *input/output* maupun banyak sekaligus (*multiple*).
2. *Input status (Input Contact)* = menyatakan fungsi *input* yang dimasukkan sebagai sebuah *output* pada hasil secara langsung secara digital.
3. *Registers* = menyatakan *input/output* yang dihasilkan dari penerimaan data dari *input* secara analog baik digital baik untuk satu *input/output* maupun banyak sekaligus (*multiple*).

Tabel 2.3 Kode Fungsi *MODBUS*.

Kode Fungsi	Fungsi	Aksi Yang Dikerjakan
1 = 01H	<i>Read Coil Status</i>	Membaca status On/Off Coil (Output Digital)
2 = 02H	<i>Read Input Status</i>	Membaca status On/Off (Input Digital)
3 = 03H	<i>Read Holding Register</i>	Membaca nilai output analog
4 = 04H	<i>Read Input Register</i>	Membaca nilai input analog
5 = 05H	<i>Force Single Coil</i>	Memberi perintah status satu coil pada keadaan On/Off
6 = 06H	<i>Preset Single Register</i>	Memberi perintah nilai pada satu output analog
15 = 0FH	<i>Force Multiple Coils</i>	Memberi perintah beberapa coil pada keadaan On/Off
16 = 10H	<i>Preset Multiple Registers</i>	Memberi perintah nilai pada beberapa output analog

#### 2.2.1.6 Data Field

Menurut M-System, (tanpa tahun:5) *Data field* pada *query* berisi kode sebagai informasi tambahan pada *function code* tentang aksi yang harus dikerjakan *Slave*. Informasi tersebut bisa berupa alamat *input-output*, jumlah *input-output*, jumlah *byte data*, atau nilai data pengesetan. Jika tidak terjadi *error*, *data field* pada respon berisi data yang diminta. Sedangkan pada *exception response*, *data field* berisi *exception code*.

#### 2.2.1.7 Exception Response

Menurut M-System, (tanpa tahun:23) Terdapat 4 proses komunikasi yang mungkin terjadi antara *Master* dan *Slave*, yaitu:

1. Jika *Slave* menerima pesan *query* tanpa adanya kesalahan komunikasi, dan *Slave* dapat menangani *query* tersebut, *Slave* akan memberikan sebuah respon normal.
2. Jika *Slave* tidak menerima *query* dikarenakan adanya kesalahan komunikasi, maka tidak ada respon yang dikirimkan. *Master* akan memberikan kondisi *time-out* untuk pengiriman *query* tersebut.
3. Jika *Slave* menerima pesan *query*, tetapi terdeteksi kesalahan komunikasi (*parity*, LRC, atau CRC), maka tidak ada respon yang dikirimkan. *Master* akan memberikan kondisi *time-out*.
4. Jika *Slave* menerima *query* tanpa adanya kesalahan komunikasi, tetapi *Slave* tidak dapat menangani perintah tersebut (contoh, perintah untuk membaca *coil* atau *register* yang tidak ada), *Slave* akan mengirimkan sebuah respon pengecualian (*exception response*) untuk memberikan informasi kepada *Master* letak kesalahan yang terjadi.

Pada sebuah *exception response*, *Slave* mengembalikan kode fungsi dengan MSB (*Most Significant Bit*) diset 1 dan *data field* diisi dengan kode pengecualian (*exception code*). Hal ini dimaksudkan agar *Master* mengetahui *exception* yang terjadi. Beberapa *exception code* berikut keterangannya ditunjukkan pada Tabel 2.4.

Tabel 2.4 *Exception code* dalam *exception response*.

Kode	Nama	Arti
1	<i>ILLEGAL FUNCTION</i>	Kode fungsi yang terdapat dalam <i>query</i> merupakan perintah ( <i>action</i> ) yang tidak diizinkan untuk <i>Slave</i> .
2	<i>ILLEGAL DATA ADDRESS</i>	Alamat data dalam <i>query</i> merupakan alamat yang tidak diizinkan untuk <i>Slave</i> .
3	<i>ILLEGAL DATA VALUE</i>	Nilai dalam <i>data field query</i> merupakan nilai yang tidak diizinkan untuk <i>Slave</i> .

### 2.2.2 MODBUS TCP (*Modbus Organization, 2006*)

*MODBUS TCP* adalah protokol *MODBUS* yang sistem pengirimannya tidak menggunakan komunikasi *serial* tetapi menggunakan pembungkus TCP/IP dan dikirimkan melalui jaringan. Sehingga dengan menggunakan protokol ini, paket yang dikirimkan dalam jaringan akan berubah.

#### 2.2.2.1 TCP/IP

TCP adalah *Transmission Control Protocol* dan IP adalah *Internet Protocol*. Protokol ini digunakan bersama dan merupakan protokol *transport* untuk internet. Ketika informasi *MODBUS* dikirim menggunakan protokol ini, data diteruskan ke TCP di mana informasi tambahan dilampirkan dan diberikan kepada IP. IP kemudian menempatkan data dalam paket (atau *datagram*) dan mengirimkannya.

TCP harus membuat koneksi sebelum mentransfer data, karena itu adalah protokol berbasis koneksi. *Master* (atau Klien di *MODBUS TCP*) menetapkan koneksi dengan *Slave* (atau *Server*). *Server* menunggu koneksi masuk dari Klien. Setelah koneksi dibuat, *Server* kemudian menanggapi pertanyaan dari Klien sampai klien menutup koneksi.

#### 2.2.2.2 MODBUS RTU over TCP

Sederhananya, ini adalah pesan *MODBUS RTU* yang dikirimkan dengan pembungkus TCP / IP dan dikirim melalui jaringan, bukan saluran *serial*. *Server* tidak memiliki *SlaveID* karena menggunakan Alamat IP.



### 2.2.2.3 ADU & PDU

Selain perbedaan utama antara koneksi *serial* dan jaringan yang disebutkan di atas, ada beberapa perbedaan dalam konten pesan. Dimulai dengan pesan *MODBUS RTU* dan menghapus *Slave ID* dari awal dan CRC dari hasil akhir di PDU, Unit Data Protokol.

Berikut ini contoh permintaan *MODBUS RTU* untuk konten *register holding output analog # 40108* hingga *40110* dari perangkat *Slave* dengan alamat *17*.

11 03 006B 0003 7687

11: Alamat *SlaveID* ( $17 = 11$  hex)

03: Kode Fungsi (baca *Analog Output Holding Register*)

006B: Alamat Data dari daftar pertama yang diminta. ( $40108 - 40001 = 107 = 6B$  hex)

0003: Jumlah total register yang diminta. (baca 3 register *40108* hingga *40110*)

7687: CRC (cyclic redundancy check) untuk pengecekan error.

Menghapus *Slave ID* dan CRC memberi PDU:

03 006B 0003

### 2.2.2.4 MBAP Header

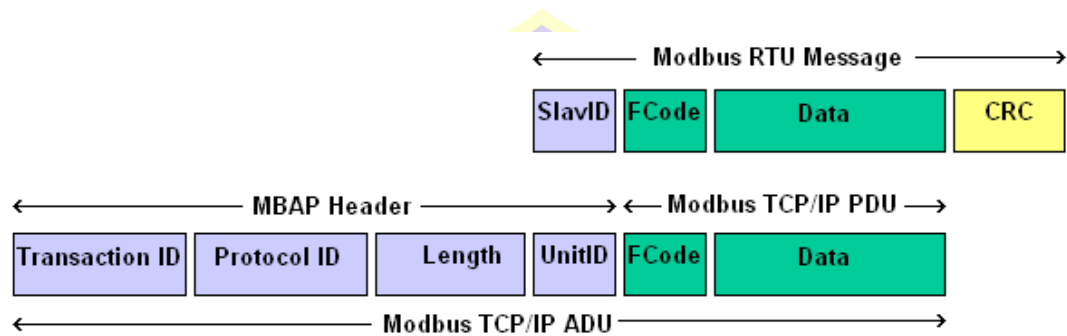
*Header 7-byte* baru yang disebut header MBAP (*MODBUS Application Header*) ditambahkan ke awal pesan. Header ini memiliki data berikut:

Pengenal Transaksi: *2 byte* yang ditetapkan oleh Klien untuk mengidentifikasi setiap permintaan secara unik. *Bytes* ini digemakan oleh *Server* karena tanggapannya mungkin tidak diterima dalam urutan yang sama dengan permintaan.

*Protocol Identifier* : 2 byte yang ditetapkan oleh Klien, selalu = 00 00

*Length* : 2 byte yang mengidentifikasi jumlah *byte* dalam pesan untuk diikuti.

*Unit Identifier* : 1 byte ditetapkan oleh Klien dan diulang oleh *Server* untuk identifikasi *remote* jarak jauh yang terhubung pada saluran *serial* atau di bus lain.



**Gambar 2.5** Paket yang akan dikirimkan melalui internet/MBAP Header

Ringkasan

Permintaan yang setara dengan contoh RTU *MODBUS* ini

11 03 006B 0003 7687

di *MODBUS* TCP adalah:

0001 0000 0006 11 03 006B 0003

0001: Pengenal Transaksi

0000: Pengenal Protokol

0006: Panjang Pesan (6 *byte* untuk diikuti)

11: Unit Identifier (17 = 11 hex)

03: Kode Fungsi (baca *Analog Output Holding Register*)

006B: Alamat Data dari daftar pertama yang diminta. (40108-40001 = 107 = 6B hex)

0003: Jumlah total *register* yang diminta. (baca 3 register 40108 hingga 40110)

### 2.2.3 WLAN (*Wireless Local Area Networks*)

Wi-Fi (*Wireless Fidelity*) atau lebih dikenal dengan WLAN (*Wireless Local Area Network*) merupakan teknologi jaringan wireless yang ditujukan untuk menghubungkan beberapa terminal berbasis IP (PC, *notebook* atau PDA) dalam suatu area LAN (*Local Area Network*). WLAN merupakan salah satu aplikasi pengembangan *wireless* untuk komunikasi data. Sesuai dengan namanya yaitu *wireless*, berarti tanpa kabel, WLAN adalah jaringan lokal yang tidak menggunakan kabel (Wibisono, 2008). Dengan performa dan keamanan yang dapat diandalkan, pengembangan jaringan WLAN menjadi tren baru pengembangan jaringan menggantikan jaringan *wired* atau jaringan kabel. Solusi dari pengembangan WLAN dapat mencakup sebuah kawasan rumah, kantor kecil, perusahaan hingga ke area-area publik (Mulyanta, 2005:147).

#### 2.2.3.1 Standarisasi *Wireless LAN*

Menurut Abdul Rokhim (tanpa tahun) ada beberapa organisasi yang telah menetapkan standard *Wireless LAN* diantaranya yaitu *Federal Communication Commission* (FCC), *Institute of Electrical and Electronic Engineers* (IEEE), *Wireless Ethernet Compatibility Alliance* (WECA), dan *European Telecommunications Standards Institute* (ETSI).

Namun standar *Wireless* yang paling sering digunakan adalah standar dari IEEE. IEEE menciptakan standar dengan aturan yang dibuat FCC. Spesifikasi yang digunakan dalam WLAN adalah 802.11 standar yang berhubungan dengan kecepatan akses data. Standar IEEE yang pernah dikeluarkan untuk *Wireless LAN* dapat dilihat melalui gambar dibawah ini.



### Spesifikasi Wi-Fi

Spesifikasi	Kecepatan	Frekuensi Band	Cocok dengan
802.11b	11 Mb/s	~2.4 GHz	b
802.11a	54 Mb/s	~2.4 GHz	a
802.11g	54 Mb/s	~2.4 GHz	b, g
802.11n	100 Mb/s	~5 GHz	b, g, n

**Gambar 2.6** Spesifikasi WiFi Standar IEEE.

Versi Wi-Fi yang paling luas dalam pasaran AS sekarang ini (berdasarkan dalam IEEE 802.11b/g) beroperasi pada 2.400 MHz sampai 2.483,50 MHz. *Channel* Wi-fi yang beroperasi di frekuensi yang dimiliki oleh standar 802.11 b/g dapat dilihat pada gambar dibawah ini.

Channel	Frequency (MHz)
Channel 1	2.412
Channel 2	2.417
Channel 3	2.422
Channel 4	2.427
Channel 5	2.432
Channel 6	2.437
Channel 7	2.442
Channel 8	2.447
Channel 9	2.452
Channel 10	2.457
Channel 11	2.462

**Gambar 2.7** *Channel* WiFi pada standar 802.11b/g.

## 2.3 Perangkat Keras

Perangkat keras (*Hardware*) memiliki macam-macam jenis dengan fungsi yang beragam. Secara umum, pengertian perangkat keras (*hardware*) adalah perangkat (modul) yang berbentuk fisik (dapat disentuh). Fungsi perangkat keras (*hardware*) adalah memberikan masukan, mengolah dan menampilkan keluaran, dan menjalankan suatu perintah.

### 2.3.1 Schneider Automation Server

*Automation Server* merupakan suatu perangkat *Server* yang dapat melakukan fungsi utama, seperti logika kontrol, *logging* tren, dan pengawasan alarm, dan mendukung komunikasi dan konektivitas ke bus I / O dan *field bus*. Intelijen terdistribusi dari solusi *SmartStruxure* memastikan toleransi kesalahan pada sistem dan menyediakan antarmuka pengguna berfitur lengkap melalui *WorkStation* dan *WebStation*.



**Gambar 2.8** Automation Server

*Automation Server* adalah perangkat yang hebat yang dapat bertindak sebagai *Server* mandiri dan juga mengendalikan modul I / O dan memantau dan mengelola perangkat bus lapangan. Dalam instalasi kecil, *Automation Server* yang disematkan bertindak sebagai *Server* mandiri, yang terpasang dengan modul I / O-nya di tapak kecil (*terminal base*). Dalam instalasi menengah dan besar, fungsionalitas didistribusikan melalui beberapa perangkat *Server SmartStruxure* yang berkomunikasi melalui TCP / IP.

*Automation Server* mampu mengkoordinasikan lalu lintas I/O dan lalu lintas bus dalam satu jaringan, AS-P dapat mengirimkan data secara langsung ke *Server* lain di luar lokasi. AS-P dapat menjalankan beberapa program pengendalian, mengelola *Input/Output*, alarm, pengguna lokal, menangani penjadwalan dan logging, dan berkomunikasi menggunakan berbagai protokol.

*Automation Server* (AS-P) memiliki beberapa port yang aktif untuk keperluan komunikasi melalui beberapa *protocol*, perangkat, dan *Server*.

AS-P memiliki beberapa port dibawah ini:

1. Dua Port *Ethernet* 10/100
2. Dua Port RS-485
3. Satu Port LonWorks TP/FT
4. Satu Port USB (*host*)
5. Satu Port USB (*device*)

Port perangkat USB memungkinkan Anda untuk melakukan *upgrade* dan berinteraksi dengan AS-P menggunakan *Device Administrator*. Port *host* USB dapat digunakan untuk memberikan tenaga dan komunikasi untuk layar sentuh AD.

Dua port *Ethernet* terhubung ke switch *Ethernet* yang tertanam. Satu port harus terhubung ke jaringan situs. Port lainnya dapat digunakan untuk menghubungkan *WorkStation* atau *WebStation* tunggal, unit *MODBUS TCP*, atau *BACnet / IPdevice*, namun bukan *Server SmartStruxure* lainnya.

*SmartStruxure Solution* menyediakan sistem perizinan yang kuat yang mudah dikelola, fleksibel, dan disesuaikan dengan semua jenis ukuran sistem. Sistem perizinan menyediakan tingkat keamanan dengan standar tertinggi. Otentikasi dilakukan terhadap sistem pengelolaan akun pengguna yang telah diverifikasi atau terhadap *Windows Active Directory Domains*. Sistem pengelolaan akun bawaan memungkinkan *administrator* menyetel kebijakan kata sandi yang memenuhi pedoman Keamanan *Cyber* yang ketat. Saat *Windows Active Directory* digunakan, biaya administrasi lebih rendah karena pengguna tidak perlu dikelola di beberapa direktori.

Melalui pengalaman apapun, pengalaman pengguna serupa terlepas dari *Server SmartStruxure* yang digunakan pengguna untuk masuk. Pengguna dapat *log* langsung ke ASP untuk insinyur, komisi, mengawasi, dan *monitor* AS-P serta modul I / O terlampir dan perangkat bus lapangan (*Field Bus*).

Salah satu landasan solusi *SmartStruxure* mendukung standar protocol terbuka. AS-P secara *native* dapat berkomunikasi dengan tiga standar paling populer untuk *Building Automation System: BACnet, LonWorks, dan MODBUS*.

AS-P secara *native* mengintegrasikan konfigurasi *Master* dan *Slave MODBUS RS-485*, juga klien dan *Server TCP*. Ini memungkinkan akses penuh ke produk pihak ketiga dan jajaran produk *Schneider Electric* yang berkomunikasi

dalam protokol *MODBUS*, seperti *power meter*, UPS, pemutus sirkuit, dan pengendali pencahayaan.

AS-P mendukung penggunaan Layanan Web berdasarkan standar terbuka, seperti *SOAP* dan *REST*, untuk mengkonsumsi data ke dalam solusi *SmartStruxure*. Gunakan data pihak ketiga yang masuk (ramalan suhu, biaya energi) berlebih Web untuk menentukan *mode*, penjadwalan, dan pemrograman situs.

AS-P dan keluarga modul I / O dirancang untuk memenuhi kebutuhan unik setiap instalasi. Bergantung pada konfigurasi, masing-masing AS-P dapat mengendalikan hingga 464 poin I / O. Karena listrik dan komunikasi dikirim bersama *field bus* terpasang, beberapa modul dapat dipasang bersamaan tanpa alat dalam proses satu langkah sederhana menggunakan konektor.

Unik untuk industri, AS-P memiliki pilihan pemrograman *Script and Function Block*. Fleksibilitas ini memastikan bahwa metode pemrograman terbaik dapat dipilih untuk aplikasi ini.

*Script* menggunakan program C dasar melalui software *Script Menta Editor*, sementara *Function Block* seperti Bahasa program pada PLC melalui software *Function Block Menta Editor*.

### **2.3.2 Power Analyzer (Power Meter) Circutor CVM-NRG96**

CVM-NRG 96 *Power Analyzer* adalah alat ukur yang dapat diprogram, menawarkan serangkaian pilihan untuk menggunakannya, yang mungkin dipilih dari menu konfigurasi pada instrumen itu sendiri. Sebelum memulai bagian analisa dengan hati-hati periksa catu daya, koneksi dan setting dan pilih bentuk operasi yang paling sesuai untuk mendapatkan data yang dibutuhkan.

CVM NRG 96 mengukur, menghitung dan menampilkan parameter listrik utama dalam tiga fase sistem industri (seimbang atau tidak seimbang). Pengukuran berada dalam nilai efektif benar, melalui tiga masukan voltase AC dan tiga masukan arus AC, masukkan arus melalui trafo CT (X/5A), dimana 5A merupakan sekunder dari toroida trafo CT.

CVM-NRG 96 memungkinkan tampilan semua parameter listrik yang ditunjukkan di atas, dengan menggunakan *display* LCD yang menyala kembali, menunjukkan 4 parameter listrik instan, maksimum atau minimum pada setiap lompatan halaman.



**Gambar 2.9** Power Analyzer Circutor CVM-NRG96

Dengan prosesornya, stasiun pengukuran memungkinkan analisis simultan dari beberapa parameter sesuai tabel dibawah ini.



Tabel 2.5 Pembacaan Parameter Listrik Pada CVM-NRG96

MAGNITUDE	UNIT	L1	L2	L3	III
Simple Voltage	V	•	•	•	
Compound voltage	V	•	•	•	
Current	A	•	•	•	••
Frequency	Hz	•			
Active power	kW	•	•	•	•
Reactive Power L	kvarL	•	•	•	•
Reactive Power C	kvarC	•	•	•	•
Apparent Power	kVA				•
Power Factor	PF	•	•	•	
Cos $\phi$	Cos $\phi$				•
Maximum Demand	Pd			•	
Neutral Current	I <sub>N</sub>			•	
Voltage THD	% THD – V	•	•	•	
Current THD	% THD – A	•	•	•	
kWh (consumption and generation)	W·h				•
kvarh.L (consumption and generation)	W·h				•
kvarh.C (consumption and generation)	W·h				•
kVAh (consumption and generation)	W·h				•
Harmonic decomposition (V and A) *	%	•	•	•	15th

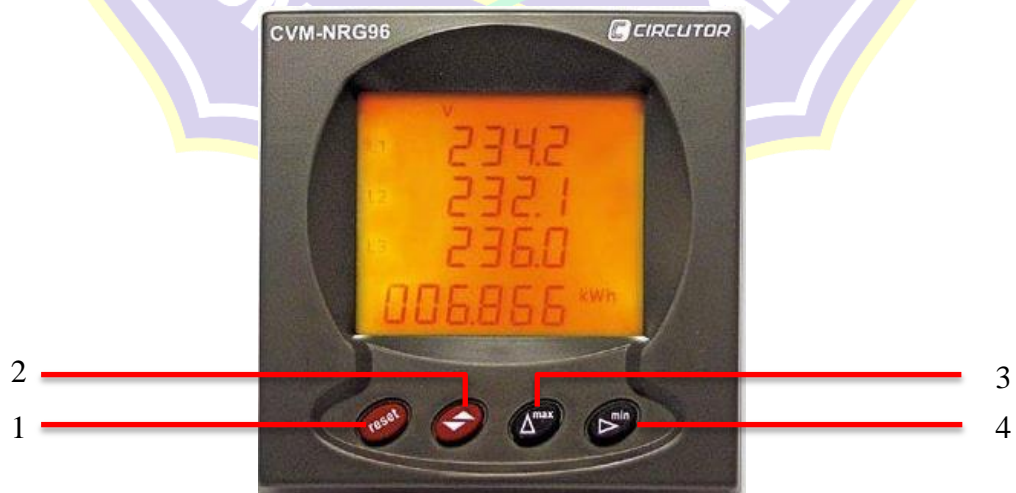
Keterangan : (•) Tersedia melalui tampilan dan komunikasi

(••) Tersedia hanya melalui komunikasi

(\*) Harmonisasi dekomposisi pada *model* HAR.





### 2.3.2.1 Bagian – Bagian Tombol Fungsi Pada CVM NRG 96

Bagian – bagian tombol fungsi pada CVM – NRG 96 dapat dilihat pada gambar dan tabel dibawah ini.



Gambar 2.10 Tampak Depan CVM NRG96

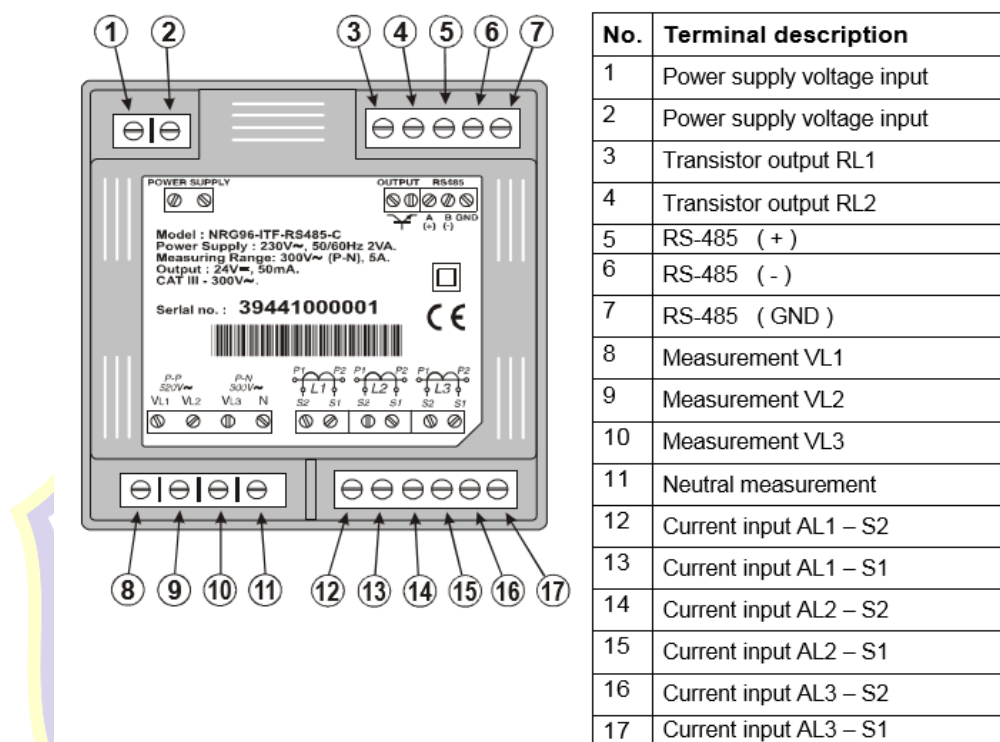
Tabel 2.6 *Function Button* pada CVM-NRG96

No	<i>Function Button</i>	Keterangan
1		<i>Key Reset</i> merupakan tombol yang berfungsi untuk hal sebagai berikut: <ol style="list-style-type: none"> <li>1. Memulai peralatan.</li> <li>2. Penghapusan nilai Maksimum dan Minimum.</li> <li>3. Melakukan pengulangan untuk settingan yang baru.</li> </ol>
2		<i>Key Display</i> merupakan tombol yang berfungsi untuk hal sebagai berikut: <ol style="list-style-type: none"> <li>1. Menampilkan semua variabel dengan penekanan berulang.</li> <li>2. Tombol fungsi dalam menu pengaturan: menekan tombol <i>Display</i> bergerak maju melalui layar yang berbeda, baik pada menu konfigurasi dan menu komunikasi.</li> <li>3. Dalam <i>mode runtime</i>, tekan lama (tahan tombol ditekan selama 2 detik), menampilkan meter energy.</li> </ol>
3		<i>Key Max</i> merupakan tombol yang berfungsi untuk melakukan tampilan maksimum atau minimum untuk setiap variabel yang ditampilkan; fungsi ini hanya berlaku saat tombol sedang ditekan. Setelah berhenti ditekan, nilai instan muncul lagi setelah lima detik.
4		<i>Key MIN</i> memilih kode atau parameter yang akan diubah dan tombol <i>MAX</i> memberikan kode dan / atau variabel yang sesuai.



### 2.3.2.2 Bagian – Bagian Terminal Pada CVM NRG96

Bagian – bagian terminal pada CVM-NRG96 dapat dilihat pada gambar dibawah ini.



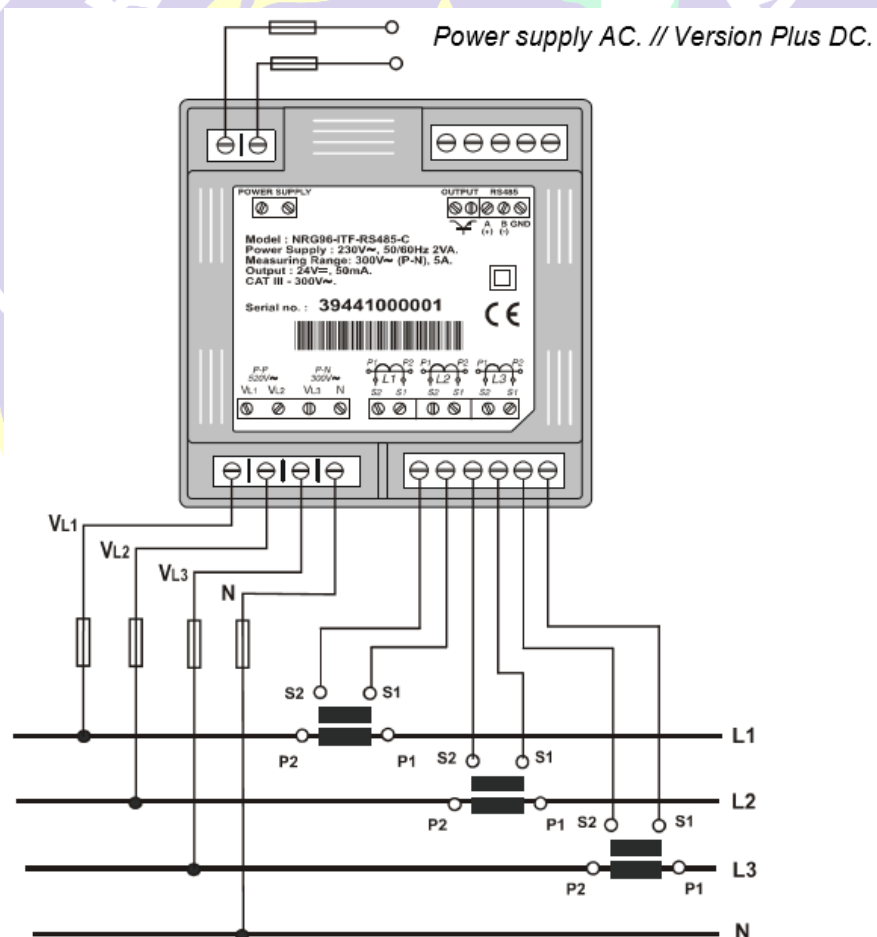
**Gambar 2.11** Bagian – Bagian Terminal Pada CVM NRG96

1. *Power Supply* ( terminal 1 dan 2) berfungsi sebagai *supply* untuk mengaktifkan modul.
2. *Transistor Output* (terminal 3 dan 4) berfungsi sebagai *output* untuk pengaturan apabila terjadi *over current* melalui referensi pengaturan.
3. RS-485 (terminal 5, 6, dan 7) berfungsi sebagai penghubung kabel komunikasi *MODBUS*.
4. *Measurement VL1, VL2, dan VL3* (terminal 8, 9, dan 10) sebagai terminal masukan voltase *system 3 fasa*.
5. *Neutral Measurement* (terminal 11) sebagai terminal masukan netral *system 3 fasa*.

6. *Current Input AL1 – S2 dan AL1 – S1* (terminal 12 dan 13) sebagai masukan arus fasa L1 melalui trafo CT.
7. *Current Input AL2 – S2 dan AL2 – S1* (terminal 14 dan 15) sebagai masukan arus fasa L2 melalui trafo CT.
8. *Current Input AL3 – S2 dan AL3 – S1* (terminal 16 dan 17) sebagai masukan arus fasa L3 melalui trafo CT.

### 2.3.2.3 Pengawatan Pada Terminal CVM-NRG96

Pengawatan pada terminal CVM-NRG96 dapat dilihat melalui gambar dibawah ini.



**Gambar 2.12** Pengawatan pada terminal CVM-NRG96

Tegangan *Power Supply* untuk modul CVM-NRG96

- a) Voltase : 230 VAC // 95....300VDC
- b) *Frequency* : 50-60 Hz
- c) *Power supply* toleransi : -15% / +10%
- d) Konsumsi peralatan : 5VA
- e) Koneksi terminal : Terminal 1-2 VAC // Terminal 1 (DC+) 2 (DC-)

#### 2.3.2.4 MODBUS Protocol pada CVM-NRG96

Satu atau beberapa instrumen CVM-NRG96 dapat dihubungkan ke komputer untuk mengotomatisasi proses produksi atau sistem kontrol energi. Seperti halnya pengoperasian masing-masing instrumen, sistem ini dapat memusatkan data pada satu titik tunggal. Untuk alasan ini CVM-NRG96 memiliki *output* komunikasi RS-485. Jika lebih dari satu instrumen dihubungkan ke satu rangkaian seri tunggal (RS-485), perlu menetapkan masing-masing nomor atau alamat (dari 01 sampai 255) sehingga komputer pusat mengirimkan permintaan yang sesuai ke alamat ini untuk setiap perangkat.

Tabel 2.7 Format untuk setiap *byte* dalam *mode* RTU.

Code	8 bit binary, <i>hexadecimal</i> 0-9, A-F 2 <i>hexadecimal</i> characters mengandung masing – masing 8- bit data pada pesan
Bits per byte	8 data bits
Field Check Error	CRC Type (Cyclical Redundancy Check)

Komposisi kabel RS-485 harus dilakukan dengan kabel layar *meshing* (minimal 3 kawat) dengan jarak tempuh maksimum 1.200 meter antara CVM-NRG96 dan *Master* unit. *Bus* ini dapat menghubungkan maksimal 32 analisa

CVM-NRG96. Dalam hal pembacaan *variable MODBUS* maka jarak antara *Slave* dan *Master* harus terukur agar tidak terjadi kerusakan data yang dikirim dan direspon oleh *Slave* (CVM-NRG96) dan *Master* (*computer*).

Dalam hal pembacaan CVM-NRG96 memiliki kode fungsi pada *MODBUS* untuk menampilkan beberapa *variable* pembacaan ke *Master* (*computer*). Berikut dibawah ini kode fungsi *MODBUS* pada CVM-NRG96.

Tabel 2.8 Kode fungsi *MODBUS* pada CVM-NRG96

Kode Fungsi	Keterangan
<i>Function 01</i>	Membaca status dari Relay
<i>Function 03 dan 04</i>	Membaca nilai parameter listrik yang diukur oleh CVM-NRG96
<i>Function 05</i>	Memerintahkan relay

*MODBUS variable* merupakan alamat – alamat *database model hexadecimal* yang akan dikirim melalui perangkat *Master* kemudian diterima oleh perangkat *Slave*. *Master* bertindak sebagai *questioner* (meminta) data, *Slave* sebagai *responder* (memberi) data.

Tabel 2.9 *MODBUS variable* pada CVM-NRG96

<i>Magnitude</i> (Besaran)	<i>Symbol</i>	<i>Variable</i>	<i>Unit</i>
<i>Voltage Phase</i>	V L1	00-01	V x 10
<i>Current</i>	A L1	02-03	mA
<i>Active Power</i>	kW L1	04-05	w
<i>Reactive Power</i>	Kvar L1	06-07	w
<i>Power Factor</i>	PF L1	08-09	x 100
<i>Voltage Phase</i>	V L2	0A-0B	V x 10
<i>Current</i>	A L2	0C-0D	mA
<i>Active Power</i>	kW L2	0E-0F	w
<i>Reactive Power</i>	Kvar L2	10-11	w
<i>Power Factor</i>	PF L2	12-13	x 100
<i>Voltage Phase</i>	V L3	14-15	V x 10

<i>Magnitude (Besaran)</i>	<i>Symbol</i>	<i>Variable</i>	<i>Unit</i>
<i>Current</i>	A L3	16-17	mA
<i>Active Power</i>	kW L3	18-19	w
<i>Reactive Power</i>	Kvar L3	1A-1B	w
<i>Power Factor</i>	PF L3	1C-1D	x 100
<i>Active Power III</i>	kW III	1E-1F	w
<i>Inductive Power III</i>	KvarL III	20-21	w
<i>Capacitive Power III</i>	KvarC III	22-23	w
<i>Cos phi III</i>	Cos phi III	24-25	x 100
<i>Power Factor III</i>	PF III	26-27	x 100
<i>Frequency</i>	Hz	28-29	Hz x 10
<i>Voltase Line L1-L2</i>	V12	2A-2B	V x 10
<i>Volatse Line L2-L3</i>	V23	2C-2D	V x 10
<i>Voltase Line L1-L3</i>	V31	2E-2F	V x 10
<i>Active Energy</i>	kW-h III	3C-3D	w-h
<i>Reactive Energy (L)</i>	kVarL-h III	3E-3F	w-h
<i>Reactive Energy (C)</i>	kVarC-h III	40-41	w-h
<i>Apparent Energy</i>	kVA-h III	56-57	w-h
<i>Active energy generated</i>	kW-h III	58-59	w-h
<i>Inductive energy generated</i>	kVarL-h III	5A-5B	w-h
<i>Capacitive energy generated</i>	kVarC-h III	5C-5D	w-h
<i>Apparent energy genarated</i>	kVA-h III	5E-5F	w-h

Contoh dari MODBUS question

QUESTION Melalui Master

**(0A 04 00 00 00 0A 71 76)**

0A	Alamat perangkat <i>Slave</i> , 10 dalam <i>decimal</i>
04	Fungsi Membaca
00 00	Merekam pembacaan dari awal mula, 0 dalam <i>decimal</i>
00 0A	Nomor yang direkam, 10 dalam <i>desimal</i>
71 76	CRC Character

RESPONSE Dari *Slave*

**(0A 04 14 00 00 08 4D 00 00 23 28 00 00 0F A0 00 00 00 90 00 00 00 60 CB 2E)**

0A	Perangkat <i>Slave</i> yang merespon, 10 dalam <i>decimal</i>
04	Fungsi membaca
14	Nomor <i>bytes</i> yang diterima (20)
00 00 08 4D	V1 x 10 (merekam <i>variable</i> 00 – 01), 2125 V dalam <i>decimal</i>

00 00 23 28	mA (merekam <i>variable</i> 02 – 03), 9000 mA dalam <i>decimal</i>
00 00 0F A0	W (merekam <i>variable</i> 04-05), 4000 w dalam <i>decimal</i>
00 00 00 90	varL (merekam <i>variable</i> 06-07), 144 varL dalam <i>decimal</i>
00 00 00 60	PF x 100 (merekam 08-09), 96 dalam desimal
CB 2E	CRC Character

### 2.3.3 Thermostat Schneider TC-300

Thermostat TC300 merupakan alat yang dirancang untuk pengendalian suhu di lingkungan industri, komersial dan rumah tangga. Bentuk fisik Thermostat TC-300 dapat dilihat pada gambar dibawah ini.



**Gambar 2.13** Thermostat TC300

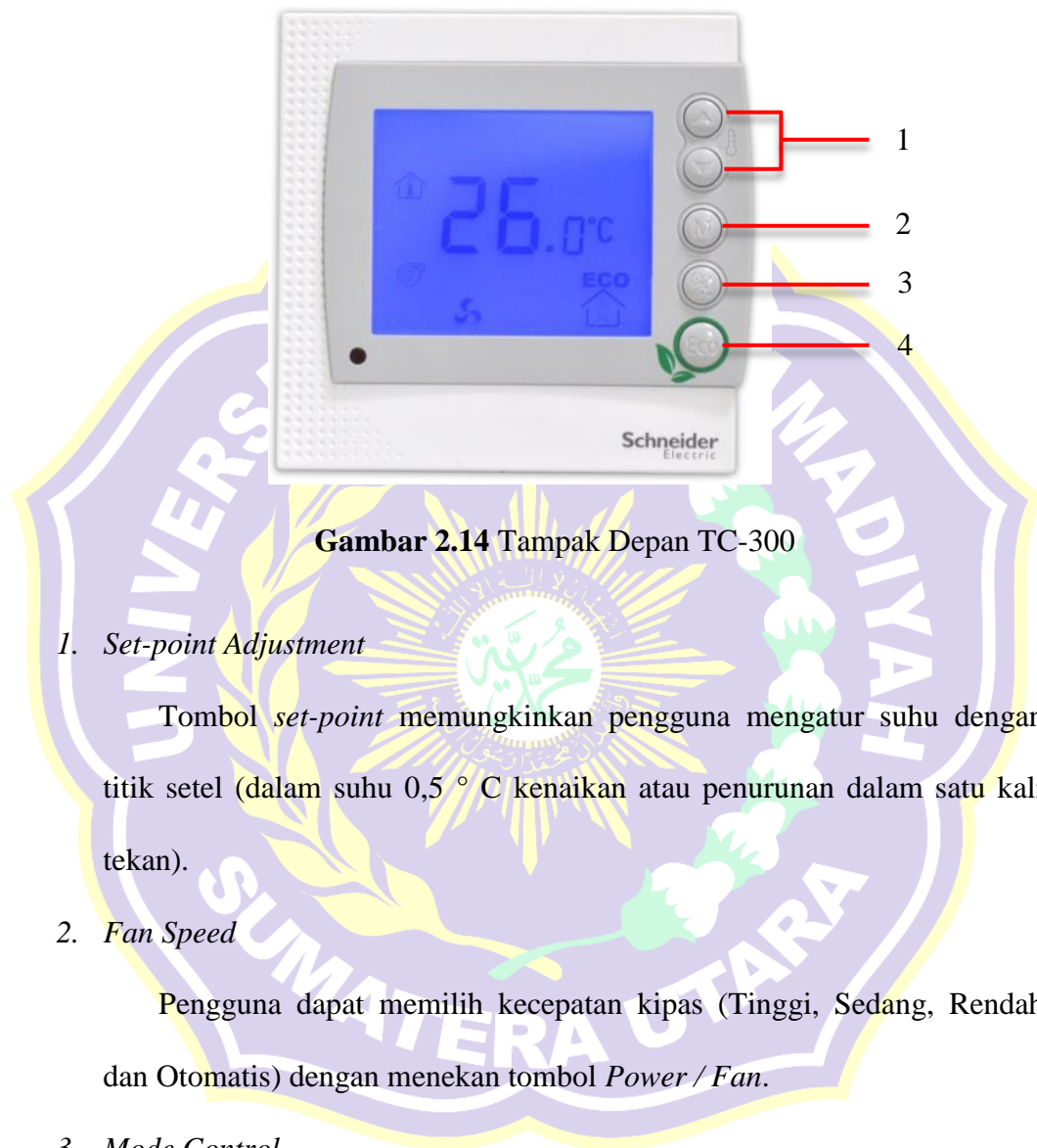
Termostat meningkatkan kenyamanan dan menghemat kontrol dengan membandingkan temperature ruangan dengan setting yang diinginkan. Informasi ini digunakan untuk mengendalikan unit koil kipas.

TC300 dilengkapi kontrol berbasis mikroprosesor dan layar LCD besar. *Mode* tampilan LCD termasuk status operasi (pendinginan, pemanasan, ventilasi), kecepatan kipas, suhu kamar, dan pengaturan suhu.



### 2.3.3.1 Bagian – Bagian *Thermostat* TC-300

Tombol – tombol pada *thermostat* memiliki defenisi berbeda Bagian – bagian *thermostat* TC-300 dapat dilihat melalui gambar dibawah ini.



**Gambar 2.14** Tampak Depan TC-300

#### 1. *Set-point Adjustment*

Tombol *set-point* memungkinkan pengguna mengatur suhu dengan titik setel (dalam suhu 0,5 ° C kenaikan atau penurunan dalam satu kali tekan).

#### 2. *Fan Speed*

Pengguna dapat memilih kecepatan kipas (Tinggi, Sedang, Rendah dan Otomatis) dengan menekan tombol *Power / Fan*.

#### 3. *Mode Control*

Pengguna dapat memilih kecepatan kipas (Tinggi, Sedang, Rendah dan Otomatis) dengan menekan tombol *Power / Fan*.

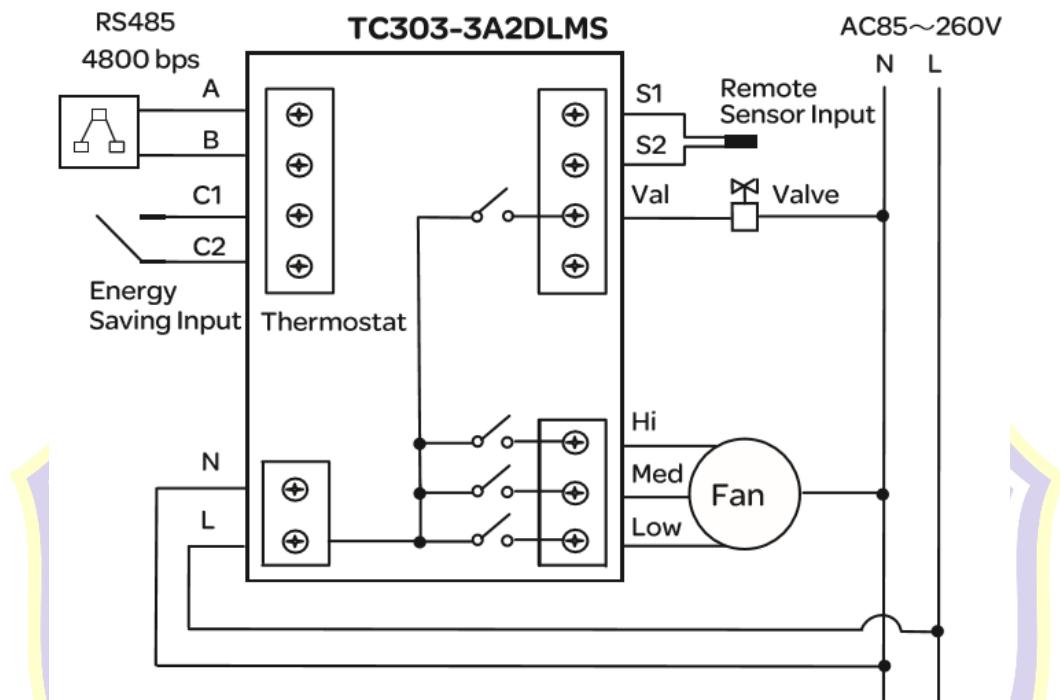
#### 4. *Eco Saving Mode*

Menekan tombol *Eco* memulai *mode* penghematan energi.



### 2.3.3.2 Pengawatan Pada Terminal *Thermostat* TC-300

Pengawatan pada terminal *thermostat* TC-300 dapat dilihat melalui gambar dibawah ini.



**Gambar 2.15** Pengawatan pada terminal *Thermostat* TC-300

### 2.3.3.3 *MODBUS* Protocol Pada *Thermostat* TC-300

Termostat TC300 hadir dengan pilihan komunikasi *MODBUS* untuk konfigurasi. *MODBUS* adalah protokol komunikasi *serial* yang terbuka, tersebar luas dan mapan yang digunakan dalam otomasi bangunan.

Dukungan komunikasi *MODBUS* memungkinkan integrasi sederhana dari termostat TC300 ke sistem manajemen bangunan dengan menggunakan komunikasi *serial MODBUS* standar.

Termostat TC300 berkomunikasi sebagai perangkat *Slave (client)*. *MODBUS* RTU melalui koneksi *serial* RS-485, yang memungkinkan transfer data *real-time*.

Parameter komunikasi RS-485 tidak dapat disesuaikan dan diperbaiki sebagai berikut: 4800bps *Baud Rate*, 8 *Data Bits*, *Odd Parity*, 1 *Stop Bit*.

Alamat *MODBUS* aktual dari termostat dapat diatur untuk setiap perangkat pada jaringan RS-485. Alamat didefinisikan melalui menu parameter pada termostat itu sendiri. Rentang alamat yang tersedia adalah dari 01 sampai 32 dan didefinisikan pada *item menu* 04 dari pengaturan parameter.

#### 2.3.3.4 *MODBUS Register Pada Thermostat TC-300*

*MODBUS Register* merupakan *register database* pada *Slave* yang nilainya akan dikirim ke *Master* secara *realtime*.

Tabel 2.10 *MODBUS variable (register) pada Thermostat TC-300*

<i>Register Address</i>	<i>Description</i>	<i>Function Code</i>	<i>Definition</i>
1	<i>Room Temperatur</i>	4	<i>Temperature</i> (0 to 50 °C)
3	<i>Thermostat Mode</i>	3 and 6	0 = <i>Off</i> 1 = <i>On</i> 2 = <i>Frost Protection</i>
4	<i>Operating Mode</i>	3 and 6	0 = <i>Cool</i> 1 = <i>Heat</i> 2 = <i>Ventilation</i>
5	<i>Room Temperature Set Point</i>	3 and 6	<i>Temperature</i> (5 to 35 °C)
6	<i>Fan Mode</i>	3 and 6	0 = <i>High</i> 1 = <i>Medium</i> 2 = <i>Low</i> 3 = <i>Auto</i>
7	<i>Unoccupied Room Temperature Set-point (Cooling Mode)</i>	3 and 6	<i>Temperature</i> (22 to 32 °C)
8	<i>Unoccupied Room Temperature Set-point (Heating Mode)</i>	3 and 6	<i>Temperature</i> (10 to 21 °C)
9	<i>Sleep Mode</i>	3 and 6	0 = <i>Disable</i> 1 = <i>Enable</i>
10	<i>Eco Mode</i>	3 and 6	0 = <i>Disable</i> 1 = <i>Enable</i>
11	<i>Occupancy Status</i>	3 and 6	0 = <i>Unoccupied</i> 1 = <i>Occupied</i>

Register Address	Description	Function Code	Definition
12	Unoccupied Fan Speed Mode	3 and 6	0 = High 1 = Medium 2 = Low
13	Keypad Status	3 and 6	0 = Unlocked 1 = Locked
3	Embedded Temperature Sensor Status	2	0 = OK, 1 = Fault
4	Remote Temperature Sensor Status	2	0 = OK, 1 = Fault
1	Cooling Valve in 4-Pipe Application	1	0 = Off 1 = On
8	Heating Valve in 4-Pipe Application	1	0 = Off 1 = On

#### 2.3.4 WELLPRO Module I/O WP9038ADAM

WP9038 merupakan perangkat I/O besutan WELLPRO, memiliki 6 saluran *input analog* (DC 0-20mA / 4-20mA), 4 saluran *digital input* berupa optoelektrik isolasi, 4 saluran *digital output* optoelektrik (transistor NPN). Bentuk fisik WELLPRO WP9038ADAM dapat dilihat melalui gambar dibawah ini.



Gambar 2.16 WP9038ADAM

Modul I/O WP9038 memiliki standar komunikasi RS485 *MODBUS* RTU. Bisa terjaring dengan perangkat lunak terkonfigurasi, PLC, dan HMI. Sirkuit komunikasi dirancang untuk meringankan proteksi dan gangguan dengan mendeteksi sinyal I/O yang masuk dan keluar melalui perangkat ini.

Dengan menggunakan *software Wellpro Debugging Software*, konfigurasi dapat dilakukan untuk menyesuaikan nilai *baudrate*, *databit*, *Slave id*, dan *parity*. Nilai *input* dan *output* dapat dilihat melalui perangkat lunak tersebut. Pada *analog input*, hanya dapat membaca *current* 4 – 20 mA.

### **2.3.5 PLC Schneider TM221ME16R**

PLC merupakan perangkat elektronik yang didesain untuk digunakan pada *industry* yang mengontrol suatu *system* ataupun sekelompok *system* baik data I/O *analog* atau *digital*. *Programmable Logic Controller* menurut Capiel (1982) adalah sistem elektronik yang beroperasi secara digital dan didisain untuk pemakaian di lingkungan industri, dimana sistem ini menggunakan memori yang dapat diprogram untuk penyimpanan secara internal instruksi-instruksi yang mengimplementasikan fungsi-fungsi spesifik seperti logika, urutan, perwaktuan, pencacahan dan operasi aritmatik untuk mengontrol mesin atau proses melalui modul-modul I/O digital maupun analog.

PLC TM221ME16R merupakan nano PLC besutan Schneider Electric yang dirancang dalam bentuk yang kompak agar mudah dalam melakukan pemetaan pada panel tetapi tidak mengurangi fungsional dari PLC tersebut. Bentuk fisik dari PLC Schneider TM221ME16R dapat dilihat melalui gambar dibawah ini.



**Gambar 2.17** PLC Schneider TM221ME16R

PLC Modicon TM221ME16R merupakan PLC model terbaru setelah model sebelumnya yaitu Twido PLC. Dalam keperluan tugas akhir ini, PLC yang digunakan memiliki 8 I/O, dimana 8 port sebagai digital input dan 8 port sebagai digital output serta memiliki 2 port analog input.

### 2.3.5.1 Bagian – Bagian PLC Modicon TM221ME16R

Bagian – bagian pada PLC Modicon TM221ME16R dapat dilihat melalui gambar dibawah ini.



**Gambar 2.18** Bagian - Bagian PLC Schneider TM221ME16R



1. *Analog Input.*

Bagian yang merupakan port *input* masukan berupa analog ( 0 – 10V).

2. *Serial Port.*

Port yang akan digunakan untuk melakukan komunikasi dengan perangkat yang lain menggunakan *protocol MODBUS RTU*. PLC dapat digunakan sebagai *Master* maupun sebagai *Slave*.

3. *Ethernet Port*

Port yang digunakan untuk menghubungkan PLC dengan perangkat yang lain menggunakan *protocol MODBUS TCP/IP* dalam satu segmen atau jaringan yang sama. PLC dapat diawasi dan dikontrol secara jarak jauh menggunakan port tersebut.

4. *SD Card Slot*

Sebagai slot masukan *SD Card* sebagai media penyimpanan program. Kapasitas *SD Card* yang dapat digunakan mencapai 32GB.

5. *Saklar Start / Stop*

Saklar ini digunakan untuk mengoperasikan PLC secara external apabila aplikasi yang ada dalam PLC tidak dalam *mode RUN*.

6. *Mini USB port*

USB port digunakan untuk mengunduh atau mengunggah program dari computer ke dalam PLC melalui perangkat lunak SoMachine Basic.

7. *Blok Terminal Masukan*

Bagian dari PLC yang berfungsi sebagai port masukan sensor dan berbagai masukan lainnya dalam bentuk masukan digital.

## 8. Blok Terminal Keluaran

Bagian dari PLC yang berfungsi sebagai port keluaran dalam bentuk keluaran digital.

### 2.3.5.2 Metode Pemrograman PLC

Untuk dapat menjalankan fungsinya sebagai peralatan *control*, PLC harus diprogram sesuai dengan fungsi *control* yang diinginkan. Pada PLC M221, perangkat lunak yang digunakan untuk melakukan pemrograman adalah *SoMachine Basic*. Pada perangkat lunak ini terdapat dua pilihan bahasa pemrograman, yaitu:

- a) IL (*Instruction List*)
- b) LD (*Ladder Diagram*)

Masing – masing dari bahasa pemrograman akan dibahas pada subbab perangkat lunak *Somachine Basic*.

Pada PLC M221 apabila telah diprogram menggunakan bahasa pemrograman *Instruction List*, program dapat langsung dikonversikan ke dalam bahasa pemrograman *Ladder Diagram* tanpa harus mengulangi pemrograman dari awal.

### 2.3.6 TP-Link MR3420 Wireless dan Router

TP-Link MR3420 berfungsi sebagai media yang menghubungkan *smart device* dengan *Automation Server* secara nirkabel (*wireless*). TP – Link bertindak sebagai *Router* yaitu perangkat *network* yang digunakan untuk menghubungkan beberapa *network*, baik *network* yang sama maupun berbeda dari segi teknologinya seperti menghubungkan *network* yang menggunakan topologi *Bus*,



*Star* dan *Ring* secara *wireless*. Bentuk fisik TP-Link MR3420 dapat dilihat melalui gambar dibawah ini.

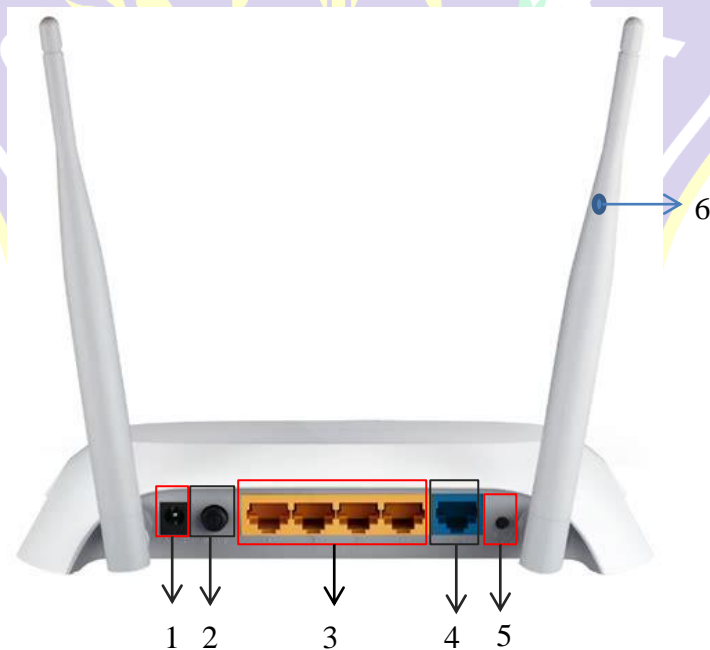


**Gambar 2.19** TP-Link MR3420 *Wireless and Router* (tampak depan).

TP-Link juga memiliki *switch* untuk menghubungkan beberapa perangkat dengan segmen IP Adress yang sama menggunakan kabel LAN.

### 2.3.6.1 Bagian – Bagian TP-Link MR3420.

Bagian – bagian TP-Link MR3420 dapat dilihat melalui gambar dibawah ini.



**Gambar 2.20** TP-Link MR3420 (tampak belakang)

Dilihat dari gambar diatas, bagian – bagian dari TP-Link MR3420 adalah sebagai berikut:

1. *Slot Power Supply.*

Berfungsi sebagai slot catu daya 12 VDC.

2. *Push On/Off.*

Berfungsi sebagai saklar untuk mengaktifkan dan menonaktifkan perangkat TP-Link MR3420.

3. *Switch Port.*

Berfungsi sebagai penghubung antar perangkat yang memiliki segmen IP Address sama menggunakan kabel LAN.

4. *WAN Port.*

Berfungsi sebagai port koneksi untuk menghubungkan perangkat penyedia jaringan internet.

5. *Reset.*

Berfungsi untuk mengatur ulang konfigurasi kedalam setelan pabrik.

6. *Antena.*

Berfungsi sebagai pemancar dan penerima koneksi secara nirkabel.

### 2.3.7 *Current Transformer (CT)*

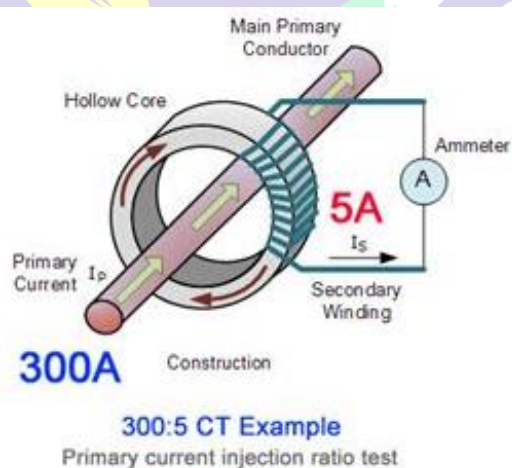
*Current Transformer* atau CT adalah salah satu *type* trafo instrumentasi yang menghasilkan arus di sekunder dimana besarnya sesuai dengan ratio dari arus primernya. Ada 2 standart yang paling banyak diikuti pada CT yaitu : IEC 60044-1 (BSEN 60044-1) & IEEE C57.13 (ANSI), meskipun ada juga standart Australia

dan Canada. Bentuk fisik *Current Transformer* dapat dilihat melalui gambar dibawah ini.



**Gambar 2.21** *Current Transformer (CT)*

Pada CT tertulis *class* dan *burden*, dimana masing masing mewakili parameter yang dimiliki oleh CT tersebut. *Class* menunjukkan tingkat akurasi CT, misalnya class 1.0 berarti CT tersebut mempunyai tingkat kesalahan 1%. *Burden* menunjukkan kemampuan CT untuk menerima sampai batas impedansi tertentu. CT standart IEC menyebutkan *burden* 1.5 VA (volt amper), 3 VA, 5 VA dst.



**Gambar 2.22** Ilustrasi Arus Mengalir Pada CT 300A/5A

CT umumnya terdiri dari sebuah inti besi yang dililiti oleh konduktor beberapa ratus kali. *Output* dari skunder biasanya adalah 1 atau 5 amper, ini ditunjukkan dengan ratio yang dimiliki oleh CT tersebut. Misal 100:1, berarti sekunder CT akan mengeluarkan *output* 1 amper jika sisi primer dilalui arus 100 Amper. Jika 300:5, berarti sekunder CT akan mengeluarkan *output* 5 ampere jika sisi primer dilalui arus 300 Ampere.

Aplikasi CT pada tugas akhir sebagai pembaca nilai arus yang akan digunakan oleh salah satu line yang diberi beban, kemudian pembacaan dari CT akan dihubungkan ke *power meter*.

## **2.4 Perangkat Lunak**

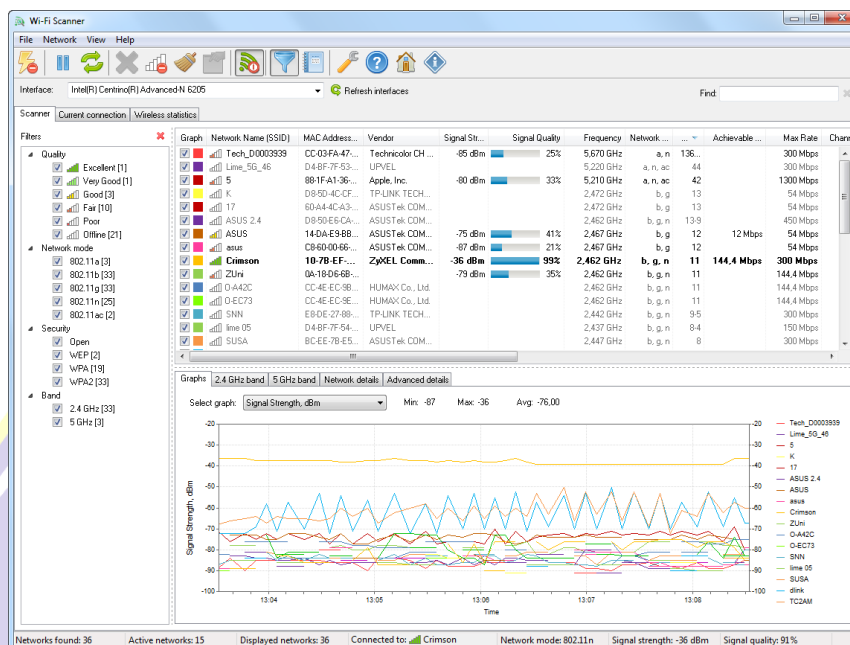
Perangkat Lunak (*Software*) adalah sekumpulan data elektronik yang disimpan dan diatur oleh komputer, data elektronik yang disimpan oleh komputer itu dapat berupa program atau instruksi yang akan menjalankan suatu perintah. Fungsi perangkat lunak pada tugas akhir ini adalah sebagai alat pendukung yang mengolah *database* dari *input / output* yang diterima oleh perangkat keras.

### **2.4.1 Lizard WiFi Scanner**

Wi-Fi *Scanner* digunakan untuk dengan mudah menemukan jaringan nirkabel yang terlihat dan informasi terkaitnya melalui *personal computer (PC)*. Alat ini mendapatkan nama jaringan (SSID), kekuatan sinyal (RSSI) dan kualitas, alamat MAC (BSSID), saluran, data *rate* maksimum dan dapat dicapai, keamanan, dan banyak lagi.

Wi-Fi *Scanner* berguna untuk pengguna jalur akses normal yang perlu mengetahui distribusi kekuatan sinyal untuk jaringan nirkabel mereka di rumah,

atau memilih posisi untuk titik akses mereka untuk kualitas sinyal yang optimal. Tampilan kerja pada perangkat lunak *Lizard Wifi Scanner* dapat dilihat dibawah ini.



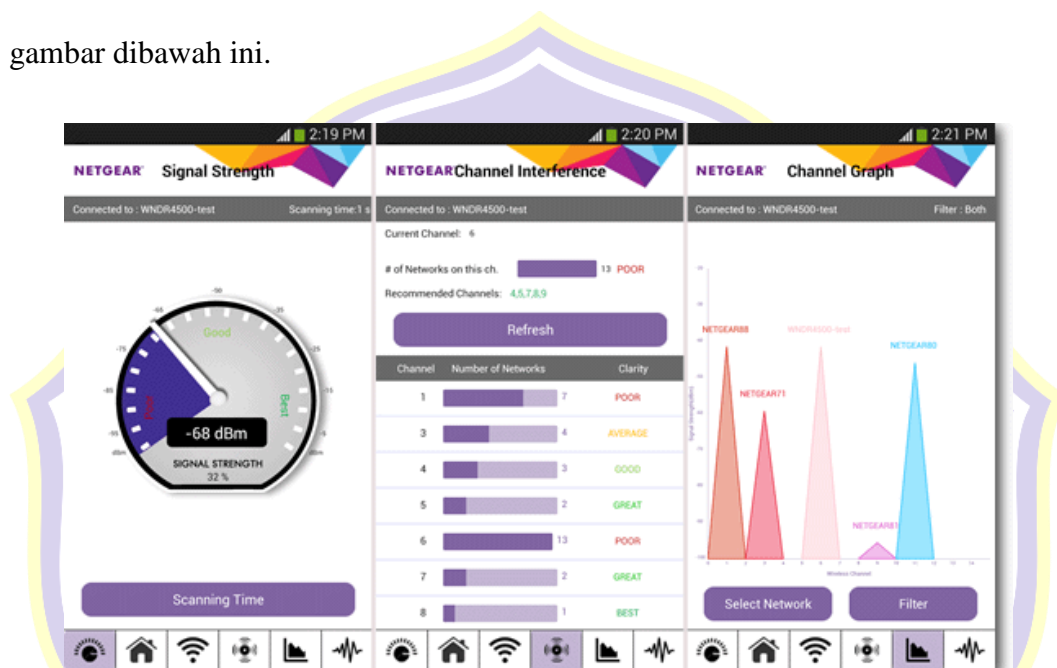
**Gambar 2.23** Tampilan Kerja Pada Software *Lizard Wifi Scanner*.

Dengan menggunakan *Wi-Fi Scanner*, Anda dapat mengevaluasi alokasi jaringan nirkabel berdasarkan saluran dan memilih *bandwidth* yang paling sedikit terpadatkan untuk jalur akses mereka, yang memungkinkan mereka untuk meningkatkan kecepatan koneksi mereka secara signifikan.

Selain itu, *Wi-Fi Scanner* adalah alat yang sangat diperlukan untuk administrator jaringan perusahaan dalam melakukan tugas-tugas seperti konfigurasi, perencanaan dan pemantauan parameter keamanan pada jaringan nirkabel bisnis.

### 2.4.2 Netgear Wifi Analytics

*Netgear Wifi Analytics* digunakan untuk memeriksa status jaringan, kekuatan sinyal WiFi, mengidentifikasi saluran WiFi ramai, menemukan saluran yang jelas dengan sedikit gangguan dan banyak lagi melalui ponsel pintar (Smart Phone). Tampilan kerja pada perangkat lunak *Netgear Wifi Analytics* dapat dilihat pada gambar dibawah ini.



**Gambar 2.24** Tampilan Kerja Pada Perangkat Lunak *Netgear Wifi Analytics*.

Dari gambar diatas, dapat dilihat tampilan kerja (*Signal Strength*, *Channel Interference*, dan *Channel Graph*) pada perangkat lunak *Netgear Wifi Analytics*.

*Meskipun* aplikasi ini dikembangkan oleh *Netgear*, aplikasi ini bekerja sempurna dengan sebagian besar *router* Wi-Fi apa pun, terlepas dari merek dan modelnya.

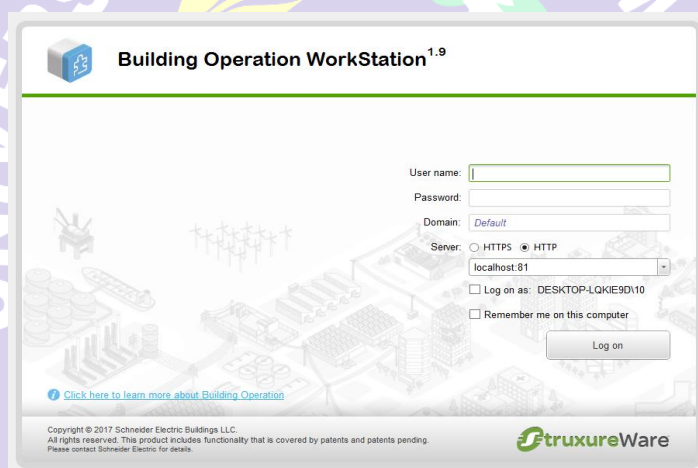
### 2.4.3 StruxureWare Building Operation Workstation

*StruxureWare Building Operation WorkStation* adalah perangkat lunak yang sepenuhnya terpilih untuk beroperasi dan mengelola semua aspek dari perangkat



lunak. WorkStation adalah jendela di mana pengguna dapat memantau penggunaan berbagai energi, air, dan lain sebagainya serta terus meningkatkan efisiensi bangunan yang ada.

*WorkStation* adalah sebuah antarmuka (*interface*) dimana pengguna dan insinyur dapat mengakses *Server SmartStruxure* mereka. Anda dapat melihat dan mengelola grafis, alarm, jadwal, tren log dan laporan. Insinyur dapat mengkonfigurasi dan memelihara semua aspek dari perangkat lunak ini. Tampilan beranda pada *StruxureWare Building Operation WorkStation* dapat dilihat melalui gambar dibawah ini.



**Gambar 2.25** Tampilan Beranda Pada Perangkat Lunak *Workstation*

*Workstation* mengharuskan setiap pengguna untuk memiliki *account*. Akses dapat melalui *account* yang dikelola oleh *SmartStruxure Solution* atau melalui *account Windows Active Directory*.

Perangkat lunak menyesuaikan bahasa yang ditampilkan, sistem pengukuran, dan tanggal/waktu format ke pengaturan sistem operasi. Bahasa dan pengukuran sistem dapat dengan mudah beralih dari dalam *WorkStation*. Terjemahan



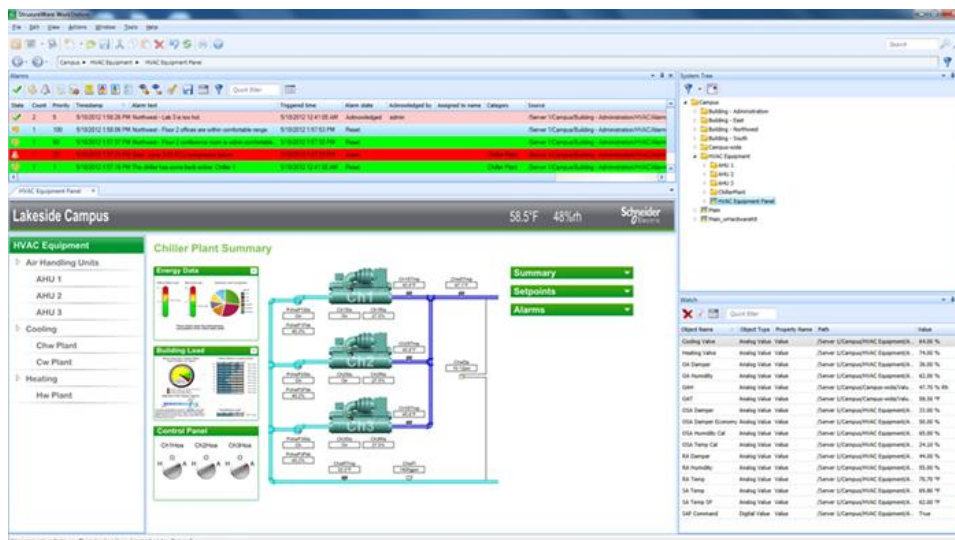
*workstation* disampaikan sebagai bagian dari instalasi atau paket bahasa yang terpisah.

*Workstation* memiliki fleksibilitas untuk sepenuhnya disesuaikan dengan preferensi pengguna individu. Antarmuka utama, disebut ruang kerja, adalah antarmuka berbasis jendela halaman dimana pengguna dapat memilih, posisi, dan mengatur ukuran berbagai komponen, seperti alarm, grafis, dan *editor*. *Workspace* ditugaskan untuk setiap akun pengguna, namun pengguna dapat dengan mudah mengubah, menyimpan, dan membuat beberapa versi serangkaian program. Ruang kerja juga dapat diubah yang diperlukan untuk mengawasi segala aktifitas.

Saat ini pengguna menginginkan lebih dari dasar kontrol. Mereka memerlukan aplikasi yang dapat disesuaikan untuk memenuhi Otomasi Gedung khusus sesuai dengan kebutuhan. Tidak seperti perangkat lunak lain yang memerlukan keterlibatan pabrik untuk non-standar atau aplikasi khusus, perangkat lunak *Workstation* dapat dengan mudah disesuaikan. Anda dapat mengubah urutan operasional menjadi kenyataan di lapangan untuk menghemat waktu dan uang pada setiap proyek.

Didalam perangkat lunak ini terdapat fitur *Editor* Grafis. Grafis yang dibuat dan disunting menggunakan *Editor* grafis: alat yang ampuh yang membantu pengguna memvisualisasikan segala sesuatu dari tingkat kontrol lapangan untuk tingkat perusahaan. *Editor* grafis menyediakan berbagai kemudahan menggunakan alat untuk membangun grafis apa pun yang diperlukan, dari gambar garis sederhana untuk foto gambar realistis. *Graphics Editor* dapat mengimpor berbagai format, termasuk .jpg dan gambar CAD. *Java Script* juga dapat

digunakan untuk lebih lanjut menyesuaikan grafis masing-masing. Tampilan *workspace* pada *workstation* dapat dilihat melalui gambar dibawah ini.



**Gambar 2.26** Tampilan Ruang Kerja Pada Perangkat Lunak *Workstation*

Animasi dapat menyiorot perubahan dalam sistem atau memudahkan navigasi. Misalnya, Anda dapat membuat pemandangan denah dengan suhu warna kode untuk setiap zona. *Editor* grafis diakses dari *WorkStation* dan memungkinkan pengguna untuk membuat dan mengedit grafis dalam *system* serta dapat disimulasikan sebelum diaplikasikan pada masing – masing gambar yang telah diatur dan dikonfigurasi.

#### 2.4.4 *SoMachine Basic*

Perangkat lunak pemrograman *SoMachine* digunakan untuk pemrograman mesin *Logic controller M221* dan telah berkembang selama beberapa tahun. *M221 Logic Controller* adalah unit baru, biaya rendah bertujuan untuk aplikasi mesin sederhana, ini dapat sangat mudah digunakan sebagai pengganti versi yang lama. *SoMachine* adalah solusi pemrograman yang mahal untuk ini sehingga *SoMachine Basic* diproduksi sebagai bentuk gratis versi *SoMachine*. Itu tidak

memiliki fungsi dari perangkat lunak *SoMachine* penuh tetapi perangkat lunak bebas masalah ini dapat digunakan untuk M221 controller. Tampilan beranda pada *Software SoMachine Basic* dapat dilihat pada gambar dibawah ini.



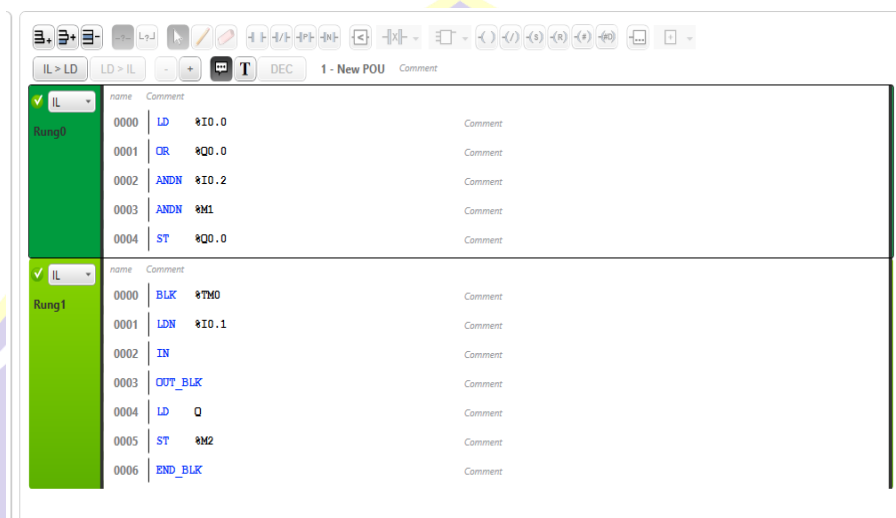
**Gambar 2.27** Tampilan Beranda Perangkat Lunak *Somachine Basic*.

*Somachine Basic* dapat bekerja pada *Operating system* apapun dan memiliki antarmuka yang begitu mudah digunakan untuk pemula dikalangan insinyur muda. Dengan fitur – fitur yang ada pada piranti lunak ini, pengontrolan sangat mudah dilakukan dan memiliki simulator sebelum diunggah pada PLC M221 sehingga tidak terjadi *trouble* apabila mengalami kesalahan dan revisi program yang terlalu banyak.

Dalam piranti lunak *Somachine Basic* terdapat fitur – fitur seperti diagram blok seperti *Timer*, *Counter*, *High Speed Counter*, *Fast Counter*, *Schedule Block*, dan *Communication Block*. *Variable – variable* pada *Somachine* tidak jauh berbeda dengan piranti lunak *Twido Suite*. Pengguna dapat melakukan proteksi dengan dua opsi yaitu semi proteksi dan proteksi penuh.

#### 2.4.4.1 Bahasa Pemrograman Intruction List

Bahasa pemrograman *Instruction List* (IL) bersifat textual. Singkatan – singkatan khusus yang disebut *mnemonic* digunakan untuk mengidentifikasi perintah yang berbeda yang sedang dijalankan ataupun tidak. Tampilan bahasa pemrograman *Instruction List* (IL) dapat dilihat melalui gambar dibawah ini.



**Gambar 2.28** Bahasa pemrograman *Intruction List* pada *Somachine Basic*

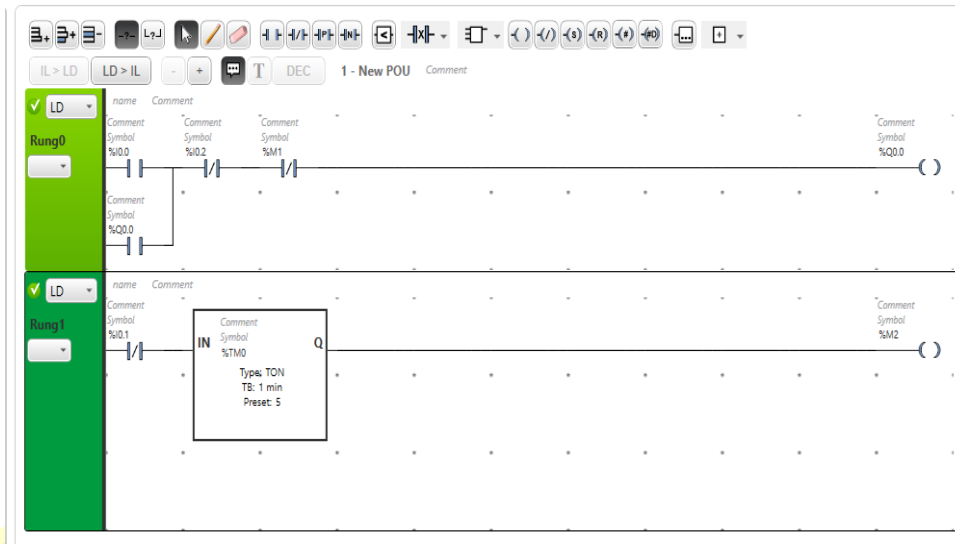
*Instruction list* memiliki persamaan dengan bahasa pemrograman dalam bentuk *script*. *Function block* pada *Somachine Basic* dapat dengan mudah diprogram karena fitur *function Block* sudah tertanam pada *mode Instruction List*.

#### 2.4.4.2 Bahasa Pemrograman *Ladder Diagram*

*Ladder Diagram* atau disebut diagram tangga adalah instruksi yang terkait dengan kondisi – kondisi seperti sebuah rangkaian kontrol listrik. Dapat dilakukan secara independen maupun dikombinasikan dengan diagram blok yang merupakan fitur dari pada *Somachine Basic*.

*Ladder Diagram* memiliki *rung* yaitu area kerja pemrograman pada satu *line* agar tidak terjadi kesulitan pada hal pemrograman. Tampilan bahasa

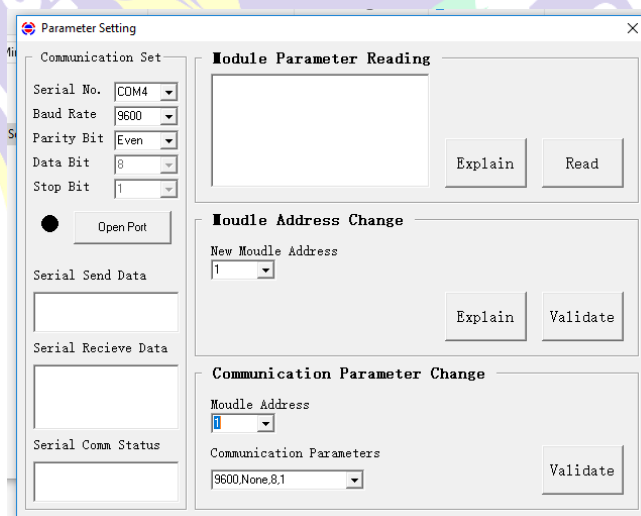
pemrograman *Ladder Diagram* pada *Somachine Basic* dapat dilihat pada gambar dibawah ini.



**Gambar 2.29** Bahasa pemrograman *Ladder Diagram* pada *Somachine Basic*.

#### 2.4.5 Wellpro Debugging Software

*Wellpro Debugging Software* merupakan piranti lunak yang dikeluarkan oleh provider wellpro modul *input/output*. Tampilan halaman pemindai pada *Wellpro Debugging* dapat dilihat pada gambar dibawah ini.

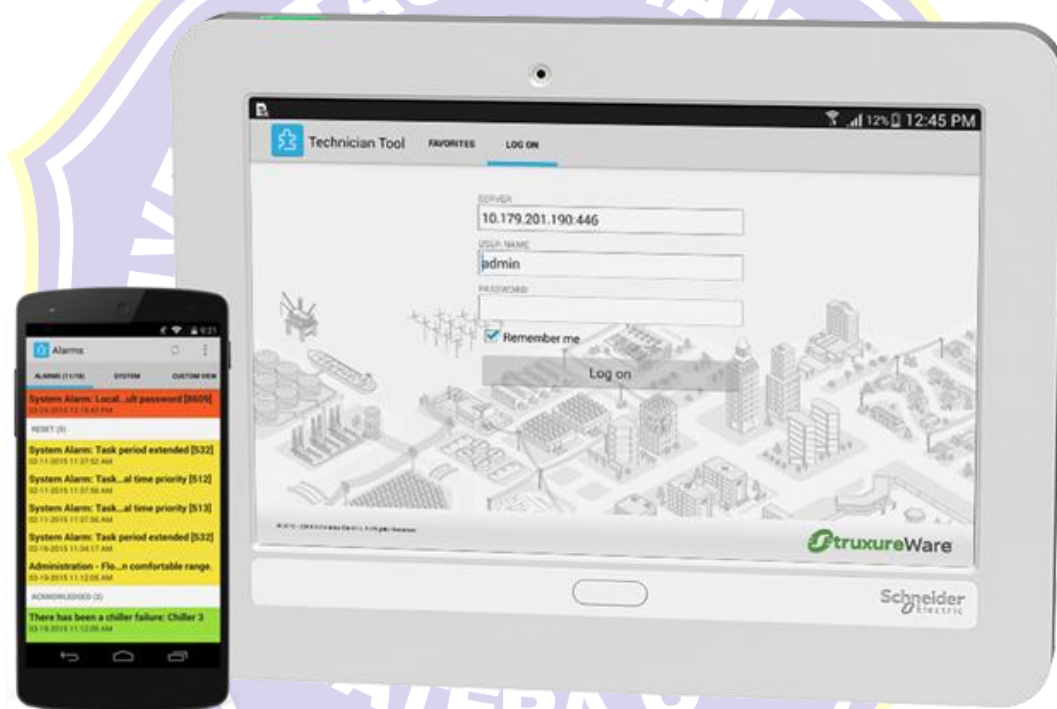


**Gambar 2.30** Tampilan Halaman Pemindai Pada *Wellpro Debugging*

Piranti lunak tersebut dapat mengolah data *input / output* pada modul melalui konfigurasi port terbuka dimana *protocol MODBUS RTU* didapatkan pertama dengan metode pemindai.

#### 2.4.6 Struxure Ware Building Operation Tech Tool

*Tech Tool* adalah aplikasi *universal* yang berjalan pada perangkat Android versi OS (4.2 dan di atasnya). Ilustrasi piranti lunak *Tech Tool* pada *Smart Device* dapat dilihat pada gambar dibawah ini.

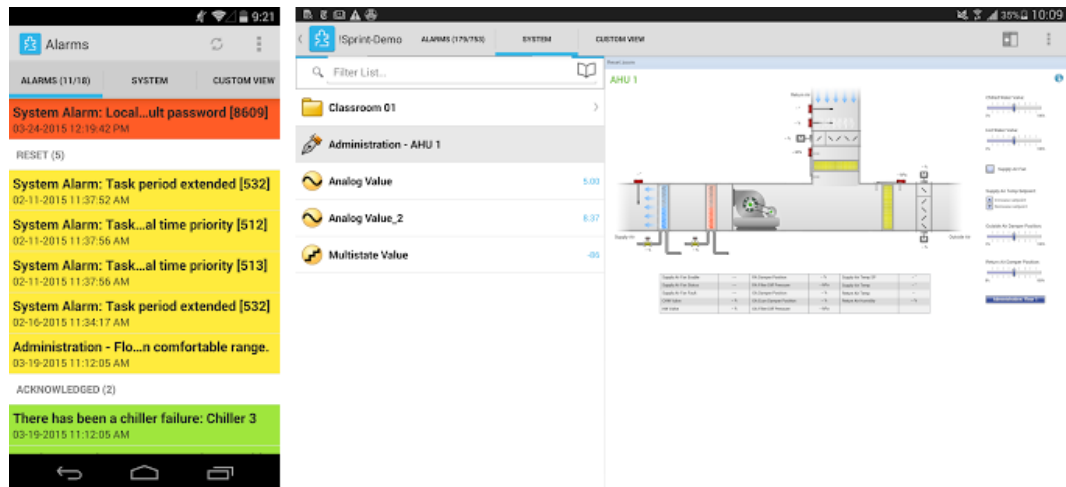


**Gambar 2.31** Ilustrasi Piranti Lunak *Tech Tool* Pada *Smart Device*

Dengan *Tech Tool* untuk Android, *Schneider Electric* menawarkan layanan antarmuka dengan cara yang nyaman untuk mengakses kontrol bangunan menggunakan WiFi secara lokal atau WiFi atau 3G melalui Internet. Menelusuri sistem untuk mendapatkan data secara berlanjut atau sejarah tren, mengubah *setpoints*, memaksa nilai - nilai *input* dan *output*, melihat dan mengatur alarm dan



mengedit jadwal waktu. Tampilan *dashboard* pada *SBO Tech Tool* dapat dilihat pada gambar dibawah ini.



**Gambar 2.32** Tampilan *Dashboard* Pada *SBO Tech Tool*

Adapun fungsi piranti *Tech Tool* sebagai berikut:

1. Akses sistem manajemen bangunan tanpa harus membawa laptop di sekitar.
2. Pemecahan masalah dan perbaikan dengan mudah.
3. Mendapatkan informasi tentang kesalahan melalui ponsel atau tablet dimanapun berada.
4. Ruang akses alamat *Server* umum digunakan dan kredensial dengan nama yang mudah dalam daftar favorit.
5. Struktur daftar favorit Anda menggunakan folder.
6. *Username* dan *Password* yang sama ketika mengakses menggunakan aplikasi *browser* yang lain.
7. *Browse* sistem dengan sistem yang terstruktur.
8. Mendapatkan akses mudah ke nilai-nilai umum digunakan melalui tab tampilan kustom.
9. Memantau aktifitas secara lanjutan, dan diperbarui secara dinamis.



10. Mengubah nilai *analog*, *digital*, dan *multistate*.
11. Mengatur nilai-nilai yang tidak dikendalikan oleh *input*, program atau *binding*.

#### 2.4.7 Struxureware Building Operation Device Administrator

*Device Administrator* merupakan piranti lunak yang digunakan untuk melakukan konfigurasi pada perangkat keras *Automation Server* untuk pertama kalinya sebelum dapat diprogram oleh piranti lunak *Workstation*. Tampilan awal *SBO Device Administrator* dapat dilihat pada gambar dibawah ini.



**Gambar 2.33** Tampilan Awal SBO Device Administrator

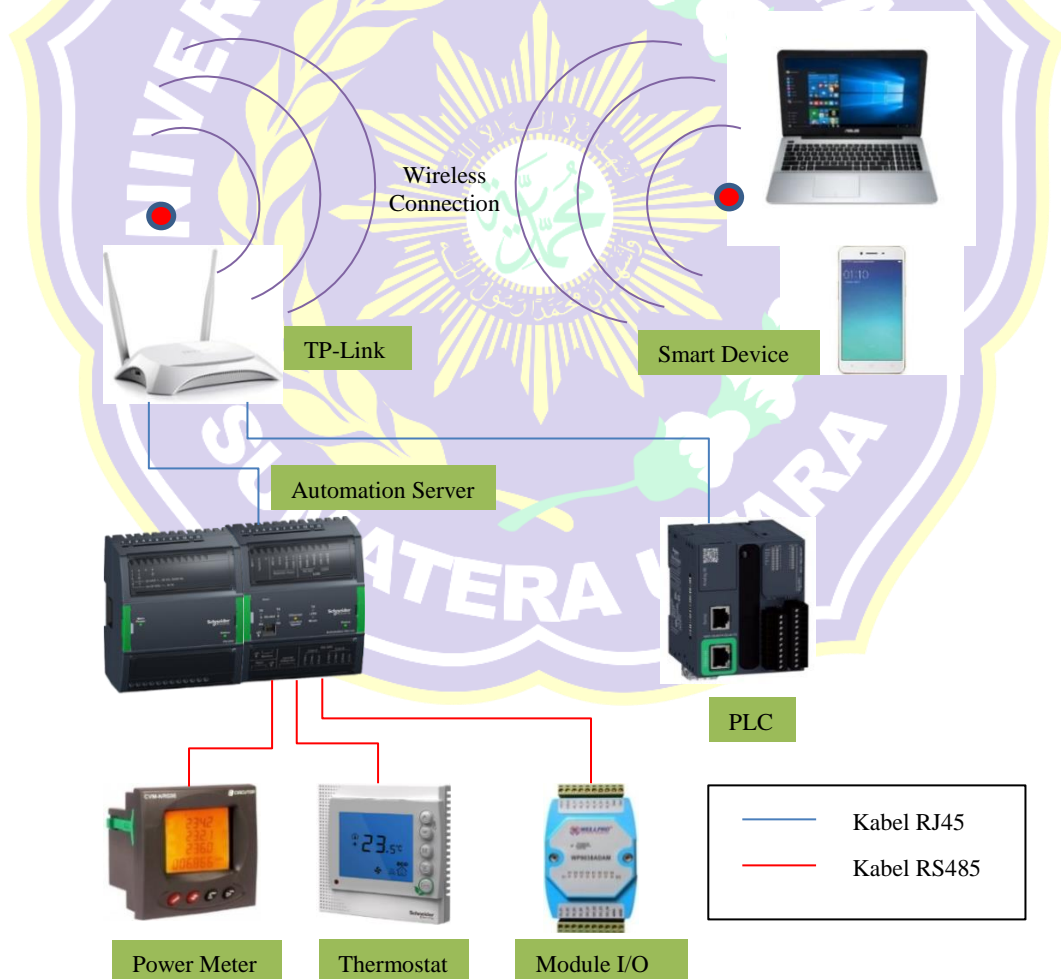
*Device Administrator* berfungsi untuk memberi informasi spesifikasi dan melakukan konfigurasi *database*, jaringan (*network*) dan *system* pendukung lainnya.

# BAB III

## PERANCANGAN SISTEM

### 3.1 Pendahuluan

Sistem *Monitoring* dan *Controlling* berfungsi sebagai pengawasan dan pengendalian menggunakan *personal computer* ataupun perangkat pintar. Pada tugas akhir ini *Building Management System* menggunakan *server* dari *Schneider Electric*, yaitu *Automation Server*. *Automation Server* akan mengawasi dan mengontrol beberapa perangkat *slave/client* seperti *PLC*, *Modul I/O*, *Thermostat*, dan *Power Meter*.



**Gambar 3.1** Blok Diagram Sistem

Dari gambar diatas ditunjukkan bahwa terdapat perangkat pintar (*smart device*) yaitu laptop dan ponsel pintar yang terhubung secara nirkabel dengan *TP-Link*. Kemudian *TP-Link* terkoneksi dengan *Automation Server* dan PLC menggunakan kabel RJ45. *Automation Server* terhubung dengan beberapa perangkat *slave/client* yang akan diawasi dan dikontrol. Laptop ataupun ponsel pintar dapat mengendalikan dan mengawasi semua perangkat yang terhubung ke *Automation Server*. Pengawatan keseluruhan dapat dilihat pada [lampiran 2](#).

### 3.2 Perancangan Sistem

Perancangan sistem dilakukan melalui dua tahap, yaitu perancangan perangkat keras (*hardware*) dan perancangan perangkat lunak (*software*).

#### 3.2.1 Perancangan *Hardware*

Pada *system* ini *hardware* terdiri dari *Smart Device* (Laptop dan Ponsel Pintar), *TP-Link*, PLC, *Module I/O*, *Thermostat*, *Power Meter*, dan *Automation Server*. Untuk melakukan koneksi keseluruhan perangkat *hardware*, maka konfigurasi IP Address dan *MODBUS Parameter* disesuaikan sebagai berikut.

Tabel 3.1 Daftar Alamat IP untuk perangkat yang didukung *MODBUS TCP/IP*

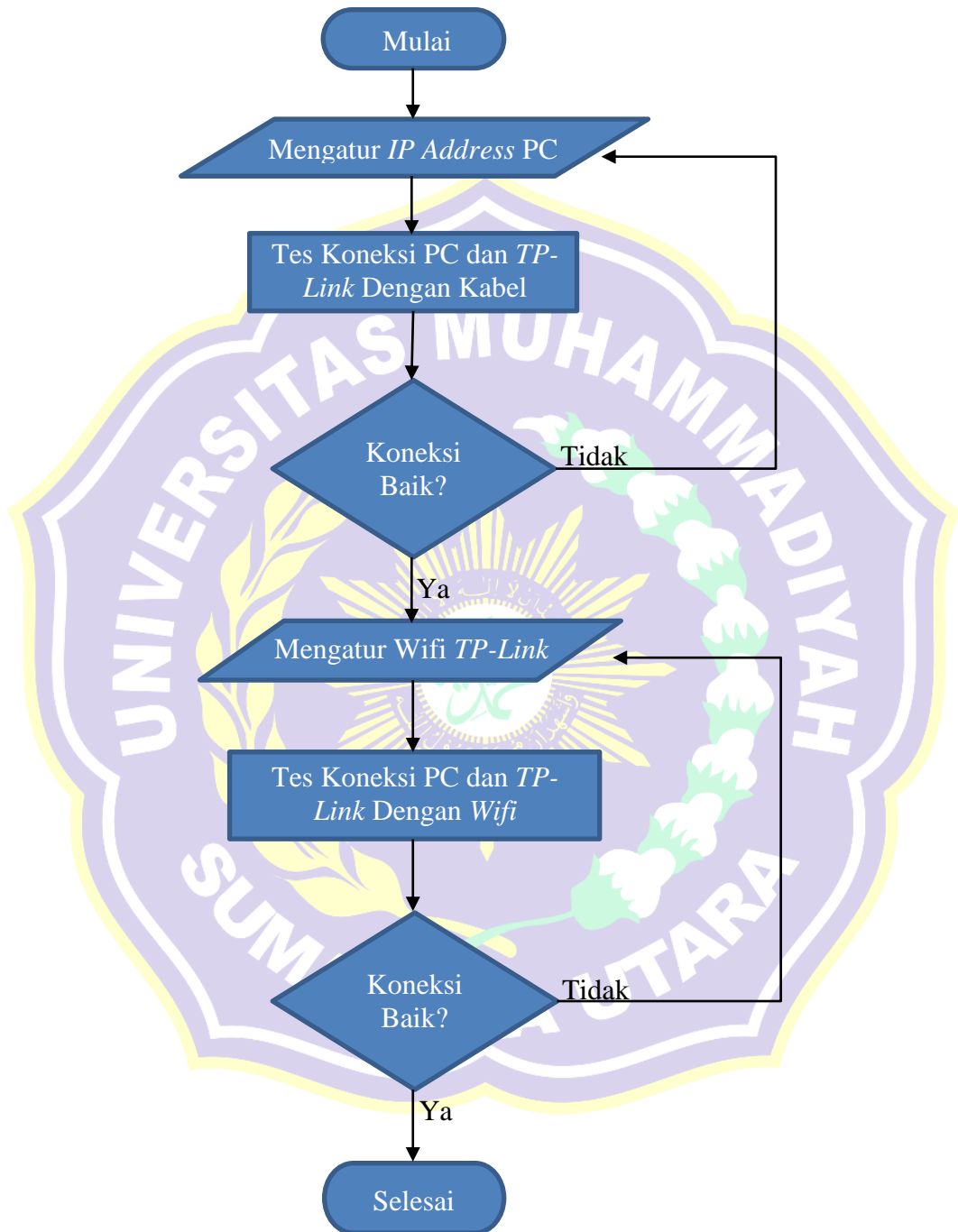
No	Perangkat	IP Address	Subnet Mask	Keterangan
1	<i>TP - Link</i>	192.168.0.1	255.255.255.0	Default
2	<i>Personal Computer</i>	192.168.0.2	255.255.255.0	Dikonfigurasi
3	PLC M221	192.168.0.3	255.255.255.0	Dikonfigurasi
4	<i>Automation Server</i>	192.168.0.4	255.255.255.0	Dikonfigurasi

Tabel 3.2 Daftar parameter untuk perangkat yang didukung *MODBUS RTU*

No	Perangkat	Baudrate	Parity	Data bit	Stop Bit	Slave	Port
1	<i>Wellpro Module I/O</i>	9600	Even	8	1	1	B
2	<i>Thermostat</i>	4800	Odd	8	1	2	A
3	<i>Power Meter</i>	4800	Odd	8	1	3	A

### 3.2.1.1 Diagram Alir Konfigurasi Pengaturan PC dan *TP-Link*

Diagram alir pengaturan PC dan *TP-Link* dapat dilihat pada gambar dibawah ini.



**Gambar 3.2** Diagram Alir Pengaturan PC dan *TP-Link*

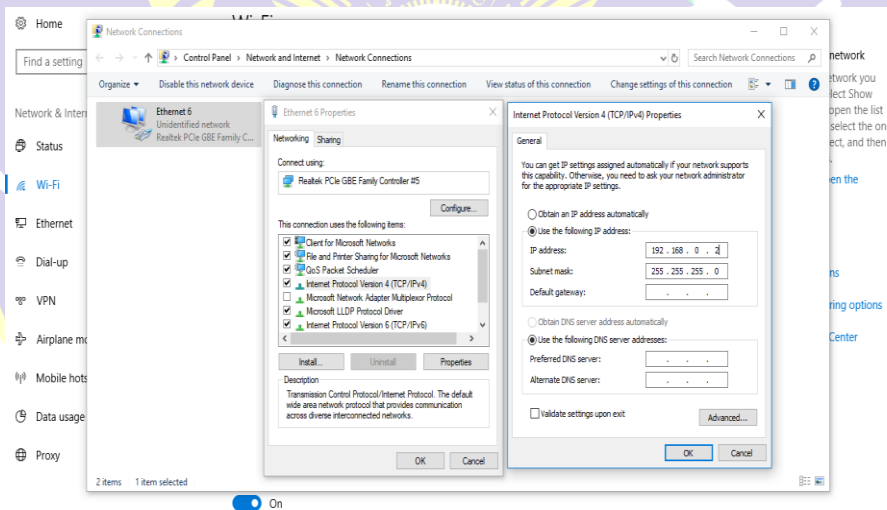
Melalui diagram alir diatas, berikut penjelasan langkah – langkah melakukan konfigurasi pada PC dan *TP-Link*.

1. Memulai dengan mempersiapkan PC dan *TP-Link* kemudian menghubungkannya dengan kabel RJ45 terlebih dahulu.
2. Membuat pengaturan *IP Address* Komputer dan *TP-Link* dalam satu segmen yang sama. *IP Address Default* pada *TP-Link* adalah 192.168.0.1 maka konfigurasi *IP Address* PC menjadi 192.168.0.2

Caranya sebagai berikut:

*Control Panel* → *Network and Internet* → *Network Connection* → *Ethernet*  
→ *Properties* → *Internet Protocol Version 4* → *Use the following IP address*.

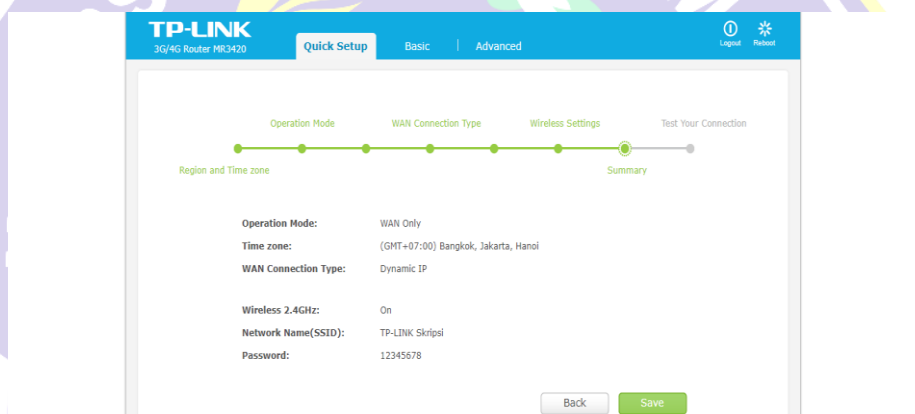
Pada kolom *IP Address* isi 192.168.0.2 dan pada kolom *Subnet mask* isi 255.255.255.0 → *OK*



**Gambar 3.3** Pengaturan *IP Address* pada PC/Laptop

3. Memeriksa koneksi antara PC dan *TP-Link* melalui tes *ping IP Address TP-Link* menggunakan *Command Prompt*.

4. Apabila koneksi baik, lanjutkan ke pengaturan *Wifi TP-Link*. Apabila gagal, maka mengatur kembali *IP Address PC*.
5. Mengatur *Wifi TP-LINK* pertama kali dengan kabel LAN sebagai berikut:
  - 1) Masuk ke *browser PC* ketik *IP Address TP-Link* 192.168.0.1 → Masukan *username* dan *password* (*username : admin , password : admin*) → *Login*
  - 2) Pada laman *Region and Time Zone* pilih *Jakarta* → *Next*  
 Pada laman *Operation Mode* pilih *WAN Only* → *Next*
  - 3) Pada laman *WAN Connection Type* pilih *Dynamic IP* → *Next*
  - 4) Pada laman *Wireless Settings*, kolom *Network Name (SSID)* ketik *TP-Link Skripsi*, kolom *Password* ketik 12345678 → *Next* → *Save* → *Finish*



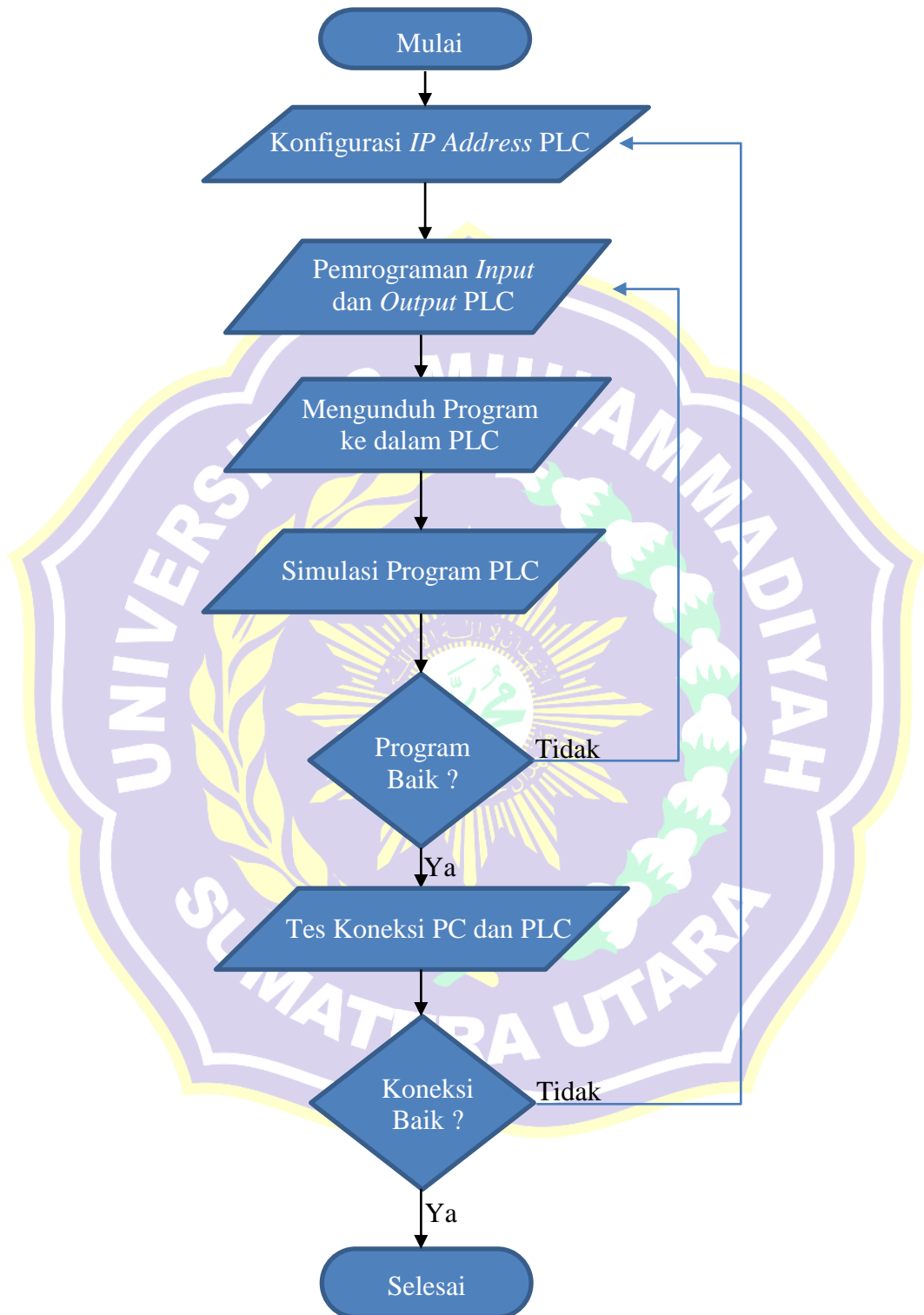
**Gambar 3.4** Pengaturan *TP-Link* sebagai pemancar *Wifi*

6. Menghubungkan *PC* ke *TP-Link* via *wifi*. Memeriksa *PC* dan *TP-Link* melalui tes *ping IP Address TP-Link* menggunakan *Command Prompt*.
7. Apabila terhubung dan *ping IP Address TP-Link* berhasil, maka konfigurasi selesai. Apabila gagal, maka melakukan pengaturan *Wifi TP-Link* kembali. Selesai



### 3.2.1.2 Diagram Alir Konfigurasi PLC M221

Diagram alir konfigurasi PLC M221 dapat dilihat pada gambar dibawah ini.

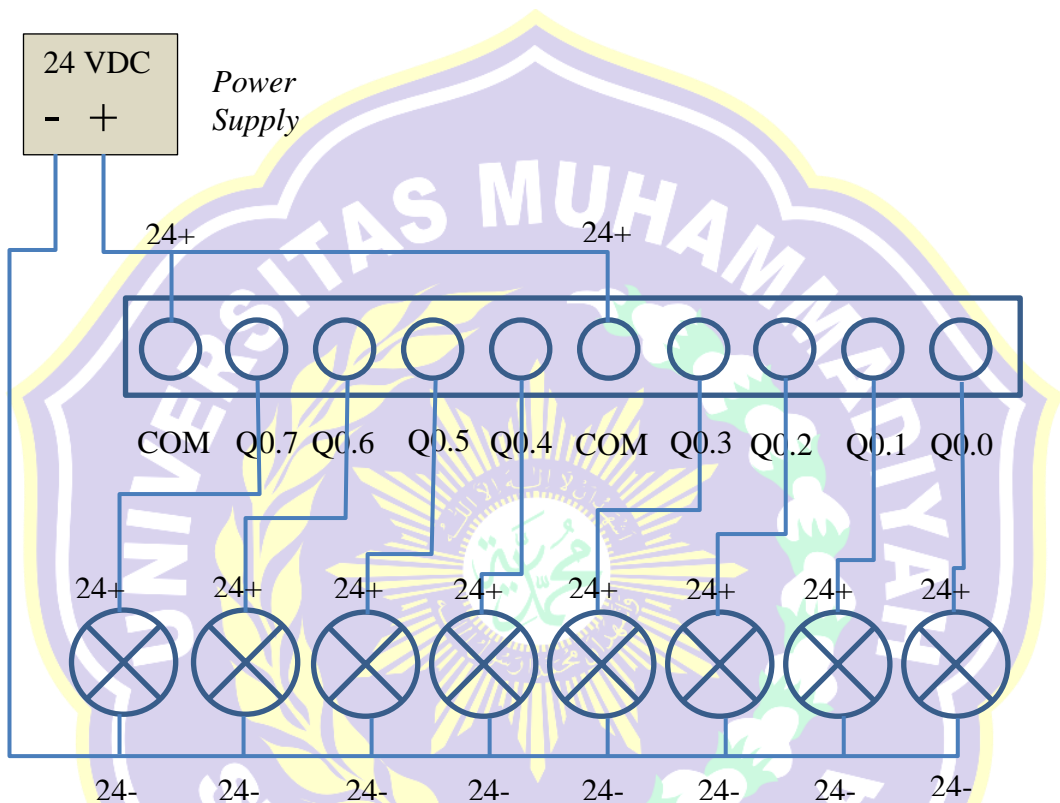


**Gambar 3.5** Diagram Alir Konfigurasi PLC M221



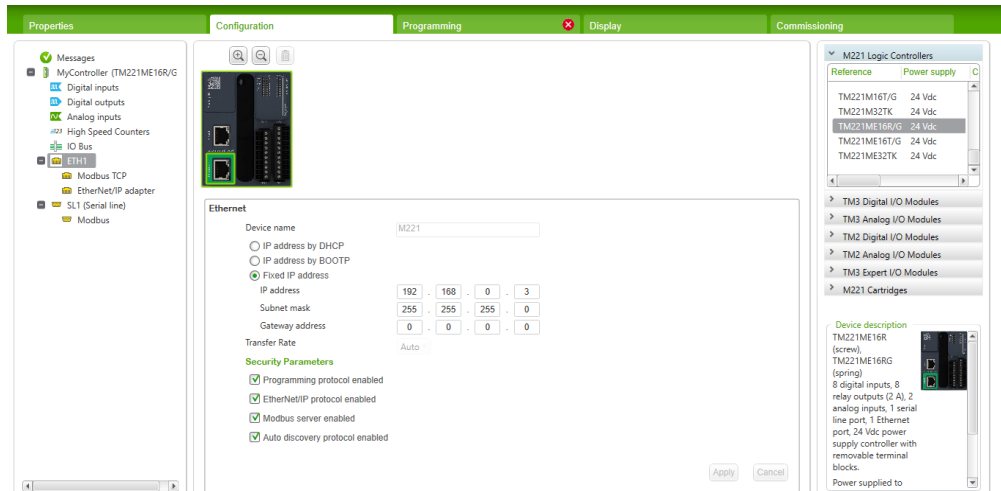
Melalui gambar diagram alir diatas, langkah – langkah konfigurasi dan pemrograman PLC adalah sebagai berikut.

1. Memulai dengan mempersiapkan PC dan PLC, membuka piranti lunak *SoMachine Basic*. Kemudian melakukan pengawatan seperti pada gambar dibawah ini.



**Gambar 3.6** Diagram Pengawatan Relay *Output* PLC M221

2. Pada laman *configuration tab m221 logic controller*, pilih PLC yang akan digunakan TM221ME16R. Dibagian menu bar pilih ETH1 untuk melakukan konfigurasi *IP Address* PLC. Centang *Fix IP Address* kemudian isi *IP address* 192.168.0.3 Subnet Mask 255.255.255.0 klik Apply.



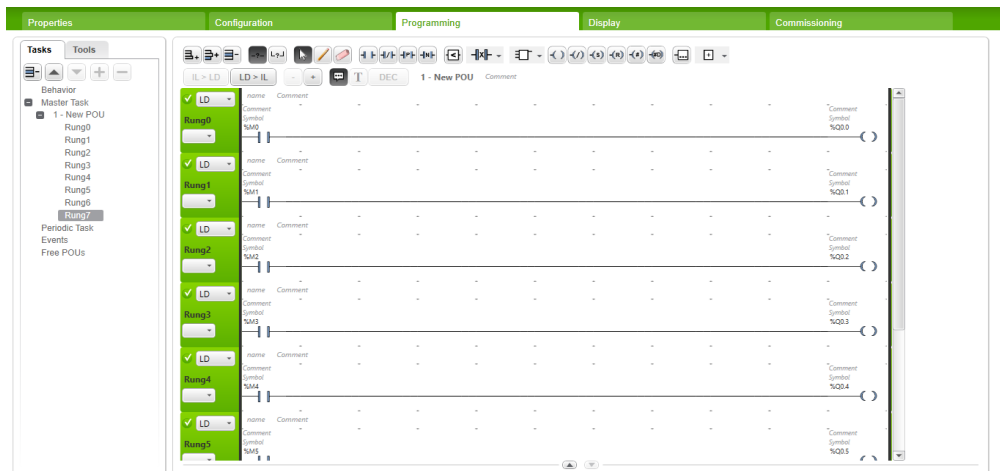
**Gambar 3.7** Pengaturan IP Address PLC di *Somachine Basic*

3. Pada laman *Programming* melakukan program untuk semua *output* pada PLC. Konfigurasi *input/output* pada pemrograman PLC M221 dapat dilihat pada tabel dibawah ini dan *Modbus register PLC Modicon M221* dapat dilihat melalui [lampiran 6](#).

**Tabel 3.3** Konfigurasi *Input/Output* Pada Pemrograman PLC M221

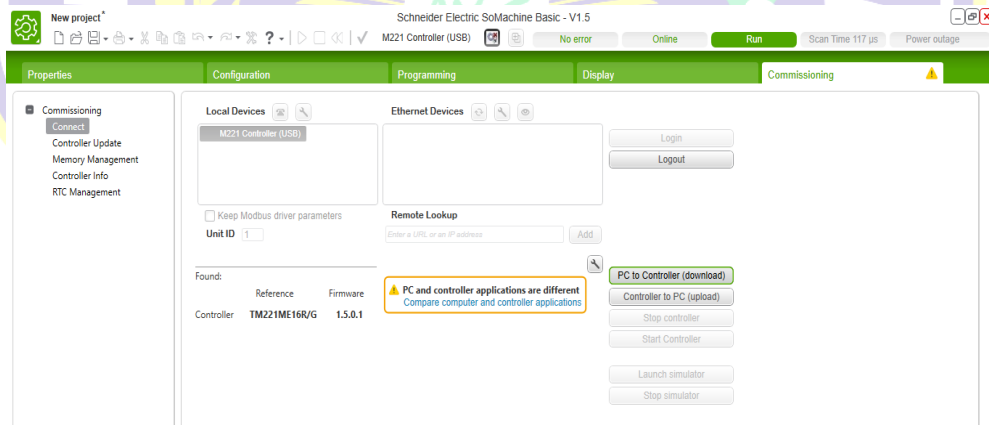
NO	<i>Input</i> dari PLC	<i>MODBUS</i> Address Untuk Automation Server	<i>Read/Write</i> Function Code	<i>Output</i> PLC
1	M0	0	02/15	Q0.0
2	M1	1	02/15	Q0.1
3	M2	2	02/15	Q0.2
4	M3	3	02/15	Q0.3
5	M4	4	02/15	Q0.4
6	M5	5	02/15	Q0.5
7	M6	6	02/15	Q0.6
8	M7	7	02/15	Q0.7

Program pada PLC dapat dilihat pada gambar dibawah ini.



**Gambar 3.8** Program PLC M221

4. Menghubungkan PC dengan PLC menggunakan kabel mini USB. Pada laman *Commisioning* pilih *M221 Controller (USB)* kemudian Login dan klik *PC to Controller* untuk mengunduh program dan konfigurasi ke PLC. Proses mengunduh program kedalam PLC dapat dilihat pada gambar dibawah ini.



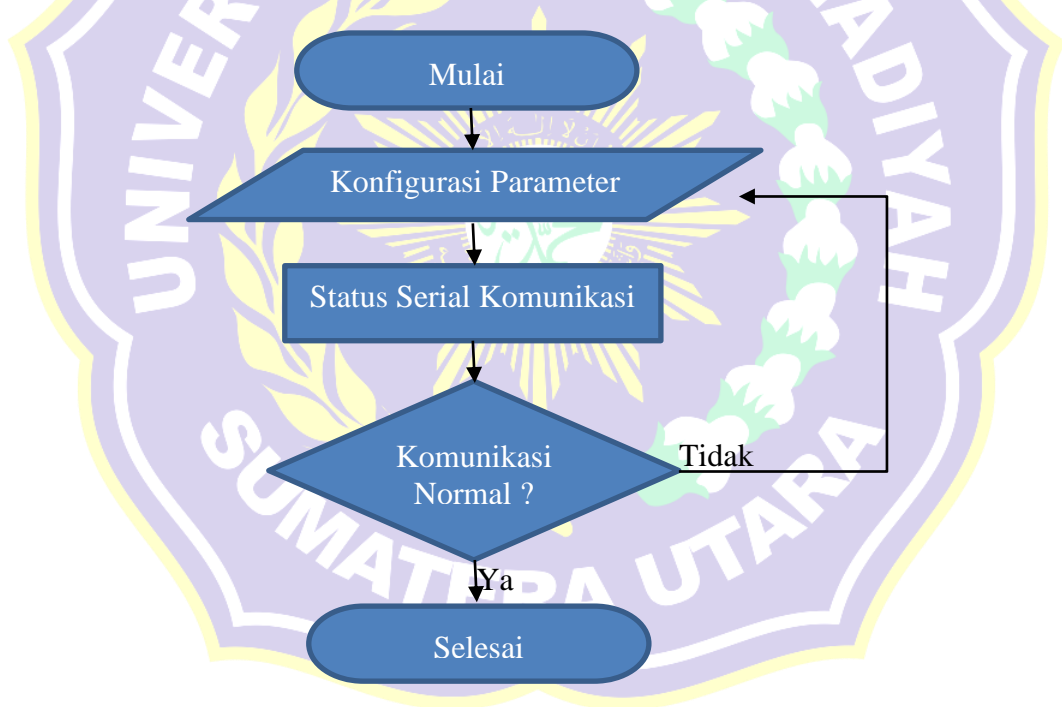
**Gambar 3.9** Proses mengunduh program kedalam PLC

5. Setelah program masuk kedalam PLC klik *Start Controller* dan mencoba (simulasi) masing – masing *output*.

6. Apabila program sesuai dan berjalan dengan baik, maka melanjutkan dengan tes koneksi PLC ke TP-Link. Apabila program gagal, kembali melakukan pemrograman ulang.
7. Menghubungkan PLC ke *TP-Link* menggunakan kabel LAN. Tes *ping IP-Address PLC* melalui PC yang terhubung *Online* via *Wifi TP-Link*.
8. Apabila berhasil, maka konfigurasi berhasil. Apabila gagal, kembali melakukan pengaturan konfigurasi PLC. Selesai.

### 3.2.1.3 Diagram Alir Konfigurasi *Wellpro Module I/O*

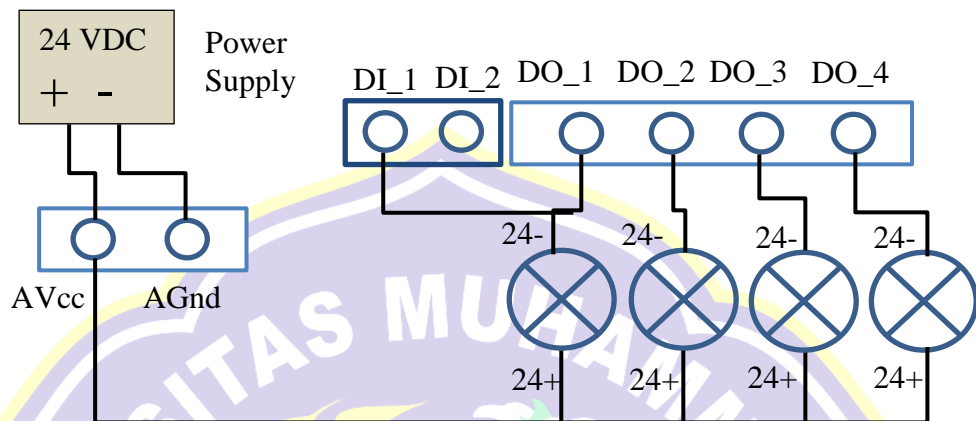
Diagram alir konfigurasi *Wellpro Module I/O* dapat dilihat pada gambar dibawah ini.



**Gambar 3.10** Diagram Alir Konfigurasi *Wellpro Module I/O*.

Melalui gambar diagram alir diatas, berikut konfigurasi pada *Wellpro Module I/O*.

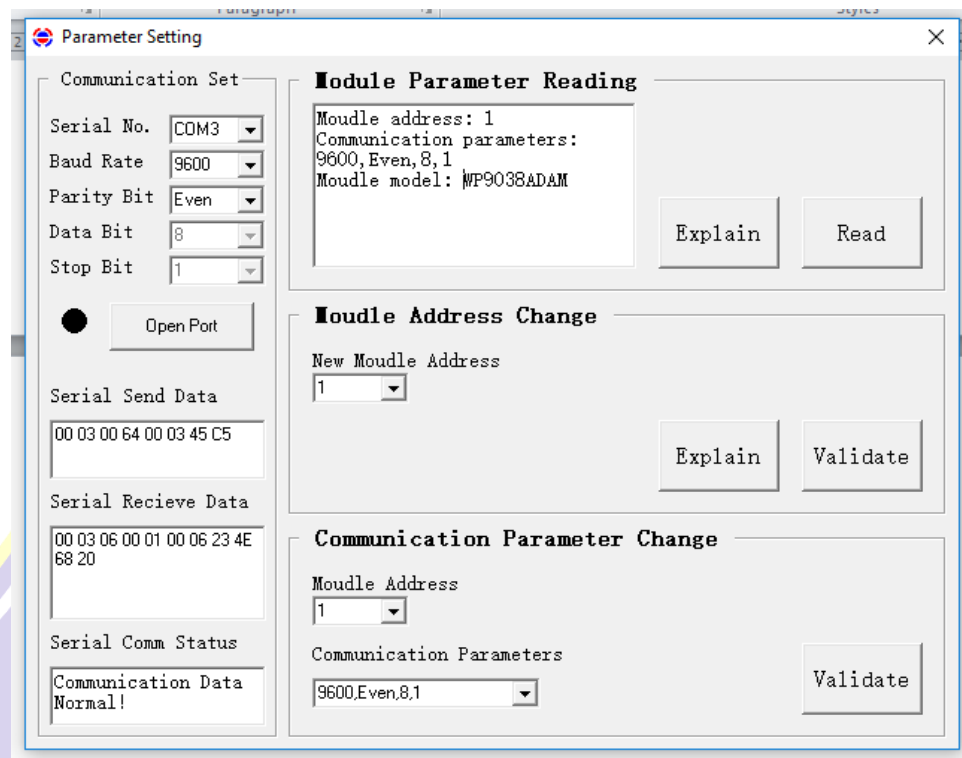
1. Memulai Menghubungkan Port 485A dan 485B yang ada pada *Wellpro Module I/O* ke *Personal Computer* menggunakan kabel U-485. Kemudian melakukan pengawatan seperti pada gambar dibawah ini.



**Gambar 3.11** Diagram Pengawatan Keluaran *Wellpro Module I/O*

2. Melakukan konfigurasi parameter pada *Wellpro* sebagai berikut.
  - 1) Buka piranti lunak *Wellpro Debugging*, kemudian pada kolom *Model Select*, pilih tipe *module I/O* yang digunakan yaitu WP9038ADAM.
  - 2) Pilih *Parameter Setting*, pada kolom *serial number* sesuaikan *port communication* yang terdaftar di *device manager*. Kemudian klik *open port*.
  - 3) Pindai (*scan/read*) perangkat *module I/O* dengan perintah *Read*, dalam mode *default* parameter modul I/O sebagai berikut. *Baudrate* : 9600, *Parity* : None, *Data bit* : 8, *Stop bit* : 1, *Slave Address* : 1.
  - 4) Mengubah *parameter* modul I/O. Pada tab *Communication Parameter Setting*, pilih parameter sebagai berikut. *Baudrate* : 9600, *Parity* : Even, *Data bit* : 8, *Stop bit* : 1 *Slave Address* : 1

Tampilan konfigurasi parameter *Wellpro Module I/O* dapat dilihat pada gambar dibawah ini.



**Gambar 3.12** Tampilan konfigurasi parameter *Wellpro Module I/O*

3. Kemudian *Validate* parameter yang telah diubah dan memindai (*scan/read*).
4. Apabila komunikasi normal dan parameter berhasil diubah, maka konfigurasi berhasil. Apabila gagal, melakukan kembali konfigurasi parameter. Selesai

#### 3.2.1.4 Konfigurasi *Thermostat TC300*

*Thermostat TC – 300* berfungsi sebagai pengatur suhu ruangan dan membaca nilai *actual* suhu pada *thermostat* berada, di tugas akhir ini *Thermostat TC-300* hanya sebagai simulasi. Parameter *MODBUS* pada *Thermostat TC-300* tidak dapat diubah karena bersifat terkunci (*lock*) dari pabrikan dan pengguna hanya bisa mengatur alamat perangkat *thermostat* dan operasional kerja *thermostat*.

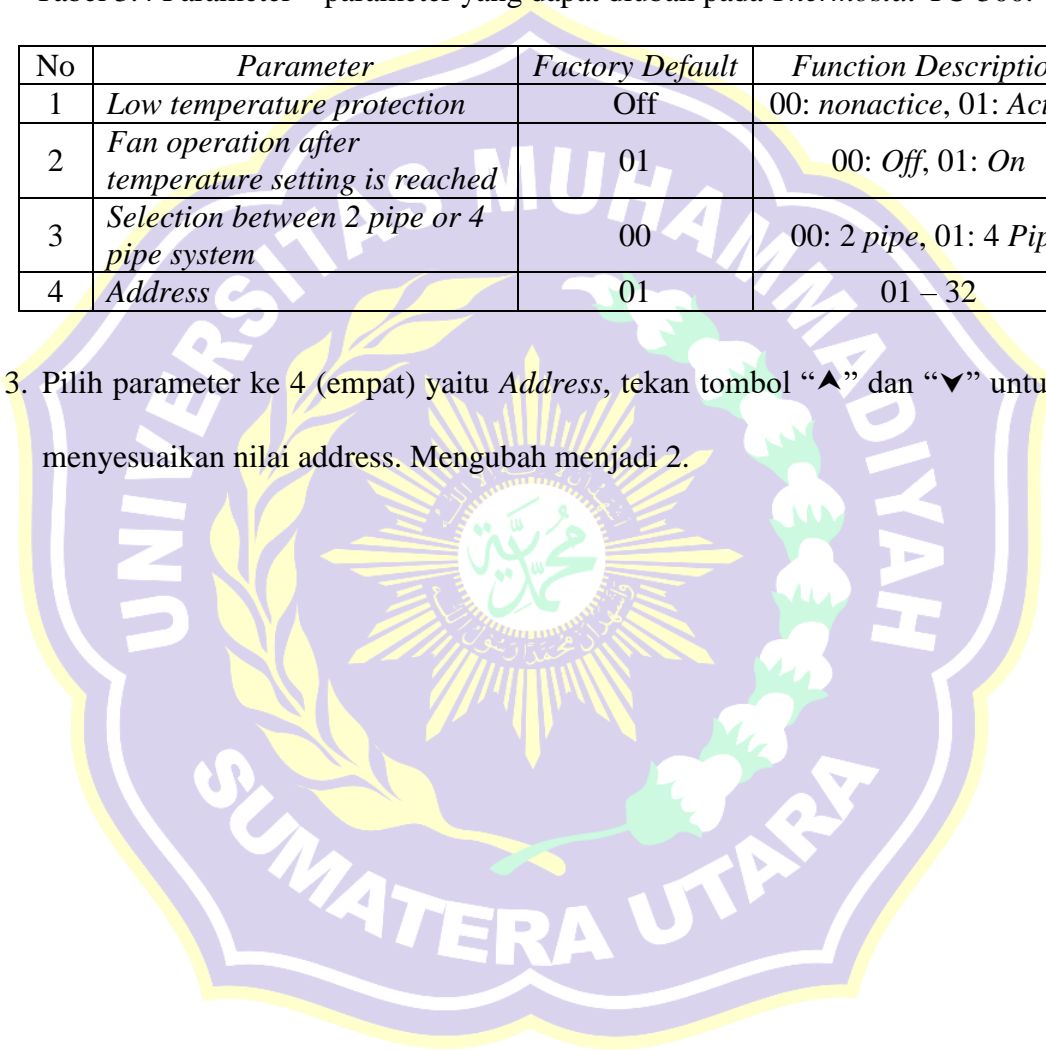
Konfigurasi pengalaman *thermostat* sebagai berikut:

1. Selama daya mati, tekan dan tahan tombol M untuk 5 detik untuk memasuki layar tampilan.
2. Tekan tombol M lagi untuk memilih parameter seperti yang ditunjukkan dalam tabel berikut.

Tabel 3.4 Parameter – parameter yang dapat diubah pada *Thermostat* TC-300.

No	<i>Parameter</i>	<i>Factory Default</i>	<i>Function Description</i>
1	<i>Low temperature protection</i>	Off	00: <i>nonactice</i> , 01: <i>Active</i>
2	<i>Fan operation after temperature setting is reached</i>	01	00: <i>Off</i> , 01: <i>On</i>
3	<i>Selection between 2 pipe or 4 pipe system</i>	00	00: <i>2 pipe</i> , 01: <i>4 Pipe</i>
4	<i>Address</i>	01	01 – 32

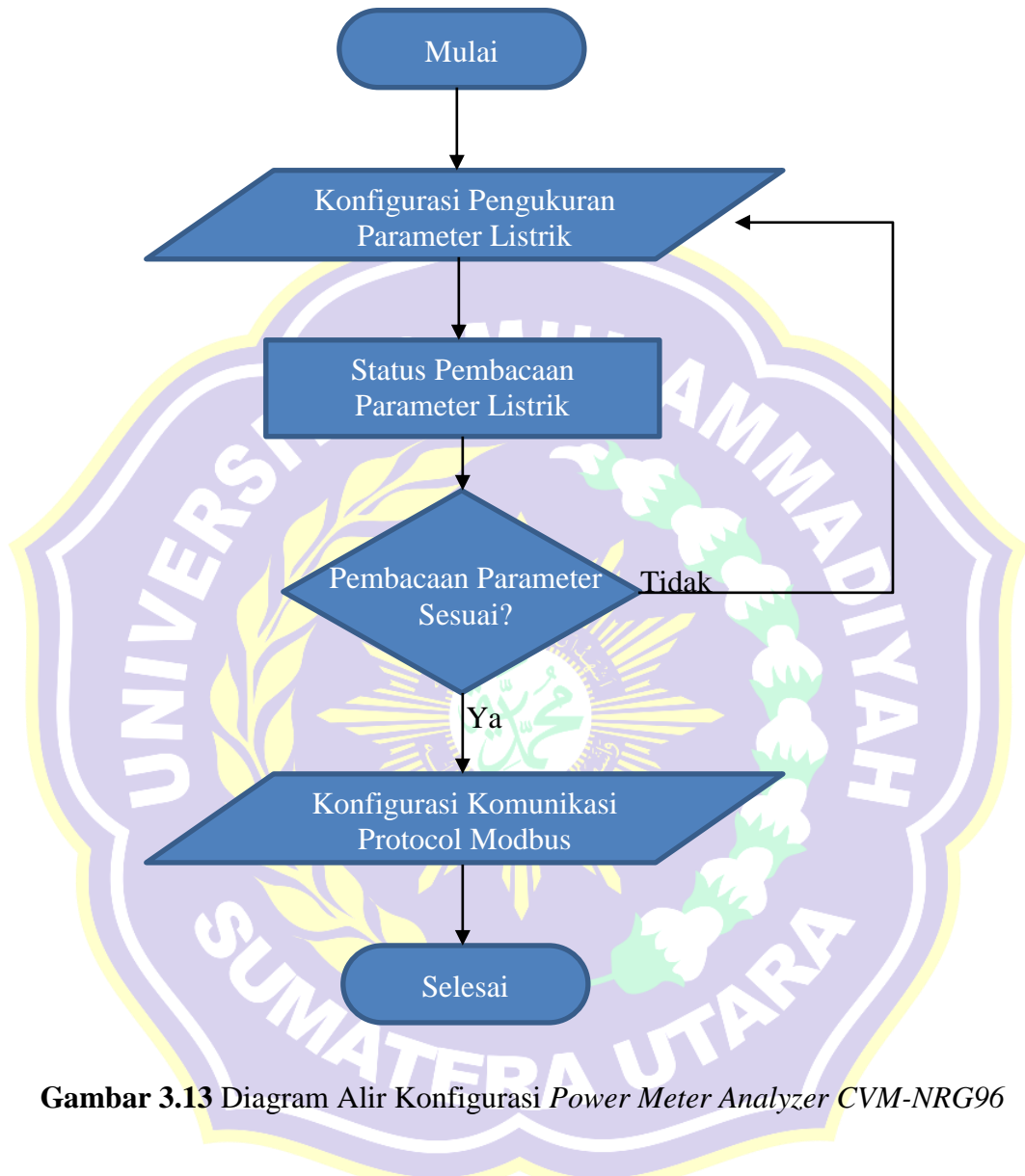
3. Pilih parameter ke 4 (empat) yaitu *Address*, tekan tombol “▲” dan “▼” untuk menyesuaikan nilai address. Mengubah menjadi 2.





### 3.2.1.5 Diagram Alir Konfigurasi *Power Meter Analyzer CVM-NRG96*

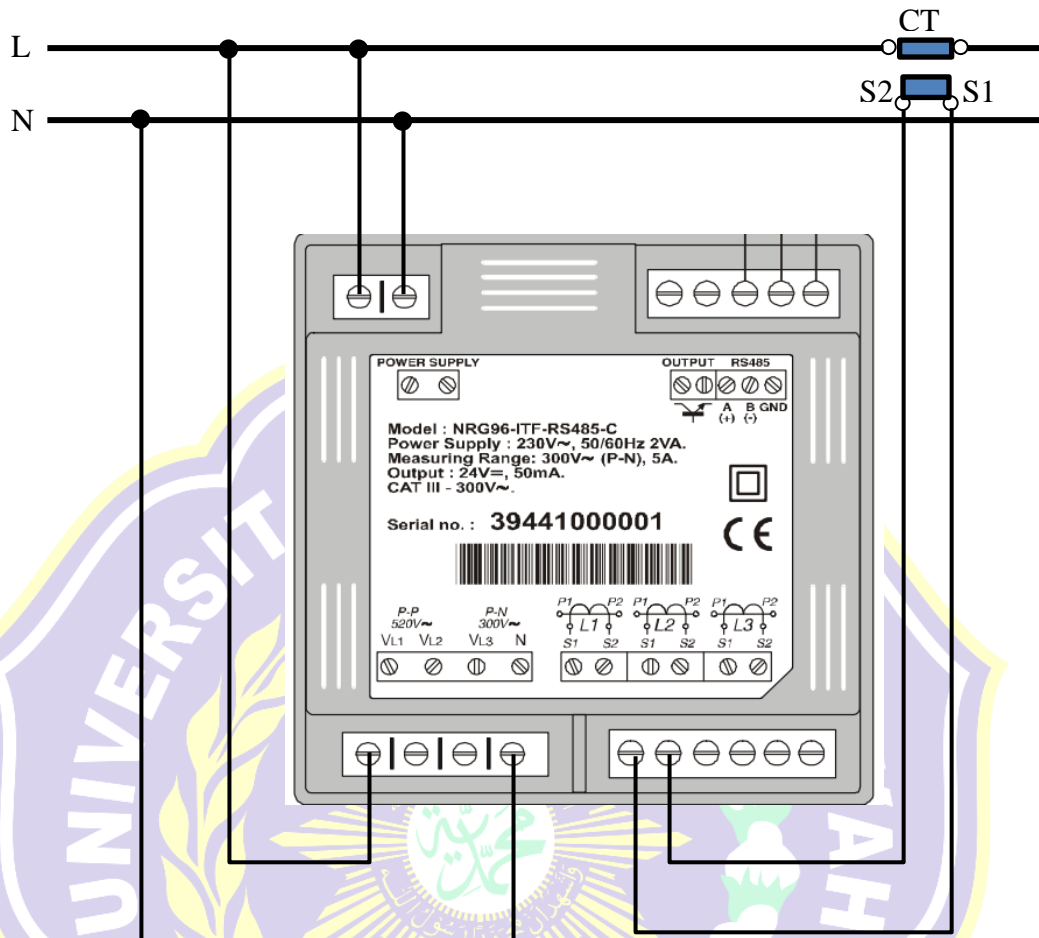
Diagram alir konfigurasi *Power Meter Analyzer CVM-NRG96* dapat dilihat pada gambar dibawah ini.





**Gambar 3.13** Diagram Alir Konfigurasi *Power Meter Analyzer CVM-NRG96*

Ditugas akhir ini *Power Analyzer* hanya sebagai simulasi untuk membaca beban 1 fasa menggunakan CT (*Current Transformer*). Melalui gambar diagram alir diatas, berikut langkah – langkah konfigurasi *Power Meter Analyzer CVM-NRG96*.

1. Memulai menyiapkan bahan (*Power Analyzer*, *CT 50/5A*, dan kabel).  
Kemudian melakukan pengawatan seperti gambar dibawah ini.



**Gambar 3.14** Diagram pengawatan *Power Analyzer*

2. Melakukan konfigurasi parameter listrik sebagai berikut.
  - 1) Untuk mengakses Halaman Pengaturan Pengukuran (*Measurement Setup*), tombol **MAX**  dan **MIN**  harus ditekan pada saat yang bersamaan sampai mode pengaturan dimasukkan.
  - 2) Saat memasuki mode pengaturan, pesan "*Setup Unlo*" atau "*Setup Loc*" ditampilkan selama beberapa detik yang menunjukkan bahwa kita sedang ingin mengatur dan meminta informasikan status perangkat (terkunci atau tidak terkunci). *Setup Unlo* yaitu parameter listrik dapat diubah. *Setup Loc*

yaitu parameter tidak dapat diubah. Untuk melanjutkan tampilan berikutnya tekan tombol *display* 




- 3) Pada tampilan pertama adalah kalibrasi tegangan yang digunakan *Simple Voltage* (U1 U2 U3) or *Compound Voltage* (U12 U23 U31), pilih menu *Simple Voltage* karena pada tugas akhir ini hanya menggunakan tegangan jala – jala 1 fasa. Tekan tombol *MAX* untuk menyesuaikan. Kemudian tekan tombol *display* untuk tampilan selanjutnya.
- 4) Pada tampilan rasio *transformator*, parameter pertama yang akan diatur adalah Trafo Tegangan Primer (*Transformer Voltage Primary*), sesuaikan menjadi “220”, gunakan tombol *Max* dan *Min* untuk menyesuaikan. Parameter kedua yang akan diatur adalah Trafo Tegangan Sekunder (*Transformer Voltage Secondary*), sesuaikan menjadi “220” gunakan tombol *Max* dan *Min* untuk menyesuaikan. Parameter ketiga adalah Trafo Arus Primer (*Transformer Current Primary*), sesuaikan dengan Primer Trafo CT yang digunakan yaitu “50A” gunakan tombol *Max* dan *Min* untuk menyesuaikan. Kemudian tekan tombol *display* untuk tampilan selanjutnya.
- 5) Pada tampilan Meteran Permintaan Daya ( *Power Demand Meter*), tekan tombol *Max* untuk menyesuaikan opsi kode “A-Ph” (Arus setiap fasa). Kemudian pilih periode 1 *minutes*. Kemudian tekan tombol *display* untuk tampilan selanjutnya.
- 6) Pada tampilan pengaturan Halaman Utama (*Main Page*) tekan tombol *MAX* untuk menyesuaikan opsi “VAR5” untuk menampilkan semua objek parameter listrik yang dibaca dengan melakukan pemindahan tampilan

secara otomatis setiap 5 detik. Kemudian tekan tombol *display* untuk tampilan selanjutnya.

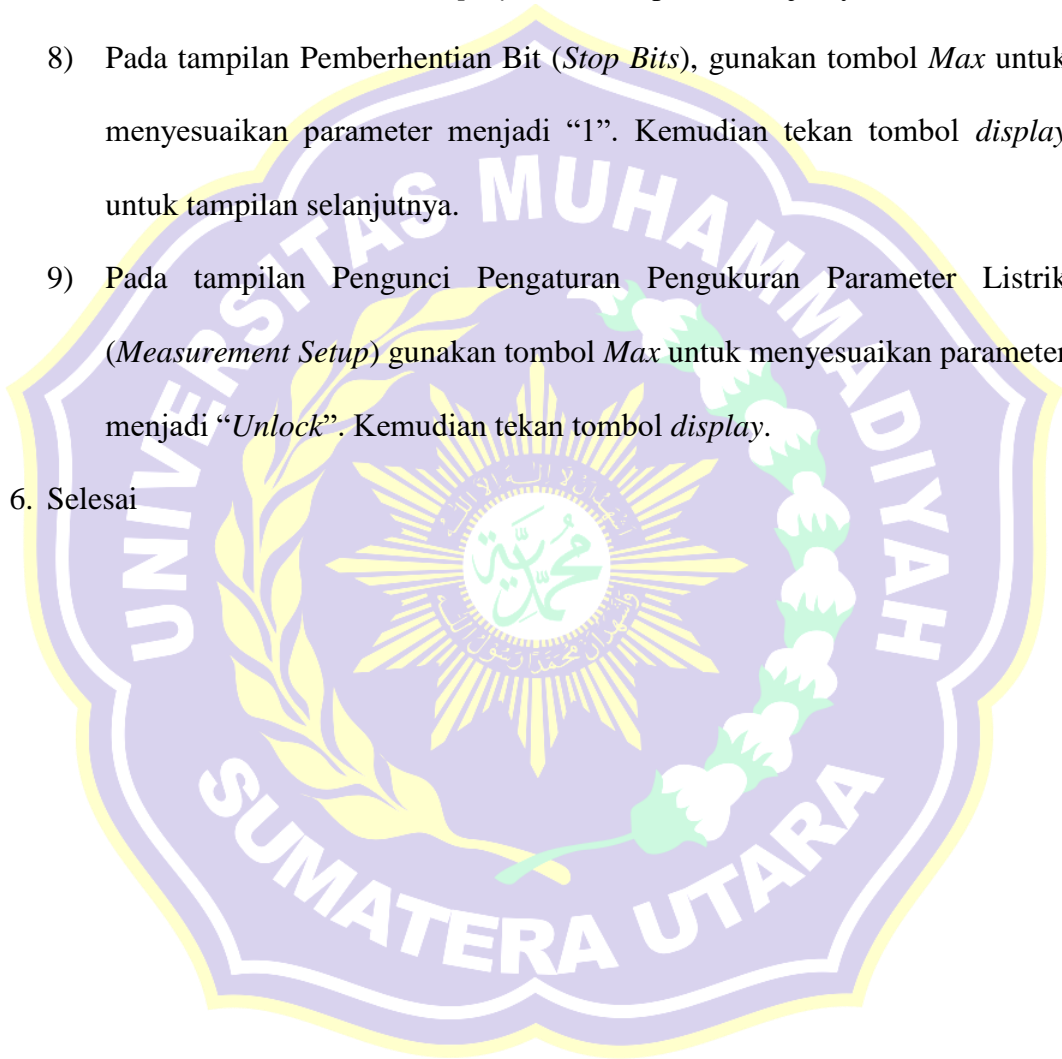
- 7) Pada tampilan Energi yang khusus dibaca (*Preferred Energy*) tekan tombol *MAX* untuk menyesuaikan opsi "*Energy kWh*" untuk menampilkan nilai Kwh. Kemudian tekan tombol *display* untuk tampilan selanjutnya.
  - 8) Pada tampilan Waktu Cahaya Latar Belakang (*Backlight Timer*), tekan tombol *MAX* dan *MIN* untuk menyesuaikan waktu cahaya menjadi permanen dengan kode "00". Kemudian tekan tombol *display* untuk tampilan selanjutnya.
  - 9) Pada tampilan Menghapus Nilai Energi (*Deletion of Energy*) tekan tombol *MAX* untuk menyesuaikan opsi "NO" (tidak menghapus nilai energi). Kemudian tekan tombol *display* untuk tampilan selanjutnya.
  - 10) Pada tampilan Analisis Distorsi Harmonis (*Harmonic Distortion Analysis*) dapat diabaikan. Kemudian tekan tombol *display* untuk tampilan selanjutnya.
  - 11) Pada tampilan Keluaran Digital Transistor (*Digital Output for Transistor*) tekan tombol *MAX* dan *MIN* untuk menyesuaikan kode "00" untuk menonaktifkan Pulse dan Alarm.
3. Menghidupkan catu daya (*power supply*) dan memeriksa parameter listrik pada layar *Power Meter*.
  4. Apabila pembacaan parameter listrik sesuai, maka melanjutkan konfigurasi *Protocol MODBUS Power Analyzer*. Apabila belum sesuai, mengamati dan memeriksa serta mengatur kembali pengaturan parameter listrik.

## 5. Konfigurasi Komunikasi *Protocol MODBUS*

Berikut pengaturan komunikasi *protocol MODBUS* pada CVM-NRG96.

- 1) Untuk mengakses Pengaturan Komunikasi (*Communication Setup*) tombol *RESET* harus ditekan (sampai peralatan dimulai) dan kemudian tekan tombol *MAX* , *MIN*  dan *display*  bersamaan harus ditekan sampai *mode* pengaturan dimasukkan.
- 2) Pada tampilan "*Set Prot Bus*" dengan menggunakan layar informasi ini, peralatan menginformasikan kepada pengguna bahwa Protokol Komunikasi melalui port seri RS-485 adalah *MODBUS*. Kemudian tekan tombol *display* untuk tampilan selanjutnya.
- 3) Pada tampilan Setelan Pabrik (*Default Setting*), terdapat dua opsi yaitu "*Yes*" dan "*No*", gunakan tombol *Max* untuk menyesuaikan parameter setelan pabrik menjadi "*No*". Kemudian tekan tombol *display* untuk tampilan selanjutnya.
- 4) Pada tampilan Nomor Periferal (*Peripheral Number*), gunakan tombol *Min* dan *Max* untuk menyesuaikan parameter menjadi "3". Ini disebut juga pengalamanan modul. Kemudian tekan tombol *display* untuk tampilan selanjutnya.
- 5) Pada tampilan Kecepatan Transmisi (*Transmission Speed*) atau biasa disebut dengan *Baud Rate*, gunakan tombol *Max* untuk menyesuaikan parameter menjadi "4800". Kemudian tekan tombol *display* untuk tampilan selanjutnya.

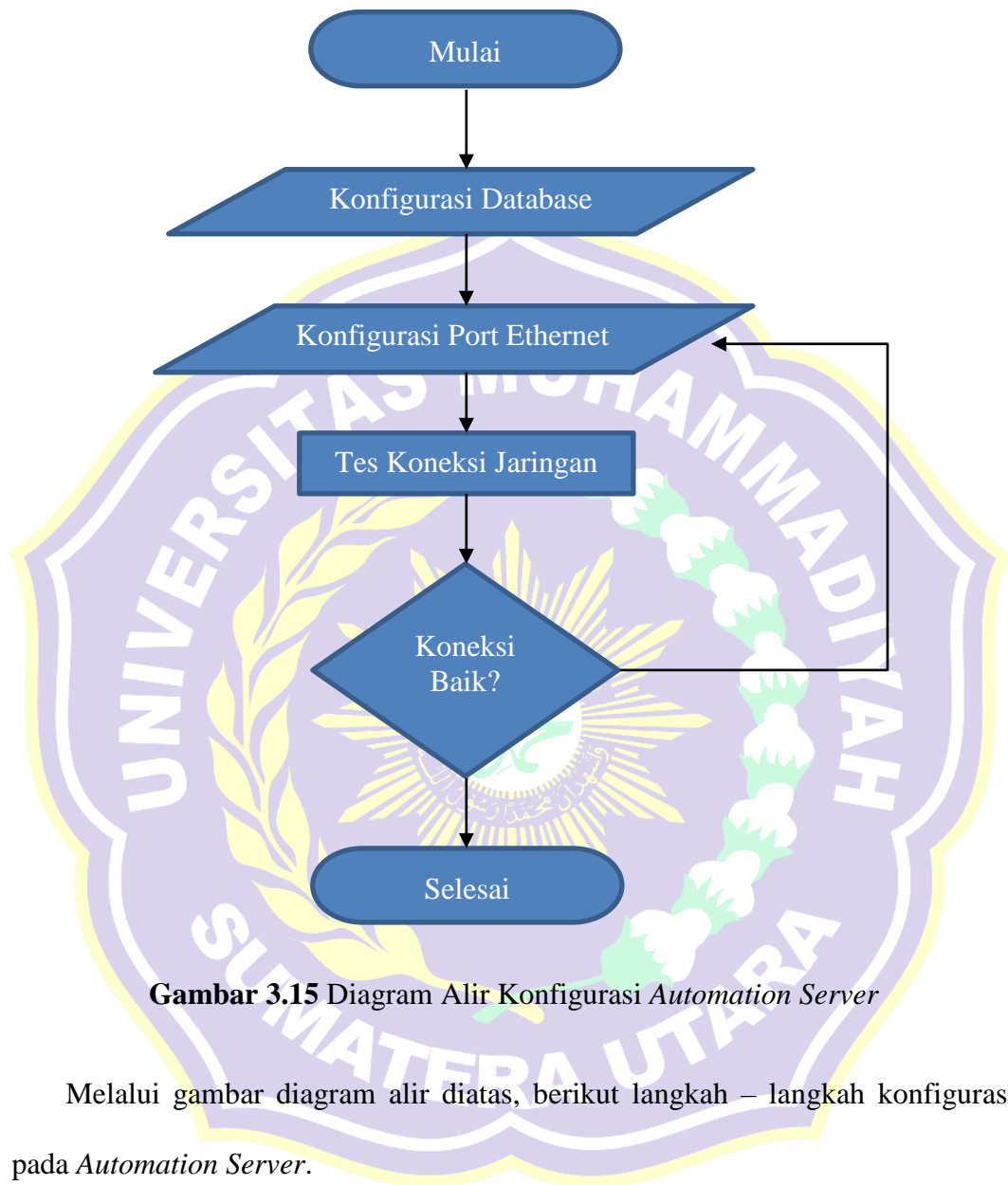
- 6) Pada tampilan Keseimbangan (*Parity*), gunakan tombol *Max* untuk menyesuaikan parameter menjadi “*odd*”. Kemudian tekan tombol *display* untuk tampilan selanjutnya.
  - 7) Pada tampilan Bit Data (*Data Bits*), layar ini hanya menginformasikan bahwa bit data berjumlah 8 bit, tidak ada pengaturan pada tampilan ini. Kemudian tekan tombol *display* untuk tampilan selanjutnya.
  - 8) Pada tampilan Pemberhentian Bit (*Stop Bits*), gunakan tombol *Max* untuk menyesuaikan parameter menjadi “1”. Kemudian tekan tombol *display* untuk tampilan selanjutnya.
  - 9) Pada tampilan Pengunci Pengaturan Pengukuran Parameter Listrik (*Measurement Setup*) gunakan tombol *Max* untuk menyesuaikan parameter menjadi “*Unlock*”. Kemudian tekan tombol *display*.
6. Selesai





### 3.2.1.6 Diagram Alir Konfigurasi *Automation Server*

Diagram alir konfigurasi *Automation Server* dapat dilihat pada gambar dibawah ini.



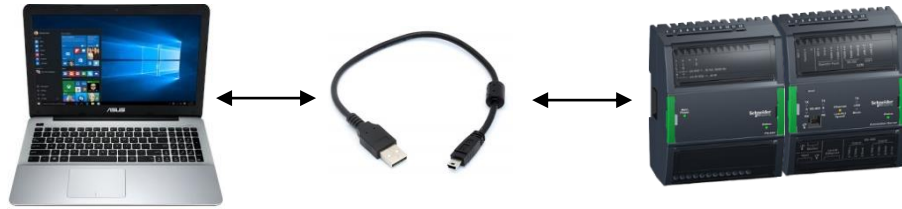
**Gambar 3.15** Diagram Alir Konfigurasi *Automation Server*

Melalui gambar diagram alir diatas, berikut langkah – langkah konfigurasi pada *Automation Server*.

1. Mempersiapkan bahan (Catu daya 24VDC, *Computer*, *Power Supply AS*, *Automation Server*). Menghidupkan catu daya 24VDC yang dihubungkan dengan perangkat *Power Supply AS*. Menghubungkan *personal computer* (PC)

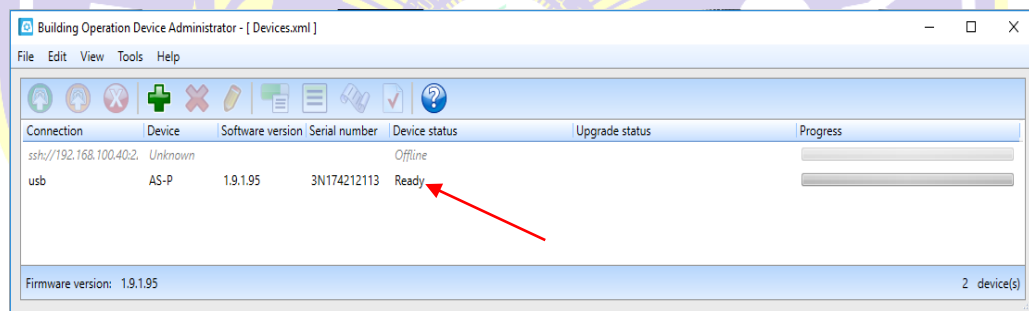


dengan *Automation Server* menggunakan kabel mini-USB. Koneksi personal komputer dengan *Automation Server* dapat dilihat pada gambar dibawah ini.



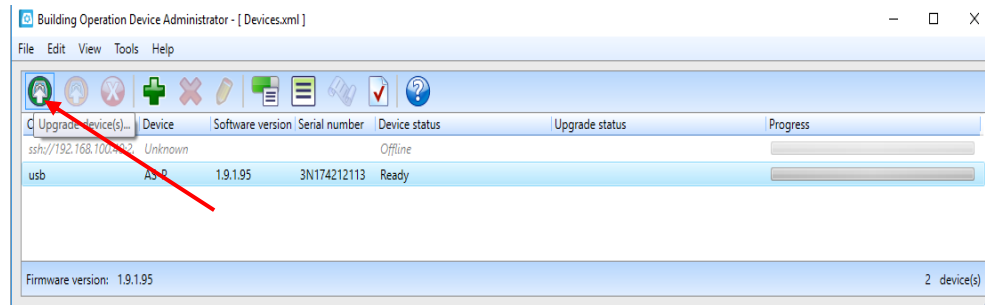
**Gambar 3.16** Koneksi Personal Komputer dengan *Automation Server*.

2. Melakukan konfigurasi *database Automation Server* seperti berikut ini.
  - 1) Membuka perangkat lunak *SBO Device Administrator*.
  - 2) Pada tampilan beranda terdapat *Automation Server* yang terdeteksi melalui kabel mini USB. Tampilan *Device Administrator Pindai Automation Server* dapat dilihat pada gambar dibawah ini.



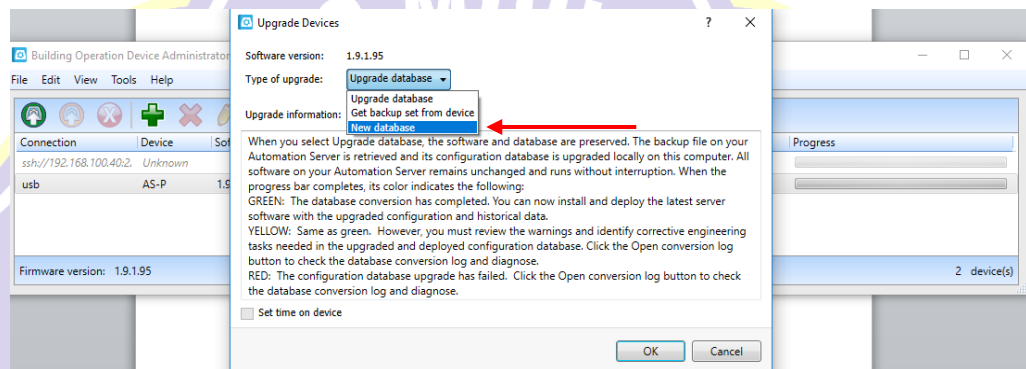
**Gambar 3.17** Tampilan *Device Administrator Pindai Automation Server*

- 3) Memilih perangkat *Automation Server* yang dikenali kemudian Klik Kanan dan pilih *Upgrade Device* atau dengan menekan tombol *Upgrade Device* seperti gambar dibawah ini.



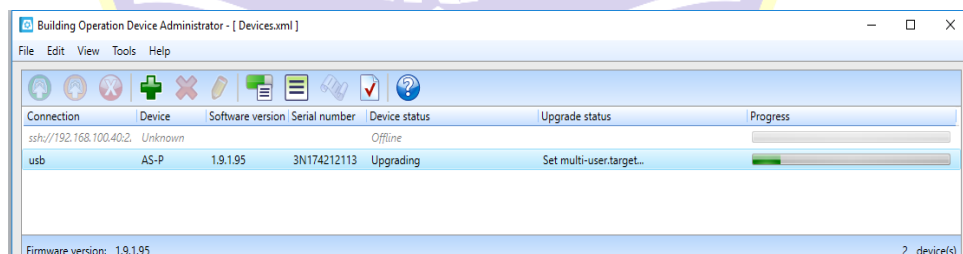
**Gambar 3.18** Tampilan Untuk *Upgrade Device* Pada *Automation Server*

- 4) Setelah *pop-up menu upgrade database* terbuka, pada opsi *type of upgrade* pilih 'New Database' sesuai gambar dibawah ini.



**Gambar 3.19** Tampilan Opsi *New Database* Pada *Automation Server*.

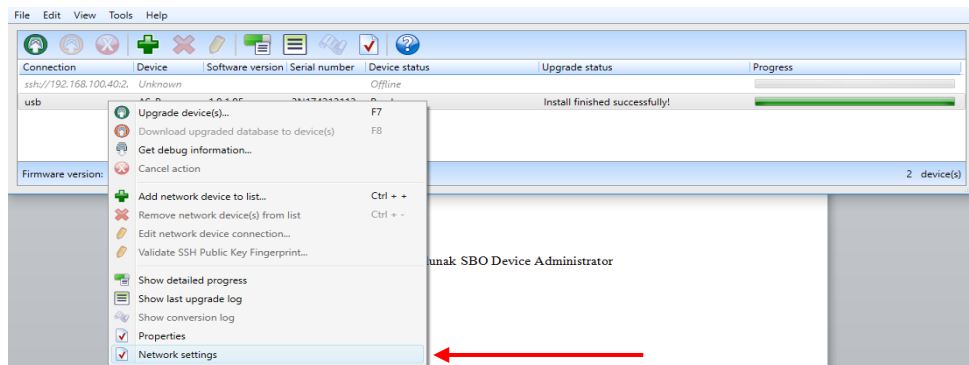
- 5) Menunggu proses *database* baru selesai pada *Automation Server*, jangan melepas kabel selama proses *upgrade database* baru di masukkan ke dalam *Automation Server*. Tampilan proses *Upgrade Device* pada *Automation Server* dapat dilihat pada gambar dibawah ini.



**Gambar 3.20** Tampilan Proses *Upgrade Device* Pada *Automation Server*

3. Setelah proses *upgrade device* selesai, melakukan konfigurasi port Ethernet seperti berikut ini.

1) Pilih perangkat kemudian klik kanan pilih '*Network Setting*'.



**Gambar 3.21** Tampilan Opsi *Network Setting*

2) Setelah *pop-up menu network setting* terbuka, pilih tab *Ethernet 1*.

Kemudian centang *Dynamic DNS*. Isi sesuai dengan nilai berikut.

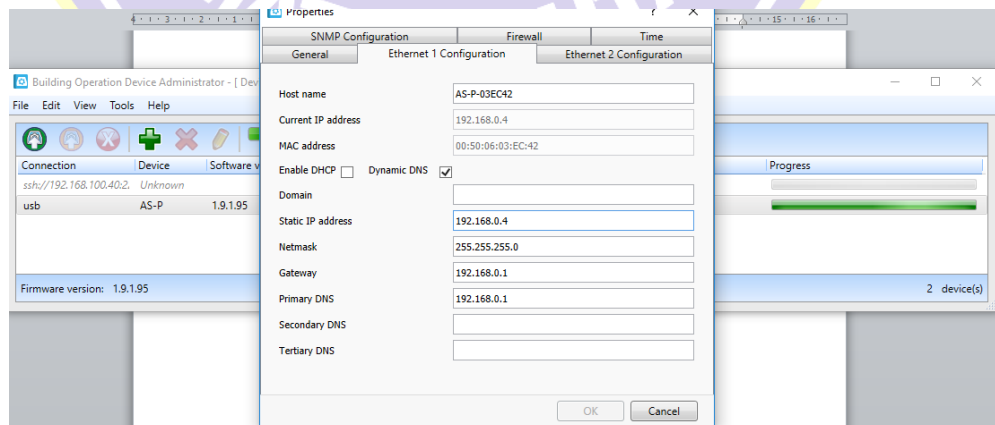
*Static IP Address* : 192.168.0.4

*Netmask* : 255.255.255.0

*Gateway* : 192.168.0.1

*Primary DNS* : 192.168.0.1

Tampilan konfigurasi *Ethernet 1* pada *Automation Server* dapat dilihat pada gambar dibawah ini.



**Gambar 3.22** Tampilan Konfigurasi *Ethernet 1* Pada *Automation Server*.

- 3) Setelah port Ethernet dikonfigurasi klik *OK*. Kemudian lepaskan kabel *mini USB*.
4. Menghubungkan *computer* dengan *Automation Server* menggunakan kabel LAN, kemudian melakukan tes *ping* jaringan melalui *command prompt*.
5. Apabila koneksi baik, maka konfigurasi *Automation Server* berhasil. Apabila koneksi gagal, maka melakukan kembali konfigurasi port Ethernet.
6. Selesai.

### 3.2.2 Perancangan Perangkat Lunak (*Software*)

Dalam perancangan perangkat lunak dan grafis bertujuan untuk dapat membuat sebuah antarmuka (*Interface*) yang dapat mengendalikan dan mengawasi semua perangkat *slave MODBUS* yang terhubung dengan *master Automation Server*. Perangkat lunak yang dibutuhkan dalam pembuatan antarmuka adalah *Struxeware Building Operation Workstation*.

Antarmuka (*Interface*) adalah sebuah sistem pengawasan dan pengendalian, dengan cara melakukan pengumpulan dan analisa data secara *real time*. Sistem ini terdiri dari 3 bagian utama yaitu *Master*, *Slave*, dan media komunikasi. Pengontrolan dengan *Interface* dilakukan dengan memanfaatkan perangkat lunak dari *Schneider Electric Struxeware Building Operation Workstation*. Antarmuka akan memvisualisasikan proses yang terjadi pada perangkat *slave MODBUS* baik nilai – nilai *input/output* digital maupun nilai – nilai *input/output* analog.

Perencanaan perangkat lunak dalam pembuatan *interface* terbagi dalam 4 bagian.

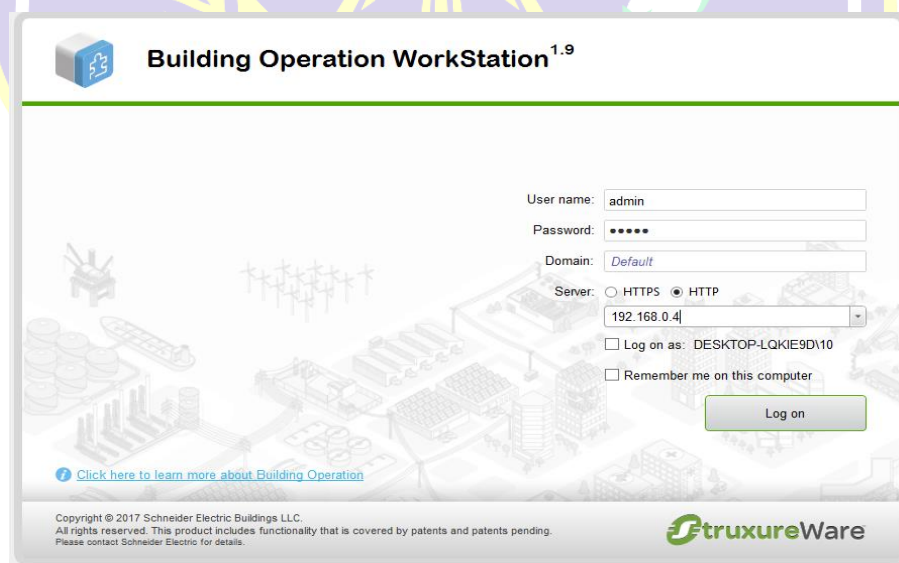
- A. Pembuatan *Interface* untuk *MODBUS Master Network Port A*
- B. Pembuatan *Interface* untuk *MODBUS Master Network Port B*
- C. Pembuatan *Interface* untuk *MODBUS TCP Network*
- D. Pemrograman dan Pembuatan Grafis
- E. Proses *Binding* Pada *Input/Output* Program dan Komponen Grafis

Berikut langkah awal sebelum memulai pembuatan antarmuka menggunakan *SBO Workstation*.

1. Menghubungkan semua perangkat *MODBUS* dengan *Automation Server*.
2. Menghubungkan *Personal Computer* dengan *TP-Link* yang terhubung.
3. Membuka aplikasi *SBO Workstation*, pada jendela awal isi kolom sebagai berikut.

*User name* : admin  
*Password* : admin  
*Server* : 192.168.0.4

Tampilan sesuai dengan gambar dibawah ini. Kemudian *Log On*.



**Gambar 3.23** Jendela Awal *SBO Workstation*

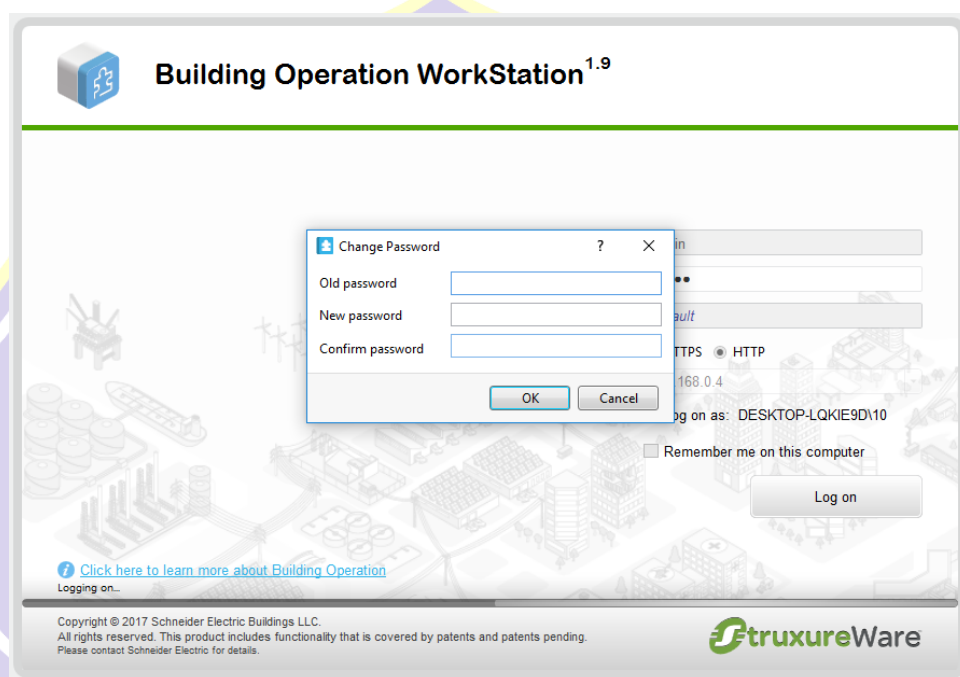
4. Setelah *Log On*, maka akan muncul permintaan untuk memasukkan password baru. Isi kolom sebagai berikut.

*Old password* : admin

*New password* : P@ssw0rd

*Confirm password* : P@ssw0rd

Tampilan konfigurasi *password* baru dapat dilihat pada gambar dibawah ini.



**Gambar 3.24** Tampilan Konfigurasi Password Baru

5. Klik *OK*. Selesai

### 3.2.2.1 Pembuatan *Interface MODBUS Master Network Port A*

Pada pembuatan *Interface MODBUS* master network port A, perangkat keras yang terhubung dengan *Automation Server Port A* adalah *Thermostat TC-300* dan *Power Meter Circutor CVM-NRG96*. Berikut langkah – langkah pembuatan *Interface MODBUS master network port A*.



1. Pada halaman kerja (*workspace*), panel *system tree*, klik kanan pada 'Server 1' pilih *New* → *Interface* untuk membuat koneksi dengan perangkat *MODBUS*. Pada bagian ini, perangkat yang pertama kali dibuat adalah untuk *Thermostat TC-300*.
2. Pada jendela *Interface*, pilih *MODBUS Master Network*. Ketik pada kolom name : 'MODBUS Master Network Port A'. Kemudian *Create*.
3. Klik kanan pada 'MODBUS Master Network Port A', pilih *Properties*.

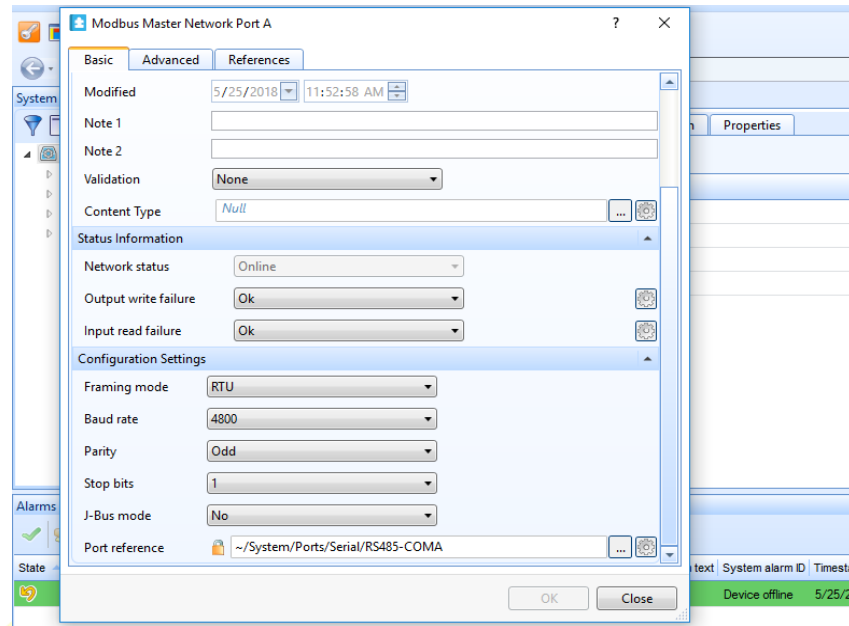
Isi kolom pada 'Configuration Setting' sebagai berikut :

<i>Framing Mode</i>	: RTU
<i>Baud Rate</i>	: 4800
<i>Parity</i>	: <i>Odd</i>
<i>Stop Bits</i>	: 1
<i>J-Bus Mode</i>	: <i>No</i>
<i>Port Reference</i>	: COM – A

Setelah selesai Klik *OK*

*Configuration Setting MODBUS Master Network Port A* dapat dilihat pada gambar dibawah ini.

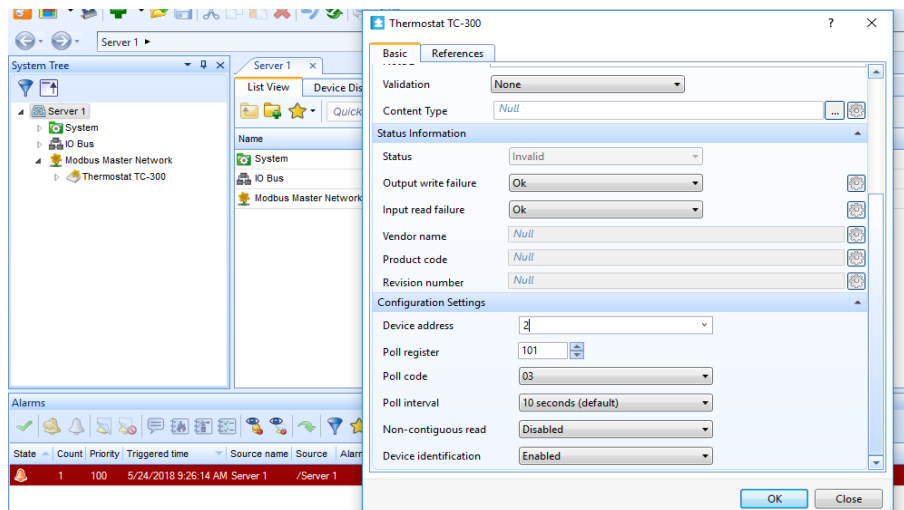




**Gambar 3.25** Configuration Setting MODBUS Master Network Port A

4. Kemudian pada panel *system tree* klik kanan 'MODBUS Master Network Port A' yang sudah di *create*, pilih *New* → *MODBUS Device*.
5. Pada jendela *create MODBUS device*, isi kolom name 'Thermostat TC-300'. Kemudian klik tombol *Create*.  
Variable MODBUS Thermostat TC-300 yang penulis gunakan untuk dibaca hanya 4 variable MODBUS yaitu *Room Temperature*, *Fan Mode*, *Eco Mode*, dan *Set Temperature*.
6. Pada panel *system tree* klik kanan pada 'Thermostat TC-300' pilih *properties*. Pilih tab *Basic*, pada *Configuration Settings* kolom *Device Adress* isi dengan nilai '2'. Klik *OK*. Periksa apakah status perangkat *Thermostat* dengan *Automation Server* sudah *Online* dengan kembali ke menu *Properties* kolom *Status Information*.

Pengaturan pengenalan alamat perangkat *Thermostat TC300* dapat dilihat pada gambar dibawah ini.



**Gambar 3.26** Pengaturan pengenalan alamat perangkat *Thermostat TC300*.

7. Kemudian pada perangkat *Thermostat TC300*, parameter *MODBUS* yang dibaca menggunakan beberapa tipe *MODBUS* point sebagai berikut.
  - a. *MODBUS Analog Input Point* :  
Digunakan untuk membaca nilai masukan analog pada *thermostat*.
  - b. *MODBUS Multistate Output Point* :  
Digunakan untuk memberi beberapa status keluaran pada *thermostat*.
  - c. *MODBUS Digital Output Point* :  
Digunakan untuk memberi perintah hanya 2 keluaran berbeda.
  - d. *MODBUS Analog Output Point* :  
Digunakan untuk memberi nilai keluaran analog pada *thermostat*.  
Semua parameter *MODBUS* baik *register number*, *register type*, *function code* dapat dilihat melalui lampiran 4.
8. Kemudian membuat masing – masing *MODBUS Point*. Klik kanan ‘*Thermostat TC-300*’ pilih *New* → *MODBUS Point* untuk membuat parameter – parameter *MODBUS Thermostat TC-300* yang akan dibaca dan dikontrol.

9. Pada jendela *MODBUS Point*, pilih '*MODBUS Analog Input Point*'. Isi kolom name dengan nama '*Room Temperature*'. Klik *OK*
10. Setelah *MODBUS point* selesai dibuat, klik kanan pilih *properties*, Pilih tab *Basic*, pada menu *Configuration Setting* isi kolom sebagai berikut.

*Register Number* : 1

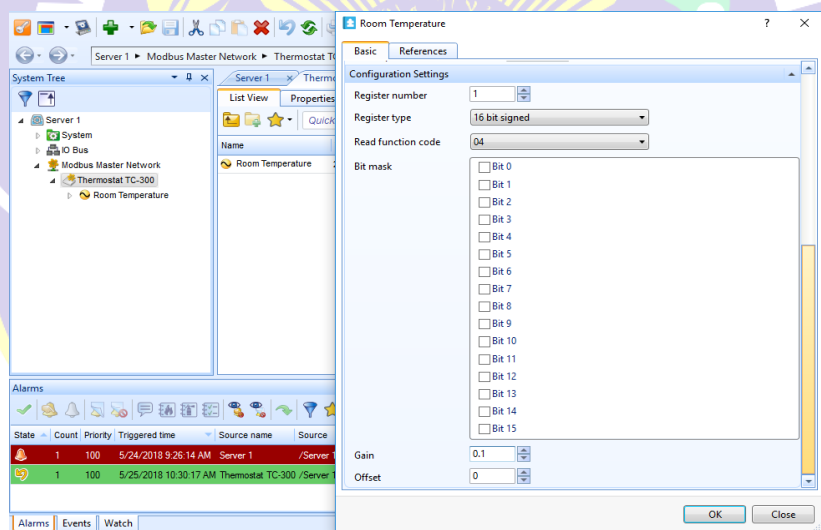
*Register Type* : Menganalisa satu persatu.

*Read Function Code* : 4

*Gain* : 0.1 (factor kali)

*Register Number* diatas digunakan untuk melihat nilai *actual temperature* yang dibaca oleh *Thermostat TC-300*.

Konfigurasi *MODBUS Point Room Temperature* dapat dilihat pada gambar dibawah ini.



**Gambar 3.27** Konfigurasi *MODBUS Point Room Temperature*.

11. Setelah selesai klik *OK*. Untuk menambah *MODBUS Point* yang lainnya, Klik kanan '*Thermostat TC-300*' pilih *New* → *MODBUS Point*
12. Pada jendela *MODBUS Point*, pilih '*MODBUS Multistate Output Point*'. Isi kolom name dengan nama '*Fan Mode*'. Klik *OK*

13. Setelah *MODBUS point* selesai dibuat, klik kanan pilih *properties*, Pilih tab *Basic*, pada menu *Configuration Setting* isi kolom sebagai berikut.

*Register Number* : 6

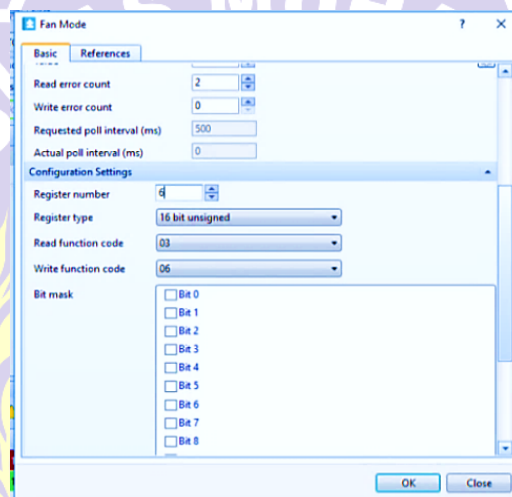
*Register Type* : Menganalisa satu persatu

*Read Function Code* : 3

*Write Function Code* : 6

*Register Number* diatas digunakan untuk mengatur kecepatan kipas.

Konfigurasi *Modbus Point Fan Mode* dapat dilihat pada gambar dibawah ini.



**Gambar 3.28** Konfigurasi Modbus Point Fan Mode

14. Setelah selesai klik *OK*. Kemudian melakukan penambahan *MODBUS Point* yang berbeda yaitu '*MODBUS Digital Output Point*' dengan cara yang sama.
15. Isi kolom name dengan nama '*Eco Mode*'. Pada menu *properties* untuk *Eco Mode*, sesuaikan *configuration settings* seperti dibawah ini.

*Register Number* : 10

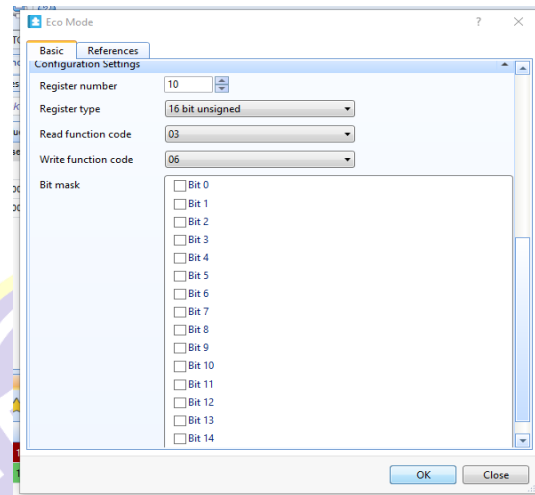
*Register Type* : Menganalisa satu persatu

*Read Function Code* : 3

*Write Function Code* : 6

*Register Number* diatas digunakan untuk mengatur kecepatan kipas sesuai dengan *mode Eco* (Penghematan Daya).

Konfigurasi *Modbus Point Eco Mode* dapat dilihat pada gambar dibawah ini.



**Gambar 3.29** Konfigurasi *Modbus Point Eco Mode*

16. Setelah selesai klik *OK*. Kemudian melakukan penambahan *MODBUS Point* yang berbeda yaitu '*MODBUS Analog Output Point*' dengan cara yang sama.
17. Isi kolom name dengan nama '*Set Temperature*'. Pada menu *properties* untuk *Set Temperature*, sesuaikan *configuration settings* seperti dibawah ini.

*Register Number* : 5

*Register Type* : Menganalisa satu persatu

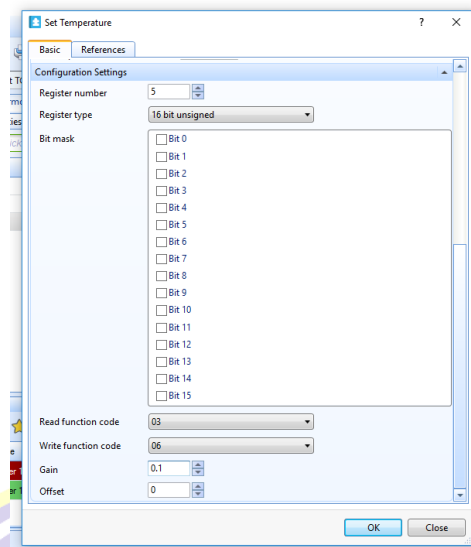
*Read Function Code* : 3

*Write Function Code* : 6

*Gain* : 0.1

*Register Number* diatas digunakan untuk mengatur nilai *temperature* yang ingin dicapai.

Konfigurasi *MODBUS Point Set Temperature* dapat dilihat pada gambar dibawah ini.



**Gambar 3.30** Konfigurasi *MODBUS Point Set Temperature*

18. Setelah selesai klik *OK*. Kemudian melihat semua *variable MODBUS* yang telah dibuat untuk *Thermostat TC-300* seperti yang ditunjukkan pada gambar dibawah ini.

Name	Value	Register number	Register type	Read function code
Eco Mode	False	10	16 bit unsigned	03
Fan Mode	1	6	16 bit unsigned	03
Room Temperature	26.00	1	16 bit unsigned	04
Set Temperature	22.00	5	16 bit unsigned	03

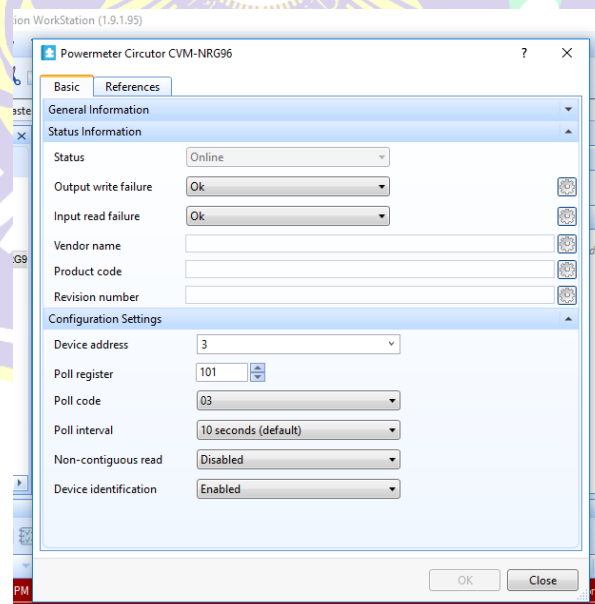
**Gambar 3.31** Tampilan Nilai – Nilai *MODBUS Point Thermostat TC-300*.

Setelah melakukan pembacaan *variable MODBUS Point* untuk *Thermostat TC-300*. Selanjutnya melakukan pembacaan *variable MODBUS* untuk *Power Meter Circutor CVM-NRG96*. Karena pada simulasi ini penulis menggunakan tegangan listrik 1 fasa, maka *variable MODBUS* perangkat *Power Meter Circutor CVM-NRG96* yang penulis gunakan untuk dibaca hanya 6 *variable MODBUS* yaitu *Voltage (V L1)*, *Current (A L1)*, *Active Power (kW L1)*, *Reactive Power*



(kVar L1), *Power Factor* (PF L1), dan *Frequency* (Hz). *Variable MODBUS Power Meter* dapat dilihat pada [lampiran 3](#). Berikut langkah – langkah untuk memasukkan nilai *variable Power Meter* ke dalam interface *MODBUS Master Network Port A*.

1. Pada panel *system tree* klik kanan ‘*MODBUS Master Network Port A*’ pilih *New* → *MODBUS Device*.
2. Pada jendela *create MODBUS device*, isi kolom name ‘*Power Meter Circutor CVM-NRG96*’. Kemudian klik tombol *Create*.
3. Pada panel *system tree* klik kanan pada ‘*Power meter Circutor CVM-NRG96*’ pilih *properties*. Pilih tab *Basic*, pada *Configuration Settings* kolom *Device Adress* isi dengan nilai ‘3’. Klik *OK*. Periksa apakah status perangkat *Power Meter* dengan *Automation Server* sudah *Online* dengan kembali ke menu *Properties* kolom *Status Information* seperti terlihat pada gambar dibawah ini.



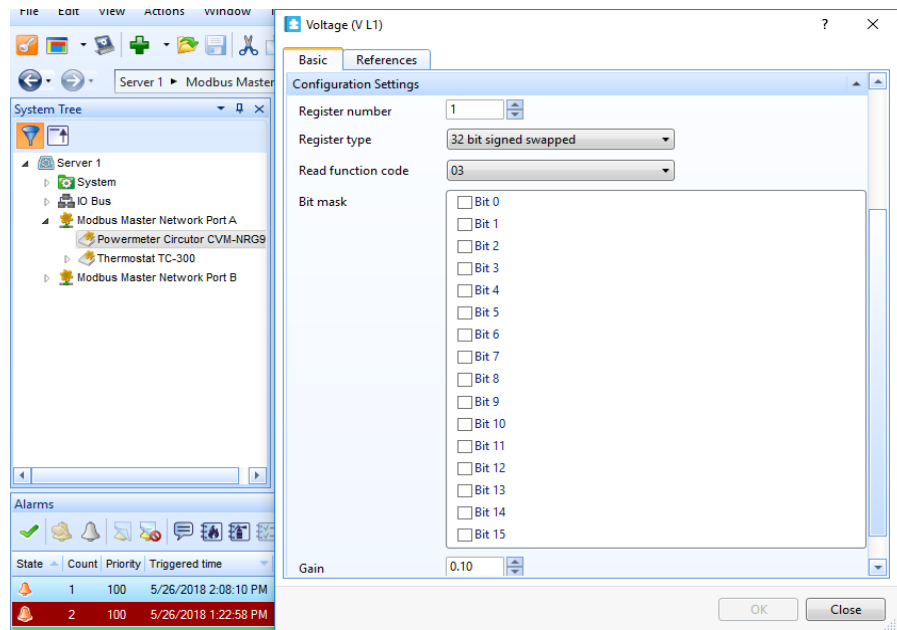
**Gambar 3.32** Pengaturan pengenalan alamat perangkat *Power Meter* Circutor CVM-NRG96



4. Kemudian pada perangkat *Power Meter Circutor CVM-NRG96* parameter *MODBUS* yang dibaca hanya menggunakan *MODBUS Analog Input Point*.  
Semua parameter *MODBUS* baik *register number*, *register type*, *function code* dapat dilihat melalui lampiran 3.
5. Kemudian membuat masing – masing *MODBUS Point*. Klik kanan ‘*Power Meter Circutor CVM-NRG96*’ pilih *New* → *MODBUS Point* untuk membuat parameter – parameter *MODBUS Power Meter Circutor CVM-NRG96* yang akan dibaca dan dikontrol.
6. Pada jendela *MODBUS Point*, pilih ‘*MODBUS Analog Input Point*’. Isi kolom name dengan nama ‘*Voltage (V L1)*’. Klik OK
7. Setelah *MODBUS point* selesai dibuat, klik kanan pilih *properties*, Pilih tab *Basic*, pada menu *Configuration Setting* isi kolom sebagai berikut.  

<i>Register Number</i>	: ( <i>Hexadecimal</i> : 00 - 01 → <i>Decimal</i> : 00 - 01 )
<i>Register Type</i>	: Menganalisa satu persatu
<i>Read Function Code</i>	: 3
<i>Gain</i>	: 0.1

*Register Number* diatas untuk membaca nilai tegangan 1 fase pada terminal L1. *Register number* yang tertera pada lampiran menggunakan bilangan *hexadecimal*, bilangan tersebut harus dikonversi kedalam bilangan *basic decimal*. Konfigurasi *MODBUS Point Voltage (V L1)* dapat dilihat pada gambar dibawah ini.



**Gambar 3.33** Konfigurasi *MODBUS Point Voltage (V L1)*

8. Setelah selesai klik *OK*. Kemudian melakukan penambahan *MODBUS Point* untuk *Current (A L1)* dengan cara yang sama.
9. Isi kolom *name* dengan nama '*Current (A L1)*'. Pada menu *properties* untuk *Current (A L1)*, sesuaikan *configuration settings* seperti dibawah ini.

*Register Number* : ( *Hexadecimal* : 02 - 03 → *Decimal* : 02 - 03 )

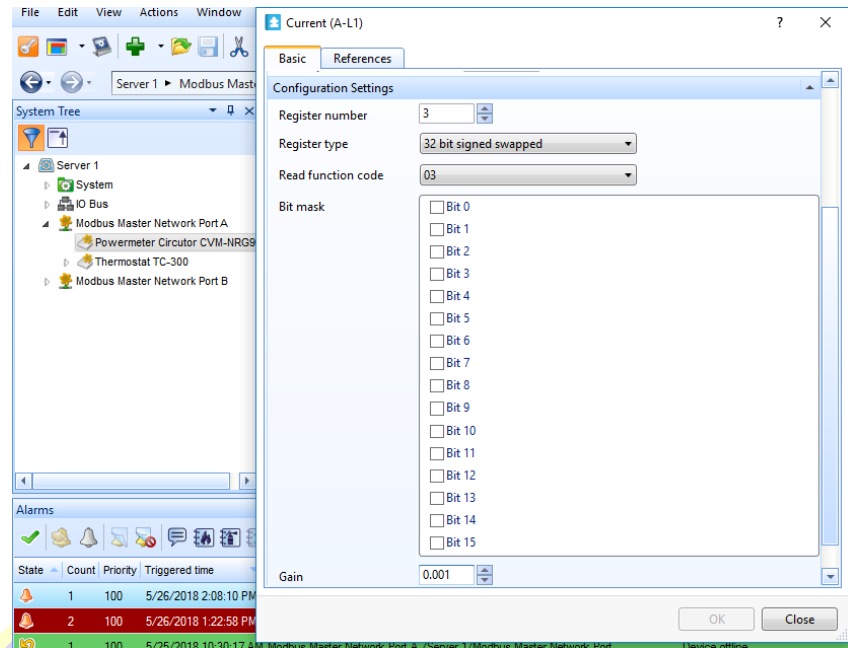
*Register Type* : Menganalisa satu persatu

*Read Function Code* : 3

*Gain* : 0.001

*Register Number* diatas untuk membaca nilai arus 1 fase pada terminal L1.

Konfigurasi *MODBUS Point Current (A L1)* dapat dilihat pada gambar dibawah ini.



**Gambar 3.34** Konfigurasi *MODBUS Point Current (A L1)*.

10. Setelah selesai klik *OK*. Kemudian melakukan penambahan *MODBUS Point* untuk *Active Power (kW L1)* dengan cara yang sama.
11. Isi kolom *name* dengan nama '*Active Power (kW L1)*'. Pada menu *properties* untuk *Active Power (kW L1)*, sesuaikan *configuration settings* seperti dibawah ini.

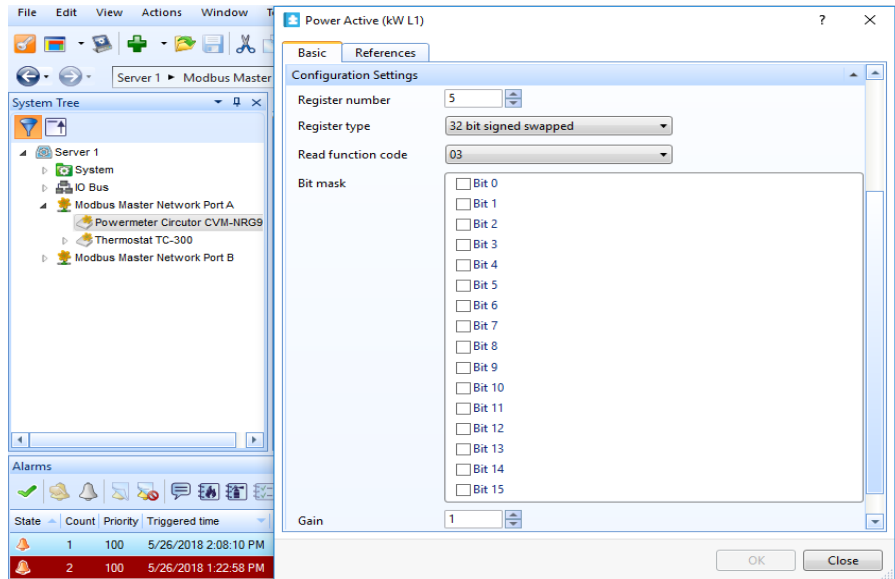
*Register Number* : ( *Hexadecimal* : 04 - 05 → *Decimal* : 04 - 05 )

*Register Type* : Menganalisa satu persatu

*Read Function Code* : 3

*Gain* : 1

*Register Number* diatas untuk membaca nilai daya aktif 1 fase pada terminal L1. Konfigurasi *MODBUS Active Power (kW L1)* dapat dilihat pada gambar dibawah ini.



**Gambar 3.35** Konfigurasi *MODBUS Point Active Power (kW L1)*

12. Setelah selesai klik *OK*. Kemudian melakukan penambahan *MODBUS Point* untuk *Reactive Power (kVar L1)* dengan cara yang sama.
13. Isi kolom *name* dengan nama '*Reactive Power (kVar L1)*'. Pada menu *properties* untuk *Reactive Power (kVar L1)*, sesuaikan *configuration settings* seperti dibawah ini.

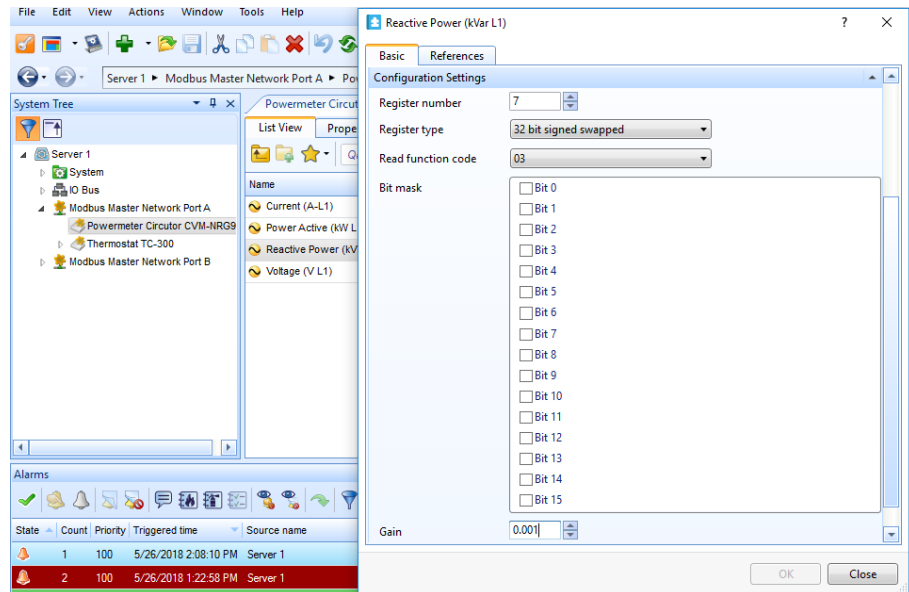
*Register Number* : ( *Hexadecimal* : 06 - 07 → *Decimal* : 06 - 07 )

*Register Type* : Mencoba satu persatu

*Read Function Code* : 3

*Gain* : 0.001

*Register Number* diatas untuk membaca nilai daya reaktif 1 fase pada terminal L1. Konfigurasi *MODBUS Point Reactive Power (kVar L1)* dapat dilihat pada gambar dibawah ini.



**Gambar 3.36** Konfigurasi *MODBUS Point Reactive Power (kVar L1)*

14. Setelah selesai klik *OK*. Kemudian melakukan penambahan *MODBUS Point* untuk *Power Factor (PF L1)* dengan cara yang sama.
15. Isi kolom *name* dengan nama '*Power Factor (PF L1)*'. Pada menu *properties* untuk *Power Factor (PF L1)*, sesuaikan *configuration settings* seperti dibawah ini.

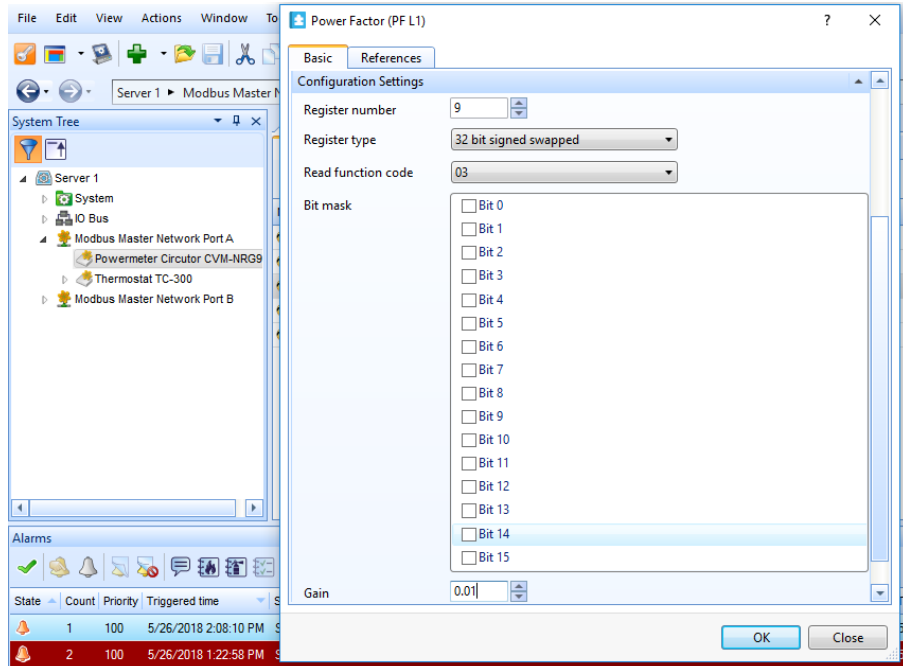
*Register Number* : ( *Hexadecimal* : 08 - 09 → *Decimal* : 08 - 09 )

*Register Type* : Menganalisa satu persatu

*Read Function Code* : 3

*Gain* : 0.01

*Register Number* diatas untuk membaca nilai faktor daya 1 fase pada terminal L1. Konfigurasi *MODBUS Point Power Factor (PF L1)* dapat dilihat pada gambar dibawah ini.



**Gambar 3.37** Konfigurasi *MODBUS Point Power Factor (PF L1)*

16. Setelah selesai klik *OK*. Kemudian melakukan penambahan *MODBUS Point* untuk *Frequency (Hz)* dengan cara yang sama.
17. Isi kolom *name* dengan nama '*Frequency (Hz)*'. Pada menu *properties* untuk *Frequency (Hz)*, sesuaikan *configuration settings* seperti dibawah ini.

*Register Number* : ( *Hexadecimal* : 28 - 29 → *Decimal* : 40 - 41 )

*Register Type* : Menganalisa satu persatu

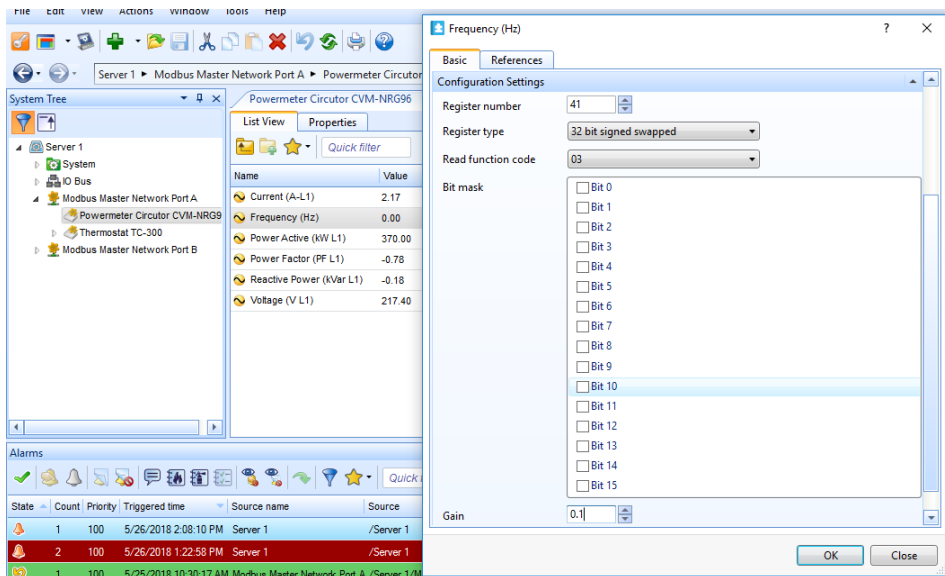
*Read Function Code* : 3

*Gain* : 0.1

*Register Number* diatas untuk membaca nilai frekuensi pada terminal L1.

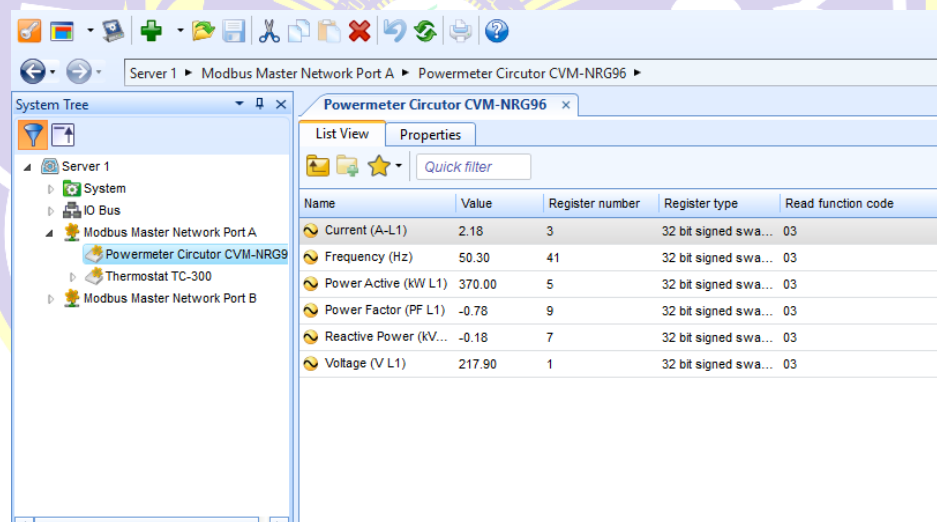
Konfigurasi *MODBUS Point Frequency (Hz)* dapat dilihat pada gambar dibawah ini.





**Gambar 3.38** Konfigurasi *MODBUS Point Frequency (Hz)*

18. Setelah selesai klik *OK*. Kemudian melihat semua *variable MODBUS* yang telah dibuat untuk perangkat *Power Meter Circutor CVM-NRG96* seperti pada gambar dibawah ini.



**Gambar 3.39** Tampilan Nilai – Nilai *MODBUS Point Power Meter Circutor CVM-NRG96*.



### 3.2.2.2 Pembuatan *Interface MODBUS Master Network Port B*

Pada pembuatan *Interface MODBUS master network port B*, perangkat keras yang terhubung dengan *Automation Server Port B* adalah *Wellpro Module I/O*. Berikut langkah – langkah pembuatan *Interface MODBUS master network port B*.

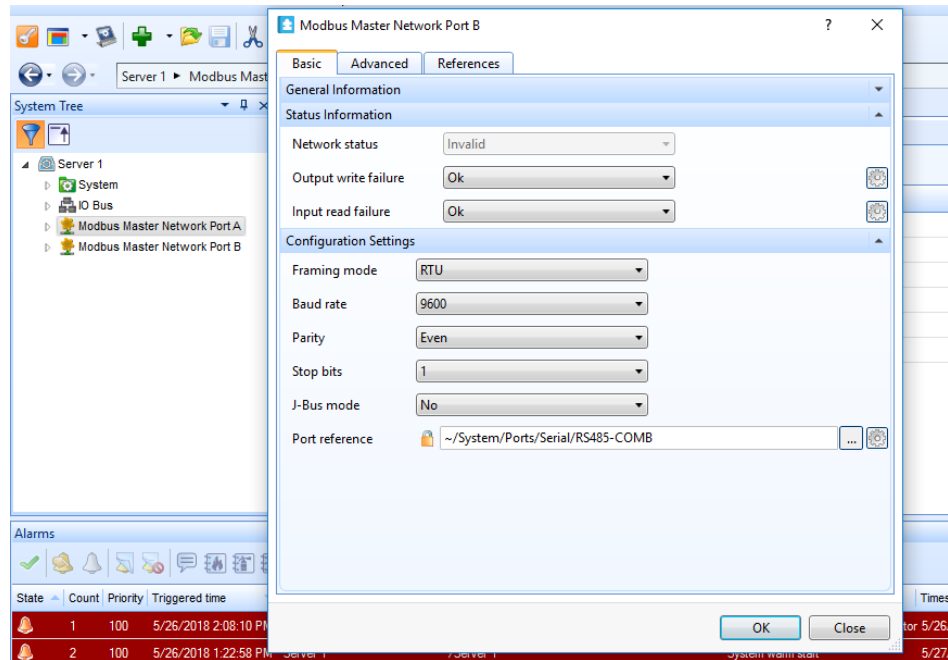
1. Pada halaman kerja (*workspace*), panel *system tree*, klik kanan pada ‘*Server 1*’ pilih *New* → *Interface* untuk membuat koneksi dengan perangkat *MODBUS*. Pada jendela *Interface*, pilih *MODBUS Master Network*. Ketik pada kolom name : ‘*MODBUS Master Network Port B*’. Kemudian *Create*.
2. Klik kanan pada ‘*MODBUS Master Network Port B*’, pilih *Properties*.

Isi kolom pada ‘*Configuration Setting*’ sebagai berikut :

<i>Framing Mode</i>	: RTU
<i>Baud Rate</i>	: 9600
<i>Parity</i>	: Even
<i>Stop Bits</i>	: 1
<i>J-Bus Mode</i>	: No
<i>Port Reference</i>	: COM – B

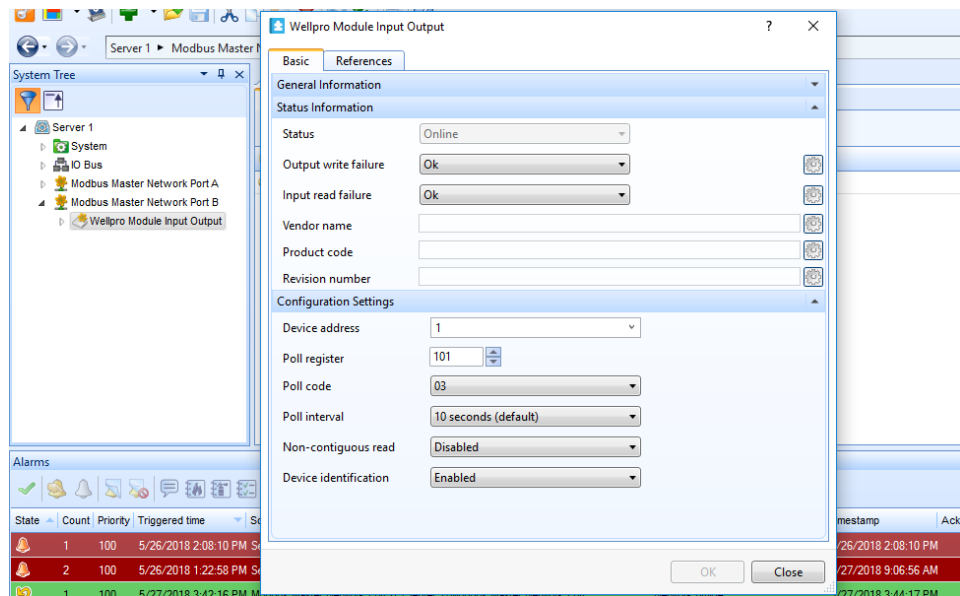
Setelah selesai Klik OK

*Configuration Setting MODBUS Master Network Port B* dapat dilihat pada gambar dibawah ini.



**Gambar 3.40** Configuration Setting MODBUS Master Network Port B

3. Kemudian pada panel *system tree* klik kanan 'MODBUS Master Network Port B' yang sudah di *create*, pilih *New* → *MODBUS Device*.
4. Pada jendela *create MODBUS device*, isi kolom name 'Wellpro Modul Input Output'. Kemudian klik tombol *Create*.  
Variable MODBUS perangkat Wellpro Module I/O yang penulis gunakan untuk dibaca hanya 2 variable MODBUS yaitu *Digital Input 1* dan *Digital Output 1*.
5. Pada panel *system tree* klik kanan pada 'Wellpro Module I/O' pilih *properties*. Pilih tab *Basic*, pada *Configuration Settings* kolom *Device Adress* isi dengan nilai '2'. Klik *OK*. Periksa apakah status perangkat Wellpro Module I/O dengan *Automation Server* sudah *Online* dengan kembali ke menu *Properties* kolom *Status Information*. Pengaturan pengenalan alamat Wellpro Module I/O dapat dilihat pada gambar dibawah ini.



**Gambar 3.41** Pengaturan Pengenalan Alamat *Wellpro Module I/O*.

6. Kemudian pada perangkat *Wellpro Module I/O*, parameter *MODBUS* yang dibaca menggunakan beberapa tipe *MODBUS point* sebagai berikut.
  - a. *MODBUS Digital Input Point* :  
Digunakan untuk membaca nilai masukan digital.
  - b. *MODBUS Digital Output Point* :  
Digunakan untuk memberi 2 status keluaran digital.

Semua parameter *MODBUS Wellpro* baik *register number*, *register type*, *function code* dapat dilihat melalui lampiran 5.
7. Kemudian membuat masing – masing *MODBUS Point*. Klik kanan ‘*Wellpro Modul Input Output*’ pilih *New* → *MODBUS Point* untuk membuat parameter – parameter *MODBUS Wellpro Module I/O* yang akan dibaca dan dikontrol.
8. Pada jendela *MODBUS Point*, pilih ‘*MODBUS Digital Input Point*’. Isi kolom name dengan nama ‘*Digital Input 1*’. Klik OK

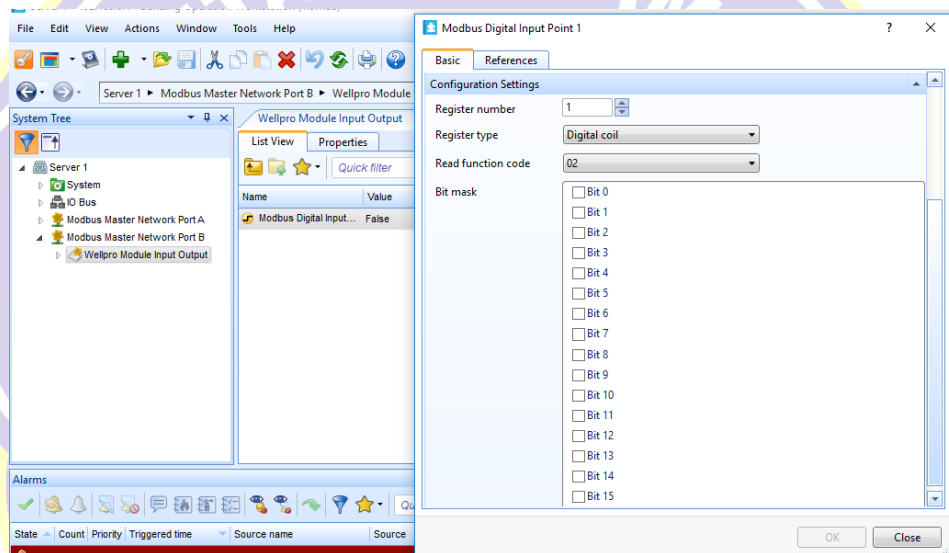
9. Setelah *MODBUS point Digital Input 1* selesai dibuat, klik kanan pilih *properties*, Pilih tab *Basic*, pada menu *Configuration Setting* isi kolom sebagai berikut.

*Register Number* : 1

*Register Type* : Menganalisa satu persatu

*Read Function Code* : 2

*Register Number* diatas digunakan untuk melihat nilai *digital input 1* pada *Wellpro Module I/O*. Konfigurasi *MODBUS Point Digital Input 1* dapat dilihat pada gambar dibawah ini.



**Gambar 3.42** Konfigurasi *MODBUS Point Digital Input 1*.

10. Setelah selesai klik *OK*. Untuk menambah *MODBUS Point* yang lainnya, Klik kanan '*Wellpro Modul Input Output*' pilih *New* → *MODBUS Point*
11. Pada jendela *MODBUS Point*, pilih '*MODBUS Digital Output Point*'. Isi kolom *name* dengan nama '*Digital Output 1*'. Klik *OK*

12. Setelah *MODBUS point Digital Output 1* selesai dibuat, klik kanan pilih *properties*, Pilih tab *Basic*, pada menu *Configuration Setting* isi kolom sebagai berikut.

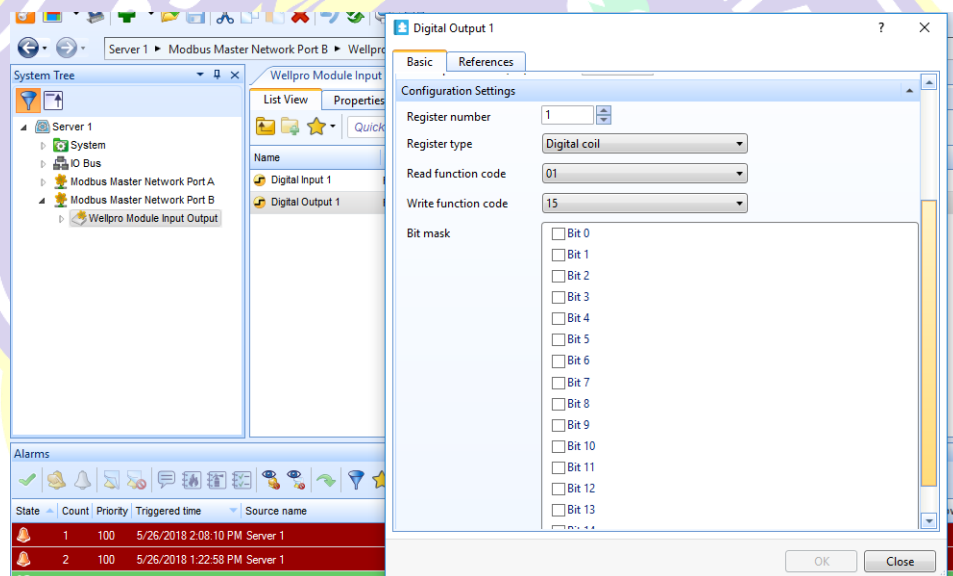
*Register Number* : 1

*Register Type* : Menganalisa satu persatu

*Read Function Code* : 1

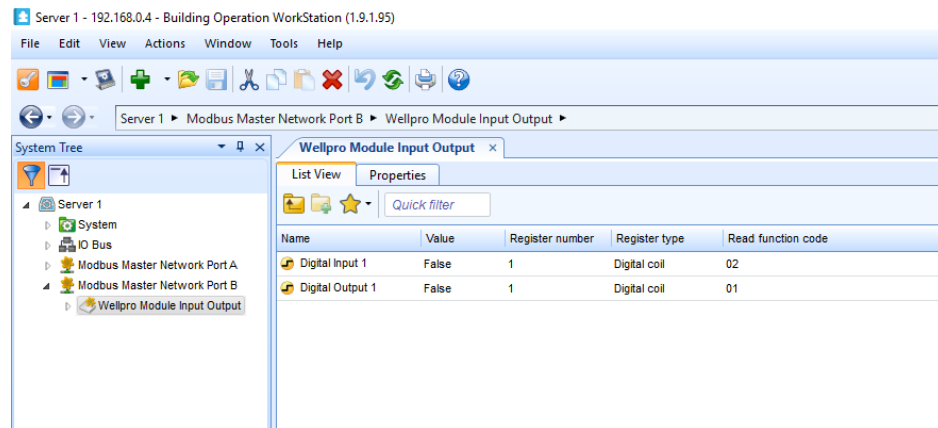
*Write Function Code* : 15

*Register Number* diatas digunakan untuk melihat dan mengendalikan nilai *digital output 1* pada *Wellpro Module I/O*. Konfigurasi *MODBUS Point Digital Output 1* dapat dilihat pada gambar dibawah ini.



**Gambar 3.43** Konfigurasi *MODBUS Point Digital Output 1*.

13. Setelah selesai klik *OK*. Kemudian melihat semua *variable MODBUS* yang telah dibuat untuk perangkat *Wellpro Module I/O* seperti pada gambar dibawah ini.



Name	Value	Register number	Register type	Read function code
Digital Input 1	False	1	Digital coil	02
Digital Output 1	False	1	Digital coil	01

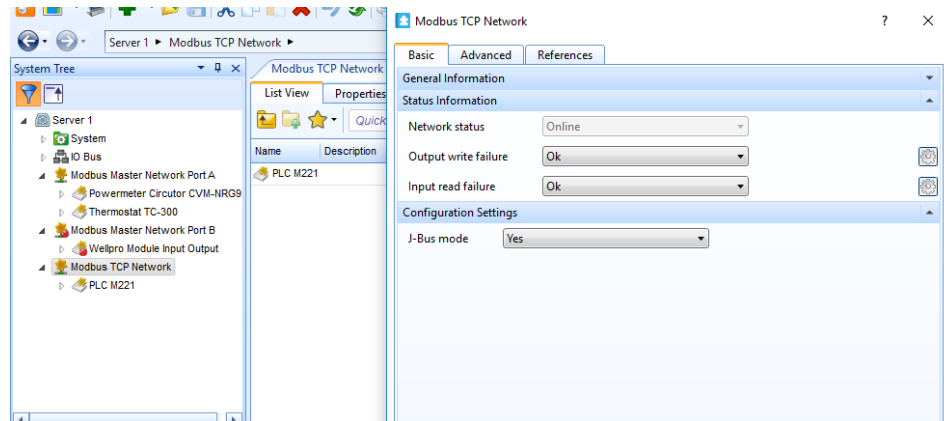
**Gambar 3.44** Tampilan Nilai – Nilai *MODBUS Point Wellpro Module I/O*.

### 3.2.2.3 Pembuatan *Interface MODBUS TCP Network*

Pada pembuatan *Interface MODBUS TCP network*, perangkat keras yang terhubung dengan *Automation Server Port Ethernet* adalah PLC M221. Berikut langkah – langkah pembuatan *Interface MODBUS TCP network port ethernet*.

1. Pada halaman kerja (*workspace*), panel *system tree*, klik kanan pada ‘*Server 1*’ pilih *New* → *Interface* untuk membuat koneksi dengan perangkat *MODBUS*. Pada jendela *Interface*, pilih *MODBUS TCP Network*. Ketik pada kolom *name* : ‘*MODBUS TCP Network Port Ethernet*’. Kemudian *Create*.
2. Klik kanan pada ‘*MODBUS TCP Network Port Ethernet*’, pilih *Properties*. Pada tab *Basic*, menu *Configuration Settings*, aktifkan *JBUS Mode*. *Configuration Setting MODBUS TCP Network Port Ethernet* dapat dilihat pada gambar dibawah ini.





**Gambar 3.45** Configuration Setting MODBUS TCP Network Port Ethernet

3. Kemudian pada panel *system tree* klik kanan 'MODBUS TCP Network Port Ethernet' yang sudah di *create*, pilih *New* → *MODBUS Device*.
4. Pada jendela *create MODBUS device*, isi kolom name 'PLC M221'. Kemudian klik tombol *Create*.

*Variable MODBUS* perangkat PLC M221 yang penulis gunakan untuk dibaca hanya 8 *variable MODBUS* yaitu *Digital Output 1* sampai dengan *Digital Output 8*.

5. Pada panel *system tree* klik kanan pada 'PLC M221' pilih *properties*. Pilih tab *Basic*, pada *Configuration Settings* isi kolom dengan nilai sebagai berikut.

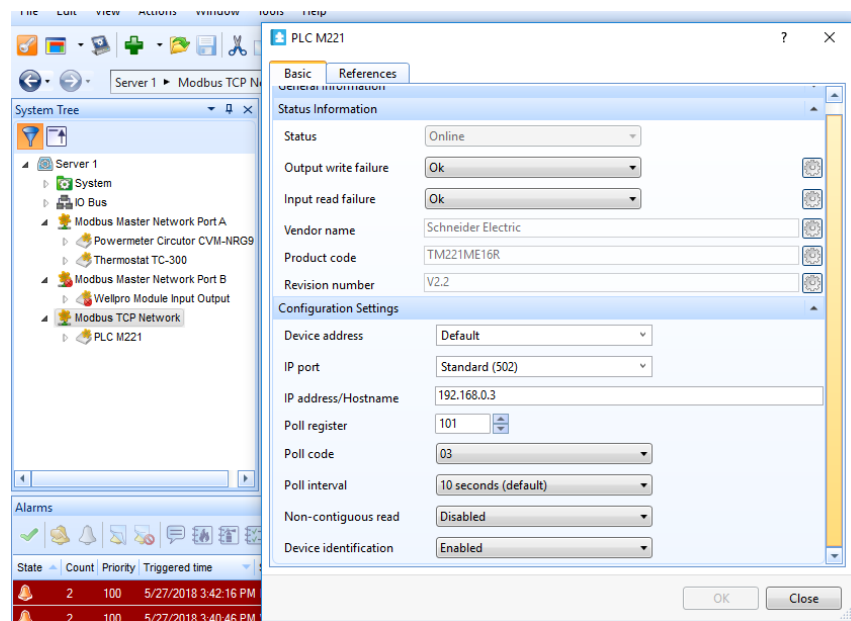
*Device Address* : Default

*IP Port* : Standard (502)

*IP Address/Hostname* : 192.168.0.3

Klik *OK*. Periksa apakah status perangkat PLC M221 dengan *Automation Server* sudah *Online* dengan kembali ke menu *Properties* kolom *Status Information* seperti pada gambar dibawah ini.





**Gambar 3.46** Pengaturan pengenalan alamat PLC M221

6. Kemudian pada perangkat PLC M221 parameter *MODBUS* yang dibaca hanya menggunakan *MODBUS Digital Output Point*.  
Semua parameter *MODBUS* baik *register number*, *register type*, *function code* dapat dilihat melalui halaman lampiran 6.
7. Kemudian membuat masing – masing *MODBUS Point*. Klik kanan ‘PLC M221’ pilih *New* → *MODBUS Point* untuk membuat parameter – parameter *MODBUS* PLC M221 yang akan dibaca dan dikontrol.
8. Pada jendela *MODBUS Point*, pilih ‘*MODBUS Digital Output Point*’. Isi kolom *name* dengan nama ‘*Digital Output 1*’. Klik *OK*
9. Setelah *MODBUS point* selesai dibuat, klik kanan pilih *properties*, Pilih tab *Basic*, pada menu *Configuration Setting* isi kolom sebagai berikut.

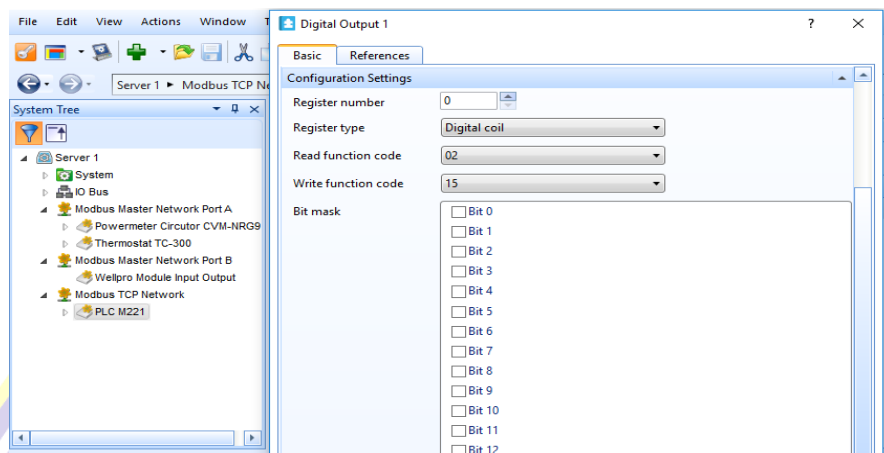
*Register Number* : 0

*Register Type* : Menganalisa satu persatu

*Read Function Code* : 2

*Write Function Code* : 15

*Register Number* diatas untuk memberi nilai keluaran digital pada terminal Q0.0 pada PLC M221. Konfigurasi *MODBUS Point Digital Output 1* (Q0.0) dapat dilihat pada gambar dibawah ini.



**Gambar 3.47** Konfigurasi *MODBUS Point Digital Output 1* (Q0.0).

10. Setelah selesai klik *OK*. Kemudian menambahkan *MODBUS Point Digital Output 2* sampai dengan 8 untuk keluaran digital terminal Q0.1 sampai Q0.7 dengan cara yang sama. *Register Number*, *Register Type*, *Read/Write Function Code* disesuaikan dengan data pada lampiran 6.
11. Setelah semua *MODBUS Point Digital Output* dibuat, kemudian melihat semua *variable MODBUS* yang telah dibuat untuk perangkat PLC M221 seperti yang ada pada gambar dibawah ini.

Name	Value	Register number	Register type	Read function code
Digital Output 1	False	0	Digital coil	02
Digital Output 2	False	1	Digital coil	02
Digital Output 3	False	2	Digital coil	02
Digital Output 4	False	3	Digital coil	02
Digital Output 5	False	4	Digital coil	02
Digital Output 6	False	5	Digital coil	02
Digital Output 7	False	6	Digital coil	02
Digital Output 8	False	7	Digital coil	02

**Gambar 3.48** Tampilan Nilai – Nilai *MODBUS Point* PLC M221

### 3.2.2.4 Pemrograman dan Pembuatan Grafis

Pembuatan program dan grafis pada simulasi ini ditujukan untuk membaca parameter listrik, proteksi jaringan listrik, dan *remote* pada peralatan – peralatan listrik dalam sebuah antarmuka yang terintegrasi. Nilai data yang digunakan oleh penulis sebagai parameter *input* tidak mengikuti standar nasional maupun internasional dikarenakan pengujian ini bersifat simulasi.

#### 1. Pembuatan Program Pengontrolan dan Pengawasan Multi Perangkat *MODBUS*.

Sebelum melakukan pemrograman pengontrolan dan pengawasan, ada beberapa tipe masukan dan keluaran dalam bahasa pemrograman. Untuk masukan dan keluaran digital disebut dengan bilangan biner (*Binary*) sedangkan untuk masukan dan keluaran analog disebut dengan bilangan nyata (*Real*). Dalam pembuatan program untuk antarmuka, *input* dan *output* dapat dilihat pada table dibawah ini.

Tabel 3.5 Pengalamatan *Input/Output* Pada Program Antarmuka

No	Nama	Tipe	Alamat Program	Perangkat
1	<i>Current (A L1)</i>	<i>Real Input</i>	<i>Analog_Input_1</i>	<i>Power Meter</i>
2	<i>Voltage (V L1)</i>	<i>Real Input</i>	<i>Analog_Input_2</i>	<i>Power Meter</i>
3	<i>Room Temperature (°C)</i>	<i>Real Input</i>	<i>Analog_Input_3</i>	<i>Thermostat</i>
4	Status Relay Proteksi Over Current	<i>Binary Input</i>	<i>Digital_Input_1</i>	Wellpro
5	Relay Proteksi Over Current	<i>Binary Output</i>	<i>Digital_Out_1</i>	PLC M221
6	Alarm Suhu Diatas 25°C	<i>Binary Output</i>	<i>Digital_Out_2</i>	PLC M221
7	Alarm Suhu Dibawah 18°C	<i>Binary Output</i>	<i>Digital_Out_3</i>	PLC M221
8	<i>Motor Booster Pump Water 1</i>	<i>Binary Output</i>	<i>Digital_Out_4</i>	PLC M221

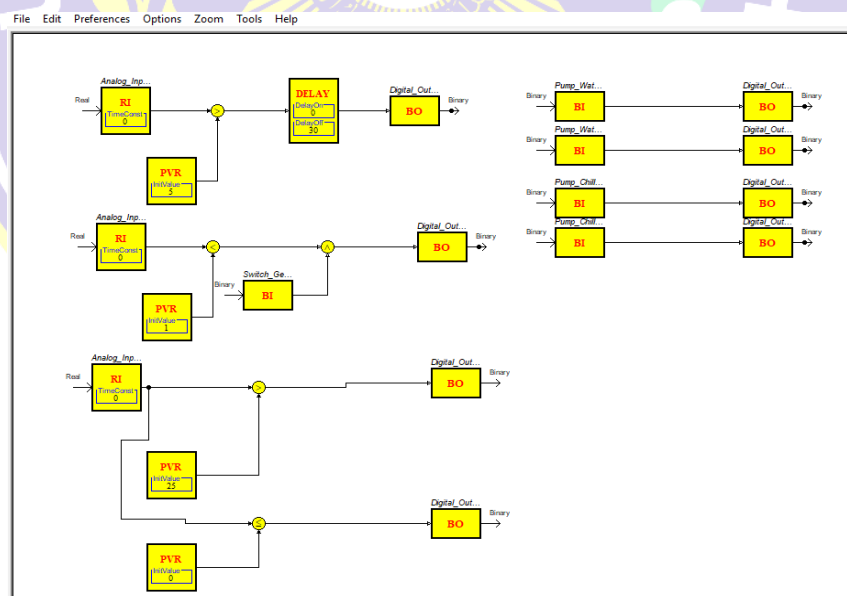
No	Nama	Tipe	Alamat Program	Perangkat
9	<i>Motor Booster Pump Water 2</i>	<i>Binary Output</i>	<i>Digital_Out_5</i>	PLC M221
10	<i>Motor Booster Pump Chiller 1</i>	<i>Binary Output</i>	<i>Digital_Out_6</i>	PLC M221
11	<i>Motor Booster Pump Chiller 2</i>	<i>Binary Output</i>	<i>Digital_Out_7</i>	PLC M221
12	<i>Starter Generator Set</i>	<i>Binary Output</i>	<i>Digital_Out_8</i>	PLC M221
13	<i>Switch Generator Set</i>	<i>Binary Input</i>	<i>Switch_Generator</i>	Grafis
14	<i>Switch Motor Booster Pump Water 1</i>	<i>Binary Input</i>	<i>Pump_Water_1</i>	Grafis
15	<i>Switch Motor Booster Pump Water 2</i>	<i>Binary Input</i>	<i>Pump_Water_2</i>	Grafis
16	<i>Switch Motor Booster Pump Chiller 1</i>	<i>Binary Input</i>	<i>Pump_Chiller_1</i>	Grafis
17	<i>Switch Motor Booster Pump Chiller 2</i>	<i>Binary Input</i>	<i>Pump_Chiller_2</i>	Grafis
18	PVR 1 ( <i>value = 5</i> )	<i>Real Input</i>	-	Grafis
19	PVR 2 ( <i>value = 150</i> )	<i>Real Input</i>	-	Grafis
20	PVR 3 ( <i>value = 25</i> )	<i>Real Input</i>	-	Grafis
21	PVR 4 ( <i>value = 18</i> )	<i>Real Input</i>	-	Grafis

Table 3.6 Diagram Blok dan *Function* pada program.

No	<i>Input</i>	Function 1	Function 2	Output
1	<i>Analog_Input_1</i>	<i>Greater Than PVR 1</i>	<i>Off Delay 30 detik</i>	<i>Digital_Out_1</i>
2	<i>Analog_Input_2</i>	<i>Less Than PVR 2</i>	<i>AND Switch_Generator</i>	<i>Digital_Out_8</i>
3	<i>Analog_Input_3</i>	<i>Greater Than PVR 3</i>	<i>Line</i>	<i>Digital_Out_2</i>
4	<i>Analog_Input_4</i>	<i>Less Than PVR 4</i>	<i>Line</i>	<i>Digital_Out_3</i>
5	<i>Pump_Water_1</i>	<i>Line</i>	<i>Line</i>	<i>Digital_Out_4</i>
6	<i>Pump_Water_2</i>	<i>Line</i>	<i>Line</i>	<i>Digital_Out_5</i>
7	<i>Pump_Chiller_1</i>	<i>Line</i>	<i>Line</i>	<i>Digital_Out_6</i>
8	<i>Pump_Chiller_2</i>	<i>Line</i>	<i>Line</i>	<i>Digital_Out_7</i>

Berikut langkah – langkah pemrograman pengontrolan perangkat *MODBUS*.

1. Pada halaman kerja (*workspace*), panel *system tree*, klik kanan pada ‘*Server 1*’ pilih *New* → *Program* untuk membuat koneksi dengan perangkat *MODBUS*.
2. Pada jendela pemilihan tipe program yang akan digunakan, pilih *Function Block*. Pada kolom name isi dengan nama ‘*Program Pengontrolan Perangkat MODBUS*’. Kemudian *Create*.
3. Kemudian klik kanan pada program yang telah dibuat. Pilih *Edit*.
4. Maka akan terbuka perangkat lunak tambahan yaitu *Function Block Editor* yang merupakan fasilitas dari perangkat lunak *SBO Workstation* untuk melakukan pembuatan program dalam bentuk bahasa program *Function Block*.
5. Membuat program seperti pada gambar dibawah ini dengan merujuk data pada table 3.5 dan 3.6.



**Gambar 3.49** Program Kontrol *Multi MODBUS Device*

6. Setelah program selesai dibuat klik ‘*Save To Server*’.



## 2. Pembuatan Grafis Pengontrolan dan Pengawasan Multi Perangkat *MODBUS*

Pembuatan grafis bertujuan untuk mengawasi dan mengontrol perangkat – perangkat dalam satu *layer* agar memudahkan insinyur ataupun operator dalam melakukan aktivitasnya pada perangkat *MODBUS* yang terhubung.

Berikut langkah – langkah pembuatan grafis simulator kontrol multi perangkat *MODBUS*.

1. Pada halaman kerja (*workspace*), panel *system tree*, klik kanan pada ‘*Server 1*’ pilih *New* → *Graphic* untuk membuat koneksi dengan perangkat *MODBUS*.
2. Kemudian jendela grafis terbuka, pada kolom *name* isi dengan nama ‘*Grafis Pengontrolan dan Pengawasan Perangkat MODBUS*’. Kemudian *Create*.
3. Kemudian klik kanan pada grafis yang telah dibuat. Pilih *Edit*.
4. Maka akan terbuka perangkat lunak tambahan yaitu *Graphics Editor* yang merupakan fasilitas dari perangkat lunak *SBO Workstation* untuk melakukan pembuatan grafis. Didalam perangkat lunak tersebut, komponen – komponen grafis memiliki *bindname* yaitu fitur yang digunakan untuk memetakan nilai – nilai pada perangkat *MODBUS* kedalam objek grafis. Berikut tabel pengalamatan *bindname* pada komponen grafis.

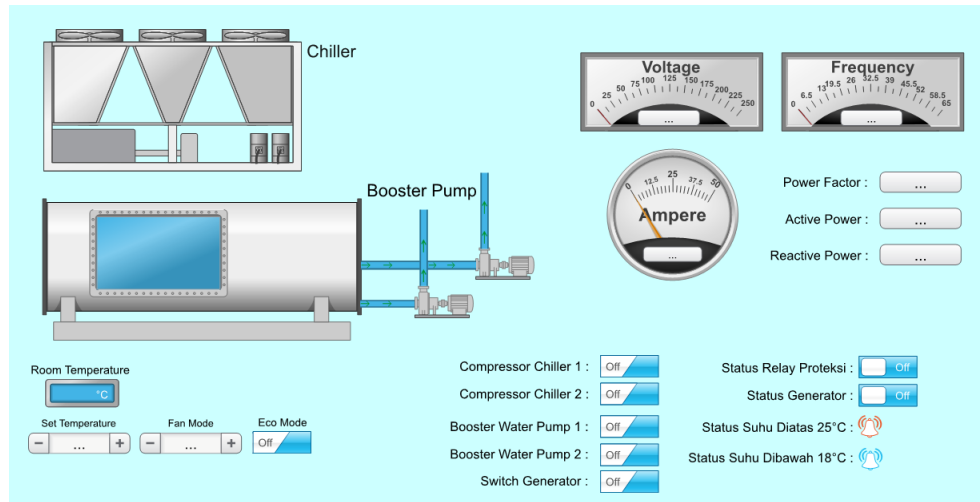
Tabel 3.7 Pengalamatan *bindname* pada komponen – komponen grafis.

No	Komponen	<i>Bindname</i>
1	<i>AC Chiller – 2 Compressor</i>	1. <i>Motor Chiller 1</i> 2. <i>Motor Chiller 2</i>
2	<i>Pump T2 Large Flow Right 1</i>	<i>Booster Pump 1</i>
3	<i>Pump T2 Large Flow Right 2</i>	<i>Booster Pump 2</i>
4	<i>DP Flow Right 1</i>	<i>Booster Pump 1</i>



No	Komponen	Bindname
5	<i>DP Flow Right 2</i>	<i>Booster Pump 2</i>
6	<i>DP Flow Up 1</i>	<i>Booster Pump 1</i>
7	<i>DP Flow Up 2</i>	<i>Booster Pump 2</i>
8	<i>Dial (Voltage)</i>	<i>Voltage</i>
9	<i>Dial (Frequency)</i>	<i>Frequency</i>
10	<i>Dial (Current)</i>	<i>Current</i>
11	<i>Analog Value 1</i>	<i>Voltage</i>
12	<i>Analog Value 2</i>	<i>Frequency</i>
13	<i>Analog Value 3</i>	<i>Current</i>
14	<i>Display</i>	<i>Room Temperature</i>
15	<i>Display</i>	<i>Power Factor</i>
16	<i>Display</i>	<i>Active Power</i>
17	<i>Display</i>	<i>Reactive Power</i>
18	<i>Slide Button Android Style 1</i>	<i>Button Comperssor Chiller_1</i>
19	<i>Slide Button Android Style 2</i>	<i>Button Comperssor Chiller_2</i>
20	<i>Slide Button Android Style 3</i>	<i>Button Booster Pump_1</i>
21	<i>Slide Button Android Style 4</i>	<i>Button Booster Pump_2</i>
22	<i>Slide Button Android Style 5</i>	<i>Button Generator</i>
23	<i>Slide Button Android Style 6</i>	<i>Button Eco Mode</i>
24	<i>Analog Set Point 1</i>	<i>Set_Temperature</i>
25	<i>Analog Set Point 2</i>	<i>Set_FanMode</i>
26	<i>Slide Button SBO Style 1</i>	Status Relay Proteksi
27	<i>Slide Button SBO Style 2</i>	Status Generator
28	<i>Alarm Bell Red</i>	<i>Temperature25</i>
29	<i>Alarm Bell Blue</i>	<i>Temperature18</i>

5. Membuat grafis seperti dibawah ini dengan merujuk data pada tabel 3.6.



**Gambar 3.50** Grafis Kontrol *Multi MODBUS Device*

6. Setelah grafis selesai dibuat klik 'Save To Server'.

### 3.2.2.5 Proses *Binding* Pada *Input/Output Program* dan *Komponen Grafis*.

Proses *Binding* digunakan untuk mengikat suatu nilai pada masukan dan keluaran perangkat *MODBUS* terhadap komponen program dan komponen grafis.

Berikut proses *Binding* nilai – nilai pada perangkat *MODBUS* kedalam komponen program dan komponen grafis.

A. *Binding* komponen program dengan perangkat *MODBUS* dan komponen Grafis.

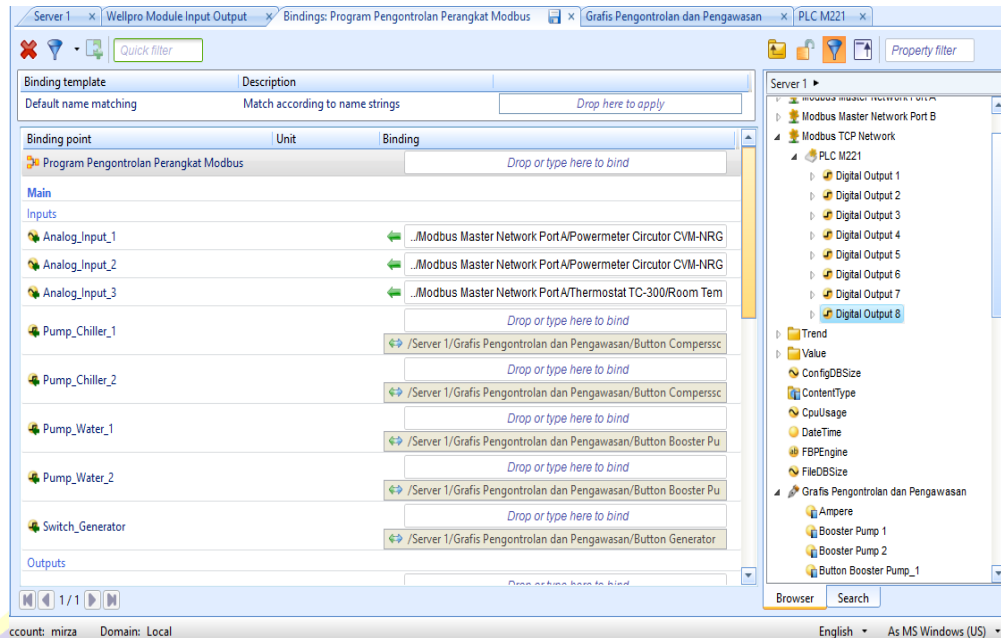
1. Pada halaman kerja (*workspace*), pada 'Server 1' pilih 'Program Pengontrolan Perangkat *MODBUS*', klik kanan, Pilih 'Edit bindings'.
2. Kemudian jendela *binding* untuk program akan tampil. *Drop* atau tarik nilai perangkat *MODBUS* yang berada pada jendela *Browser* sesuai dengan data pada tabel dibawah ini.

Tabel 3.8 Pengalamanan *Binding* komponen program dengan perangkat *MODBUS* dan komponen grafis.

No	<i>Binding Point</i>	<i>Binding</i>
1	<i>Analog_Input_1</i>	<i>Power Meter – Current (A L1) - Value</i>
2	<i>Analog_Input_2</i>	<i>Power Meter – Voltage (V L1) - Value</i>
3	<i>Analog_Input_3</i>	<i>Thermostat – Room Temperature - Value</i>
4	<i>Pump_Chiller_1</i>	Grafis Pengontrolan – <i>Button Compressor Chiller_1</i>
5	<i>Pump_Chiller_2</i>	Grafis Pengontrolan – <i>Button Compressor Chiller_2</i>
6	<i>Pump_Water_1</i>	Grafis Pengontrolan – <i>Button Booster Pump_1</i>
7	<i>Pump_Water_2</i>	Grafis Pengontrolan – <i>Button Booster Pump_2</i>
8	<i>Switch_Generator</i>	Grafis Pengontrolan – <i>Button Generator</i>
9	<i>Digital_Out_1</i>	1. PLC M221 – <i>Digital Output 1 – Value</i> 2. Wellpro Module I/O – <i>Digital Output 1 - Value</i>
10	<i>Digital_Out_2</i>	1. PLC M221 – <i>Digital Output 2 – Value</i> 2. Grafis Pengontrolan – <i>Temperature25</i>
11	<i>Digital_Out_3</i>	1. PLC M221 – <i>Digital Output 3 – Value</i> 2. Grafis Pengontrolan – <i>Temperature18</i>
12	<i>Digital_Out_4</i>	PLC M221 – <i>Digital Output 4 - Value</i>
13	<i>Digital_Out_5</i>	PLC M221 – <i>Digital Output 5 - Value</i>
14	<i>Digital_Out_6</i>	PLC M221 – <i>Digital Output 6 - Value</i>
15	<i>Digital_Out_7</i>	PLC M221 – <i>Digital Output 7 - Value</i>
16	<i>Digital_Out_8</i>	PLC M221 – <i>Digital Output 8 - Value</i>

3. Setelah selesai melakukan *drop binding point* komponen program dengan perangkat *MODBUS* dan komponen grafis. Klik *Save* untuk menyimpan. Selesai.

Tampilan *Binding* Komponen Program dapat dilihat pada gambar dibawah ini.



**Gambar 3.51** Tampilan *Binding* Komponen Program

B. *Binding* komponen grafis dengan perangkat *MODBUS* dan komponen program.

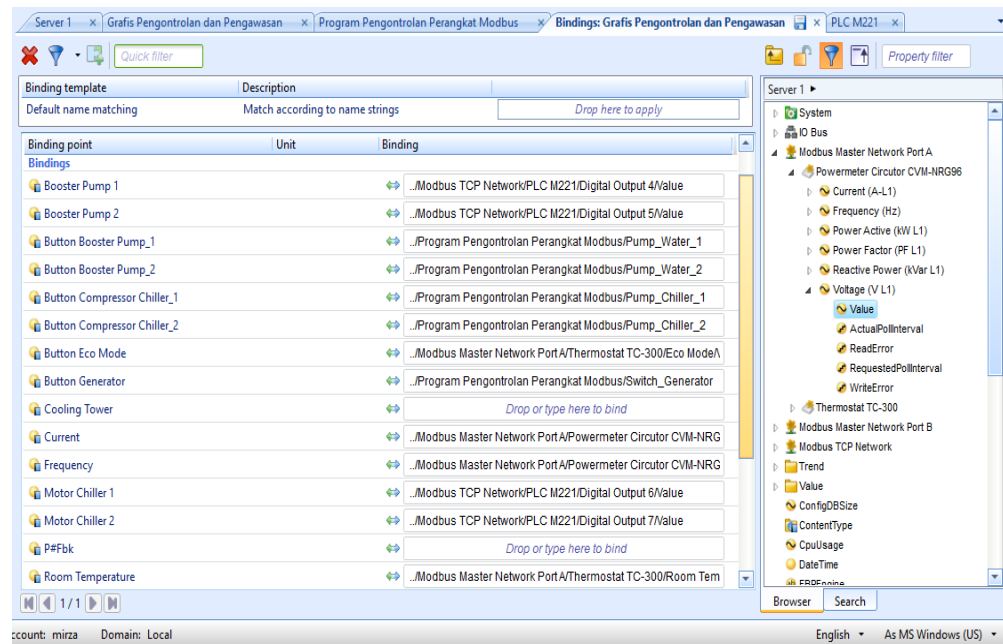
1. Pada halaman kerja (*workspace*), pada 'Server 1' pilih 'Grafis Pengawasan dan Pengontrolan' yang telah dibuat sebelumnya, kemudian klik kanan, Pilih 'Edit bindings'.
2. Kemudian jendela *binding* untuk program akan tampil. *Drop* atau tarik nilai perangkat *MODBUS* yang berada pada jendela *Browser* sesuai dengan data pada tabel dibawah ini.

Tabel 3.9 Pengalamanan *Binding* komponen grafis dengan perangkat *MODBUS* dan komponen program.

No	<i>Binding Point</i>	<i>Binding</i>
1	<i>Active Power</i>	<i>Power Meter – Active Power - Value</i>
2	<i>Booster Pump 1</i>	PLC M221 – <i>Digital Output 4 - Value</i>
3	<i>Booster Pump 2</i>	PLC M221 – <i>Digital Output 5 - Value</i>
4	<i>Button Booster Pump_1</i>	Program Pengontrolan – <i>Pump_Water_1</i>
5	<i>Button Booster Pump_2</i>	Program Pengontrolan – <i>Pump_Water_2</i>
6	<i>Button Compressor Chiller_1</i>	Program Pengontrolan – <i>Pump_Chiller_1</i>
7	<i>Button Compressor Chiller_2</i>	Program Pengontrolan – <i>Pump_Chiller_2</i>
8	<i>Button Eco Mode</i>	<i>Thermostat – Eco Mode - Value</i>
9	<i>Button Generator</i>	Program Pengontrolan – <i>Switch_Generator</i>
10	<i>Current</i>	<i>Power Meter – Current (A L1) - Value</i>
11	<i>Frequency</i>	<i>Power Meter – Frequency (Hz) - Value</i>
12	<i>Motor Chiller 1</i>	PLC M221 – <i>Digital Output 6 - Value</i>
13	<i>Motor Chiller 2</i>	PLC M221 – <i>Digital Output 7 - Value</i>
14	<i>Power Factor</i>	<i>Power Meter – Power Factor - Value</i>
15	<i>Reactive Power</i>	<i>Power Meter – Reactive Power - Value</i>
16	<i>Room Temperature</i>	<i>Thermostat – Room Temperature - Value</i>
17	<i>Set_FanMode</i>	<i>Thermostat – Fan Mode - Value</i>
18	<i>Set_Temperature</i>	<i>Thermostat – Set Temperature - Value</i>
19	<i>Status Generator</i>	PLC M221 – <i>Digital Output 8 - Value</i>
20	<i>Status Relay Proteksi</i>	<i>Wellpro – Digital Input 1 - Value</i>
21	<i>Temperature18</i>	Program Pengontrolan – <i>Digital_Out_3</i>
22	<i>Temperature25</i>	Program Pengontrolan – <i>Digital_Out_2</i>
23	<i>Voltage</i>	<i>Power Meter – Voltage (V L1) - Value</i>

- Setelah selesai melakukan *drop binding point* komponen grafis dengan perangkat *MODBUS* dan komponen program. Klik *Save* untuk menyimpan. Selesai.

Tampilan *Binding* Komponen Grafis dapat dilihat pada gambar dibawah ini.



**Gambar 3.52** Tampilan *Binding* Komponen Grafis





## BAB IV

### ANALISA DAN PENGUJIAN

Pada bab ini akan dibahas analisa dan pengujian kekuatan sinyal, konfigurasi, program dan grafis untuk melakukan pengawasan dan pengendalian nilai – nilai masukan dan keluaran perangkat – perangkat *MODBUS* yang terhubung (*Online*) dengan *Automation Server* melalui PC yang memiliki koneksi dengan perangkat *TP-Link* secara nirkabel (*Wireless*) dalam mode jaringan *Local Area Network*. Hasil dari pengujian yang dilakukan akan dianalisa sesuai dengan tolak ukur yang ada. Berdasarkan data – data hasil pengujian dari simulasi ini, maka dapat diambil suatu kesimpulan dari sistem alat tugas akhir yang telah dibuat.

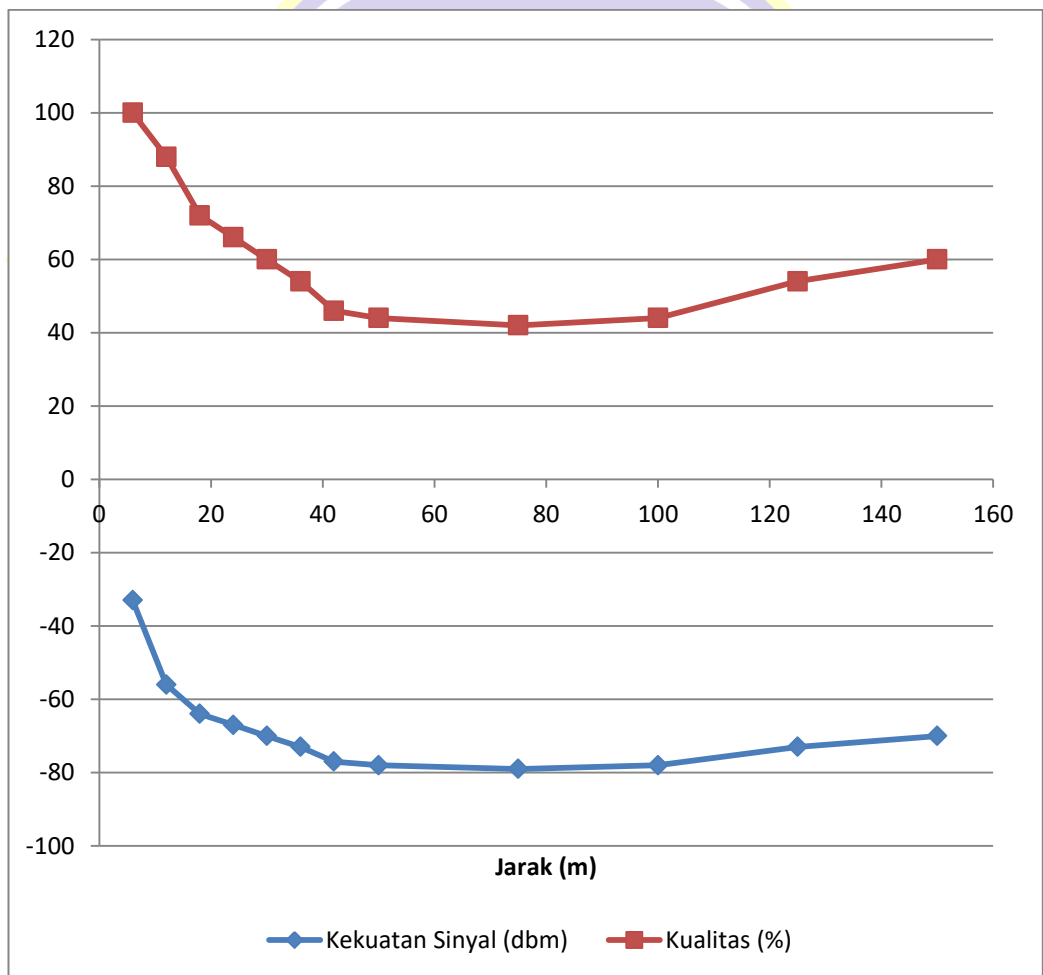
#### 4.1 Pengujian Kekuatan Sinyal WLAN (*Wireless Local Area Network*)

Pengujian kekuatan sinyal dilakukan untuk mengetahui kekuatan sinyal nirkabel antara *smart device* dan *TP-Link* untuk melakukan pengendalian dan pengawasan multi *MODBUS Device* yang terhubung dengan *Automation Server*. Pengujian dilakukan menggunakan perangkat lunak *Lizard Wifi Scanner* pada PC dan *Netgear Wifi Analytics* pada *smartphone*, pengujian dilakukan pada kondisi minim hambatan. Hasil pengujian dapat dilihat pada tabel dan gambar grafik dibawah ini.

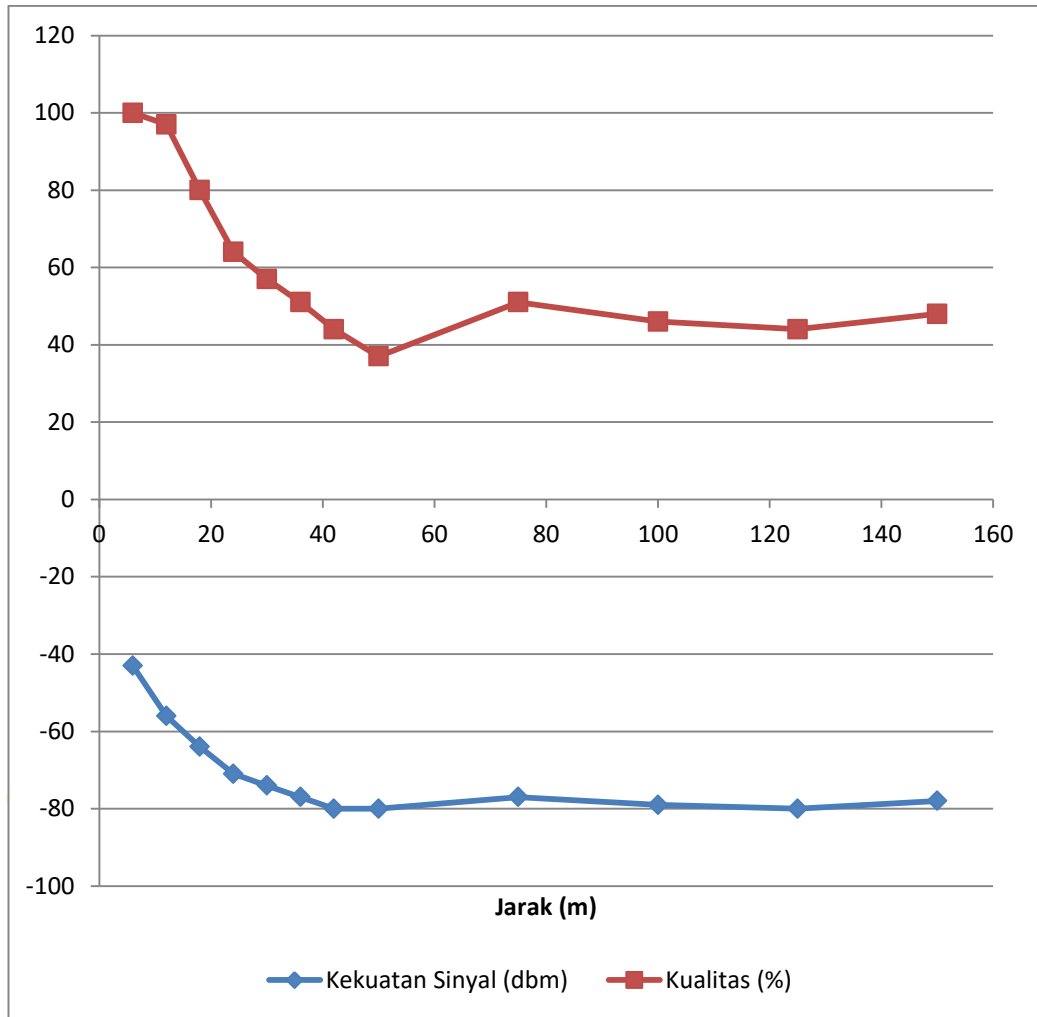
Tabel 4.1 Hasil kekuatan sinyal WLAN

Jarak(meter)	<i>Personal Computer</i>		<i>Smartphone</i>	
	Kekuatan sinyal (dbm)	Kualitas (%)	Kekuatan sinyal (dbm)	Kualitas (%)
6	-33	100	-43	100
12	-56	88	-56	97
18	-64	72	-64	80

Jarak(meter)	Personal Computer		Smartphone	
	Kekuatan sinyal (dbm)	Kualitas (%)	Kekuatan sinyal (dbm)	Kualitas (%)
24	-67	66	-71	64
30	-70	60	-74	57
36	-73	54	-77	51
42	-77	46	-80	44
50	-78	44	-80	37
75	-79	42	-77	51
100	-78	44	-79	46
125	-73	54	-80	44
150	-70	60	-78	48



**Gambar 4.1** Grafik kekuatan sinyal dan kualitas sinyal antara laptop dan *TP-Link MR3420*.



**Gambar 4.2** Grafik kekuatan sinyal dan kualitas sinyal antara *Smartphone* dan *TP-Link MR3420*.

Melalui dua gambar grafik diatas, penurunan kekuatan sinyal rata – rata dapat ditentukan pada persamaan umum dibawah ini.

Rata – rata kekuatan sinyal pada jarak 6 meter.

$$\text{Mean (6)} = \frac{((-33) + (-43))}{2} = -38$$

Rata – rata kekuatan sinyal pada jarak 150 meter.

$$\text{Mean (150)} = \frac{((-70) + (-78))}{2} = -74$$

Maka persentase penurunan kekuatan sinyal sebagai berikut.

$$x = \frac{-74}{-38} \times 100 = 194,7$$

Persentase penurunan kekuatan sinyal (%) =  $194,7 - 100 = 94,7\%$

Melalui dua gambar grafik diatas, penurunan kualitas sinyal rata – rata dapat ditentukan pada persamaan umum dibawah ini.

Rata – rata kualitas sinyal pada jarak 6 meter.

$$Mean (6) = \frac{(100 + 100)}{2} = 100$$

Rata – rata kualitas sinyal pada jarak 150 meter.

$$Mean (150) = \frac{(60 + 43)}{2} = 51,5$$

Maka persentase penurunan kualitas sinyal sebagai berikut.

$$x = \frac{51,5}{100} \times 100 = 51,5$$

Persentase penurunan kualitas sinyal (%) =  $100 - 51,5 = 48,5\%$

#### 4.2 Pengujian Konfigurasi Kecepatan *Automation Server* dan *Register Type*

Pengujian konfigurasi dilakukan untuk menguji keberhasilan konfigurasi perangkat terhadap nilai *type register*, batas *poll transmit* dan *receive Modbus Point* masing – masing port komunikasi A dan B serta *Port Ethernet* antara *Master* dan *Slave*. Sebelum melakukan pengujian terhadap konfigurasi yang telah diatur, maka penulis melakukan analisa terhadap *receive timeout* untuk masing – masing port komunikasi agar nilai *Modbus* yang diuji tidak terjadi *crash* (kerusakan).

#### 4.2.1 Analisa kecepatan *Receive Timeout* (ms) pada *Port COM A*

*Receive Timeout* (ms) pada komunikasi *Port COM A* diperlukan untuk perangkat *Master* menyesuaikan nilai pembacaan dan pengendalian sesuai jumlah *Modbus Point* pada masing – masing perangkat.

Dengan mengabaikan *Silence Characters*, *Transmit Guard bits*, *Point poll delay*, dan *Poll duty cycle* dengan nilai standar pabrikan (*default*). Nilai *receive timeout* yang dibutuhkan untuk *Port Com – A* pada *baud rate* 4800 setelah melakukan inisialisasi nilai *Modbus Point* pada perangkat *Modbus* dapat dihitung melalui persamaan (2.1) dan persamaan (2.2) sebagai berikut:

$$T_{3,5} = 3,5 \times \frac{\text{Jumlah bit tiap karakter}}{\text{Baud Rate Com-A}} \dots\dots\dots(2.1)$$

$$T_{3,5} = 3,5 \times \frac{11 \text{ bit}}{4800 \text{ bps}}$$

$$T_{3,5} = 0,008020833333 \text{ s} = 8,02 \text{ ms}$$

*Port COM – A* memiliki 2 *slave* yaitu *Power meter* dan *Thermostat* yang masing – masing memiliki jumlah *Modbus point* sebanyak (6) enam dan (4) empat. Apabila kuantitas karakter sama dengan kuantitas *Modbus point*, maka nilai *receive timeout* dapat dihitung melalui persamaan (2.2) sebagai berikut:

$$\text{Receive Timeout} = T_{3,5} \times \text{Jumlah Karakter} \dots\dots\dots(2.2)$$

$$\text{Receive Timeout} = 8,02 \times ( 6 + 4 )$$

$$\text{Receive Timeout} = 8,02 \times 10 = 80,2 \text{ ms}$$

Dengan hasil perhitungan diatas, maka konfigurasi *Receive Timeout* (ms) pada *Port COM- A* adalah diatas 80,2 ms. Sehingga konfigurasi pada perangkat lunak ditetapkan dengan nilai *Receive Timeout* (ms) = 100.

#### 4.2.2 Analisa kecepatan *Receive Timeout* (ms) pada *Port COM B*

*Receive Timeout* (ms) pada komunikasi *Port COM B* diperlukan untuk perangkat *Master* menyesuaikan nilai pembacaan dan pengendalian sesuai jumlah *Modbus Point* pada masing – masing perangkat.

Dengan mengabaikan *Silence Characters*, *Transmit Guard bits*, *Point poll delay*, dan *Poll duty cycle* dengan nilai standar pabrikan (*default*). Nilai *receive timeout* yang dibutuhkan untuk *Port Com – B* pada *baud rate* 9600 setelah melakukan inisialisasi nilai *Modbus Point* pada perangkat *Modbus* dapat dihitung melalui persamaan (2.1) dan persamaan (2.2) sebagai berikut:

$$T_{3,5} = 3,5 \times \frac{\text{Jumlah bit tiap karakter}}{\text{Baud Rate Com-B}} \dots\dots\dots(2.1)$$

$$T_{3,5} = 3,5 \times \frac{11 \text{ bit}}{9600 \text{ bps}}$$

$$T_{3,5} = 0,00401 \text{ s} = 4,01 \text{ ms}$$

*Port COM – B* memiliki 1 *slave* yaitu *Wellpro Module I/O* yang memiliki jumlah *Modbus point* sebanyak (2) dua . Apabila nilai karakter sama dengan nilai *Modbus point*, maka nilai *receive timeout* sebagai berikut.

$$\text{Receive Timeout} = T_{3,5} \times \text{Jumlah Karakter} \dots\dots\dots(2.2)$$

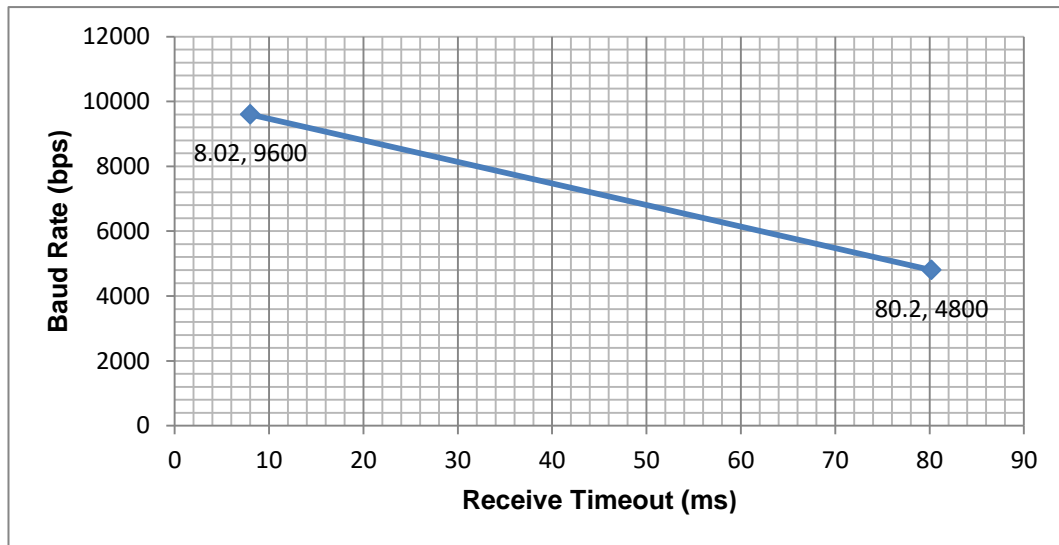
$$\text{Receive Timeout} = 4,01 \times ( 2 ) = 8,02 \text{ ms}$$

Dengan hasil perhitungan diatas, maka konfigurasi *Receive Timeout* (ms) pada *Port COM- B* adalah diatas 8,02 ms. Sehingga konfigurasi pada perangkat lunak ditetapkan dengan nilai *Receive Timeout* (ms) = 100.

Tabel 4.2 Perbandingan *receive timeout* port komunikasi *MODBUS COM-A* dan *MODBUS COM-B*.

Port	Jumlah Karakter	<i>Baud Rate</i> (bps)	<i>Receive Timeout</i> (ms)
COM-A	10	4800	80,2
COM-B	2	9600	8,02





**Gambar 4.3** Grafik Perbandingan *receive timeout* port komunikasi *MODBUS COM-A* dan *MODBUS COM-B*.

#### 4.2.3 Analisa kecepatan *Receive Timeout* (ms) pada *Port Ethernet*

*Receive Timeout* (ms) pada komunikasi *Port Ethernet* diperlukan untuk perangkat *Master* menyesuaikan nilai pembacaan dan pengendalian sesuai jumlah *Modbus Point* pada masing – masing perangkat.

Pada *Modbus TCP* parameter *Baud rate*, *Silence Characters*, *Transmit Guard bits*, *Point poll delay*, dan *Poll duty cycle* tidak ada. Untuk nilai *Baud rate* diubah dengan *transfer rate* standar *Modbus TCP* adalah 10/100 Mbps, tetapi pada Bab III, program PLC untuk nilai *transfer rate* adalah Auto, maka akan ada 2 perhitungan untuk nilai *Receive Timeout* (ms) pada *Port Ethernet*.

Dengan merujuk teori pada Bab II, *Modbus TCP* menggunakan karakter yang sama dengan *Modbus RTU* yaitu karakter CRC dan nilai paket data unit ditambah 7 bytes =56 bit. Nilai *receive timeout* yang dibutuhkan untuk *Port Ethernet* setelah melakukan inialisasi nilai *Modbus Point* pada perangkat *Modbus* dapat dihitung melalui persamaan (2.1) dan persamaan (2.2) sebagai berikut:

A. Pada *transfer rate* 10 mbps.

$$T_{3,5} = 3,5 \times \frac{\text{Jumlah bit tiap karakter} + \text{Packet data unit}}{\text{Transfer Rate}} \dots\dots\dots(2.1)$$

$$T_{3,5} = 3,5 \times \frac{11 \text{ bit} + 56 \text{ bit}}{10 \text{ mbps}}$$

$$T_{3,5} = 0,00002345 \text{ s} = 0,02345 \text{ ms}$$

B. Pada *transfer rate* 100 mbps.

$$T_{3,5} = 3,5 \times \frac{\text{Jumlah bit tiap karakter} + \text{Packet data unit}}{\text{Transfer Rate}} \dots\dots\dots(2.1)$$

$$T_{3,5} = 3,5 \times \frac{11 \text{ bit} + 56 \text{ bit}}{100 \text{ mbps}}$$

$$T_{3,5} = 0,000002345 \text{ s} = 0,002345 \text{ ms}$$

Port Ethernet memiliki 1 *slave* yaitu PLC M221 yang memiliki jumlah *Modbus point* sebanyak (8) delapan . Apabila nilai karakter sama dengan nilai *Modbus point*, maka nilai *receive timeout* sebagai berikut.

A. Pada *transfer rate* 10 mbps.

$$\text{Receive Timeout} = T_{3,5} \times \text{Jumlah Karakter} \dots\dots\dots(2.2)$$

$$\text{Receive Timeout} = 0,02345 \times ( 8 ) = 0,1876 \text{ ms}$$

B. Pada *transfer rate* 100 mbps.

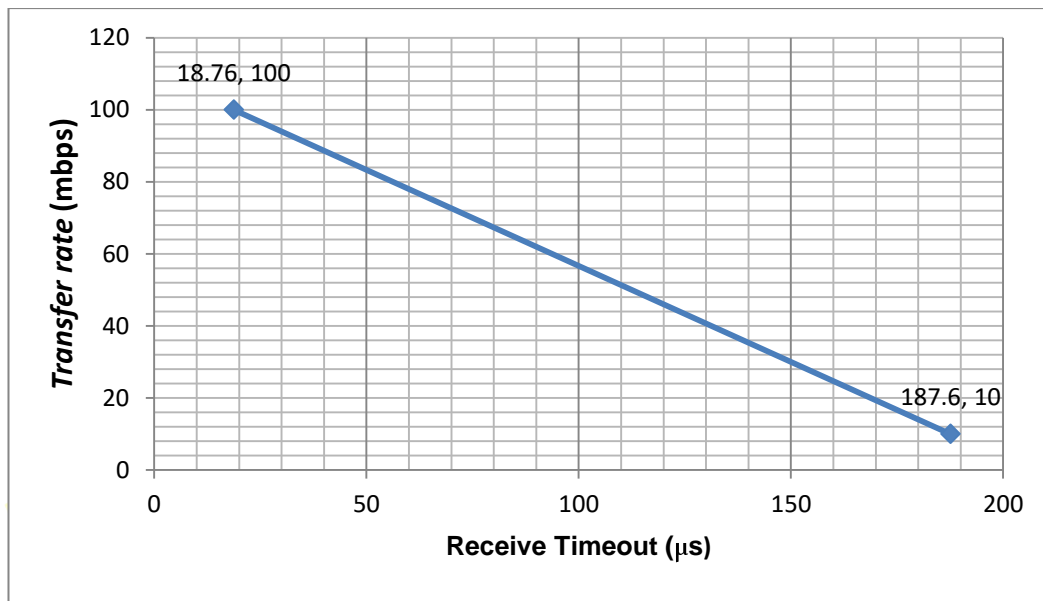
$$\text{Receive Timeout} = T_{3,5} \times \text{Jumlah Karakter} \dots\dots\dots(2.2)$$

$$\text{Receive Timeout} = 0,002345 \times ( 8 ) = 0,01876 \text{ ms}$$

Dengan hasil perhitungan diatas, maka konfigurasi *Receive Timeout* (ms) pada Port Ethernet adalah diatas 0,1876 ms. Sehingga konfigurasi pada perangkat lunak ditetapkan dengan nilai *Receive Timeout* (ms) = 100.

Tabel 4.3 Perbandingan *receive timeout* pada port Ethernet masing – masing *transfer rate*.

<i>Transfer rate</i> (mbps)	Jumlah Karakter	<i>Receive Timeout</i> ( $\mu$ s)
10	8	187,6
100	8	18,76

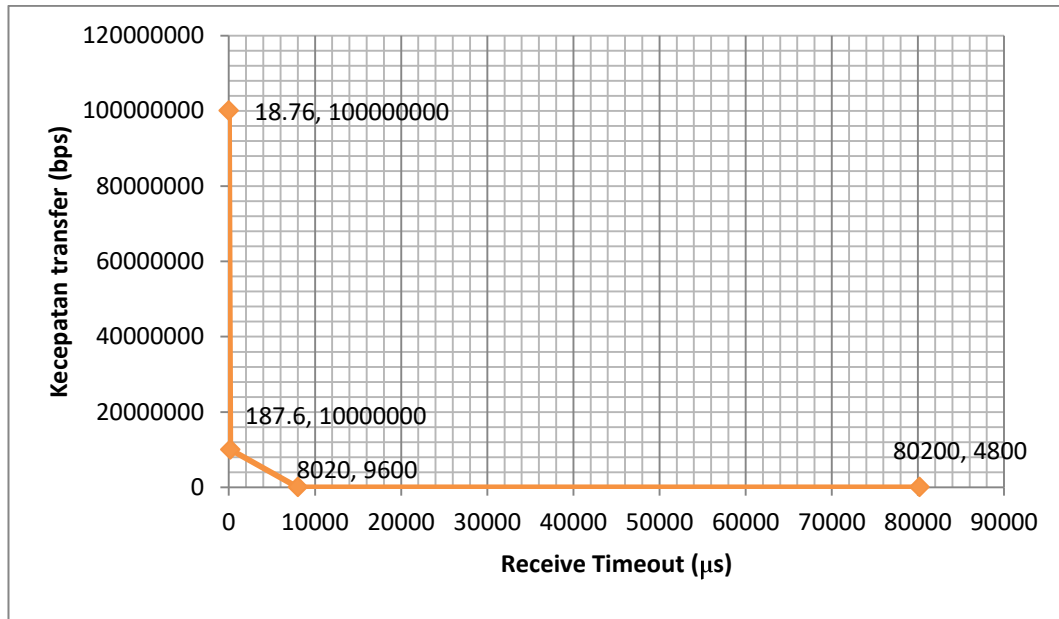


Gambar 4.4 Grafik Perbandingan *receive timeout* pada Port Ethernet masing – masing *transfer rate*.

Setelah melakukan analisa diatas untuk semua port komunikasi *MODBUS COM-A*, *COM-B*, dan *ETHERNET*. Maka perbandingan *receive timeout* pada masing – masing port sesuai kuantitas karakternya dapat dilihat pada tabel dibawah ini.

Tabel 4.4 Perbandingan *receive timeout* pada masing – masing port komunikasi *MODBUS*.

Port	Jumlah karakter	Kecepatan transfer (bps)	<i>Receive timeout</i> ( $\mu$ s)
COM- A	10	4.800	80200
COM- B	2	9.600	8020
Ethernet	8	10.000.000	187,6
	8	100.000.000	18,7



**Gambar 4.5** Grafik Perbandingan *receive timeout* pada masing – masing port komunikasi *MODBUS*.

#### 4.2.4 Pengujian *Register Type* Pada Masing – Masing Perangkat *Modbus*.

Pengujian *register type* dilakukan untuk menguji dan menyesuaikan nilai pembacaan sesuai referensi nilai yang tampil pada perangkat keras *Modbus*. Hasil pengujian masing – masing *register type* pada masing – masing *MODBUS Point Power Meter* dan *Thermostat* dapat dilihat pada tabel dibawah ini.

Tabel 4.5 Pengujian *register type* pada perangkat *Power meter Circutor CVM-NRG96*

<i>Modbus Point</i>	Kode Fungsi	<i>Register Number</i>	Tipe Register	Keterangan Nilai
<i>Active Power (kW)</i>	03	05	<i>16 bit unsigned</i>	Tidak Sesuai
			<i>16 bit signed</i>	Tidak Sesuai
			<i>32 bit unsigned</i>	Tidak Sesuai
			<i>32 bit unsigned swapped</i>	Sesuai
			<i>32 bit signed</i>	Tidak Sesuai
			<i>32 bit signed swapped</i>	Sesuai
			<i>32 bit real</i>	Tidak Sesuai
			<i>32 bit real swapped</i>	Tidak Sesuai
<i>Current (A L1)</i>	03	03	<i>16 bit unsigned</i>	Tidak Sesuai
			<i>16 bit signed</i>	Tidak Sesuai

<i>Modbus Point</i>	<i>Kode Fungsi</i>	<i>Register Number</i>	<i>Type Register</i>	<i>Keterangan Nilai</i>
			<i>32 bit unsigned</i>	Tidak Sesuai
			<i>32 bit unsigned swapped</i>	Sesuai
			<i>32 bit signed</i>	Tidak Sesuai
			<i>32 bit signed swapped</i>	Sesuai
			<i>32 bit real</i>	Tidak Sesuai
			<i>32 bit real swapped</i>	Tidak Sesuai
<i>Frequency (Hz)</i>	03	41	<i>16 bit unsigned</i>	Tidak Sesuai
			<i>16 bit signed</i>	Tidak Sesuai
			<i>32 bit unsigned</i>	Tidak Sesuai
			<i>32 bit unsigned swapped</i>	Sesuai
			<i>32 bit signed</i>	Tidak Sesuai
			<i>32 bit signed swapped</i>	Sesuai
			<i>32 bit real</i>	Tidak Sesuai
<i>32 bit real swapped</i>	Tidak Sesuai			
<i>Power Factor (PF L1)</i>	03	09	<i>16 bit unsigned</i>	Tidak Sesuai
			<i>16 bit signed</i>	Tidak Sesuai
			<i>32 bit unsigned</i>	Tidak Sesuai
			<i>32 bit unsigned swapped</i>	Sesuai
			<i>32 bit signed</i>	Tidak Sesuai
			<i>32 bit signed swapped</i>	Sesuai
			<i>32 bit real</i>	Tidak Sesuai
			<i>32 bit real swapped</i>	Tidak Sesuai
<i>Reactive Power (kVar)</i>	03	07	<i>16 bit unsigned</i>	Tidak Sesuai
			<i>16 bit signed</i>	Tidak Sesuai
			<i>32 bit unsigned</i>	Tidak Sesuai
			<i>32 bit unsigned swapped</i>	Sesuai
			<i>32 bit signed</i>	Tidak Sesuai
			<i>32 bit signed swapped</i>	Sesuai
			<i>32 bit real</i>	Tidak Sesuai
			<i>32 bit real swapped</i>	Tidak Sesuai
<i>Voltage (V L1)</i>	03	01	<i>16 bit unsigned</i>	Tidak Sesuai
			<i>16 bit signed</i>	Tidak Sesuai
			<i>32 bit unsigned</i>	Tidak Sesuai
			<i>32 bit unsigned swapped</i>	Sesuai
			<i>32 bit signed</i>	Tidak Sesuai
			<i>32 bit signed swapped</i>	Sesuai
			<i>32 bit real</i>	Tidak Sesuai
			<i>32 bit real swapped</i>	Tidak Sesuai

Tabel 4.6 Pengujian *register type* pada perangkat *Thermostat TC-300*

<i>Modbus Point</i>	Kode Fungsi ( <i>read</i> )	Kode Fungsi ( <i>write</i> )	<i>Register Number</i>	Tipe <i>register</i>	Keterangan Nilai ( <i>read</i> )	Keterangan Nilai ( <i>write</i> )
<i>Eco Mode</i>	03	06	10	<i>Digital Coil</i>	Tidak Sesuai	Tidak Dapat Dikendalikan
				<i>16 bit unsigned</i>	Sesuai	Dapat Dikendalikan
<i>Fan Mode</i>	03	06	06	<i>16 bit unsigned</i>	Sesuai	Dapat Dikendalikan
				<i>16 bit signed</i>	Sesuai	Dapat Dikendalikan
				<i>32 bit unsigned</i>	Tidak Sesuai	Dapat Dikendalikan
				<i>32 bit unsigned swapped</i>	Tidak Sesuai	Tidak Dapat Dikendalikan
<i>Room Temperature</i>	04	-	01	<i>16 bit unsigned</i>	Sesuai	-
				<i>16 bit signed</i>	Sesuai	-
				<i>32 bit unsigned</i>	Sesuai	-
				<i>32 bit unsigned swapped</i>	Tidak Sesuai	-
<i>Set Temperature</i>	03	06	05	<i>16 bit unsigned</i>	Sesuai	Dapat Dikendalikan
				<i>16 bit signed</i>	Sesuai	Dapat Dikendalikan
				<i>32 bit unsigned</i>	Sesuai	Dapat Dikendalikan
				<i>32 bit unsigned swapped</i>	Tidak Sesuai	Tidak Dapat Dikendalikan



Tabel 4.7 Pengujian *register type* pada perangkat *Wellpro Module I/O*

<i>Modbus Point</i>	Kode Fungsi ( <i>read</i> )	Kode Fungsi ( <i>write</i> )	<i>Register Number</i>	Tipe <i>register</i>	Keterangan Nilai ( <i>read</i> )	Keterangan Nilai ( <i>write</i> )
<i>Digital Input 1</i>	02	None	01	<i>Digital Coil</i>	Sesuai	-
				<i>16 bit unsigned</i>	Sesuai	-
<i>Digital Output 1</i>	01	15	1	<i>Digital Coil</i>	Sesuai	Dapat Dikendalikan
				<i>16 bit unsigned</i>	Sesuai	Tidak Dapat Dikendalikan

Tabel 4.8 Pengujian *register type* pada perangkat PLC M221

<i>Modbus Point</i>	Kode Fungsi ( <i>read</i> )	Kode Fungsi ( <i>write</i> )	<i>Register Number</i>	Tipe <i>register</i>	Keterangan Nilai ( <i>read</i> )	Keterangan Nilai ( <i>write</i> )
<i>Digital Output 1</i>	02	15	00	<i>Digital Coil</i>	Sesuai	Dapat Dikendalikan
				<i>16 bit unsigned</i>	Sesuai	Tidak Dapat Dikendalikan
<i>Digital Output 2</i>	02	15	01	<i>Digital Coil</i>	Sesuai	Dapat Dikendalikan
				<i>16 bit unsigned</i>	Sesuai	Tidak Dapat Dikendalikan
<i>Digital Output 3</i>	02	15	02	<i>Digital Coil</i>	Sesuai	Dapat Dikendalikan
				<i>16 bit unsigned</i>	Sesuai	Tidak Dapat Dikendalikan
<i>Digital Output 4</i>	02	15	03	<i>Digital Coil</i>	Sesuai	Dapat Dikendalikan
				<i>16 bit unsigned</i>	Sesuai	Tidak Dapat Dikendalikan
<i>Digital Output 5</i>	02	15	04	<i>Digital Coil</i>	Sesuai	Dapat Dikendalikan
				<i>16 bit unsigned</i>	Sesuai	Tidak Dapat Dikendalikan
<i>Digital Output 6</i>	02	15	05	<i>Digital Coil</i>	Sesuai	Dapat Dikendalikan

<i>Modbus Point</i>	Kode Fungsi ( <i>read</i> )	Kode Fungsi ( <i>write</i> )	<i>Register Number</i>	Tipe <i>register</i>	Keterangan Nilai ( <i>read</i> )	Keterangan Nilai ( <i>write</i> )
				<i>16 bit unsigned</i>	Sesuai	Tidak Dapat Dikendalikan
<i>Digital Output 7</i>	02	15	06	<i>Digital Coil</i>	Sesuai	Dapat Dikendalikan
				<i>16 bit unsigned</i>	Sesuai	Tidak Dapat Dikendalikan
<i>Digital Output 7</i>	02	15	07	<i>Digital Coil</i>	Sesuai	Dapat Dikendalikan
				<i>16 bit unsigned</i>	Sesuai	Tidak Dapat Dikendalikan

### 4.3 Pengujian Program dan Grafis

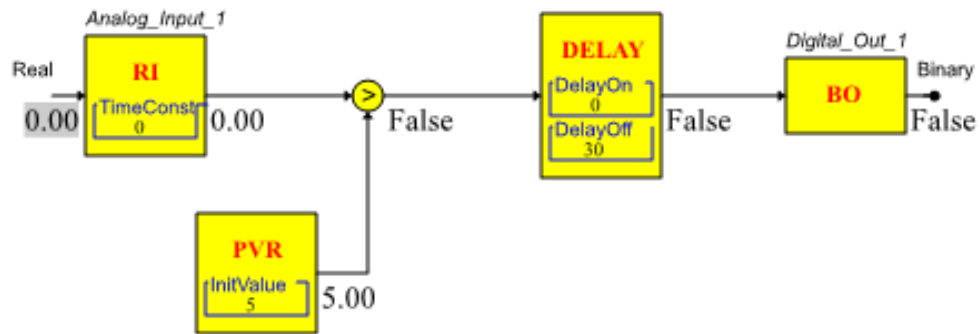
#### 4.3.1 Pengujian Program

Pengujian program dilakukan untuk menguji keberhasilan Fungsi Diagram Blok (*Function Block Diagram*) program dan fungsi dasar *Modbus* pada tiap perangkat yang dilakukan dalam mode telekontrol protokol *Modbus* melalui komunikasi antara *Master* dan *Slave*.

##### 4.3.1.1 Pengujian Blok Diagram Fungsi dan Fungsi Dasar *Modbus* Pada Program *Over Current* (Arus Lebih).

Pengujian dilakukan dengan cara mengamati nilai arus yang dibaca oleh *power meter* dan memberikan nilai keluaran pada PLC dan Modul I/O. Hasil dari pengujian Fungsi dasar *Modbus* dan Diagram Blok Fungsi program dapat dilihat pada gambar dan tabel dibawah ini.

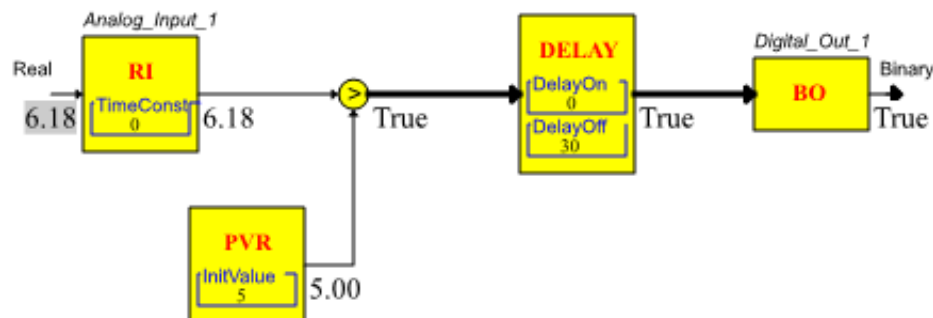
Program sebelum masukan nilai arus yang dibaca oleh *Power meter* masih dibawah 5 Amper dapat dilihat pada gambar dibawah ini.



**Gambar 4.6** Program sebelum masukan nilai arus dibawah 5 Ampere.

Dari gambar diatas, nilai keluaran *Digital\_Out\_1* masih dalam status *False* (0) dikarenakan nilai masukan *Analog\_Input\_1* masih dibawah 5.

Program setelah masukan nilai arus yang dibaca oleh *Power meter* bernilai diatas 5 Ampere dapat dilihat pada gambar dibawah ini.



**Gambar 4.7** Program setelah masukan nilai arus diatas 5 Ampere.

Dari gambar diatas, nilai keluaran *Digital\_Out\_1* dalam status *True* (1) dikarenakan nilai masukan *Analog\_Input\_1* diatas 5.

Tabel 4.9 Hasil pengujian respon fungsi dasar *Modbus* program *Over Current*.

Nama	Alamat Binding	Perangkat Modbus	Kode Fungsi	Register Number	Respon
<i>Analog_Input_1</i>	<i>Current (A L1)</i>	<i>Power Meter</i>	03	03	<i>OK</i>
<i>Digital_Out_1</i>	<i>Digital Output 1</i>	PLC M221	02	00	<i>OK</i>

Nama	Alamat Binding	Perangkat Modbus	Kode Fungsi	Register Number	Respon
<i>Digital_Out_1</i>	<i>Digital Output 1</i>	<i>Wellpro</i>	01	01	<i>OK</i>

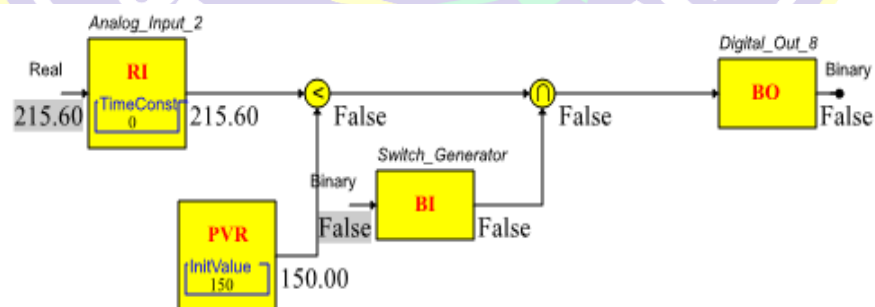
Tabel 4.10 Hasil pengujian Diagram Blok Fungsi Program *Over Current*.

Keluaran	Nilai <i>Analog_Input_1</i>	Status Keluaran	Indikator Lampu PLC	Indikator Lampu <i>Wellpro</i>
<i>Digital_Out_1</i>	Dibawah 5 A (< 5A)	<i>False</i>	Mati	Mati
<i>Digital_Out_1</i>	Diatas 5 A (> 5A)	<i>True</i>	Hidup	Hidup

#### 4.3.1.2 Pengujian Blok Diagram Fungsi dan Fungsi Dasar *Modbus* Pada Program *Starting Generator*.

Pengujian dilakukan dengan cara mengamati nilai tegangan yang dibaca oleh *power meter* dan memberikan nilai keluaran pada PLC. Hasil dari pengujian Fungsi dasar *Modbus* dan Diagram Blok Fungsi program dapat dilihat pada gambar dan tabel dibawah ini.

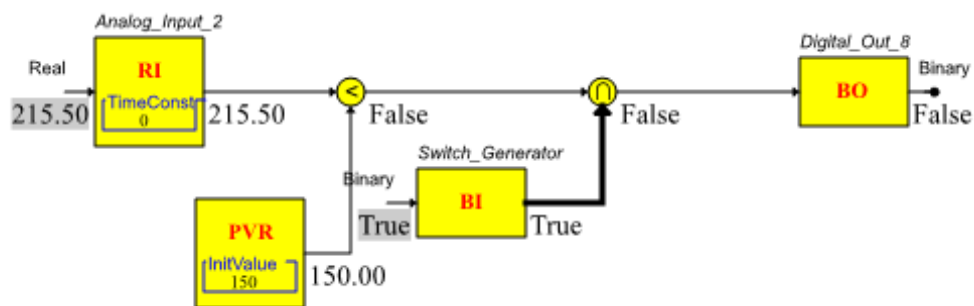
Program ketika nilai tegangan diatas 150 Volt dan *Switch Generator* dalam keadaan *OFF* dapat dilihat pada gambar dibawah ini.



**Gambar 4.8** Program ketika nilai tegangan diatas 150 Volt dan *Switch Generator* dalam keadaan *OFF*.

Dari gambar diatas, nilai keluaran *Digital\_Out\_8* bernilai *false* (0), dikarenakan nilai masukan *Analog\_Input\_2* bernilai diatas 150 dan *Switch Generator* bernilai *false* (0).

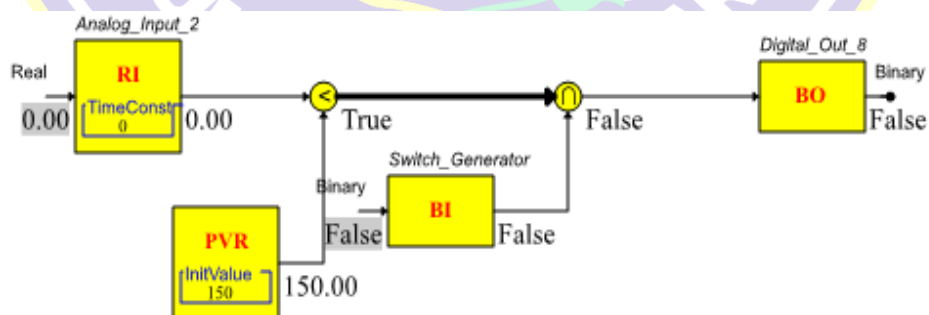
Program ketika nilai tegangan diatas 150 Volt dan *Switch Generator* dalam keadaan *ON* dapat dilihat pada gambar dibawah ini.



**Gambar 4.9** Program ketika nilai tegangan diatas 150 Volt dan *Switch Generator* dalam keadaan *ON*.

Dari gambar diatas, nilai keluaran *Digital\_Out\_8* bernilai *false* (0), dikarenakan nilai masukan *Analog\_Input\_2* bernilai diatas 150 dan *Switch Generator* bernilai *true* (1).

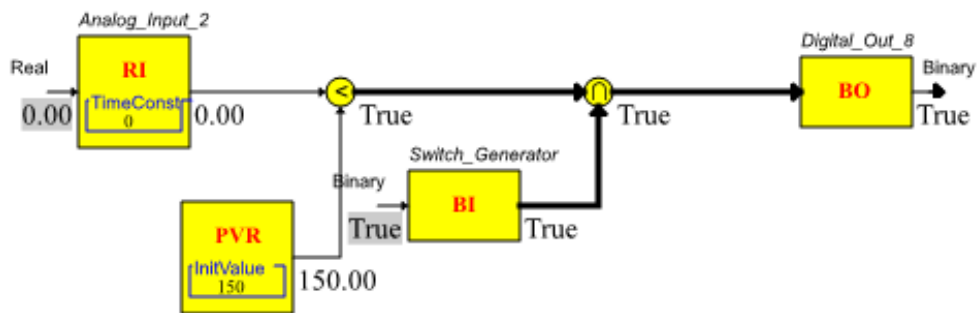
Program ketika nilai tegangan dibawah 150 Volt dan *Switch Generator* dalam keadaan *OFF* dapat dilihat pada gambar dibawah ini.



**Gambar 4.10** Program ketika nilai tegangan dibawah 150 Volt dan *Switch Generator* dalam keadaan *OFF*.

Dari gambar diatas, nilai keluaran *Digital\_Out\_8* bernilai *false* (0), dikarenakan nilai masukan *Analog\_Input\_2* bernilai dibawah 150 dan *Switch Generator* bernilai *false* (0).

Program ketika nilai tegangan dibawah 150 Volt dan *Switch Generator* dalam keadaan *ON* dapat dilihat pada gambar dibawah ini.



**Gambar 4.11** Program ketika nilai tegangan dibawah 150 Volt dan *Switch Generator* dalam keadaan *ON*.

Dari gambar diatas, nilai keluaran *Digital\_Out\_8* bernilai *true* (1), dikarenakan nilai masukan *Analog\_Input\_2* bernilai dibawah 150 dan *Switch Generator* bernilai *true* (1).

Tabel 4.11 Hasil pengujian respon fungsi dasar *Modbus* program *Starter Generator*.

Nama	Alamat Binding	Perangkat Modbus	Kode Fungsi	Register Number	Respon
<i>Analog_Input_2</i>	Voltage (V L1)	Power Meter	03	01	OK
<i>Digital_Out_8</i>	Digital Output 8	PLC M221	02	07	OK



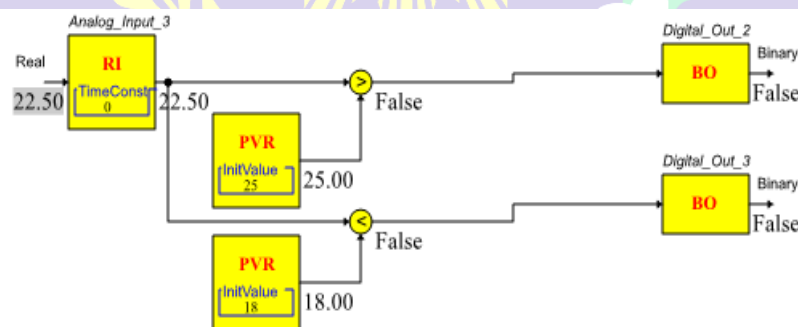
Tabel 4.12 Hasil pengujian Diagram Blok Fungsi Program *Starter Generator*.

Keluaran	Nilai <i>Analog_Input_2</i>	Kondisi <i>Switch_Generator</i>	Status Keluaran	Indikator Lampu PLC
<i>Digital_Out_8</i>	Diatas 150 Volt	<i>OFF</i>	<i>False</i>	Mati
<i>Digital_Out_8</i>	Diatas 150 Volt	<i>ON</i>	<i>False</i>	Mati
<i>Digital_Out_8</i>	Dibawah 150 Volt	<i>OFF</i>	<i>False</i>	Mati
<i>Digital_Out_8</i>	Dibawah 150 Volt	<i>ON</i>	<i>True</i>	Hidup

#### 4.3.1.3 Pengujian Blok Diagram Fungsi dan Fungsi Dasar *Modbus* Pada Program *Temperature Alarm*.

Pengujian dilakukan dengan cara mengamati nilai suhu yang dibaca oleh *Thermostat* dan memberikan nilai keluaran pada PLC. Hasil dari pengujian Fungsi dasar *Modbus* dan Diagram Blok Fungsi program dapat dilihat pada gambar dan tabel dibawah ini.

Program ketika nilai suhu berada diatas 18°C dan dibawah 25°C dapat dilihat pada gambar dibawah ini.

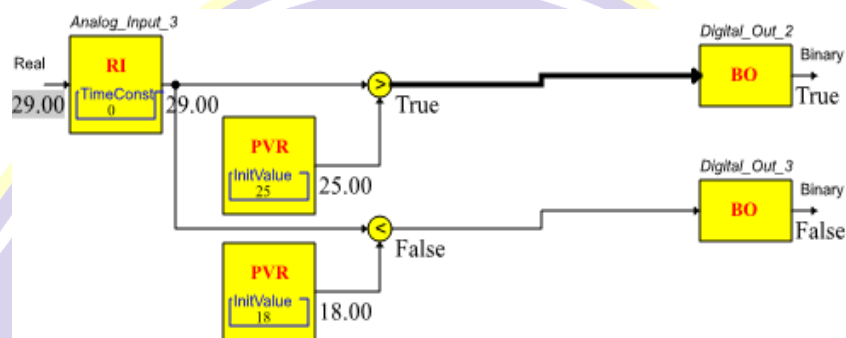


**Gambar 4.12** Program saat nilai suhu berada diatas 18°C dan dibawah 25°C.

Dari gambar diatas, nilai keluaran *Digital\_Out\_2* bernilai *false* (0), dikarenakan nilai masukan *Analog\_Input\_3* bernilai dibawah 25°C. Nilai keluaran

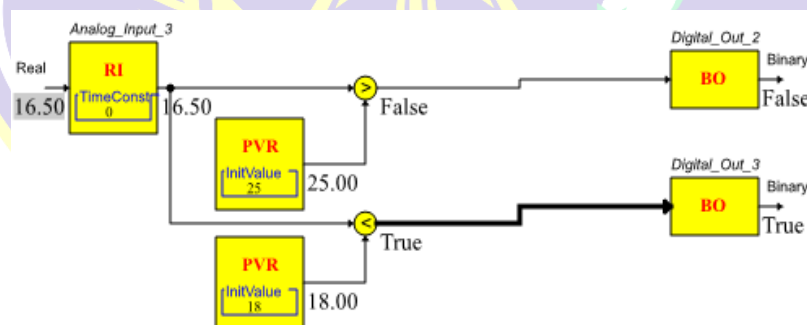
*Digital\_Out\_3* bernilai *false* (0), dikarenakan nilai masukan *Analog\_Input\_3* bernilai diatas  $18^{\circ}\text{C}$ .

Program ketika nilai suhu berada diatas  $25^{\circ}\text{C}$  dapat dilihat pada gambar (4.13). Dari gambar tersebut, nilai keluaran *Digital\_Out\_2* bernilai *true* (1), dikarenakan nilai masukan *Analog\_Input\_3* bernilai diatas  $25^{\circ}\text{C}$ . Nilai keluaran *Digital\_Out\_3* bernilai *false* (0), dikarenakan nilai masukan *Analog\_Input\_3* bernilai diatas  $18^{\circ}\text{C}$ .



**Gambar 4.13** Program ketika nilai suhu berada diatas  $25^{\circ}\text{C}$

Program ketika nilai suhu berada dibawah  $18^{\circ}\text{C}$  dapat dilihat pada gambar dibawah ini.



**Gambar 4.14** Program ketika nilai suhu berada dibawah  $18^{\circ}\text{C}$

Dari gambar diatas, nilai keluaran *Digital\_Out\_2* bernilai *false* (0), dikarenakan nilai masukan *Analog\_Input\_3* bernilai dibawah  $25^{\circ}\text{C}$ . Nilai keluaran

*Digital\_Out\_3* bernilai *true* (1), dikarenakan nilai masukan *Analog\_Input\_3* bernilai dibawah 18°C.

Tabel 4.13 Hasil pengujian respon fungsi dasar *Modbus* program *Temperature Alarm*.

Nama	Alamat Binding	Perangkat Modbus	Kode Fungsi (Read)	Register Number	Respon
<i>Analog_Input_3</i>	<i>Room Temperature</i>	<i>Thermostat</i>	04	01	<i>OK</i>
<i>Digital_Out_2</i>	<i>Digital Output 2</i>	PLC M221	02	01	<i>OK</i>
<i>Digital_Out_3</i>	<i>Digital Output 3</i>	PLC M221	02	02	<i>OK</i>

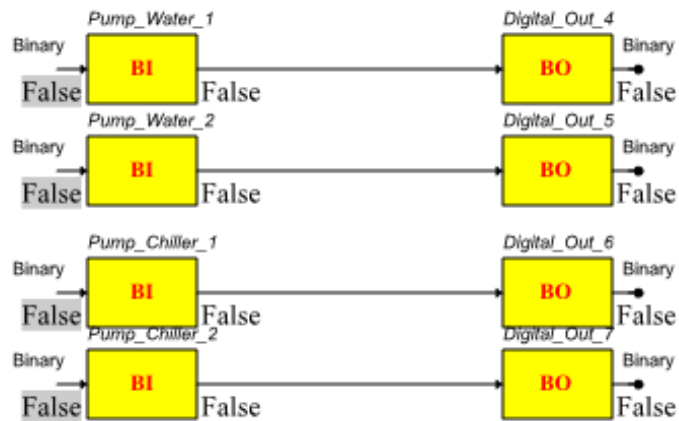
Tabel 4.14 Hasil pengujian Diagram Blok Fungsi Program *Temperature Alarm*.

Keluaran	Nilai <i>Analog_Input_3</i>	Status Keluaran	Indikator Lampu PLC
<i>Digital_Out_2</i>	Dibawah 25°C	<i>False</i>	Mati
<i>Digital_Out_2</i>	Diatas 25°C	<i>True</i>	Hidup
<i>Digital_Out_3</i>	Dibawah 18°C	<i>True</i>	Hidup
<i>Digital_Out_3</i>	Diatas 18°C	<i>False</i>	Mati

#### 4.3.1.4 Pengujian Blok Diagram Fungsi dan Fungsi Dasar Modbus Pada *Button Booster Water Pump dan Compressor Chiller*.

Pengujian dilakukan dengan cara mengamati nilai keluaran PLC apabila *Button* pada grafis dikendalikan. Hasil dari pengujian Fungsi dasar Modbus dan Diagram Blok Fungsi program dapat dilihat pada gambar –gambar dan table dibawah ini.

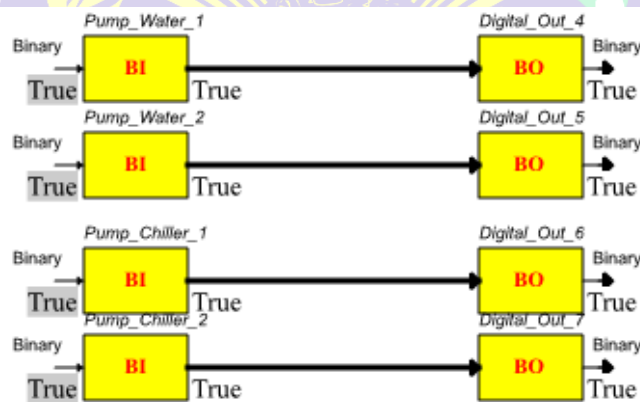
Program ketika semua *Switch* (*Pump\_Water\_1*, *Pump\_Water\_2*, *Pump\_Chiller\_1*, dan *Pump\_Chiller\_2*) dalam keadaan *Off* (*False*).



**Gambar 4.15** Program *Button Booster Water Pump dan Compressor Chiller* pada saat semua *button* dalam keadaan *OFF*

Pada gambar diatas, nilai keluaran (*Digital\_Out\_4*, *Digital\_Out\_5*, *Digital\_Out\_6*, dan *Digital\_Out\_7*) bernilai *false* atau dalam keadaan *off*.

Program ketika semua *Switch* (*Pump\_Water\_1*, *Pump\_Water\_2*, *Pump\_Chiller\_1*, dan *Pump\_Chiller\_2*) dalam keadaan *On (True)*.



**Gambar 4.16** Program *Button Booster Water Pump dan Compressor Chiller* pada saat semua *button* dalam keadaan *ON*.

Pada gambar diatas, nilai keluaran (*Digital\_Out\_4*, *Digital\_Out\_5*, *Digital\_Out\_6*, dan *Digital\_Out\_7*) bernilai *true* atau dalam keadaan *On*.

Tabel 4.15 Hasil pengujian respon fungsi dasar Modbus program *Button Booster Water Pump dan Compressor Chiller*.

Nama	Alamat Binding	Perangkat Modbus	Kode Fungsi (Read)	Register Number	Respon
<i>Pump_Water_1</i>	<i>Button Booster Pump_1</i>	Grafis	-	-	<i>OK</i>
<i>Pump_Water_2</i>	<i>Button Booster Pump_2</i>	Grafis	-	-	<i>OK</i>
<i>Pump_Chiller_1</i>	<i>Button Compressor Chiller_1</i>	Grafis	-	-	<i>OK</i>
<i>Pump_Chiller_2</i>	<i>Button Compressor Chiller_2</i>	Grafis	-	-	<i>OK</i>
<i>Digital_Out_4</i>	<i>Digital Output 4</i>	PLC M221	02	03	<i>OK</i>
<i>Digital_Out_5</i>	<i>Digital Output 5</i>	PLC M221	02	05	<i>OK</i>
<i>Digital_Out_6</i>	<i>Digital Output 6</i>	PLC M221	02	06	<i>OK</i>
<i>Digital_Out_7</i>	<i>Digital Output 7</i>	PLC M221	02	07	<i>OK</i>

Tabel 4.16 Hasil pengujian Diagram Blok Fungsi Program *Button Booster Water Pump dan Compressor Chiller*.

Keluaran	Nilai <i>Button Booster Pump_1</i>	Nilai <i>Button Booster Pump_1</i>	Nilai <i>Button Compressor Chiller_1</i>	Nilai <i>Button Compressor Chiller_2</i>	Indikator Lampu PLC
<i>Digital_Out_4</i>	<i>False</i>	-	-	-	Mati
<i>Digital_Out_4</i>	<i>True</i>	-	-	-	Hidup
<i>Digital_Out_5</i>	-	<i>False</i>	-	-	Mati
<i>Digital_Out_5</i>	-	<i>True</i>	-	-	Hidup
<i>Digital_Out_6</i>	-	-	<i>False</i>	-	Mati
<i>Digital_Out_6</i>	-	-	<i>True</i>	-	Hidup
<i>Digital_Out_7</i>	-	-	-	<i>False</i>	Mati

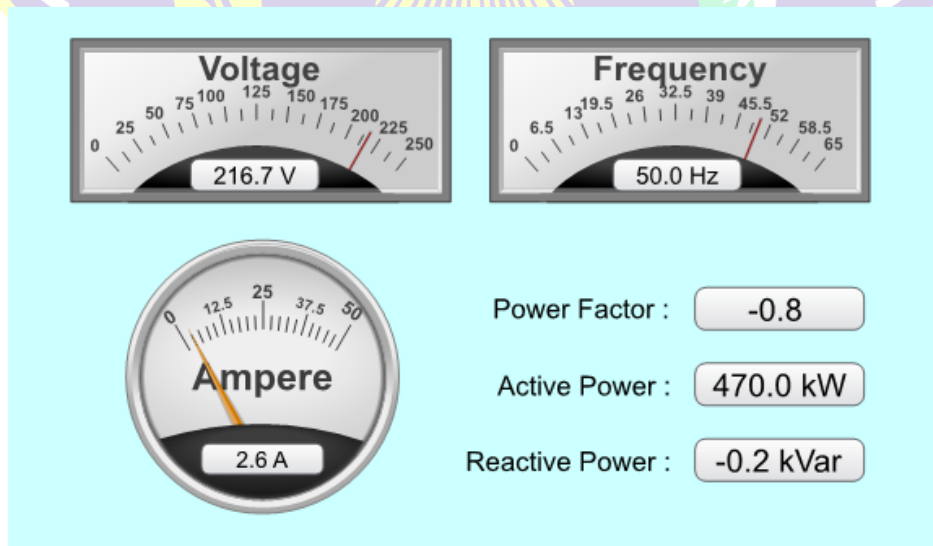
Keluaran	Nilai <i>Button Booster Pump_1</i>	Nilai <i>Button Booster Pump_1</i>	Nilai <i>Button Compressor Chiller_1</i>	Nilai <i>Button Compressor Chiller_2</i>	Indikator Lampu PLC
<i>Digital_Out_7</i>	-	-	-	<i>True</i>	Hidup

### 4.3.2 Pengujian Grafis

Pengujian grafis dilakukan untuk menguji keberhasilan grafis dalam melakukan pengawasan dan pengendalian terhadap nilai – nilai pada perangkat *Modbus* yang terhubung dengan *Automation Server* dalam satu antarmuka (*Interface*).

#### 4.3.2.1 Pengujian Grafis Terhadap Perangkat *Power meter CVM-NRG96*.

Pengujian dilakukan untuk menguji nilai – nilai *Modbus Point* pada *Power meter* yang dibaca oleh grafis sesuai dengan pembacaan pada perangkat kerasnya. Hasil pengujian nilai *Modbus Point* yang dibaca oleh grafis dapat dilihat pada gambar dan table dibawah ini.



**Gambar 4.17** Tampilan Pembacaan Nilai – Nilai *Modbus Point Power meter CVM-NRG96* Pada Grafis.



Dari gambar diatas, dapat dilihat nilai – nilai *Modbus Point Power Meter* yang dibaca melalui grafis.

Tabel 4.17 Hasil pengujian Fungsi Dasar *Modbus Point* pada *Power meter* untuk dibaca dan dikendalikan melalui Grafis.

<i>Modbus Point</i>	<i>Function Code (Read)</i>	<i>Function Code (Write)</i>	<i>Register Number</i>	Keterangan
<i>Active Power</i>	3	-	5	Hanya Dibaca
<i>Current (A L1)</i>	3	-	3	Hanya Dibaca
<i>Frequency (Hz)</i>	3	-	41	Hanya Dibaca
<i>Power Factor</i>	3	-	9	Hanya Dibaca
<i>Reactive Power</i>	3	-	7	Hanya Dibaca
<i>Voltage (V L1)</i>	3	-	1	Hanya Dibaca

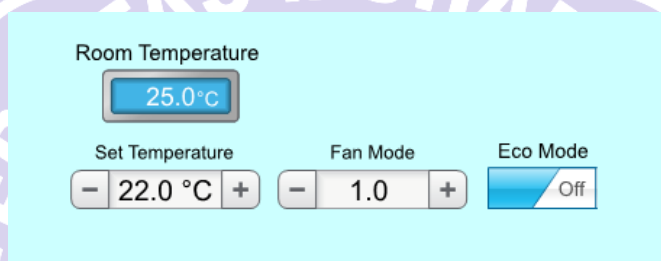
Tabel 4.18 Hasil pengujian Grafis terhadap nilai – nilai *Modbus Point Powermeter CVM-NRG96*.

Komponen Grafis	<i>Alamat Binding</i>	<i>Modbus Point</i>	Keterangan
<i>Dial (Voltage)</i>	<i>Voltage</i>	<i>Voltage (V L1)</i>	Sesuai
<i>Dial (Frequency)</i>	<i>Frequency</i>	<i>Frequency (Hz)</i>	Sesuai
<i>Dial (Current)</i>	<i>Current</i>	<i>Current (A L1)</i>	Sesuai
<i>Analog Value 1</i>	<i>Voltage</i>	<i>Voltage (V L1)</i>	Sesuai
<i>Analog Value 2</i>	<i>Frequency</i>	<i>Frequency (Hz)</i>	Sesuai
<i>Analog Value 3</i>	<i>Current</i>	<i>Current (A L1)</i>	Sesuai
<i>Display</i>	<i>Power Factor</i>	<i>Power Factor</i>	Sesuai
<i>Display</i>	<i>Active Power</i>	<i>Active Power</i>	Sesuai

Komponen Grafis	Alamat <i>Binding</i>	<i>Modbus Point</i>	Keterangan
<i>Display</i>	<i>Reactive Power</i>	<i>Reactive Power</i>	Sesuai

#### 4.3.2.2 Pengujian Grafis Terhadap Perangkat *Thermostat TC-300*.

Pengujian dilakukan untuk menguji dan mengendalikan nilai – nilai *Modbus Point* pada *Thermostat* melalui grafis dan menguji pembacaan sesuai pada perangkat kerasnya. Hasil pengujian nilai *Modbus Point* yang dibaca dan dikendalikan oleh grafis dapat dilihat pada gambar dan tabel dibawah ini.



**Gambar 4.18** Tampilan Pembacaan dan Pengendalian Nilai – Nilai *Modbus Point Thermostat TC-300* Pada Grafis.

Dari gambar diatas, dapat dilihat nilai – nilai *Modbus Point Thermostat TC300* yang dibaca melalui grafis.

Tabel 4.19 Hasil pengujian Fungsi Dasar *Modbus Point* pada *Thermostat* untuk dikendalikan melalui Grafis.

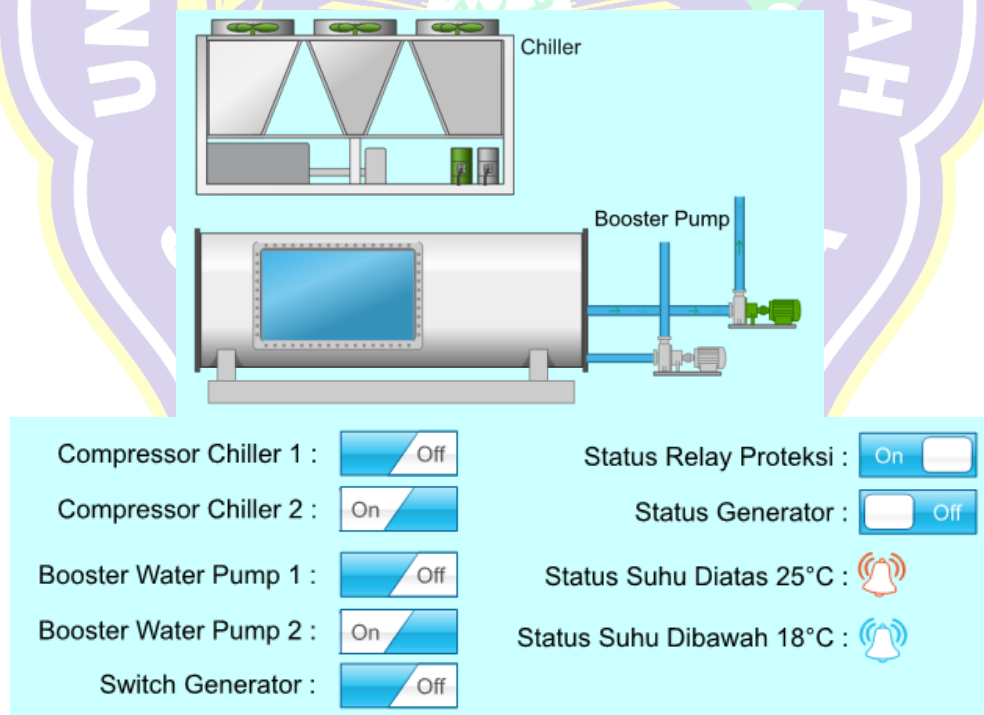
<i>Modbus Point</i>	<i>Function Code (Read)</i>	<i>Function Code (Write)</i>	<i>Register Number</i>	Keterangan
<i>Eco Mode</i>	3	6	10	Dapat Dikendalikan
<i>Fan Mode</i>	3	6	6	Dapat Dikendalikan
<i>Room Temperature</i>	4	<i>None</i>	1	Hanya Dibaca
<i>Set Temperature</i>	3	6	5	Dapat Dikendalikan

Tabel 4.20 Hasil pengujian Grafis terhadap nilai – nilai *Modbus Point Thermostat*.

Komponen Grafis	Alamat <i>Binding</i>	<i>Modbus Point</i>	Keterangan
<i>Analog Set Point 1</i>	<i>Set_Temperature</i>	<i>Set Temperature</i>	Sesuai
<i>Analog Set Point 2</i>	<i>Set_FanMode</i>	<i>Fan Mode</i>	Sesuai
<i>Slide Button Android Style 6</i>	<i>Button Eco Mode</i>	<i>Eco Mode</i>	Sesuai
<i>Display</i>	<i>Room Temperature</i>	<i>Room Temperature</i>	Sesuai

#### 4.3.2.3 Pengujian Grafis Terhadap Perangkat PLC M221 dan *Wellpro Modul Input/Output*.

Pengujian dilakukan untuk menguji dan mengendalikan nilai – nilai *Modbus Point* pada PLC dan Modul I/O melalui grafis dan menguji pembacaan sesuai pada perangkat kerasnya. Hasil pengujian nilai *Modbus Point* yang dibaca dan dikendalikan oleh grafis dapat dilihat pada gambar dan tabel dibawah ini.



**Gambar 4.19** Tampilan Pembacaan dan Pengendalian Nilai – Nilai *Modbus Point* PLC M221 dan *Wellpro Modul I/O* Pada Grafis.

Dari gambar diatas, status dan *selector switch* untuk melakukan pengawasn dan pengendalian terhadap nilai – nilai *Modbus Point* pada PLC M221 dan *Wellpro Module I/O* dapat dilihat dan dilakukan pada sebuah grafis.

Tabel 4.21 Hasil pengujian Fungsi Dasar *Modbus Point* pada PLC M221 dan *Wellpro Modul I/O* untuk dibaca dan dikendalikan melalui Grafis.

<i>Modbus Point</i>	Perangkat	<i>Function Code (Read)</i>	<i>Function Code (Write)</i>	<i>Register Number</i>	Keterangan
<i>Digital Input 1</i>	<i>Wellpro</i>	2	None	1	Hanya Dibaca
<i>Digital Output 1</i>	<i>Wellpro</i>	1	15	1	Dapat Dikendalikan
<i>Digital Output 1</i>	PLC M221	2	15	0	Dapat Dikendalikan
<i>Digital Output 2</i>	PLC M221	2	15	1	Dapat Dikendalikan
<i>Digital Output 3</i>	PLC M221	2	15	2	Dapat Dikendalikan
<i>Digital Output 4</i>	PLC M221	2	15	3	Dapat Dikendalikan
<i>Digital Output 5</i>	PLC M221	2	15	4	Dapat Dikendalikan
<i>Digital Output 6</i>	PLC M221	2	15	5	Dapat Dikendalikan
<i>Digital Output 7</i>	PLC M221	2	15	6	Dapat Dikendalikan
<i>Digital Output 8</i>	PLC M221	2	15	7	Dapat Dikendalikan

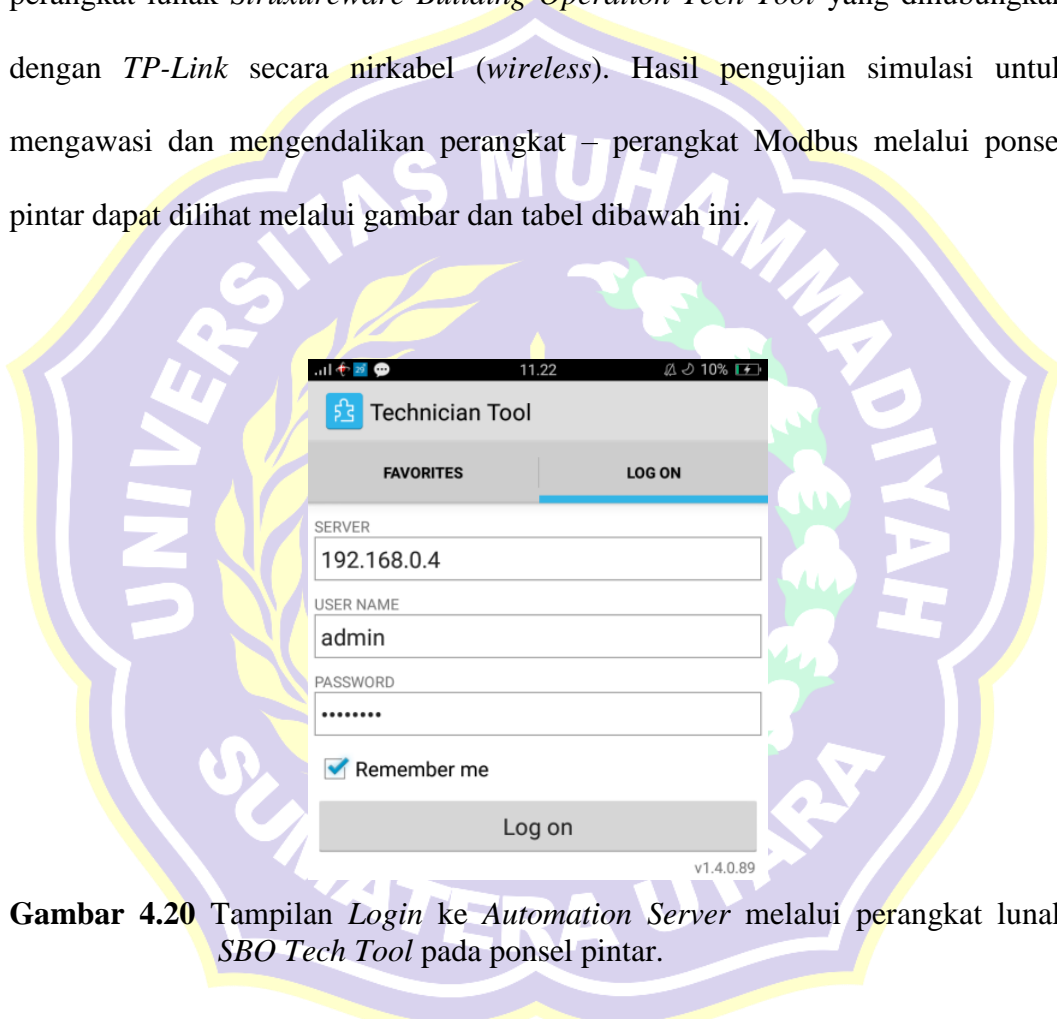
Tabel 4.22 Hasil pengujian Grafis terhadap nilai – nilai *Modbus Point* PLC M221 dan *Wellpro Module I/O*.

Komponen Grafis	<i>Alamat Binding</i>	<i>Modbus Point</i>	Perangkat	Keterangan
<i>AC Chiller – 2 Compressor</i>	<i>Motor Chiller 1</i>	<i>Digital Output 6</i>	PLC M221	Sesuai
<i>AC Chiller – 2 Compressor</i>	<i>Motor Chiller 2</i>	<i>Digital Output 7</i>	PLC M221	Sesuai

Komponen Grafis	Alamat Binding	Modbus Point	Perangkat	Keterangan
<i>Pump T2 Large Flow Right 1</i>	<i>Booster Pump 1</i>	<i>Digital Output 4</i>	PLC M221	Sesuai
<i>Pump T2 Large Flow Right 2</i>	<i>Booster Pump 2</i>	<i>Digital Output 5</i>	PLC M221	Sesuai
<i>DP Flow Right 1</i>	<i>Booster Pump 1</i>	<i>Digital Output 4</i>	PLC M221	Sesuai
<i>DP Flow Right 2</i>	<i>Booster Pump 2</i>	<i>Digital Output 5</i>	PLC M221	Sesuai
<i>DP Flow Up 1</i>	<i>Booster Pump 1</i>	<i>Digital Output 4</i>	PLC M221	Sesuai
<i>DP Flow Up 2</i>	<i>Booster Pump 2</i>	<i>Digital Output 5</i>	PLC M221	Sesuai
<i>Slide Button Android Style 1</i>	<i>Button Comperssor Chiller_1</i>	-	*Program	Sesuai
<i>Slide Button Android Style 2</i>	<i>Button Comperssor Chiller_2</i>	-	*Program	Sesuai
<i>Slide Button Android Style 3</i>	<i>Button Booster Pump_1</i>	-	*Program	Sesuai
<i>Slide Button Android Style 4</i>	<i>Button Booster Pump_2</i>	-	*Program	Sesuai
<i>Slide Button Android Style 5</i>	<i>Button Generator</i>	-	*Program	Sesuai
<i>Slide Button SBO Style 1</i>	Status Relay Proteksi	<i>Digital Input 1</i>	Wellpro	Sesuai
<i>Slide Button SBO Style 1</i>	Status Relay Proteksi	<i>Digital Output 1</i>	PLC M221	Sesuai
<i>Slide Button SBO Style 2</i>	Status Generator	<i>Digital Output 8</i>	PLC M221	Sesuai
<i>Alarm Bell Red</i>	<i>Temperature 25</i>	<i>Digital Output 2</i>	PLC M221	Sesuai
<i>Alarm Bell Blue</i>	<i>Temperature 18</i>	<i>Digital Output 1</i>	PLC M221	Sesuai

#### 4.3.2.4 Pengujian Program dan Grafis Menggunakan Ponsel Pintar (*Smartphone*)

Pengujian program dan grafis dilakukan untuk menguji keberhasilan grafis dalam melakukan pengawasan dan pengendalian terhadap nilai – nilai pada perangkat *Modbus* yang terhubung dengan *Automation Server* dalam satu antarmuka (*Interface*) menggunakan ponsel pintar (*smartphone*) melalui perangkat lunak *Struxureware Building Operation Tech Tool* yang dihubungkan dengan *TP-Link* secara nirkabel (*wireless*). Hasil pengujian simulasi untuk mengawasi dan mengendalikan perangkat – perangkat *Modbus* melalui ponsel pintar dapat dilihat melalui gambar dan tabel dibawah ini.



**Gambar 4.20** Tampilan *Login* ke *Automation Server* melalui perangkat lunak *SBO Tech Tool* pada ponsel pintar.

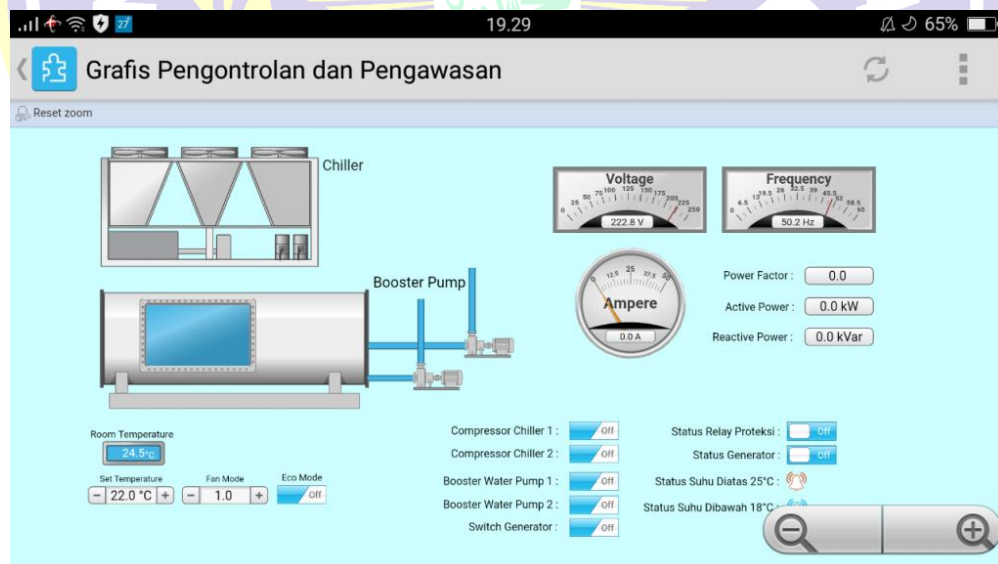
Dari gambar diatas, dapat dilihat *menu Log On* pada perangkat lunak *SBO Tech Tool* untuk masuk kedalam *Automation Server*.





**Gambar 4.21** Tampilan beranda *tab system* pada perangkat lunak *SBO Tech Tool* setelah *login* ke *Automation Server*.

Dari gambar diatas, dapat dilihat tampilan beranda setelah berhasil *Login* kedalam *Automation Server*.



**Gambar 4.22** Tampilan grafis pengawasan dan pengendalian perangkat – perangkat *Modbus* melalui perangkat lunak *SBO Tech Tool*.

Dari gambar diatas dapat dilihat tampilan keseluruhan grafis untuk melakukan pengendalian dan pengawasan perangkat – perangkat *Modbus* yang terhubung (*Online*) dengan *Automation Server*.

Tabel 4.23 Hasil pengujian grafis dalam melakukan pengawasan dan pengendalian melalui ponsel pintar.

Komponen Grafis	Alamat Binding	Modbus Point	Perangkat	Keterangan
<i>Dial (Voltage)</i>	<i>Voltage</i>	<i>Voltage (V L1)</i>	<i>Power meter</i>	Sesuai
<i>Dial (Frequency)</i>	<i>Frequency</i>	<i>Frequency (Hz)</i>	<i>Power meter</i>	Sesuai
<i>Dial (Current)</i>	<i>Current</i>	<i>Current (A L1)</i>	<i>Power meter</i>	Sesuai
<i>Analog Value 1</i>	<i>Voltage</i>	<i>Voltage (V L1)</i>	<i>Power meter</i>	Sesuai
<i>Analog Value 2</i>	<i>Frequency</i>	<i>Frequency (Hz)</i>	<i>Power meter</i>	Sesuai
<i>Analog Value 3</i>	<i>Current</i>	<i>Current (A L1)</i>	<i>Power meter</i>	Sesuai
<i>Display</i>	<i>Power Factor</i>	<i>Power Factor</i>	<i>Power meter</i>	Sesuai
<i>Display</i>	<i>Active Power</i>	<i>Active Power</i>	<i>Power meter</i>	Sesuai
<i>Display</i>	<i>Reactive Power</i>	<i>Reactive Power</i>	<i>Power meter</i>	Sesuai
<i>Analog Set Point 1</i>	<i>Set_Temperature</i>	<i>Set Temperature</i>	<i>Thermostat</i>	Sesuai
<i>Analog Set Point 2</i>	<i>Set_FanMode</i>	<i>Fan Mode</i>	<i>Thermostat</i>	Sesuai
<i>Slide Button Android Style 6</i>	<i>Button Eco Mode</i>	<i>Eco Mode</i>	<i>Thermostat</i>	Sesuai
<i>AC Chiller – 2 Compressor</i>	<i>Motor Chiller 1</i>	<i>Digital Output 6</i>	PLC M221	Sesuai
<i>AC Chiller – 2 Compressor</i>	<i>Motor Chiller 2</i>	<i>Digital Output 7</i>	PLC M221	Sesuai
<i>Pump T2 Large Flow Right 1</i>	<i>Booster Pump 1</i>	<i>Digital Output 4</i>	PLC M221	Sesuai

Komponen Grafis	Alamat Binding	Modbus Point	Perangkat	Keterangan
<i>Pump T2 Large Flow Right 2</i>	<i>Booster Pump 2</i>	<i>Digital Output 5</i>	PLC M221	Sesuai
<i>DP Flow Right 1</i>	<i>Booster Pump 1</i>	<i>Digital Output 4</i>	PLC M221	Sesuai
<i>DP Flow Right 2</i>	<i>Booster Pump 2</i>	<i>Digital Output 5</i>	PLC M221	Sesuai
<i>DP Flow Up 1</i>	<i>Booster Pump 1</i>	<i>Digital Output 4</i>	PLC M221	Sesuai
<i>DP Flow Up 2</i>	<i>Booster Pump 2</i>	<i>Digital Output 5</i>	PLC M221	Sesuai
<i>Slide Button Android Style 1</i>	<i>Button Comperssor Chiller_1</i>	-	*Program	Sesuai
<i>Slide Button Android Style 2</i>	<i>Button Comperssor Chiller_2</i>	-	*Program	Sesuai
<i>Slide Button Android Style 3</i>	<i>Button Booster Pump_1</i>	-	*Program	Sesuai
<i>Slide Button Android Style 4</i>	<i>Button Booster Pump_2</i>	-	*Program	Sesuai
<i>Slide Button Android Style 5</i>	<i>Button Generator</i>	-	*Program	Sesuai
<i>Slide Button SBO Style 1</i>	Status Relay Proteksi	<i>Digital Input 1</i>	Wellpro	Sesuai
<i>Slide Button SBO Style 1</i>	Status Relay Proteksi	<i>Digital Output 1</i>	PLC M221	Sesuai
<i>Slide Button SBO Style 2</i>	Status Generator	<i>Digital Output 8</i>	PLC M221	Sesuai
<i>Alarm Bell Red</i>	<i>Temperature 25</i>	<i>Digital Output 2</i>	PLC M221	Sesuai
<i>Alarm Bell Blue</i>	<i>Temperature 18</i>	<i>Digital Output 1</i>	PLC M221	Sesuai

## BAB V

### PENUTUP

#### 5.1 Kesimpulan

Dari hasil pengujian simulasi telekontrol menggunakan modul *Schneider Automation Server* sebagai kontrol multi *Modbus Device*, maka dapat diambil kesimpulan sebagai berikut:

1. Semakin jauh jarak antara *smart device* dan *TP-Link MR3420*, sensitivitas kekuatan sinyal (dbm) semakin kecil. Kekuatan sinyal mengalami penurunan 94,7% pada jarak 150 meter. Semakin jauh jarak antara *smart device* dan *TP-Link MR3420*, kualitas sinyal semakin kecil. Kualitas sinyal mengalami penurunan 48,5% pada jarak 150 meter.
2. Nilai untuk *receive timeout* pada masing – masing port komunikasi dapat diatur lebih besar nilainya dari hasil *Timer 3,5 Character (T<sub>3,5</sub>)*. Pada Port COM-A (4800bps) diatur dengan nilai 100ms dimana standar perhitungan adalah 80,2ms. Pada Port COM-B (9600bps) diatur dengan nilai 100ms dimana standar perhitungan adalah 8,02ms. Pada Port Ethernet = Auto dimana standar perhitungan minimal sebesar 187,6 $\mu$ s. Semakin besar nilai *baudrate* dan *transfer rate*, maka semakin kecil nilai *receive timeout*.

Pada perangkat – perangkat *slave Modbus*, untuk pembacaan nilai digital dan analog dilakukan dengan cara mengubah tipe register pada masing – masing *Modbus Point*.

3. Pemrograman dan pembuatan grafis serta kombinasi dengan beberapa perangkat *slave Modbus* berhasil dilakukan dengan perangkat lunak internal

dan teknik Binding. Indikator keberhasilan program dan grafis dinilai melalui penyesuaian pembacaan dan pengendalian *Input/Output* baik digital maupun analog pada perangkat-perangkat *slave Modbus*.

## 5.2 Saran

1. Perangkat *Wellpro Module Input/Output* yang penulis gunakan memiliki kelemahan yaitu perangkat tidak akan *online* sebelum pengguna menampilkan salah satu nilai data pada perangkat tersebut di perangkat lunak *SBO Workstation*. Diharapkan penelitian selanjutnya menggunakan metode pembukaan port komunikasi agar dapat *online* tanpa menampilkan salah satu data.
2. Percobaan simulasi yang penulis gunakan untuk indikator keluaran hanya berupa lampu indikator dan tidak langsung diaplikasikan pada perangkat yang sebenarnya.
3. Simulasi telekontrol yang penulis gunakan pada simulasi ini menggunakan jaringan lokal *Wireless Local Area Network (WLAN)*. Diharapkan penelitian selanjutnya menggunakan jaringan luas *Wide Area Network (WAN)* yang mengharuskan penggunaan IP Publik (*Publish*).
4. Diharapkan penelitian selanjutnya menggunakan parameter pengaruh panjang kabel antara *Master* dan perangkat – perangkat *Slave* untuk mencegah kerusakan data.



## DAFTAR PUSTAKA

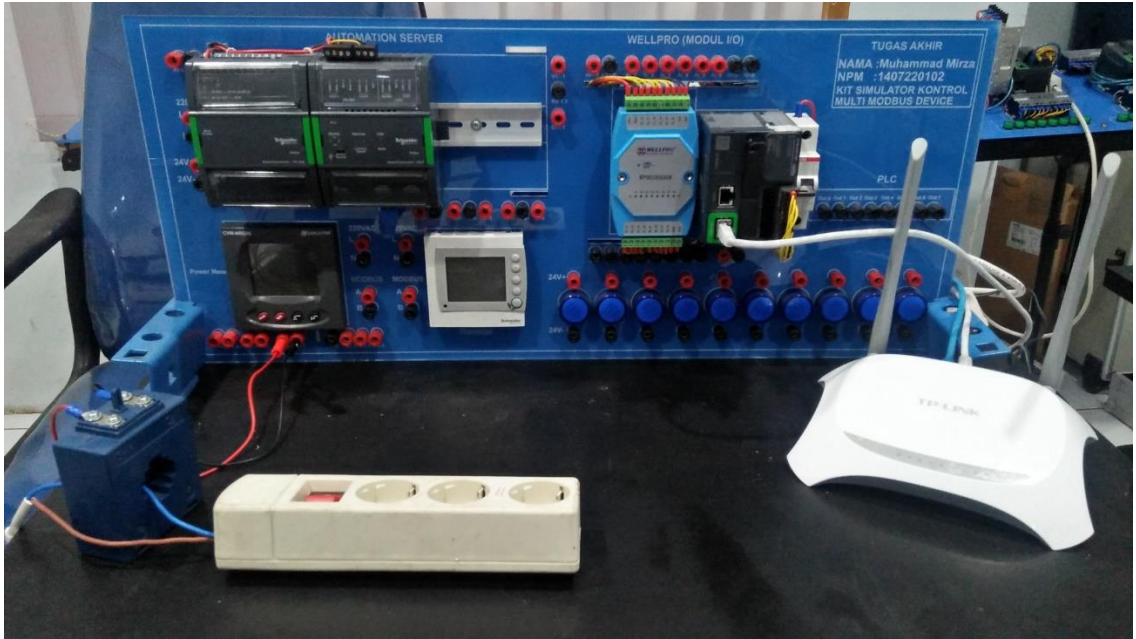
- Adriansyah, Andi dan Rizally Priatmadja. (2015). *Rancang Bangun Protocol MODBUS Pada KWH Meter Elektronik Tipe ION 8600 Untuk Memonitor Besaran Energi listrik Trafo Dengan Menggunakan Aplikasi Citect SCADA*. Jakarta. Repository Mercu Buana.
- Apriyanto, Benediktus Tri. (2016). *Aplikasi PLC Modicon M221 Untuk Smart Home Dengan HMI Berbasis Android*. Yogyakarta. Universitas Sanata Dharma.
- Ashtekar, Anjali S dkk. (2013). *Application of MODBUS to Communicate the PLC and Lab VIEW for Real Time Process Control*. India. *International Journal of Emerging Science and Engineering (IJESE)*.
- Circutor, SA. *Power Analyzer CVM-NRG96 : User Manual*. Barcelona, Spain.
- Fang, Yinlan dkk. (2013). *Research and Implementation of Collision Detection Based on Modbus Protocol*. Beijing, China. North China University of Technology.
- Gumilang, Ferdina Iqra dkk. *Rancang Bangun Jaringan Komunikasi Multi PLC dengan Platform Sistem SCADA-DCS*. Bandung. Politeknik Manufaktur Negeri Bandung.
- M-System CO., Ltd. *Modbus Protocol Reference Guide*. Minamitsumori. Osaka, Japan
- Marshal, Citra. (2012). *Rancang Bangun Sistem Monitoring Daya Pada Rumah Cerdas*. Depok. Universitas Indonesia.
- MODBUS Organization. (2002). *MODBUS over Serial Line Specification & Implementation guide*.
- MODBUS Organization. (2006). *MODBUS Messaging on TCP/IP Implementation Guide V1.0b*.
- MODBUS Organization. (2012). *MODBUS APPLICATION PROTOCOL SPECIFICATION V1.1b3*.
- MODICON, Inc., Industrial Automation Systems. (1996), *Modicon Modbus Protocol Reference Guide*. North Andover, Massachusetts 01845.
- Mulyanta, Edi S. (2005). *Pengenalan Protokol Jaringan Wireless Komputer*. Yogyakarta. Penerbit ANDI



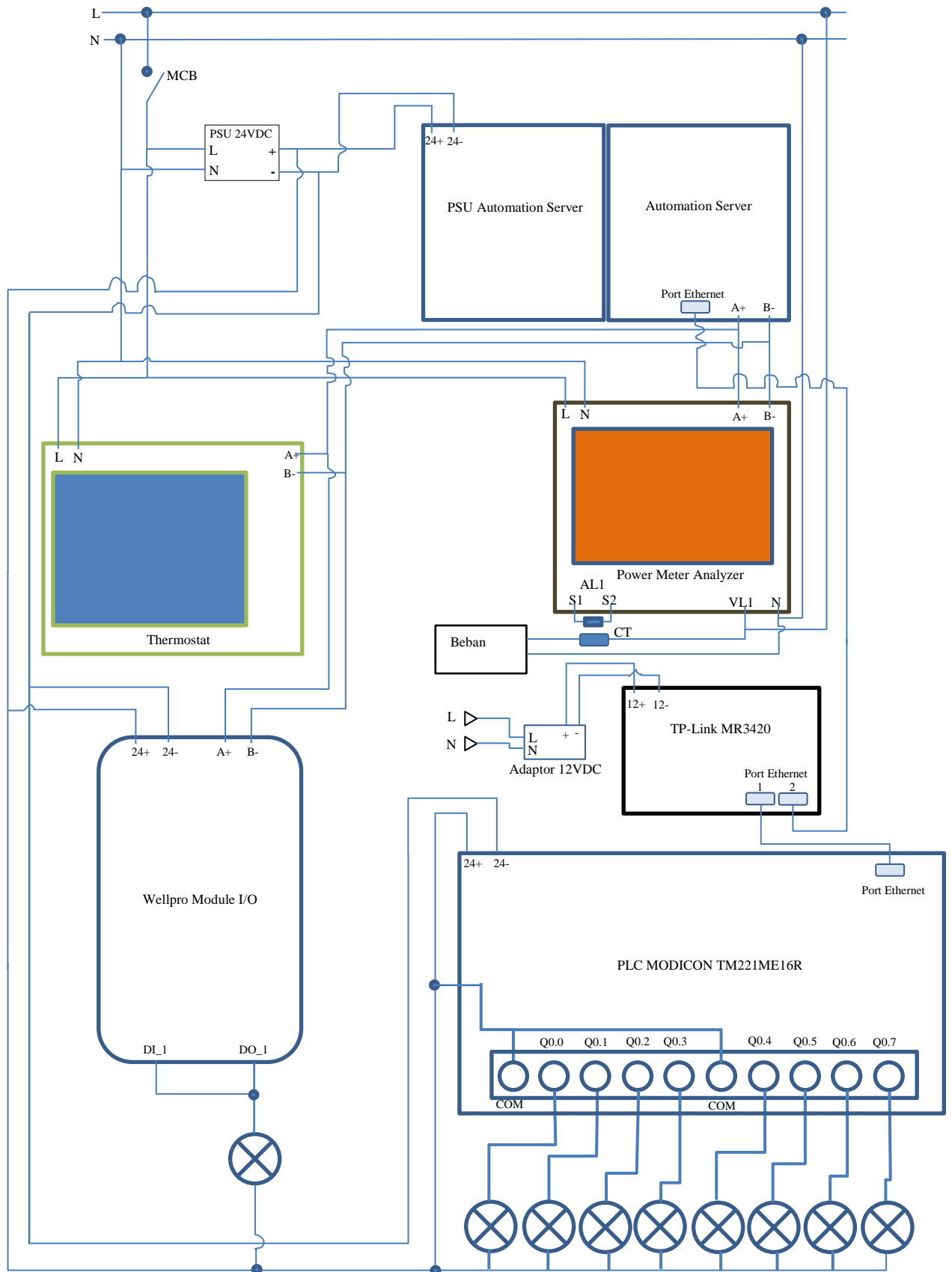
- Nurhidayat, Eka. (2010). *Pengaruh Temperatur Lingkungan Kerja dan Harmonisa Terhadap Kinerja Transformator Arus*. Depok. Universitas Indonesia.
- Nurpadmi. *Studi Tentang Modbus Protokol Pada Sistem Kontrol Vol. 01 No. 02*. Cepu. Pusdiklat Migas.
- Pancoro, Bayu. (2009). *Curent Transformer (CT)*. Wordpress.
- Pangaribowo, Triyanto dan Hibnu Yulianda. (2016). *Sistem Monitoring Suhu Melalui Sistem Komunikasi Programmable Logic Controller To Personal Computer*. Jakarta. Repository Mercu Buana
- Putranto, Ichsan Edi dkk. (2015). *Implementasi Dan Analisis Protocol Modbus TCP Pada Smart Building Berbasis OPENMTC*. Indonesia. Telkom University.
- Ramandha, Audy. (2015). *Implementasi Scada Pada Android Dengan Simulator MODBUSPAL Studi Kasus Perusahaan X*. Jakarta. Universitas Islam Negeri Syarif Hidayatullah.
- Rockhim, Abdul. *Modul Konsep Jaringan BAB XV (WLAN)*. STMIK Yadika Bangil.
- Schneider Electric, Inc. (2015). *Guide to Open Protocols in Building Automation*. Polytec, France.
- Schneider Electric, Inc (2015). *Modicon M221 Logic Controllers*. Polytec, France.
- Schneider Electric, Inc (2016). *SmartX Controller : Hardware Reference Guide*. Polytec, France.
- Schneider Electric Asia Pasific, Inc (2011). *TC300 Series Networking Digital Fan Coil Thermostat*. Singapore.
- Schneider Electric, Inc (2016). *TC300 Series Modulating Fan Coil Thermostat : Installation Intruction*. Polytec, France.
- Schneider Electric, Inc (2015). *SoMachine Basic : Operating Guide*. Polytec, France.
- Shanghai Wellpro Electrical Technology Co., Ltd. *WP9038ADAM : User Manual*. Shanghai.
- Shanghai Wellpro Electrical Technology Co., Ltd. *WELLPRO MODULE : MODBUS RTU REGISTER ADDRESS*. Shanghai.
- Tiyono, Agus dkk. (2007). *Sistem Telekontrol SCADA Dengan Fungsi Dasar Modbus Menggunakan Mikrokontroler AT89S51 Dan Komunikasi Serial RS-485*. Semarang. Diponegoro University.

Wibisono, Gunawan dan Gunadi. 2008. *Mobile Broadband Tren Teknologi Wireless Saat ini dan Masa Datang*. Bandung. Penerbit Informatika Bandung.





Lampiran (1) Modul Simulasi Telekontrol *Multi Modbus Device* Keseluruhan



Lampiran (2) Pengawatan Keseluruhan Modul Telekontrol Modbus Device

## MODBUS® memory map

Magnitude	Symbol	MODBUS VARIABLES			Unit
		Instant	Maximum	Minimum	
Voltage Phase	V L1	00-01	60-61	C0-C1	V x10
Current	A L1	02-03	62-63	C2-C3	mA
Active Power	kW L1	04-05	64-65	C4-C5	w
Reactive Power	Kvar L1	06-07	66-67	C6-C7	w
Power Factor	PF L1	08-09	68-69	C8-C9	x 100
Voltage Phase	V L2	0A-0B	6A-6B	CA-CB	V x10
Current	A L2	0C-0D	6C-6D	CC-CD	mA
Active Power	kW L2	0E-0F	6E-6F	CE-CF	w
Reactive Power	Kvar L2	10-11	70-71	D0-D1	w
Power Factor	PF L2	12-13	72-73	D2-D3	x 100
Voltage Phase	V L3	14-15	74-75	D4-D5	V x10
Current	A L3	16-17	76-77	D6-D7	mA
Active Power	kW L3	18-19	78-79	D8-D9	W
Reactive Power	Kvar L3	1A-1B	7A-7B	DA-DB	W
Power Factor	PF L3	1C-1D	7C-7D	DC-DD	x 100
Active Power III	kW III	1E-1F	7E-7F	DE-DF	w
Inductive Power III	KvarL III	20-21	80-81	E0-E1	w
Capacitive Power III	KvarC III	22-23	82-83	E2-E3	w
Cos φ III	Cos φ III	24-25	84-85	E4-E5	x 100
Power Factor III	PF III	26-27	86-87	E6-E7	x 100
Frequency	Hz	28-29	88-89	E8-E9	Hz x 10
Voltage Line L1-L2	V12	2A-2B	8A-8B	EA-EB	V x10
Voltage Line L2-L3	V23	2C-2D	8C-8D	EC-ED	V x10
Voltage Line L3-L1	V31	2E-2F	8E-8F	EE-EF	V x10
% THD V L1	%THD VL1	30-31	90-91	F0-F1	% x 10
% THD V L2	%THD VL2	32-33	92-93	F2-F3	% x 10
% THD V L3	%THD VL3	34-35	94-95	F4-F5	% x 10
% THD A L1	%THD AL1	36-37	96-97	F6-F7	% x 10
% THD A L2	%THD AL2	38-39	98-98	F8-F9	% x 10
% THD A L3	%THD AL3	3A-3B	9A-9B	FA-FB	% x 10
Apparent Power III	KvaIII	42-43	A2-A3	102-103	w
Maximum Demand	Md (Pd)	44-45	A4-A5	104-105	w/VA/mA
Three Phase Current (average)	A_AVG	46-47	A6-A7	106-107	mA
Neutral Current	In	48-49	A8-A9	108-109	mA
Maximum Demand A2	Md (Pd)	52-53	B2-B3	112-113	mA
Maximum Demand A3	Md (Pd)	54-55	B4-B5	114-115	mA

MODBUS VARIABLES					
Magnitude	Symbol	Instant	Maximum	Minimum	Unit
Active Energy	<i>kWh III</i>	3C-3D	9C-CD	FC-FD	w-h
Inductive Reactive Energy	<i>kvarL-h III</i>	3E-3F	9E-9F	FE-FF	w-h
React. Energy Capacitive	<i>kvarC-h III</i>	40-41	A0-A1	100-101	w-h
Apparent Energy	<i>kVA-h III</i>	56-57	B6-B7	116-117	w-h
Active energy Generated	<i>kWh III (-)</i>	58-59	B8-B9	118-119	w-h
Inductive energy generated	<i>kvarL-h III (-)</i>	5A-5B	BA-BB	11A-11B	w-h
Capacit. Energy Generated	<i>kvarC-h III (-)</i>	5C-5D	BC-BD	11C-11D	w-h
Apparent Energy Generated	<i>kVA-h III (-)</i>	5E-5F	BE-BF	11E-11F	w-h

*Recordings available in HAR model					
MODBUS VARIABLES					
Magnitude	Symbol	L1	L2	L3	Unit
<b>Harmonic decomposition in VOLTAGE</b>		Instant			
RMS Current	V	2AE-2AF	2CC-2CD	2EA-2EB	Vx10
Harmonic 2		2B0-2B1	2CE-2CF	2EC-2ED	%
Harmonic 3		2B2-2B3	2D0-2D1	2EE-2EF	%
Harmonic 4		2B4-2B5	2D2-2D3	2F0-2F1	%
Harmonic 5		2B6-2B7	2D4-2D5	2F2-2F3	%
Harmonic 6		2B8-2B9	2D6-2D7	2F4-2F5	%
Harmonic 7		2BA-2BB	2D8-2D9	2F6-2F7	%
Harmonic 8		2BC-2BD	2DA-2DB	2F8-2F9	%
Harmonic 9		2BE-2BF	2DC-2DD	2FA-2FB	%
Harmonic 10		2C0-2C1	2DE-2DF	2FC-2FD	%
Harmonic 11		2C2-2C3	2E0-2E1	2FE-2FF	%
Harmonic 12		2C4-2C5	2E2-2E3	300-301	%
Harmonic 13		2C6-2C7	2E4-2E5	302-303	%
Harmonic 14		2C8-2C9	2E6-2E7	304-305	%
Harmonic 15		CA-CB	2E8-2E9	306-307	%
<b>Harmonic decomposition in CURRENT</b>		Instant			
RMS current	A	1F4-1F5	212-213	230-231	mA
Harmonic 2		1F6-1F7	214-215	232-233	%
Harmonic 3		1F8-1F9	216-217	234-235	%
Harmonic 4		1FA-1FB	218-219	236-237	%
Harmonic 5		1FC-1FD	21A-21B	238-239	%
Harmonic 6		1FE-1FF	21C-21D	23A-23B	%
Harmonic 7		200-201	21E-21F	23C-23D	%
Harmonic 8		202-203	220-221	23E-23F	%
Harmonic 9		204-205	222-223	240-241	%
Harmonic 10		206-207	224-225	242-243	%
Harmonic 11		208-209	226-227	244-245	%
Harmonic 12		20A-20B	228-229	246-247	%
Harmonic 13		20C-20D	22A-22B	248-249	%
Harmonic 14		20E-20F	22C-22D	24A-24B	%
Harmonic 15		210-211	22E-22F	24C-24D	%



### Modbus Transport Protocols

The TC303 thermostats come with the option of Modbus communication for both 2- and 4-pipe configurations. Modbus is an open, widespread and well established serial communication protocol used within building automation. The support of Modbus communication allows simple integration of the TC303 thermostats to a building management system, using standard Modbus serial communications. The following models provide support for Modbus connectivity:

- TC303-3A2DPMS
- TC303-3A4DPMS

The TC303 thermostats communicate as a Modbus RTU slave device over a serial RS-485 connection, allowing for the transfer of real time data. The RS-485 communication parameters cannot be adjusted and are fixed as follows:

- 4800 bps Baud Rate
- 8 Data Bits
- Odd Parity
- 1 Stop Bit

The actual Modbus address of the thermostat can be set for each individual device on the RS-485 network. The address is defined via the parameter menu on the thermostat itself. The address range available is from 01 to 32 and is defined under menu item 04 of the parameter settings. For full details on this and other parameter setting options available, please refer to the relevant section of this installation guide.

Connection to the RS-485 network is made via dedicated terminals on the back of the thermostat and marked A (+) and B (-).

The following Modbus register types and formats are supported:

Function Code	Register Type	Data Type	Data Format
01	Read Coils	Boolean	Binary/Digital
02	Read Discrete Inputs	Boolean	Binary/Digital
03	Read Holding Registers	Word	16-Bit Unsigned Integer
04	Read Input Registers	Word	16-Bit Unsigned Integer
06	Write Single Register	Word	16-Bit Unsigned Integer

### Function Code 01

Reg. Address	Register Description	Value Definition
5	Fan Speed Status - High	0 = Off, 1 = On
6	Fan Speed Status - Medium	0 = Off, 1 = On
7	Fan Speed Status - Low	0 = Off, 1 = On

### Function Code 02

Reg. Address	Register Description	Value Definition
3	Embedded Temperature Sensor Status	0 = OK, 1 = Fault
4	Remote Temperature Sensor Status	0 = OK, 1 = Fault

### Function Codes 03 and 06

Reg. Address	Register Description	Definition
3	Thermostat Mode	0 = Off 1 = On 2 = Frost Protection <sup>1</sup>
4	Operating Mode	1 = Cool 2 = Heat 3 = Ventilation
5	Room Temperature Set-point	Temperature (5 to 35 °C)
6	Fan Mode	0 = High 1 = Medium 2 = Low 3 = Auto
7	Unoccupied Room Temperature Set-point (Cooling Mode)	Temperature (22 to 32 °C)
8	Unoccupied Room Temperature Set-point (Heating Mode)	Temperature (10 to 21 °C)
9	Sleep Mode	0 = Disable 1 = Enable
10	Eco Mode	0 = Disable 1 = Enable
11 <sup>1</sup>	Occupancy Status	0 = Unoccupied 1 = Occupied
12	Unoccupied Fan Speed Mode	0 = High 1 = Medium 2 = Low
13	Keypad Status	0 = Unlocked 1 = Locked
19 <sup>1</sup>	Heating Control Signal	Output Voltage 0 ... 10 Vdc
20 <sup>1</sup>	Cooling Control Signal	Output Voltage 0 ... 10 Vdc

1. Read only.

### Function Code 04

Reg. Address	Register Description	Definition
1	Actual Room Temperature	Temperature (0 to 50 °C)

## WELLPRO MODULE

### MODBUS RTU REGISTER ADDRESS

Version 1.42A

MODULE	TYPE	FUNCTION CODE	REGISTER TYPE	REGISTER ADDRESS
WP8024ADAM	DI	02	1X	10001-10008
	RY	01/05/15	0X	00001-00004
WP8025ADAM	RY	01/05/15	0X	00001-00008
WP8026ADAM	DI	02	1X	10001-10016
WP8027ADAM	DO	01/05/15	0X	00001-00016
WP8028ADAM	DI	02	1X	10001-10008
	DO	01/05/15	0X	00001-00008
WP3066ADAM	DS	03	4X	40001-40008
WP3076ADAM	AO	03/06	4X	40001
WP3082ADAM	AI	03	4X	40001-40008
WP3084ADAM	VI	03	4X	40001-40008
WP9038ADAM	AI	03	4X	40001-40006
	DI	02	1X	10001-10004
	DO	01/05/15	0X	00001-00004
WP9065ADAM	T	03	4X	40001-40004
	DI	02	1X	10001-10004
	DO	01/05/15	0X	00001-00004

上海纬朴电气科技有限公司  
[www.shwellpro.com](http://www.shwellpro.com)

Lampiran (5) Modbus Register Wellpro WP9038ADAM

Drive Saya - Google Drive x PDF ke Gambar - Ubah P x modbus m221 - Penelus x What are the supported x terjemahan - Penelusur x

https://www.schneider-electric.com/en/faqs/FA249614/

€69.48 +0.12% My Favorites My Documents Global EN FR Partner Portal

Life Is On | **Schneider Electric** | Search

PRODUCTS SOLUTIONS SERVICES SUPPORT ABOUT US

## What are the supported Modbus Function Codes for a M221 controller?

This table lists the function codes supported by both serial Modbus and Modbus TCP and their effect on controller memory variables:

Supported Modbus Function Code	Supported Sub-Function Code	Description
1 (0x01)	-	Read multiple internal bits %M
2 (0x02)	-	Read multiple internal bits %M
3 (0x03)	-	Read multiple internal registers %MW
4 (0x04)	-	Read multiple internal registers %MW
5 (0x05)	-	Force single internal bit %M
6 (0x06)	-	Write single internal register %MW
8 (0x08)	0 (0x00), 10 (0x0A)...18 (0x12)	Diagnostics
15 (0x0F)	-	Write multiple internal bits %M
16 (0x10)	-	Write multiple internal registers %MW
23 (0x17)	-	Read/write multiple internal registers %MW
43 (0x2B)	14 (0x0E)	Read device identification (regular service)

**Note:** For function code 5 and 6 you must use the EXCH function. These two function codes are not available with the WRITE\_VAR function block

Lampiran (6) *Modbus Register PLC Modicon M221*

# Perancangan Simulator Telekontrol Menggunakan Modul *Schneider Automation Server* Sebagai Kontrol *Multi Modbus Device*

Muhammad Mirza<sup>1)</sup>, Rohana<sup>2)</sup>, Muhammad Syafril<sup>3)</sup>

<sup>1)</sup>Mahasiswa Teknik Elektro Universitas Muhammadiyah Sumatera Utara

<sup>2,3)</sup>Dosen Pembimbing Teknik Elektro Universitas Muhammadiyah Sumatera Utara Medan

Email: [mirzaanwar96@gmail.com](mailto:mirzaanwar96@gmail.com)

## ABSTRAK

*MODBUS Protocol* merupakan protokol komunikasi yang berperan penting terhadap alat – alat kendali elektronis. Penelitian ini bertujuan untuk menganalisa reliabilitas/kehandalan *Automation Server* dan kekuatan sinyal nirkabel dalam mengawasi dan mengendalikan perangkat *slave* (*Power meter, Module I/O, Thermostat, PLC*) serta menganalisa penyesuaian pembacaan *Modbus point* masing – masing register type dari masing – masing perangkat *slave*. Simulasi yang dilakukan dengan mengatur nilai baudrate dan transfer rate serta jumlah *Modbus Point* pada masing – masing perangkat *slave* yang berbeda-beda setiap port-nya pada *Automation Server* dan simulasi dilakukan dengan mengatur jarak nirkabel antara smart device dan *TP-Link* serta simulasi dilakukan dengan pemrograman dan grafis yang sederhana untuk melakukan pengawasan dan pengendalian. Hasil penelitian ini menunjukkan bahwa (1) semakin jauh jarak antara smart device dan *TP-Link MR3420*, sensitivitas kekuatan sinyal (*dbm*) semakin kecil. Kekuatan sinyal mengalami penurunan 94,7% pada jarak 150 meter. Semakin jauh jarak antara smart device dan *TP-Link MR3420*, kualitas sinyal semakin kecil. Kualitas sinyal mengalami penurunan 48,5% pada jarak 150 meter (2) semakin besar nilai baud rate dan transfer rate maka receive timeout semakin kecil, dan untuk penyesuaian pembacaan nilai input/output baik digital maupun analog pada perangkat *slave* adalah dengan mengatur tipe register pada masing – masing *Modbus Point* (3) pemrograman dan pembuatan grafis serta kombinasi dengan beberapa perangkat *slave Modbus* berhasil dilakukan dengan perangkat lunak internal dan teknik *Binding*. Indikator keberhasilan program dan grafis dinilai melalui penyesuaian pembacaan dan pengendalian *Input/Output* baik digital maupun analog pada perangkat-perangkat *slave Modbus*.

**Kata kunci :** *MODBUS Protocol, Receive Timeout, Kekuatan sinyal nirkabel*

## I. PENDAHULUAN

### 1. Latar Belakang

Di era serba otomatisasi dan *monitoring* saat ini, perangkat kendali dan perangkat *monitoring* merupakan sebuah komponen utama yang paling banyak digunakan dalam dunia industri dan gedung komersial. Dalam penggunaan peralatan kendali dan peralatan *monitoring* terkadang menimbulkan biaya besar yaitu pada hal operasional pengkabelan yang menghubungkan antara perangkat – perangkat *monitoring* dan perangkat – perangkat kendali ke komputer operator. Pengawasan dan pengendalian pada perangkat kendali dan perangkat *monitoring* tersebut dapat dilakukan

dengan dua metode, yaitu menggunakan sistem pengkabelan (*wired*) dan menggunakan sistem nirkabel (*wireless*). Untuk melakukan pengawasan dan pengendalian secara nirkabel dibutuhkan sebuah alat yang biasa disebut ‘*Router*’, dimana akan dihubungkan dengan ‘*Automation Server*’ yang akan mengawasi dan mengendalikan perangkat *slave Modbus* melalui sebuah protokol komunikasi yaitu ‘*MODBUS Protocol*’. Penelitian ini ditujukan untuk mensimulasikan sebuah alat berupa modul *Automation Server* yang dibuat oleh pabrikan *Schneider Electric* untuk melakukan pengawasan (*monitoring*) dan pengendalian (*controlling*) multi perangkat

*Modbus* secara nirkabel melalui router TP-Link MR3420.

## 2. Tujuan Penelitian

Tujuan dari penelitian ini adalah sebagai berikut:

1. Menganalisa nilai kekuatan dan kualitas sinyal (*signal strength*) nirkabel antara *Smart Device* dan *TP-Link MR3420* dalam mempermudah pengawasan dan pengendalian.
2. Menganalisa kecepatan *Automation Server* sebagai *Master* dalam merespon multi perangkat *MODBUS* dan menganalisa *Register Type* perangkat *slave* dalam proses *monitoring* (pengawasan) dan *controlling* (pengendalian) untuk dapat dibaca serta mencegah terjadinya *crash timeout error respon*.
3. Menganalisa pemrograman dan pembuatan grafis serta mengkombinasikannya dengan beberapa perangkat *slave MODBUS* untuk melakukan pengawasan dan pengendalian.

## II. TINJAUAN PUSTAKA

### 1. Modbus

*MODBUS* adalah protokol komunikasi *serial* yang diterbitkan oleh Modicon pada 1979 untuk diaplikasikan pada PLC. Kemudian protokol ini telah menjadi standar protokol komunikasi di industri, dan sekarang *MODBUS* merupakan protokol komunikasi dua-arah yang paling umum digunakan sebagai media penghubung dengan perangkat industri atau media elektronik lainnya dengan computer (*Schneider Electric*, 2015).

### 2. WLAN

Wi-Fi (*Wireless Fidelity*) atau lebih dikenal dengan WLAN (*Wireless Local Area Network*) merupakan teknologi jaringan wireless yang ditujukan untuk menghubungkan beberapa terminal berbasis IP (PC, *notebook* atau PDA) dalam suatu area LAN (*Local Area Network*). Sesuai dengan namanya yaitu *wireless*, berarti tanpa kabel, WLAN adalah jaringan lokal yang tidak menggunakan kabel (Wibisono, 2008).

### 3. Perangkat Keras

Fungsi perangkat keras (*hardware*) adalah memberikan masukan, mengolah dan menampilkan keluaran, dan menjalankan suatu perintah. Berikut perangkat keras yang digunakan.

- a. *Automation Server*
- b. *Power Analyzer CVM NRG-96*
- c. *Thermostat TC300*
- d. *WELLPRO WP9038ADAM*
- e. *PLC M221*
- f. *TP-LINK MR3420*
- g. *Current Transfotmer*

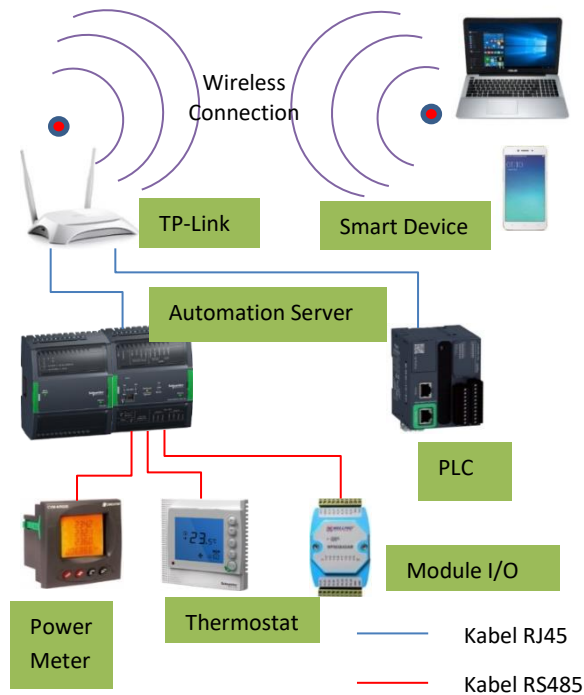
### 4. Software

Perangkat Lunak (*Software*) adalah sekumpulan data elektronik yang disimpan dan diatur oleh komputer, dapat berupa program atau instruksi yang akan menjalankan suatu perintah. Berikut perangkat lunak yang digunakan.

- a. *Lizard Wifi Scanner*
- b. *Wifi NetGear Analytic*
- c. *SBO Workstation*
- d. *SoMachine Basic*
- e. *Wellpro Debugging*
- f. *SBO Tech Tool*
- g. *SBO Device Administration*

### III. PERANCANGAN SISTEM

*Automation Server* akan mengawasi dan mengontrol beberapa perangkat *slave/client* seperti PLC, Modul I/O, *Thermostat*, dan *Power Meter*.



Gambar 1. Blok Diagram Sistem

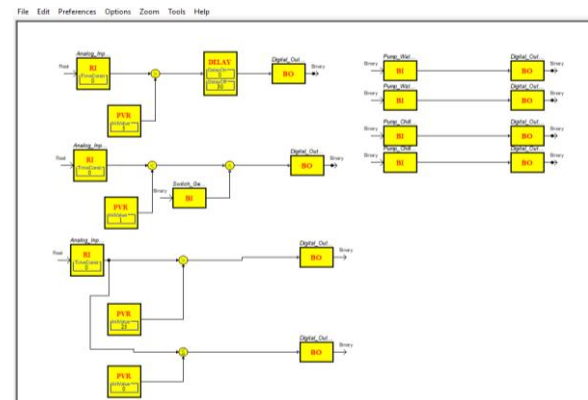
Dari gambar diatas ditunjukkan bahwa terdapat perangkat pintar (*smart device*) yaitu laptop dan ponsel pintar yang terhubung secara nirkabel dengan *TP-Link*. Kemudian *TP-Link* terkoneksi dengan *Automation Server* dan PLC menggunakan kabel RJ45. *Automation Server* terhubung dengan beberapa perangkat *slave/client* yang akan diawasi dan dikontrol. Laptop ataupun ponsel pintar dapat mengendalikan dan mengawasi semua perangkat yang terhubung ke *Automation Server*.

Perancangan sistem dilakukan melalui tiga tahap, yaitu sebagai berikut.

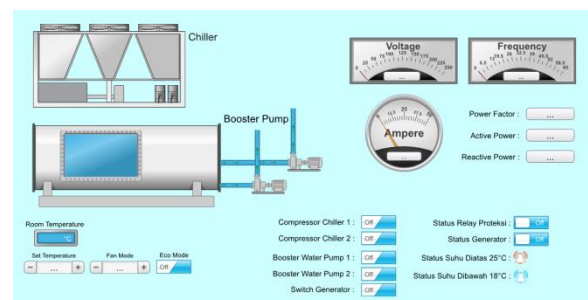
1. Perancangan perangkat keras (*hardware*).

2. Perancangan perangkat lunak (*software*).
3. Pemrograman dan pembuatan grafis.

Pembuatan program dan grafis pada simulasi ini ditujukan untuk membaca parameter pada masing – masing perangkat. Berikut program dan grafis pengendalian multi Modbus *device*.



Gambar 2. Program Kontrol MODBUS Device



Gambar 3. Grafis Kontrol MODBUS Device

### IV. ANALISA DAN HASIL

#### 1. Pengujian Kekuatan Sinyal WLAN (*Wireless Local Area Network*)

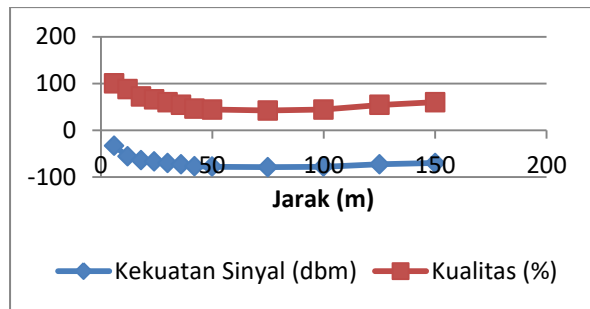
Pengujian kekuatan sinyal dilakukan untuk mengetahui kekuatan sinyal nirkabel antara *smart device* dan *TP-Link* untuk melakukan pengendalian dan pengawasan multi *MODBUS Device* yang terhubung dengan *Automation Server*. Pengujian dilakukan menggunakan perangkat lunak *Lizard Wifi Scanner* pada PC dan *Netgear Wifi Analytics* pada *smartphone*, pengujian dilakukan pada kondisi minim hambatan.



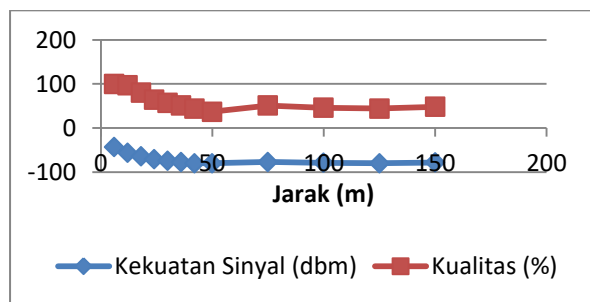
Hasil pengujian dapat dilihat pada tabel dan gambar grafik dibawah ini.

Tabel 1. Hasil kekuatan sinyal WLAN

Jarak (meter)	Personal Computer		Smartphone	
	Kekuatan sinyal (dbm)	Kualitas (%)	Kekuatan sinyal (dbm)	Kualitas (%)
6	-33	100	-43	100
12	-56	88	-56	97
18	-64	72	-64	80
24	-67	66	-71	64
30	-70	60	-74	57
36	-73	54	-77	51
42	-77	46	-80	44
50	-78	44	-80	37
75	-79	42	-77	51
100	-78	44	-79	46
125	-73	54	-80	44
150	-70	60	-78	48



Gambar 4. Grafik kekuatan sinyal dan kualitas sinyal antara laptop dan TP-Link MR3420.



Gambar 5. Grafik kekuatan sinyal dan kualitas sinyal antara smartphone dan TP-Link MR3420.

## 2. Pengujian Konfigurasi Kecepatan Automation Server dan Register Type.

Pengujian konfigurasi dilakukan untuk menguji keberhasilan konfigurasi perangkat terhadap nilai *type register*,

batas *poll transmit* dan *receive Modbus Point* masing – masing port komunikasi A dan B serta *Port Ethernet* antara *Master* dan *Slave*.

### a) Analisa kecepatan *Receive Timeout* (ms) pada *Port COM A*

Nilai *receive timeout* yang dibutuhkan untuk Port Com – A pada baud rate 4800 setelah melakukan inisialisasi nilai *Modbus Point* pada perangkat *Modbus* dapat dihitung melalui persamaan (2.1) dan persamaan (2.2) sebagai berikut:

$$T_{3,5} = 3,5 \times \frac{\text{Jumlah bit tiap karakter}}{\text{Baud Rate Com-A}} \dots\dots\dots(2.1)$$

$$T_{3,5} = 3,5 \times \frac{11 \text{ bit}}{4800 \text{ bps}}$$

$$T_{3,5} = 0,008020833333 \text{ s} = 8,02 \text{ ms}$$

Port COM – A memiliki 2 slave yaitu Power meter dan Thermostat yang masing – masing memiliki jumlah *Modbus point* sebanyak (6) enam dan (4) empat. Apabila kuantitas karakter sama dengan kuantitas *Modbus point*, maka nilai *receive timeout* dapat dihitung melalui persamaan (2.2) sebagai berikut:

$$\text{Receive Timeout} = T_{3,5} \times \text{Jumlah Karakter} \dots\dots\dots(2.2)$$

$$\text{Receive Timeout} = 8,02 \times (6 + 4)$$

$$\text{Receive Timeout} = 8,02 \times 10 = 80,2 \text{ ms}$$

### b) Analisa kecepatan *Receive Timeout* (ms) pada *Port COM B*

Nilai *receive timeout* yang dibutuhkan untuk Port Com – B pada baud rate 9600 setelah melakukan inisialisasi nilai *Modbus Point* pada perangkat *Modbus* dapat dihitung melalui persamaan (2.1) dan persamaan (2.2) sebagai berikut:

$$T_{3,5} = 3,5 \times \frac{\text{Jumlah bit tiap karakter}}{\text{Baud Rate Com-B}} \dots\dots\dots(2.1)$$

$$T_{3,5} = 3,5 \times \frac{11 \text{ bit}}{9600 \text{ bps}}$$

$$T_{3,5} = 0,00401 \text{ s} = 4,01 \text{ ms}$$

Port COM – B memiliki 1 slave yaitu Wellpro Module I/O yang memiliki jumlah Modbus point sebanyak (2) dua . Apabila nilai karakter sama dengan nilai Modbus point, maka nilai receive timeout sebagai berikut.

$$Receive\ Timeout = T_{3,5} \times \text{Jumlah Karakter} \dots (2.2)$$

$$Receive\ Timeout = 4,01 \times ( 2 ) = 8,02\ ms$$

**c) Analisa kecepatan *Receive Timeout* (ms) pada *Port Ethernet***

Nilai receive timeout yang dibutuhkan untuk Port Ethernet setelah melakukan inisialisasi nilai Modbus Point pada perangkat Modbus dapat dihitung melalui persamaan (2.1) dan persamaan (2.2) sebagai berikut:

Pada *transfer rate* 10 mbps.

$$T_{3,5} = 3,5 \times \frac{\text{Jumlah bit per karakter} + \text{Packet data unit}}{\text{Transfer Rate}} \quad (2.1)$$

$$T_{3,5} = 3,5 \times \frac{11\ bit + 56\ bit}{10\ mbps}$$

$$T_{3,5} = 0,00002345\ s = 0,02345\ ms$$

Pada *transfer rate* 100 mbps.

$$T_{3,5} = 3,5 \times \frac{\text{Jumlah bit per karakter} + \text{Packet data unit}}{\text{Transfer Rate}} \quad (2.1)$$

$$T_{3,5} = 3,5 \times \frac{11\ bit + 56\ bit}{100\ mbps}$$

$$T_{3,5} = 0,000002345\ s = 0,002345\ ms$$

Port Ethernet memiliki 1 slave yaitu PLC M221 yang memiliki jumlah Modbus point sebanyak (8) delapan . Apabila nilai karakter sama dengan nilai Modbus point, maka nilai receive timeout sebagai berikut.

Pada *transfer rate* 10 mbps.

$$Receive\ Timeout = T_{3,5} \times \text{Jumlah Karakter} \dots (2.2)$$

$$Receive\ Timeout = 0,02345 \times ( 8 ) = 0,1876\ ms$$

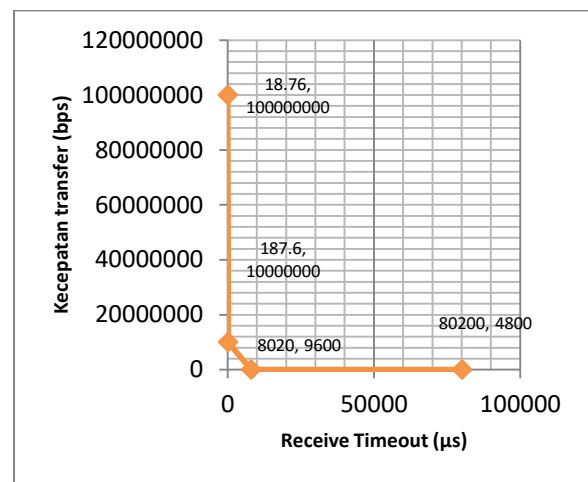
Pada *transfer rate* 100 mbps.

$$Receive\ Timeout = T_{3,5} \times \text{Jumlah Karakter} \dots (2.2)$$

$$Receive\ Timeout = 0,002345 \times ( 8 ) = 0,01876\ ms$$

Tabel 2. Perbandingan receive timeout pada masing – masing port komunikasi MODBUS.

Port	Jumlah karakter	Kecepatan transfer (bps)	Receive timeout (µs)
COM- A	10	4.800	80200
COM- B	2	9.600	8020
Ethernet	8	10.000.000	187,6
	8	100.000.000	18,7



Gambar 6. Grafik Perbandingan *receive timeout* pada masing – masing port komunikasi MODBUS.

**d) Pengujian *Register Type***

Pengujian *register type* dilakukan untuk menguji dan menyesuaikan nilai pembacaan sesuai referensi nilai yang tampil pada perangkat keras *Modbus*.

Tabel 3. Pengujian register type pada perangkat Power meter Circutor CVM-NRG96.

Modbus Point	Kode Fungsi	Register Number	Tipe Register
Active Power (kW)	03	05	32 bit unsigned swapped
Current (A L1)	03	03	32 bit unsigned swapped
Frequency (Hz)	03	41	32 bit unsigned swapped
Power Factor (PF L1)	03	09	32 bit unsigned swapped
Reactive Power (kVar)	03	07	32 bit unsigned swapped

Modbus Point	Kode Fungsi	Register Number	Tipe Register
Voltage (V L1)	03	01	32 bit unsigned swapped

Tabel 4. Pengujian *register type* pada perangkat *Thermostat TC-300*

Modbus Point	Kode Fungsi (read)	Register Number	Tipe register
Eco Mode	03	10	16 bit unsigned
Fan Mode	03	06	16 bit unsigned
Room Temperature	04	01	16 bit unsigned
Set Temperature	03	05	16 bit unsigned

Tabel 5. Pengujian *register type* pada perangkat *Wellpro Module I/O*

Modbus Point	Kode Fungsi (read)	Kode Fungsi (write)	Register Number	Tipe register
Digital Input 1	02	None	01	Digital Coil 16 bit unsigned
Digital Output 1	01	15	1	Digital Coil 16 bit unsigned

Tabel 6. Pengujian *register type* pada perangkat *PLC M221*

Modbus Point	Kode Fungsi (read)	Kode Fungsi (write)	Register Number	Tipe register
Digital Output 1	02	15	00	Digital Coil
Digital Output 2	02	15	01	Digital Coil
Digital Output 3	02	15	02	Digital Coil
Digital Output 4	02	15	03	Digital Coil
Digital Output 5	02	15	04	Digital Coil
Digital Output 6	02	15	05	Digital Coil
Digital Output 7	02	15	06	Digital Coil
Digital Output 7	02	15	07	Digital Coil

### 3. Pengujian Program dan Grafis

#### a) Pengujian Program

Pengujian program dilakukan untuk menguji keberhasilan Fungsi Diagram Blok (*Function Block Diagram*) program dan fungsi dasar *Modbus* pada tiap perangkat.

Tabel 7. Hasil pengujian respon fungsi dasar *Modbus* program *Over Current*.

Nama	Alamat Binding	Perangkat Modbus	Respon
Analog_Input_1	Current (A L1)	Power Meter	OK
Digital_Out_1	Digital Output 1	PLC M221	OK
Digital_Out_1	Digital Output 1	Wellpro	OK

Tabel 8. Hasil pengujian respon fungsi dasar *Modbus* program *Starter Generator*.

Nama	Alamat Binding	Perangkat Modbus	Respon
Analog_Input_2	Voltage (V L1)	Power Meter	OK
Digital_Out_8	Digital Output 8	PLC M221	OK

Tabel 9. Hasil pengujian respon fungsi dasar *Modbus* program *Temperature Alarm*.

Nama	Alamat Binding	Perangkat Modbus	Respon
Analog_Input_3	Room Temperature	Thermostat	OK
Digital_Out_2	Digital Output 2	PLC M221	OK
Digital_Out_3	Digital Output 3	PLC M221	OK

Tabel 10. Hasil pengujian respon fungsi dasar *Modbus* program *Button Booster Water Pump dan Compressor Chiller*.

Nama	Alamat Binding	Perangkat Modbus	Respon
Pump_Water_1	Button Booster Pump_1	Grafis	OK
Pump_Water_2	Button Booster Pump_2	Grafis	OK
Pump_Chiller_1	Button Compressor Chiller_1	Grafis	OK
Pump_Chiller_2	Button Compressor Chiller_2	Grafis	OK
Digital_Out_4	Digital Output 4	PLC M221	OK
Digital_Out_5	Digital Output 5	PLC M221	OK
Digital_Out_6	Digital Output 6	PLC M221	OK
Digital_Out_7	Digital Output 7	PLC M221	OK

#### b) Pengujian Grafis

Pengujian grafis dilakukan untuk menguji keberhasilan grafis dalam melakukan pengawasan dan pengendalian terhadap nilai – nilai pada perangkat *Modbus*.

Tabel 11. Hasil pengujian grafis dalam melakukan pengawasan dan pengendalian melalui smartphone.

Komponen Grafis	Alamat Binding	Modbus Point	Perangkat	Hasil
Dial (Voltage)	Voltage	Voltage (V L1)	Power meter	Sesuai
Dial (Frequency)	Frequency	Frequency (Hz)	Power meter	Sesuai
Dial (Current)	Current	Current (A L1)	Power meter	Sesuai
Analog Value 1	Voltage	Voltage (V L1)	Power meter	Sesuai
Analog Value 2	Frequency	Frequency (Hz)	Power meter	Sesuai
Analog Value 3	Current	Current (A L1)	Power meter	Sesuai
Display	Power Factor	Power Factor	Power meter	Sesuai
Display	Active Power	Active Power	Power meter	Sesuai
Display	Reactive Power	Reactive Power	Power meter	Sesuai
Analog Set Point 1	Set Temperature	Set Temperature	Termostat	Sesuai
Analog Set Point 2	Set_FanMode	Fan Mode	Termostat	Sesuai
Slide Button Android Style 6	Button Eco Mode	Eco Mode	Termostat	Sesuai
AC Chiller – 2 Compressor	Motor Chiller 1	Digital Output 6	PLC M221	Sesuai
AC Chiller – 2 Compressor	Motor Chiller 2	Digital Output 7	PLC M221	Sesuai
Pump T2 Large Flow Right 1	Booster Pump 1	Digital Output 4	PLC M221	Sesuai
Pump T2 Large Flow Right 2	Booster Pump 2	Digital Output 5	PLC M221	Sesuai
DP Flow Right 1	Booster Pump 1	Digital Output 4	PLC M221	Sesuai
DP Flow Right 2	Booster Pump 2	Digital Output 5	PLC M221	Sesuai
DP Flow Up 1	Booster Pump 1	Digital Output 4	PLC M221	Sesuai
DP Flow Up 2	Booster Pump 2	Digital Output 5	PLC M221	Sesuai
Slide Button Android Style 1	Button Compressor Chiller_1	-	*Program	Sesuai
Slide Button Android Style 2	Button Compressor Chiller_2	-	*Program	Sesuai
Slide Button Android Style 3	Button Booster Pump_1	-	*Program	Sesuai
Slide Button Android Style 4	Button Booster Pump_2	-	*Program	Sesuai
Slide Button Android Style 5	Button Generator	-	*Program	Sesuai

## V. KESIMPULAN

- 1) Semakin jauh jarak antara smart device dan TP-Link MR3420, sensitivitas kekuatan sinyal (dbm) semakin kecil. Kekuatan sinyal mengalami penurunan 94,7% pada jarak 150 meter. Semakin jauh jarak antara smart device dan TP-Link MR3420, kualitas sinyal semakin kecil. Kualitas sinyal mengalami penurunan 48,5% pada jarak 150 meter.
- 2) Nilai untuk *receive timeout* pada masing – masing port komunikasi dapat diatur lebih besar nilainya dari hasil *Timer 3,5 Character* ( $T_{3,5}$ ). Pada Port COM-A (4800bps) diatur dengan nilai 100ms dimana standar perhitungan adalah 80,2ms. Pada Port COM-B (9600bps) diatur dengan nilai 100ms dimana standar perhitungan adalah 8,02ms. Pada Port Ethernet = Auto dimana standar perhitungan minimal sebesar 187,6 $\mu$ s. Semakin besar nilai *baudrate* dan *transfer rate*, maka semakin kecil nilai *receive timeout*.
- 3) Pemrograman dan pembuatan grafis serta kombinasi dengan beberapa perangkat *slave Modbus* berhasil dilakukan dengan perangkat lunak internal dan teknik Binding. Indikator keberhasilan program dan grafis dinilai melalui penyesuaian pembacaan dan pengendalian *Input/Output* baik digital maupun analog pada perangkat-perangkat *slave Modbus*.

## DAFTAR PUSTAKA

- Adriansyah, Andi dan Rizally Priatmadja. (2015). *Rancang Bangun Protocol MODBUS Pada KWH Meter Elektronik Tipe ION 8600 Untuk Memonitor Besaran Energi listrik Trafo Dengan Menggunakan Aplikasi Citect SCADA*. Jakarta. Repository Mercu Buana.
- Apriyanto, Benediktus Tri. (2016). *Aplikasi PLC Modicon M221 Untuk Smart Home Dengan HMI Berbasis Android*. Yogyakarta. Universitas Sanata Dharma.
- Ashtekar, Anjali S dkk. (2013). *Application of MODBUS to Communicate the PLC and Lab VIEW for Real Time Process Control*. India. *International Journal of Emerging Science and Engineering (IJESE)*.
- Circutor, SA. *Power Analyzer CVM-NRG96 : User Manual*. Barcelona, Spain.
- Fang, Yinlan dkk. (2013). *Research and Implementation of Collision Detection Based on Modbus Protocol*. Beijing, China. *North China University of Technology*.
- Gumilang, Ferdina Iqra dkk. *Rancang Bangun Jaringan Komunikasi Multi PLC dengan Platform Sistem SCADA-DCS*. Bandung. Politeknik Manufaktur Negeri Bandung.
- M-System CO., Ltd. *Modbus Protocol Reference Guide*. Minamitsumori. Osaka, Japan
- Marshal, Citra. (2012). *Rancang Bangun Sistem Monitoring Daya Pada Rumah Cerdas*. Depok. Universitas Indonesia.
- MODBUS Organization. (2002). *MODBUS over Serial Line Specification & Implementation guide*.
- MODBUS Organization. (2006). *MODBUS Messaging on TCP/IP Implementation Guide V1.0b*.
- MODBUS Organization. (2012). *MODBUS APPLICATION PROTOCOL SPECIFICATION V1.1b3*.
- MODICON, Inc., Industrial Automation Systems. (1996), *Modicon Modbus Protocol Reference Guide*. North Andover, Massachusetts 01845.
- Mulyanta, Edi S. (2005). *Pengenalan Protokol Jaringan Wireless Komputer*. Yogyakarta. Penerbit ANDI
- Nurhidayat, Eka. (2010). *Pengaruh Temperatur Lingkungan Kerja dan Harmonisa Terhadap Kinerja Transformator Arus*. Depok. Universitas Indonesia.
- Nurpadmi. *Studi Tentang Modbus Protokol Pada Sistem Kontrol Vol. 01 No. 02*. Cepu. Pusdiklat Migas.
- Pancoro, Bayu. (2009). *Curent Transformer (CT)*. Wordpress.
- Pangaribowo, Triyanto dan Hibnu Yulianda. (2016). *Sistem Monitoring Suhu Melalui Sistem Komunikasi Programmable Logic Controller To Personal Computer*. Jakarta. Repository Mercu Buana
- Putranto, Ichsan Edi dkk. (2015). *Implementasi Dan Analisis Protocol Modbus TCP Pada Smart Building Berbasis OPENMTC*. Indonesia. Telkom University.
- Ramandha, Audy. (2015). *Implementasi Scada Pada Android Dengan Simulator MODBUSPAL Studi Kasus Perusahaan X*. Jakarta. Universitas Islam Negeri Syarif Hidayatullah.
- Rockhim, Abdul. *Modul Konsep Jaringan BAB XV (WLAN)*. STMIK Yadika Bangil.
- Schneider Electric, Inc. (2015). *Guide to Open Protocols in Building Automation*. Polytec, France.
- Schneider Electric, Inc (2015). *Modicon M221 Logic Controllers*. Polytec, France.

Schneider Electric, Inc (2016). *SmartX Controller : Hardware Reference Guide*. Polytec, France.

Schneider Electric Asia Pasific, Inc (2011). *TC300 Series Networking Digital Fan Coil Thermostat*. Singapore.

Schneider Electric, Inc (2016). *TC300 Series Modulating Fan Coil Thermostat : Installation Intruaction*. Polytec, France.

Schneider Electric, Inc (2015). *SoMachine Basic : Operating Guide*. Polytec, France.

Shanghai Wellpro Electrical Technology Co., Ltd. *WP9038ADAM : User Manual*. Shanghai.

Shanghai Wellpro Electrical Technology Co., Ltd. *WELLPRO MODULE : MODBUS RTU REGISTER ADDRESS*. Shanghai.

Tiyono, Agus dkk. (2007). *Sistem Telekontrol SCADA Dengan Fungsi Dasar Modbus Menggunakan Mikrokontroler AT89S51 Dan Komunikasi Serial RS-485*. Semarang. Diponegoro University.

Wibisono, Gunawan dan Gunadi. 2008. *Mobile Broadband Tren Teknologi Wireless Saat ini dan Masa Datang*. Bandung. Penerbit Informatika Bandung.

### **Tentang Penulis**



Muhammad Mirza, lahir di Medan 08 Juni 1996. Menempuh pendidikan Sekolah Dasar di SD Negeri 060827 Medan Lulus tahun 2008 kemudian Sekolah Menengah

Pertama di SMP Negeri 3 Medan Lulus tahun 2011 dan Sekolah Menengah Atas di SMA Negeri 13 Medan Lulus tahun 2014. Kemudian melanjutkan Pendidikan Akademik di Universitas Muhammadiyah Sumatera Utara Program Studi Teknik

Elektro Strata-1 Lulus pada tahun 2018. Penulis dapat dihubungi via email di : [mirzaanwar96@gmail.com](mailto:mirzaanwar96@gmail.com)