

Rancang Bangun Kontrol *Fan Cooling Tower* Dimonitoring Oleh SCADA Wonderware

Fatahillah Fauzan Fradana^{1*}, Agung Yoke Basuki²

¹Teknik Elektro, Universitas Mercu Buana, Jakarta

²Teknik Elektro, Universitas Mercu Buana, Jakarta

*fauzanfradana@gmail.com

Abstrak—Penggunaan mesin Chiller dengan beban tidak konstan mengakibatkan mesin start dan stop secara berulang dengan waktu yang relatif cepat karena *set point* suhu yang tercapai. Pada dasarnya mesin terdiri dari tiga bagian utama yaitu bagian penghembus udara dingin, pemindah panas air suplai ke air *cooling tower* dan *cooling tower*. Pola seperti ini dikhawatirkan dapat menyebabkan *lifetime* mesin berkurang. Salah satu rancangan untuk mengurangi resiko ini adalah menggunakan sistem kontrol pada *fan cooling tower* yang dapat berputar sesuai dengan suhu air. Beberapa komponen seperti komponen pengontrol, pengindra dan unit pemonitor dibutuhkan dalam sistem ini. Pengontrol dapat berupa PLC, pengindra dapat berupa sensor RTD PT100 dan SCADA Wonderware untuk memonitor kinerja sistem. Agar PLC dapat berkomunikasi dengan SCADA, alamat memori PLC di *pooling* ke OPC melalui protokol Modbus RTU RS-485. Berdasarkan hasil pengujian yang telah dilakukan, akurasi dari sensor RTD yang digunakan memiliki rata-rata 0,11% dengan penyimpangan terbesar 0,28%. Nilai keluaran sensor 4-20 mA dapat *discaling* menjadi nilai display pada SCADA 0-100 °C dengan gap rata-rata 0,04 °C dan gap maksimal sebesar 0,12 °C jika dibandingkan dengan alat ukur. Sistem kontrol *fan cooling tower* dapat termonitoring oleh SCADA Wonderware dengan menampilkan suhu, alarm, pengoperasian yang dilakukan dan kecepatan *fan* yang dapat berubah sesuai keadaan suhu air *cooling tower* dengan melakukan pengaturan pada parameter PID secara *trial and error*.

Kata kunci—Cooling tower, Modbus RTU, OPC, RTD PT100, SCADA Wonderware.

DOI: 10.22441/jte.2025.v16i3.004

I. PENDAHULUAN

Banyak proses yang dilakukan didalam dunia industri. Proses tersebut tidak akan lepas dengan kebutuhan penunjang seperti pemanas, udara bertekanan, pendingin dll. Sebagai contoh, mesin Chiller yang dapat mensuplai udara dingin untuk berbagai kebutuhan seperti pendinginan panel, ruangan produksi dll. Secara umum, mesin Chiller mendinginkan air yang akan dikirim ke area yang membutuhkan suplai udara dingin. Air ini akan menjadi panas karena hawa dingin sudah dihembuskan oleh *evaporator*. Air panas ini selanjutnya bersirkulasi kembali ke mesin Chiller. Didalam mesin, panas air ditransfer ke air pendinginan. Air pendinginan akan disirkulasikan ke *cooling tower* untuk membuang hawa panas keudara oleh *fan cooling tower* [1]. Siklus ini terus berjalan sampai *set point* mesin tercapai.

Sejak mesin Chiller memiliki sistem otomatis yang dapat berhenti bekerja ketika output sesuai atau kurang dari *set point* banyak konstruksi *fan cooling tower* dioperasikan dengan cara manual untuk menyalakan atau mematikan [13]. Setelah dilakukan studi dan pengamatan, kelemahan sistem ini adalah ketika kebutuhan udara dingin di area suplai tidak konstan, mesin akan mengalami *start* dan *stop* secara berulang dengan waktu yang relatif cepat. Jika hal ini dibiarkan dikhawatirkan akan mengurangi *lifetime* mesin. Salah satu cara yang diusulkan untuk mengurangi resiko *start* dan *stop* ini adalah membuat sistem kontrol yang dapat membuat *fan cooling tower* berputar berdasarkan suhu air *cooling tower*. Beberapa komponen seperti pengontrol (PLC), sensor suhu (RTD PT100) dan SCADA Wonderware untuk memantau kinerja sistem digunakan dalam penelitian ini.

II. PENELITIAN TERKAIT

Pada bab ini dibahas mengenai penelitian sebelumnya, dasar teori dan penjelasan dari setiap komponen yang digunakan untuk mengetahui karakteristik komponen.

Setelah dilakukan pengamatan secara langsung, mesin Chiller akan stop jika *set point* sudah tercapai kemudian akan start lagi jika suhu menjauhi *set point*. Siklus start dan stop yang terjadi relatif cepat biasanya terjadi saat beban mesin ringan seperti saat waktu istirahat atau tidak ada kegiatan produksi. Karena *fan cooling tower* berputar dengan putaran yang konstan sedangkan beban mesin Chiller ringan menyebabkan mesin relatif mudah mencapai *set point*.

Sebagai pengontrol dari sistem yang akan dibuat digunakan PLC (*Programmable Logic Control*) karena dapat diprogram sesuai keinginan pengguna [6]. Untuk mengukur suhu digunakan sensor RTD PT100 karena termasuk sensor suhu yang dapat memberikan data ukur secara berkelanjutan yang dapat digunakan sebagai umpan balik ke sistem [10].

Penelitian ini merupakan pengembangan dari penelitian sebelumnya dimana konsepnya adalah membuat kontrol *fan cooling tower* yang awalnya dioperasikan secara manual dan berputar secara konstan menjadi otomatis berputar berdasarkan suhu air *cooling*. Hal yang membedakan dari penelitian sebelumnya adalah adanya integrasi dengan SCADA Wonderware dengan protokol komunikasi Modbus RTU dan OPC melalui serial RS-485. Beberapa komponen yang digunakan dalam penelitian ini dapat dijelaskan seperti berikut :

A. Power Supply

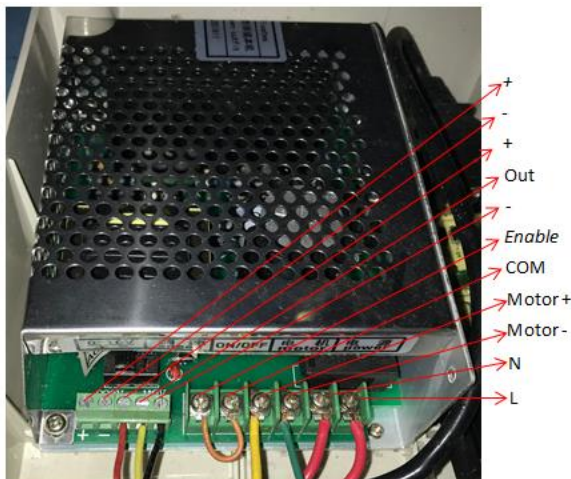
Power Supply digunakan untuk mencatu daya peralatan elektronik. Terdapat 2 power supply yang digunakan :

Power Supply 24 VDC, digunakan untuk mencatu daya PLC, sensor RTD PT 100 dan relay.



Gambar 1. Power Supply 24 VDC

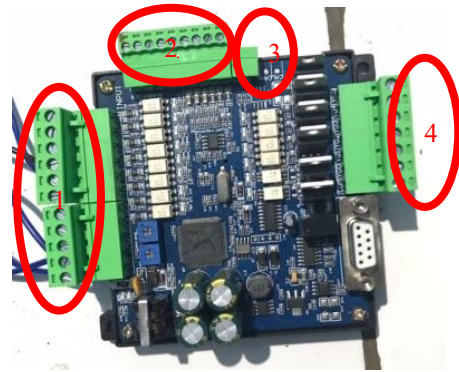
Power Supply motor DC, digunakan untuk mencatu daya motor DC yang memutar *fan cooling tower*. Power supply jenis ini memiliki pin analog untuk mengatur daya yang dikeluarkan.



Gambar 2. Power Supply Motor DC

B. PLC LE3U-14MT6AI2AO

Programmable Logic Control digunakan sebagai pengontrol karena kemampuannya yang dapat diprogram sesuai keinginan pengguna, dukungan beberapa bahasa pemrograman dan kemampuannya dalam berkomunikasi dengan perangkat lain [6]. PLC yang digunakan dalam penelitian ini adalah PLC Lolette LE3U-14MT6AI2AO karena memiliki fitur seperti gambar berikut :



Gambar 3. PLC LE3U-14MT6AI2AO

Keterangan :

1. Digital input, untuk menerima signal digital *high* atau *low* [14] yang nantinya akan dihubungkan dengan tombol start, stop dan *emergency*.
2. Analog input output, untuk menerima masukan signal analog 0-10 V atau 0-20 mA [14] yang nantinya dihubungkan dengan sensor RTD PT100 dan mengeluarkan signal analog 0-10 V [14] untuk mendrive power supply motor DC. PLC ini memiliki resolusi pembacaan signal analog sebesar 12-bit dimana range 0-10 V atau 0-20 mA dibaca sebagai nilai logika 0-4095.
3. RS-485, untuk komunikasi serial RS-485 [14] yang nantinya dihubungkan ke SCADA Wonderware dengan protokol Modbus RTU.
4. Digital output, untuk mengeluarkan signal digital *high* atau *low* [14] yang nantinya dihubungkan ke relay.

C. Sensor RTD PT100

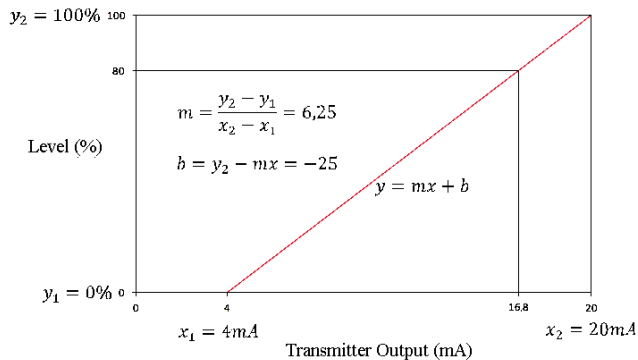
Resistance Temperature Detector adalah sensor suhu yang pengukurannya berdasarkan perubahan hambatan listrik pada logam yang dipengaruhi oleh perubahan suhu. Dikatakan PT100 karena sensor ini dikalibrasi pada 0°C dengan resistansi 100 Ω . Prinsip kerja dari sensor ini adalah mengubah nilai resistansi menjadi arus listrik. Perubahan nilai resistansi terhadap suhu terjadi secara linear. Artinya semakin suhu meningkat maka nilai resistansi juga meningkat [17]. Sensor RTD yang digunakan dalam penelitian ini mengeluarkan kuat arus 4-20 mA dengan range pengukuran 0-100 °C.



Gambar 4. Sensor RTD PT100

Keluaran sensor yang berupa 4-20 mA dihubungkan ke PLC yang kemudian dibaca sebagai nilai logika 0-4095 (resolusi 12-bit). Nilai logika inilah yang nanti dapat di *scaling* agar dapat

menampilkan suhu dengan range 0-100 °C. Rumus untuk melakukan *scaling* dapat dilihat pada ilustrasi berikut :



Gambar 5. Ilustrasi scaling

Dari gambar diatas dapat diamati bahwa nilai dengan range 4-20 mA dapat discaling menjadi nilai dengan range 0-100%. Sebagai contoh, 16,8 mA sama dengan nilai 80%. Perhitungan ni dapat dijelaskan dengan persamaan berikut

$$y = mx + b \quad (2.1)$$

Diketahui :

$$x_1 (\text{minimum mA}) = 4, \quad x_2 (\text{maksimum mA}) = 20$$

$$y_1 (\text{minimum logika}) = 0, \quad y_2 (\text{maksimum logika}) = 100$$

Maka:

$$m = \frac{y_2 - y_1}{x_2 - x_1}, \quad b = y_2 - mx_2$$

$$m = \frac{100 - 0}{20 - 4}, \quad b = 100 - (6,25 \cdot 20)$$

$$m = 6,25, \quad b = -25$$

$$y = mx + b$$

$$y = 6,25 \cdot 16,8 + (-25)$$

$$y = 105 + (-25)$$

$$y = 80$$

Dari dasar rumus diatas dapat juga dituliskan untuk melakukan *scaling* nilai 4-20 mA menjadi nilai logika 0-4095 :

$$y = \frac{(Data \text{ mA} - x_1)}{x_2 - x_1} \times (y_2 - y_1) + y_1 \quad (2.2)$$

Atau dengan menggunakan persamaan 2.2 dapat dijelaskan sebagai berikut :

$$y = \frac{(Data \text{ mA} - x_1)}{x_2 - x_1} \times (y_2 - y_1) + y_1$$

$$y = \frac{(16,8 - 4)}{20 - 4} \times (100 - 0) + 0$$

$$y = \frac{12,8}{16} \times 100 + 0$$

$$y = 80$$

D. Tombol / Button

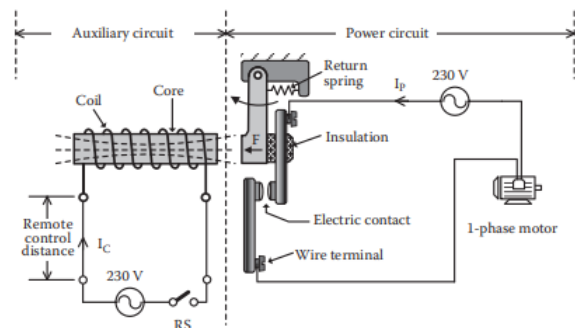


Gambar 6. Tombol

Tombol digunakan untuk mengoperasikan rangkaian listrik dengan cara menghubungkan atau memutuskan aliran arus listrik. Istilah ini biasa disebut dengan NO (*Normally Open*) adalah keadaan dimana dalam keadaan normal / tidak ditekan, tombol berada dalam keadaan tidak terhubung dan NC (*Normally Close*) adalah keadaan dimana dalam keadaan normal / tidak ditekan, tombol berada dalam keadaan terhubung [6]. Terdapat beberapa jenis tombol yang digunakan untuk penelitian ini, yaitu *Push Button*, tombol dengan cara pengoperasian ditekan kemudian akan otomatis kembali ke keadaan normal jika tidak ditekan. *Switch*, disebut juga dengan saklar dimana cara pengoprasiannya diputar dan keadaan ini akan ditahan sampai diputar kembali ke keadaan semula. *Emergency Button*, tombol dimana cara pengoprasiannya ditekan dan keadaan ini akan ditahan sampai *direlease* dengan cara memutar tombol.

E. Relay 24 VDC

Relay digunakan untuk menghandle peralatan listrik dengan suplai listrik yang relatif besar dengan kontrol listrik yang relative lebih kecil [6]. Jenis relay yang digunakan untuk penelitian ini adalah jenis elektromekanikal dengan bagian penting berupa :



Gambar 7. Konstruksi Relay

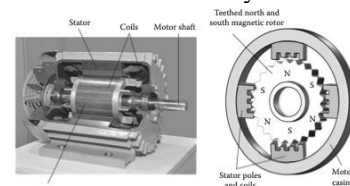
(Sumber : *Introduction To Industrial Automation*, Stamatios Manesis, George Nikolakopoulos : 2018)

Auxiliary Circuit, bagian ini terdiri dari koil dengan tegangan 24 VDC. Koil ini jika dialiri arus listrik akan menimbulkan elektromagnet untuk menarik *spring* dan menggerakkan kontak hubung pada bagian *power circuit*.

Power circuit, bagian ini terdiri dari *spring*, dan kontak hubung untuk menghubungkan sumber listrik ke beban

F. Motor DC

Motor DC adalah peralatan listrik yang mampu mengubah energi listrik menjadi energi gerak [6]. Motor DC mampu mengubah energi listrik menjadi energi gerak karena memiliki konstruksi khusus. Berikut ilustrasinya :



Gambar 8. Konstruksi Motor

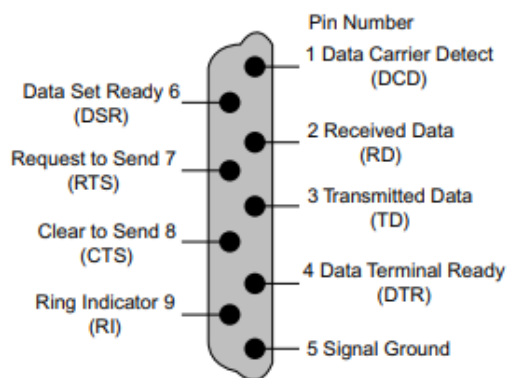
(Sumber : *Introduction To Industrial Automation*, Stamatiou Manesis, George Nikolakopoulos : 2018)

Bagian paling penting adalah adanya *stator* dan *rotor*. *Stator* adalah bagian motor yang tidak bergerak meliputi rangka, *yoke* dan kumparan medan jika motor tidak memiliki magnet permanen. *Rotor* adalah bagian motor yang bergerak meliputi armature, *shaft*, dan komutator. Listrik dialirkan ke *brush* yang terbuat dari karbon atau graphite menuju komutator. Komutator yang berupa *slip ring* dengan beberapa segmen yang terpisah oleh insulator yang sesuai dengan jumlah belitan pada armature diteruskan ke armature. Slip ring juga menjadi *switching* kutub. Pada bagian *stator* juga terjadi medan magnet (jika menggunakan kumparan medan, kumparan medan perlu mendapat suplai listrik tetapi jika menggunakan magnet permanen, suplai listrik tidak dibutuhkan karena sudah terjadi kemagnetan). Interaksi antar medan magnet stator dan rotorlah yang dimanfaatkan untuk menghasilkan putaran [6].

G. Serial RS-232

Komunikasi serial RS-232 pada penelitian ini digunakan untuk melakukan programming PLC yang dihubungkan ke komputer. Berikut beberapa karakteristik dari komunikasi RS-232 yang berhubungan dengan penelitian ini :

- Merupakan jenis *point-to-point*, hanya dapat terhubung ke satu peralatan dengan satu peralatan lain. Satu peralatan berperan sebagai *transmitter* (TD) dan satu peralatan berperan sebagai *receiver* (RD) [5].
- Panjang kabel maksimal mencapai 50 kaki (15,24 meter) [5].
- Memiliki kecepatan 20 – 115 Kb/s dengan *baud rate* 75, 110, 300, 1200, 2400, 4800, 9600, 19200, 38400, 57600, dan 115200 bps [5].
- Memiliki pengkabelan 2 Kabel (TD dan GND), 3 Kabel (TD, RD, dan GND), 5 Kabel (TD, RD, RTS, CTS, dan GND) [5].
- Jenis konektor yang digunakan adalah tipe DB9 [5].



Gambar 9. DB9

(Sumber : *Handbook Of Serial Communications Interfaces*, Louis E. Frenzel Jr, 2016)

Agar komputer dapat berkomunikasi dengan PLC melalui komunikasi RS-232, pada umumnya digunakan konverter RS-232 to USB.



Gambar 10. Konverter RS-232 to USB

H. Serial RS-485

Saat menggunakan komunikasi serial RS-232 masalah timbul karena PLC yang digunakan harus dihubungkan secara bersamaan dengan peralatan lain sedangkan RS-232 termasuk koneksi *point-to-point*. Disisi lain PLC yang digunakan juga memiliki port RS-485 untuk koneksi tambahan. Berikut karakteristik dari komunikasi RS-485 yang digunakan untuk penelitian ini :

- Merupakan jenis *multi-drop*, yaitu dapat dihubungkan dengan beberapa peralatan [5].
- Panjang kabel maksimal mencapai 4000 kaki (1200 meter) [5].
- Memiliki kecepatan 100 Kb/s – 10 Mb/s [5].
- Tidak memiliki standart konektor khusus [5].

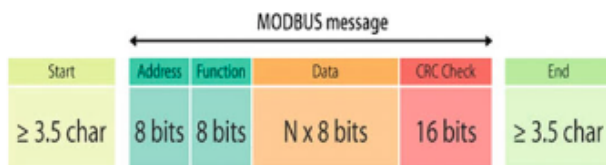
Agar komputer dapat berkomunikasi dengan PLC melalui komunikasi RS-485, pada umumnya digunakan konverter RS-485 to USB.



Gambar 11. Konverter USB to RS-485

I. Modbus Communication

Komunikasi Modbus merupakan komunikasi umum yang sering digunakan terutama saat beberapa peralatan dibuat oleh beberapa vendor karena biasanya setiap vendor memiliki standar komunikasi untuk ekosistemnya masing-masing atau dengan sederhananya komunikasi Modbus menjadi bahasa universal. Komunikasi Modbus memiliki respon format secara umum seperti gambar berikut :



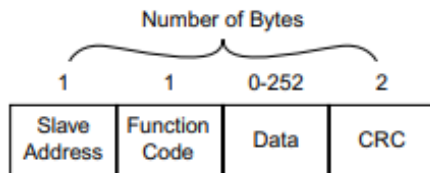
Gambar 12. Format Modbus

(Sumber : *Industrial Sensors and Controls in Communication Networks*, Dong-Seong Kim, Hoa Tran-Dang : 2019)

Karena sifatnya yang menjadi Bahasa universal dan mendukung *interface* RS-485 [5]. Modbus serial bekerja dengan cara satu peralatan sebagai *master* dan peralatan yang lain sebagai *slave*, hanya *master* yang mampu melakukan kontrol sedangkan *slave* hanya merespon. Modbus memiliki 3 mode dalam berkomunikasi yaitu TCP / IP, ASCII dan RTU (*Remote Terminal Unit*). Pada penelitian ini digunakan mode RTU karena merupakan mode komunikasi yang lebih banyak digunakan dan paling mudah pada komunikasi serial dengan format yang dilengkapi CRC (*Cyclic Redundancy Check error*). Setiap 8-bit dalam satu *byte* dalam satu pesan akan berisi 2x4-bit karakter heksadesimal dengan format 11 bit :

- 1 *start bit*.
- 8 *data bit*.
- 1 *bit* untuk *even / odd parity* dan tidak ada bit jika menggunakan *none parity*.
- 1 *stop bit* jika menggunakan *parity* dan 2 *bit* jika *none parity*.

Berikut format untuk komunikasi data dengan mode RTU :



Gambar 13. Format Modbus RTU

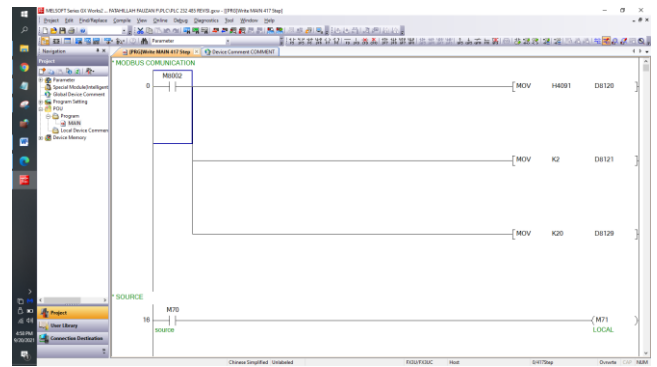
(Sumber : *Handbook Of Serial Communications Interfaces*, Louis E. Frenzel Jr, 2016)

Didalam Modbus terdapat kode untuk mewakili fungsinya. Lebih dari 20 kode terdapat dalam kode fungsi Modbus akan tetapi dalam penelitian ini hanya digunakan :

- 01 = *Read Status Coil*
- 03 = *Read Holding Registers*

J. GX-Work 2

Software yang digunakan untuk melakukan pemrograman pada PLC yang digunakan pada penelitian ini adalah GX-Work 2. *Software* ini sebenarnya digunakan untuk memprogram PLC merk Mitsubishi, akan tetapi dapat digunakan untuk memprogram PLC LE3U.

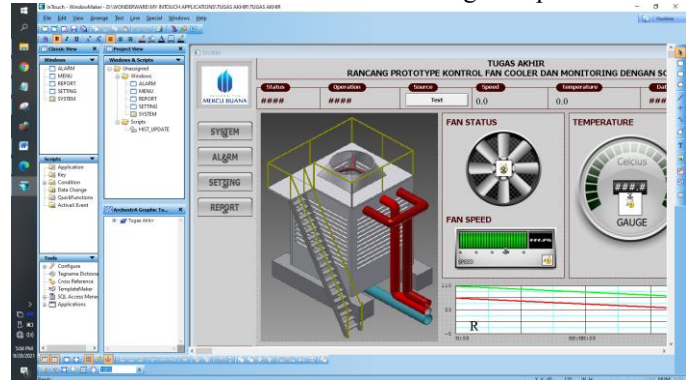


Gambar 14. Gx-Work 2

K. Wonderware Intouch 2014 R2 SP1

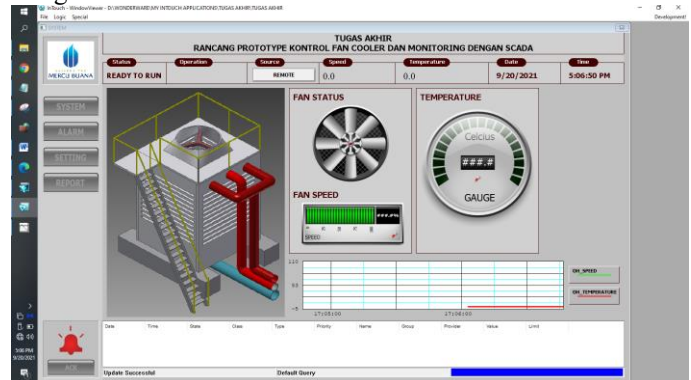
Software SCADA yang digunakan pada penelitian ini adalah Wonderware Intouch dari Schneider Electric *Software*, LLC. *Software* ini memiliki sub-*software* diantaranya :

Intouch WindowMaker untuk mendesign tampilan.



Gambar 15. Window Maker

Intouch WindowViewer untuk menjalankan *runtime* hasil design.

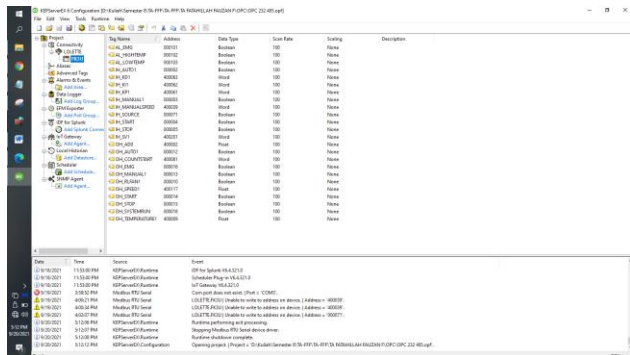


Gambar 16. Window Viewer

L. KepServerEx 6

Software ini digunakan untuk membuat tag dari PLC kemudian dihubungkan ke SCADA menggunakan komunikasi OPC (*OLE Process Control*). Pada penelitian ini OPC akan diatur untuk berkomunikasi dengan PLC dengan protokol

Modbus RTU kemudian tag akan diteruskan ke SCADA agar dapat diterima menjadi input output.



Tag Name	Address	Data Type	Scale Factor	Scaling	Description
Setpoint	10000	Real	100	None	Setpoint
Output	10001	Real	100	None	Output
Input	10002	Real	100	None	Input
Temperature	10003	Real	100	None	Temperature
Pressure	10004	Real	100	None	Pressure
Flow	10005	Real	100	None	Flow
Level	10006	Real	100	None	Level
Speed	10007	Real	100	None	Speed
Position	10008	Real	100	None	Position
Count	10009	Real	100	None	Count
Time	10010	Real	100	None	Time
Angle	10011	Real	100	None	Angle
Weight	10012	Real	100	None	Weight
Force	10013	Real	100	None	Force
Power	10014	Real	100	None	Power
Energy	10015	Real	100	None	Energy
Frequency	10016	Real	100	None	Frequency
Period	10017	Real	100	None	Period
Acceleration	10018	Real	100	None	Acceleration
Velocity	10019	Real	100	None	Velocity
Displacement	10020	Real	100	None	Displacement
Angle	10021	Real	100	None	Angle
Speed	10022	Real	100	None	Speed
Position	10023	Real	100	None	Position
Count	10024	Real	100	None	Count
Time	10025	Real	100	None	Time
Angle	10026	Real	100	None	Angle
Weight	10027	Real	100	None	Weight
Force	10028	Real	100	None	Force
Power	10029	Real	100	None	Power
Energy	10030	Real	100	None	Energy
Frequency	10031	Real	100	None	Frequency
Period	10032	Real	100	None	Period
Acceleration	10033	Real	100	None	Acceleration
Velocity	10034	Real	100	None	Velocity
Displacement	10035	Real	100	None	Displacement
Angle	10036	Real	100	None	Angle
Speed	10037	Real	100	None	Speed
Position	10038	Real	100	None	Position
Count	10039	Real	100	None	Count
Time	10040	Real	100	None	Time
Angle	10041	Real	100	None	Angle
Weight	10042	Real	100	None	Weight
Force	10043	Real	100	None	Force
Power	10044	Real	100	None	Power
Energy	10045	Real	100	None	Energy
Frequency	10046	Real	100	None	Frequency
Period	10047	Real	100	None	Period
Acceleration	10048	Real	100	None	Acceleration
Velocity	10049	Real	100	None	Velocity
Displacement	10050	Real	100	None	Displacement
Angle	10051	Real	100	None	Angle
Speed	10052	Real	100	None	Speed
Position	10053	Real	100	None	Position
Count	10054	Real	100	None	Count
Time	10055	Real	100	None	Time
Angle	10056	Real	100	None	Angle
Weight	10057	Real	100	None	Weight
Force	10058	Real	100	None	Force
Power	10059	Real	100	None	Power
Energy	10060	Real	100	None	Energy
Frequency	10061	Real	100	None	Frequency
Period	10062	Real	100	None	Period
Acceleration	10063	Real	100	None	Acceleration
Velocity	10064	Real	100	None	Velocity
Displacement	10065	Real	100	None	Displacement
Angle	10066	Real	100	None	Angle
Speed	10067	Real	100	None	Speed
Position	10068	Real	100	None	Position
Count	10069	Real	100	None	Count
Time	10070	Real	100	None	Time
Angle	10071	Real	100	None	Angle
Weight	10072	Real	100	None	Weight
Force	10073	Real	100	None	Force
Power	10074	Real	100	None	Power
Energy	10075	Real	100	None	Energy
Frequency	10076	Real	100	None	Frequency
Period	10077	Real	100	None	Period
Acceleration	10078	Real	100	None	Acceleration
Velocity	10079	Real	100	None	Velocity
Displacement	10080	Real	100	None	Displacement
Angle	10081	Real	100	None	Angle
Speed	10082	Real	100	None	Speed
Position	10083	Real	100	None	Position
Count	10084	Real	100	None	Count
Time	10085	Real	100	None	Time
Angle	10086	Real	100	None	Angle
Weight	10087	Real	100	None	Weight
Force	10088	Real	100	None	Force
Power	10089	Real	100	None	Power
Energy	10090	Real	100	None	Energy
Frequency	10091	Real	100	None	Frequency
Period	10092	Real	100	None	Period
Acceleration	10093	Real	100	None	Acceleration
Velocity	10094	Real	100	None	Velocity
Displacement	10095	Real	100	None	Displacement
Angle	10096	Real	100	None	Angle
Speed	10097	Real	100	None	Speed
Position	10098	Real	100	None	Position
Count	10099	Real	100	None	Count
Time	10100	Real	100	None	Time
Angle	10101	Real	100	None	Angle
Weight	10102	Real	100	None	Weight
Force	10103	Real	100	None	Force
Power	10104	Real	100	None	Power
Energy	10105	Real	100	None	Energy
Frequency	10106	Real	100	None	Frequency
Period	10107	Real	100	None	Period
Acceleration	10108	Real	100	None	Acceleration
Velocity	10109	Real	100	None	Velocity
Displacement	10110	Real	100	None	Displacement
Angle	10111	Real	100	None	Angle
Speed	10112	Real	100	None	Speed
Position	10113	Real	100	None	Position
Count	10114	Real	100	None	Count
Time	10115	Real	100	None	Time
Angle	10116	Real	100	None	Angle
Weight	10117	Real	100	None	Weight
Force	10118	Real	100	None	Force
Power	10119	Real	100	None	Power
Energy	10120	Real	100	None	Energy
Frequency	10121	Real	100	None	Frequency
Period	10122	Real	100	None	Period
Acceleration	10123	Real	100	None	Acceleration
Velocity	10124	Real	100	None	Velocity
Displacement	10125	Real	100	None	Displacement
Angle	10126	Real	100	None	Angle
Speed	10127	Real	100	None	Speed
Position	10128	Real	100	None	Position
Count	10129	Real	100	None	Count
Time	10130	Real	100	None	Time
Angle	10131	Real	100	None	Angle
Weight	10132	Real	100	None	Weight
Force	10133	Real	100	None	Force
Power	10134	Real	100	None	Power
Energy	10135	Real	100	None	Energy
Frequency	10136	Real	100	None	Frequency
Period	10137	Real	100	None	Period
Acceleration	10138	Real	100	None	Acceleration
Velocity	10139	Real	100	None	Velocity
Displacement	10140	Real	100	None	Displacement
Angle	10141	Real	100	None	Angle
Speed	10142	Real	100	None	Speed
Position	10143	Real	100	None	Position
Count	10144	Real	100	None	Count
Time	10145	Real	100	None	Time
Angle	10146	Real	100	None	Angle
Weight	10147	Real	100	None	Weight
Force	10148	Real	100	None	Force
Power	10149	Real	100	None	Power
Energy	10150	Real	100	None	Energy
Frequency	10151	Real	100	None	Frequency
Period	10152	Real	100	None	Period
Acceleration	10153	Real	100	None	Acceleration
Velocity	10154	Real	100	None	Velocity
Displacement	10155	Real	100	None	Displacement
Angle	10156	Real	100	None	Angle
Speed	10157	Real	100	None	Speed
Position	10158	Real	100	None	Position
Count	10159	Real	100	None	Count
Time	10160	Real	100	None	Time
Angle	10161	Real	100	None	Angle
Weight	10162	Real	100	None	Weight
Force	10163	Real	100	None	Force
Power	10164	Real	100	None	Power
Energy	10165	Real	100	None	Energy
Frequency	10166	Real	100	None	Frequency
Period	10167	Real	100	None	Period
Acceleration	10168	Real	100	None	Acceleration
Velocity	10169	Real	100	None	Velocity
Displacement	10170	Real	100	None	Displacement
Angle	10171	Real	100	None	Angle
Speed	10172	Real	100	None	Speed
Position	10173	Real	100	None	Position
Count	10174	Real	100	None	Count
Time	10175	Real	100	None	Time
Angle	10176	Real	100	None	Angle
Weight	10177	Real	100	None	Weight
Force	10178	Real	100	None	Force
Power	10179	Real	100	None	Power
Energy	10180	Real	100	None	Energy
Frequency	10181	Real	100	None	Frequency
Period	10182	Real	100	None	Period
Acceleration	10183	Real	100	None	Acceleration
Velocity	10184	Real	100	None	Velocity
Displacement	10185	Real	100	None	Displacement
Angle	10186	Real	100	None	Angle
Speed	10187	Real	100	None	Speed
Position	10188	Real	100	None	Position
Count	10189	Real	100	None	Count
Time	10190	Real	100	None	Time
Angle	10191	Real	100	None	Angle
Weight	10192	Real	100	None	Weight
Force	10193	Real	100	None	Force
Power	10194	Real	100	None	Power
Energy	10195	Real	100	None	Energy
Frequency	10196	Real	100	None	Frequency
Period	10197	Real	100	None	Period
Acceleration	10198	Real	100	None	Acceleration
Velocity	10199	Real	100	None	Velocity
Displacement	10200	Real	100	None	Displacement

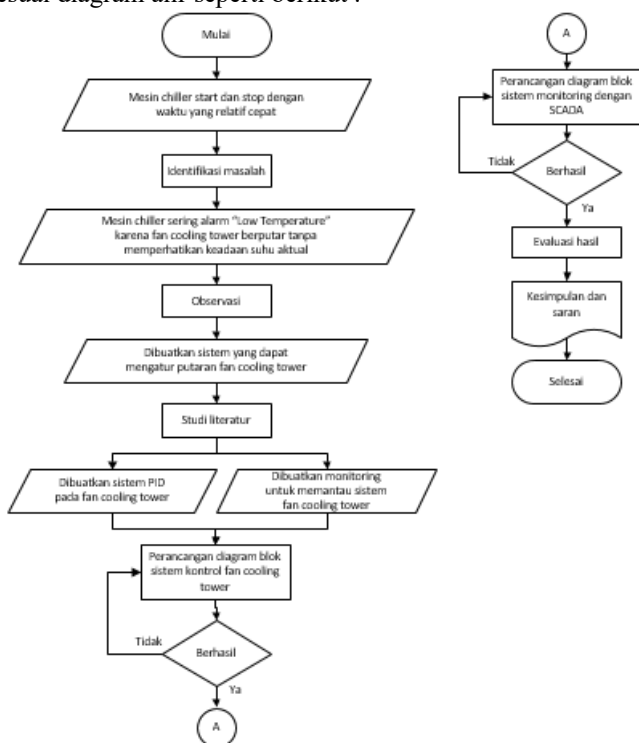
Gambar 17. KepServerEx

III. METODOLOGI PENELITIAN

Metode penelitian ini digunakan untuk menghasilkan rancangan sistem kontrol *fan cooling tower* yang dapat dimonitoring oleh SCADA Wonderware secara *real time*.

A. Diagram Alir

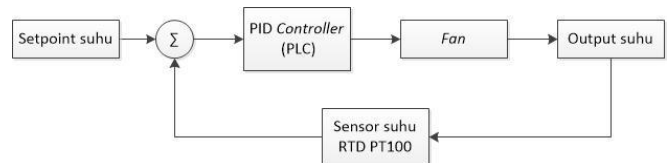
Pelaksanaan penelitian ini dilaksanakan dengan tahapan sesuai diagram alir seperti berikut :



Gambar 18. Diagram Alir

B. Diagram Blok Sistem Kontrol

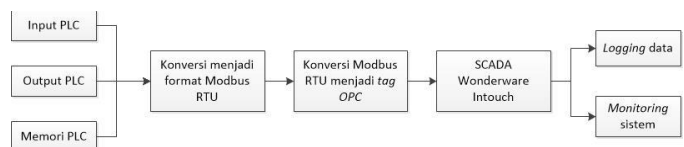
Diagram blok sistem kontrol menjelaskan bagaimana sistem kontrol bekerja.

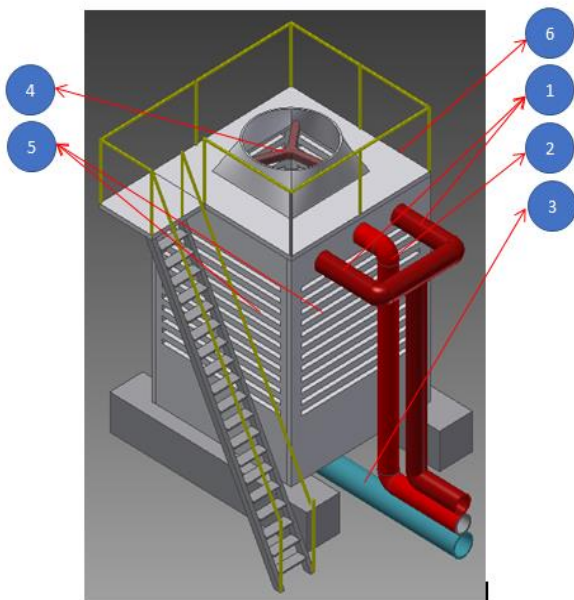


Gambar 19. Diagram Blok Sistem Kontrol

C. Diagram Blok Sistem Komunikasi dan Monitoring

Diagram blok sistem komunikasi dan monitoring menjelaskan bagaimana sistem berkomunikasi dan dapat termonitoring oleh SCADA.

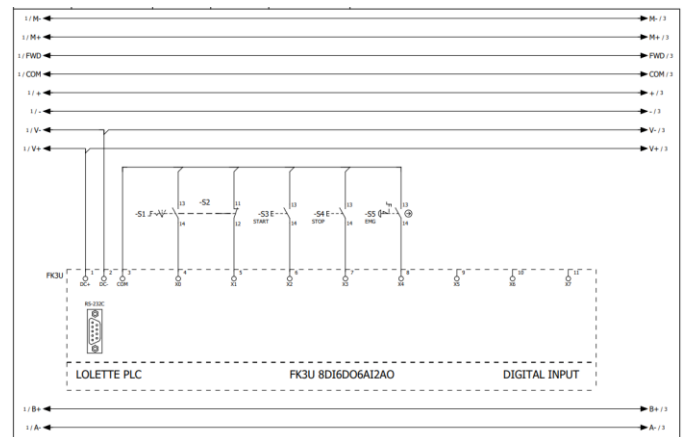




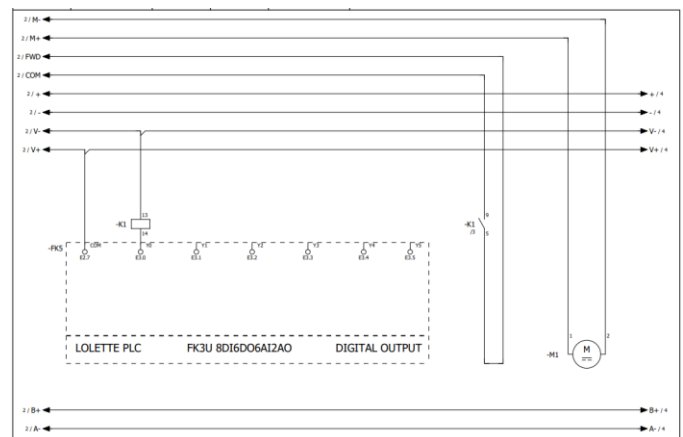
Gambar 22. Basin Cooling Tower

Keterangan :

1. Jalur air panas dari mesin Chiller.
2. Jalur *overflow* jika terjadi luapan air berlebih.
3. Jalur air dingin yang sudah didinginkan.
4. Fan dengan arah buangan udara ke atas untuk membuang panas air ke udara.
5. Lubang kisi untuk sirkulasi udara fan.
6. *Mounting* sensor suhu.



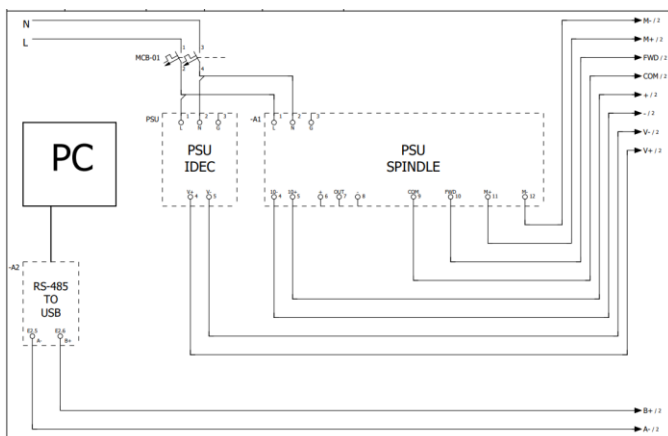
Gambar 24. Wiring Digital Input



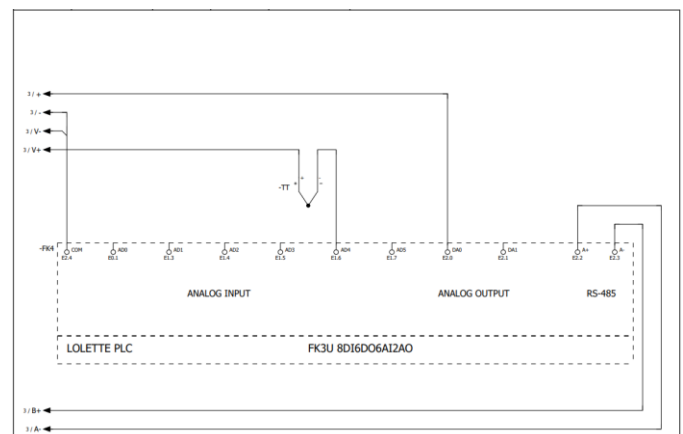
Gambar 25. Wiring Digital Output

F. Perancangan Wiring

Wiring digunakan untuk mempermudah implementasi pengkabelan dan referensi jika suatu saat terjadi *abnormal*.



Gambar 23. Wiring Power



Gambar 26. Wiring Input Output Analog

G. Perancangan Program PLC

Untuk memudahkan pemrograman, pisahkan program menjadi beberapa sub yang terdiri :

- *Modbus Communication*, berisi program untuk mengatur komunikasi PLC dengan protokol Modbus.
- *Source*, berisi program untuk mengatur pengontrolan

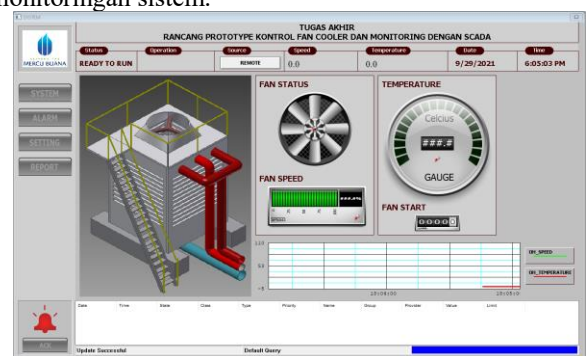
sistem agar dijalankan secara lokal menggunakan tombol atau secara remot dari SCADA.

- Input Digital, berisi program untuk mendefinisikan input yang digunakan berupa tombol untuk *auto*, *manual*, *start*, *stop*, dan *emergency*.
- Output, berisi program untuk mendefinisikan output yang digunakan berupa relay untuk menghidupkan *fan*.
- Sensor, berisi program untuk membaca sensor dan menskala pembacaan PLC 0-4053 menjadi 0-100 derajat Celcius.
- *Speed Scaling*, berisi program untuk menskala logika PLC 0-400 (0-0.99 V) menjadi 0-100% yang akan digunakan untuk keperluan tampilan di SCADA.
- *System Run*, berisi logika yang menyatakan sistem siap atau tidak untuk dijalankan.
- Manual, berisi logika jika sistem akan dijalankan secara manual.
- Auto, berisi logika jika sistem akan dijalankan secara otomatis.
- *Countstart*, berisi program untuk menghitung jumlah fan melakukan *starting*.
- SV 0-100 to 0-4053, berisi program untuk menskala nilai *set point* dari masukan tampilan menjadi nilai 0-4053 untuk keperluan logika PLC.

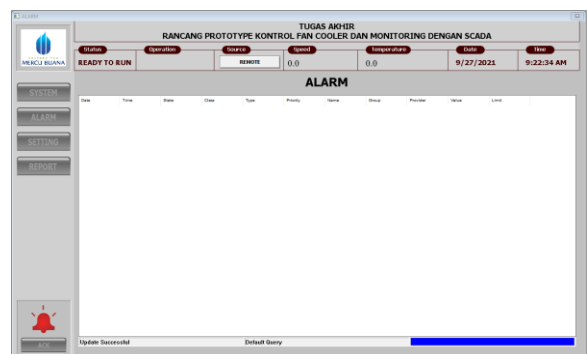
IH_AUTO1	I/O Discrete	OPC_TA	PLC_IO	AccessLevel
IH_KD1	I/O Integer	OPC_TA	PLC_IO	AccessLevel
IH_KI1	I/O Integer	OPC_TA	PLC_IO	AccessLevel
IH_KP1	I/O Integer	OPC_TA	PLC_IO	AccessLevel
IH_MANUAL1	I/O Discrete	OPC_TA	PLC_IO	AccessLevel
IH_MANUALSPEED	I/O Integer	OPC_TA	PLC_IO	AccessLevel
IH_SOURCE	I/O Discrete	OPC_TA	PLC_IO	AccessLevel
IH_START	I/O Discrete	OPC_TA	PLC_IO	AccessLevel
IH_STOP	I/O Discrete	OPC_TA	PLC_IO	AccessLevel
IH_SV1	I/O Real	OPC_TA	PLC_IO	AccessLevel
OH_AUTO1	I/O Discrete	OPC_TA	PLC_IO	AccessLevel
OH_EMG	I/O Discrete	OPC_TA	PLC_IO	AccessLevel
OH_MANUAL1	I/O Discrete	OPC_TA	PLC_IO	AccessLevel
OH_PV	I/O Real	OPC_TA	PLC_IO	AccessLevel
OH_RLFAN1	I/O Discrete	OPC_TA	PLC_IO	AccessLevel
OH_SPEED1	I/O Real	OPC_TA	PLC_IO	AccessLevel
OH_START	I/O Discrete	OPC_TA	PLC_IO	AccessLevel
OH_STOP	I/O Discrete	OPC_TA	PLC_IO	AccessLevel
OH_SYSTEMRUN	I/O Discrete	OPC_TA	PLC_IO	AccessLevel
OH_TEMPERATURE1	I/O Real	OPC_TA	PLC_IO	AccessLevel

Gambar 27. Tag SCADA

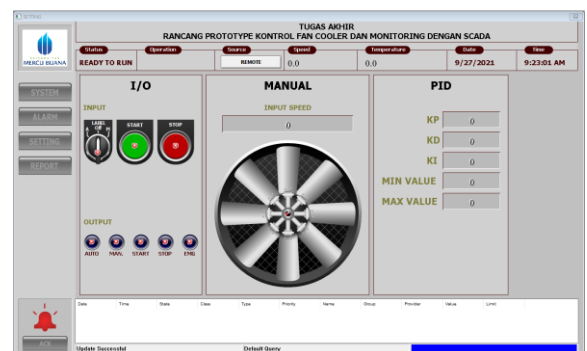
Kemudian buat beberapa tampilan untuk memudahkan pemantauan sistem.



Gambar 28. System Window



Gambar 29. Alarm Window



Gambar 30. Setting Window

H. Perancangan Tag PLC Pada KepServerEx

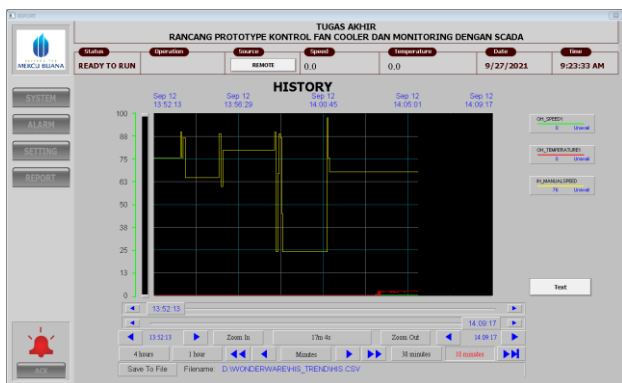
Agar input, output dan data pada PLC dapat terhubung ke SCADA menggunakan komunikasi Modbus RTU dengan sistem OPC perlu dibuat tag.

Tabel 1. Tag KepServerEx

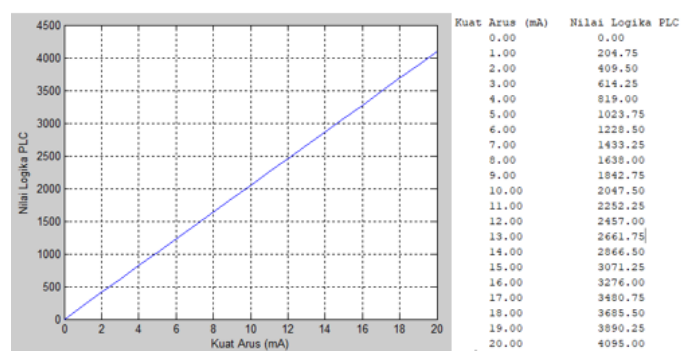
Tag Name	Address	Data Type	Respect Data Type	Client Access	Scan Rate
IH_AUTO1	000002	Boolean	1	R/W	100
IH_KD1	400063	Word	1	R/W	100
IH_KI1	400062	Word	1	R/W	100
IH_KP1	400061	Word	1	R/W	100
IH_MANUAL1	000003	Boolean	1	R/W	100
IH_MANUALSPEED	400039	Word	1	R/W	100
IH_SOURCE	000071	Boolean	1	R/W	100
IH_START	000004	Boolean	1	R/W	100
IH_STOP	000005	Boolean	1	R/W	100
IH_SV1	400201	Word	1	R/W	100
OH_PV	400002	Float	1	RO	100
OH_AUTO1	000012	Boolean	1	R/W	100
OH_EMG	000016	Boolean	1	RO	100
OH_MANUAL1	000013	Boolean	1	R/W	100
OH_RLFAN1	000010	Boolean	1	RO	100
OH_SPEED1	400117	Float	1	RO	100
OH_START	000014	Boolean	1	RO	100
OH_STOP	000015	Boolean	1	RO	100
OH_SYSTEMRUN	000018	Boolean	1	RO	100
OH_TEMPERATURE1	400009	Float	1	RO	100

I. Perancangan SCADA Wonderware

Pada bagian SCADA perlu dirancang tag yang mendefinisikan data yang akan dikomunikasi dengan KepServerEx.



Gambar 31. Report Window

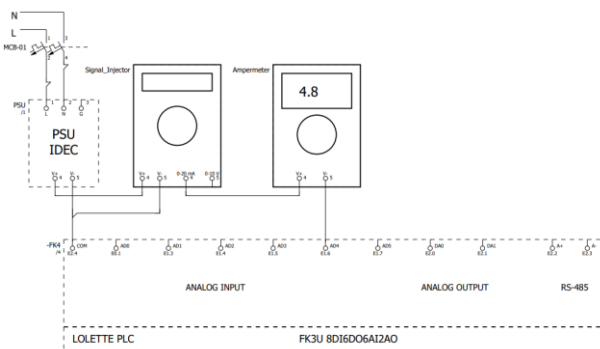


Gambar 33. Scaling 0-20 mA menjadi nilai logika PLC 0-4095 menggunakan Matlab

IV. HASIL DAN ANALISA

Untuk mendapatkan akurasi hasil pembacaan sensor dilakukan pengetesan pada PLC dengan metode injeksi signal ke terminal analog PLC dan kalibrasi sensor menggunakan alat ukur suhu yang telah terkalibrasi.

Berdasarkan manual *book* PLC, pada terminal AD4 menerima signal analog 0-20 mA kemudian terbaca sebagai nilai logika di PLC menjadi 0-4095. Untuk melakukan pengetesan pembacaan PLC dilakukan dengan metode injeksi signal.



Gambar 32. Injeksi Signal

Nilai kuat arus 0-20 mA yang terbaca sebagai nilai logika di PLC 0-4095 dapat di jelaskan dengan rumus scaling:

$$\frac{(Data\ mA - minimum\ mA)}{Resolusi\ mA} \times Resolusi\ logic + minimum\ logic = Data\ logic$$

Berikut contoh penggunaan rumus diatas :

$$\frac{(0-0)}{20} \times 4095 + 0 = 0 \quad , \quad \frac{(1-0)}{20} \times 4095 + 0 = 204,75$$

Dengan menggunakan Matlab, perhitungan diatas didapatkan hasil seperti berikut :

Kemudian dilakukan pengukuran aktual untuk membandingkan teori dengan rumus diatas benar atau tidak.

Tabel 2. Perbandingan hasil pembacaan nilai logika PLC dengan menggunakan rumus dan aktual

Kuat Arus (mA)	Nilai Logika PLC		Gap	Persentase
	Teori	Aktual		
0	0,00	0	0,00	0,00
1	204,75	205	0,25	100,12
2	409,50	409	0,50	99,88
3	614,25	609	5,25	99,15
4	819,00	814	5,00	99,39
5	1023,75	1015	8,75	99,15
6	1228,50	1218	10,50	99,15
7	1433,25	1419	14,25	99,01
8	1638,00	1622	16,00	99,02
9	1842,75	1823	19,75	98,93
10	2047,50	2027	20,50	99,00
11	2252,25	2229	23,25	98,97
12	2457,00	2432	25,00	98,98
13	2661,75	2638	23,75	99,11
14	2866,50	2838	28,50	99,01
15	3071,25	3038	33,25	98,92
16	3276,00	3241	35,00	98,93
17	3480,75	3444	36,75	98,94
18	3685,50	3644	41,50	98,87
19	3890,25	3851	39,25	98,99
20	4095,00	4053	42,00	98,97
Rata-rata				94,40

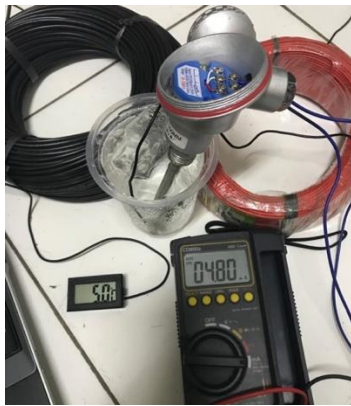
Dari tabel diatas dapat diambil kesimpulan rata-rata akurasi 94,40% dengan maksimal gap sebesar 42 dan minimal gap sebesar 0. Konversi nilai logika PLC 0-4095.

A. Hasil Akurasi Sensor Suhu RTD

Karena spesifikasi sensor adalah output 4-20 mA dengan resolusi ukur 0-100 °C. Karena keterbatasan alat zero point yang digunakan adalah 5 °C . Untuk menentukan harga kuat arus pada suhu 5 °C dilakukan *scaling* 0-100 °C ke 4-20 mA dengan cara :

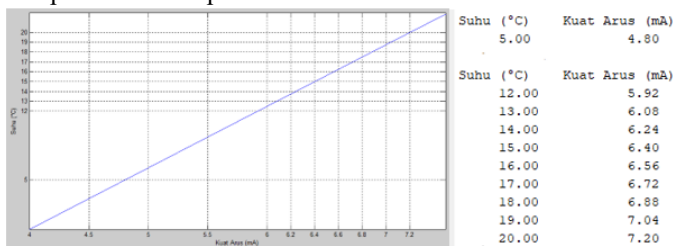
$$\frac{(Data\ suhu - minimum\ suhu)}{Resolusi\ suhu} \times Resolusi\ mA + minimum\ mA = Data\ mA$$

$$\frac{(5-0)}{100} \times 16 + 4 = 4.80\ mA$$



Gambar 34. Kalibrasi Sensor RTD

Sampling pengukuran pada suhu tertentu kemudian bandingkan hasil keluaran signal sensor secara aktual dengan teori. Dengan menggunakan Matlab, perhitungan diatas didapatkan hasil seperti berikut :



Gambar 35. scaling 0-100 °C ke 4-20 mA menggunakan Matlab

Tabel 3. Perbandingan keluaran sensor 4-20mA pada resolusi 0-100 °C

Suhu (°C)	Nilai Kuat arus (mA)		Gap (mA)	Persentase
	Teori	Aktual		
5	4,80	4,80	0,00	0,00
12	5,91	5,91	0,01	0,17
13	6,08	6,08	0,00	0,00
14	6,24	6,25	0,01	0,16
15	6,40	6,41	0,01	0,16
16	6,56	6,55	0,01	0,15
17	6,72	6,72	0,00	0,00
18	6,88	6,89	0,01	0,15
19	7,04	7,04	0,00	0,00
20	7,20	7,22	0,02	0,28
Rata-rata			0,01	0,11

Dari table diatas dapat disimpulkan bahwa perhitungan secara teori dan pengukuran aktual memiliki rata-rata gap sebesar 0,01 mA dengan rata-rata persentase sebesar 0,11% dengan penyimpangan maksimal 0,28%.

B. Hasil Scaling Pembacaan Sensor RTD ke Display Monitoring

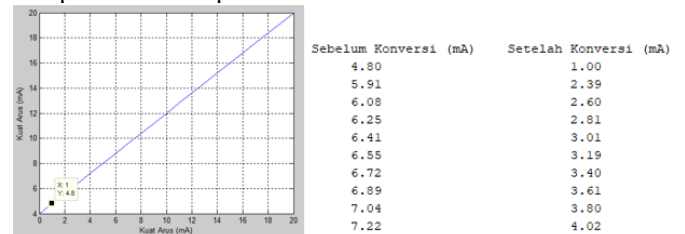
Karena range pembacaan PLC berbeda dengan sensor RTD, perlu dilakukan *scaling* agar pembacaan antara PLC dan sensor RTD sama. Berikut hasilnya :

Mengkonversi Nilai Range Sensor RTD ke Nilai Range PLC :

Langkah ini menkonversi nilai range sensor RTD (4-20 mA) menjadi nilai range PLC (0-20 mA). Dengan menggunakan rumus *scaling* :

$$\frac{(Data\ mA.S - minimum\ mA.S)}{Resolusi\ mA.S} \times Resolusi\ mA.P + minimum\ mA.P = Data\ mA.S$$

Dengan menggunakan Matlab, perhitungan diatas didapatkan hasil seperti berikut :



Gambar 36. Konversi nilai range 4-20 mA ke 0-20 mA menggunakan matlab

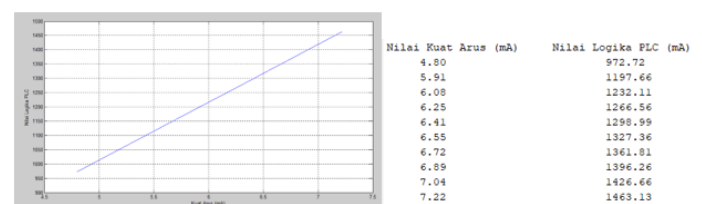
Dari gambar diatas dapat diartikan bahwa 4,80 mA sebelum konversi (pada nilai range 4-20 mA) sama dengan 1,00 mA setelah dikonversi (pada nilai range 0-20 mA) dan seterusnya. Nilai konversi pada gambar diatas akan disebut dengan nilai "range1".

Mengkonversi Nilai Range Sensor RTD ke Nilai Range Logika PLC :

Karena nilai range sensor (4-20 mA) dan PLC (0-20 mA) berbeda, perlu membandingkan hasil nilai logika PLC (0-4053) pada keduanya. Pertama konversi nilai "range1 sebelum konversi" ke nilai range logika PLC (0-4053). Dengan menggunakan rumus *scaling* :

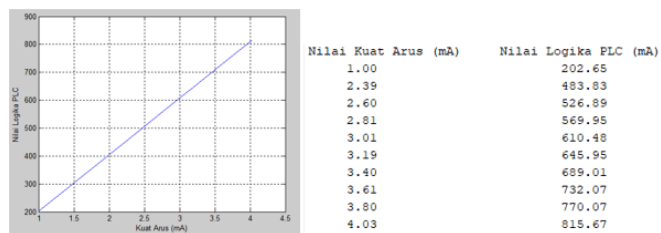
$$\frac{(Data\ range1 - min.range1)}{Resolusi\ range1} \times resolusi\ logic + min.logic = Data\ logic$$

Dengan menggunakan Matlab, perhitungan diatas didapatkan hasil seperti berikut :



Gambar 37. konversi nilai "range1 sebelum konversi" ke nilai range logika PLC (0-4053)

Gambar 37 merupakan hasil aktual sebelum perbedaan nilai range sensor RTD dan PLC disamakan. Kemudian konversi nilai "range1 setelah konversi" ke nilai range logika PLC (0-4053). Dengan menggunakan rumus dan metode yang sama menggunakan Matlab, didapatkan hasil seperti berikut :



Gambar 38. konversi nilai “range1 setelah konversi” ke nilai range logika PLC (0-4053) menggunakan Matlab

Gambar 38 merupakan nilai standar/nilai seharusnya atau dapat diartikan jika sensor RTD dengan range 4-20 mA mengeluarkan output sebesar 4,80 mA dihubungkan ke PLC dengan range 0-20 mA maka seharusnya trigger yang diterima PLC adalah 1 mA bukan 4,80 mA.

Tabel 4. Perbandingan konversi kuat arus ke nilai logika PLC 0-4053

Suhu (°C)	Nilai Kuat Arus Sensor (mA)	Nilai Logika PLC		Gap
		Hasil	Seharusnya	
5	4,80	972,72	202,65	770,07
12	5,91	1197,66	483,83	713,83
13	6,08	1232,11	526,89	705,22
14	6,25	1266,56	569,95	696,61
15	6,41	1298,99	610,48	688,51
16	6,55	1327,36	645,95	681,41
17	6,72	1361,81	689,01	672,80
18	6,89	1396,26	732,07	664,19
19	7,04	1426,66	770,07	656,59
20	7,22	1463,13	815,67	647,46

Dari table diatas dapat disimpulkan bahwa jika kita menghubungkan sensor RTD secara langsung ke PLC akan terjadi kesalahan pembacaan akibat perbedaan range diantara keduanya.

Mencari Nilai Zero Point Untuk Pembacaan Logika PLC :

Sejak nilai range sensor RTD dan PLC memiliki perbedaan, untuk memanipulasi agar kedua range memiliki kesamaan perlu dilakukan penyamaan nilai zero point. Dengan menggunakan perbandingan konversi 0-20 mA ke nilai logika 0-4053 vs 4-20 mA ke nilai logika 0-4053 yang dihitung menggunakan rumus *scaling* :

$$\frac{(Data\ mA - min.\ mA)}{Resolusi\ mA} \times resolusi\ logic + min.\ logic = Data\ logic$$

$$\frac{(4-0)}{20} \times 4053 + 0 = 810,6$$

$$\frac{(4-4)}{16} \times 4053 + 0 = 0$$

$$\frac{(5-0)}{20} \times 4053 + 0 = 1013,25$$

$$\frac{(5-4)}{16} \times 4053 + 0 = 253,3125$$

Dari perhitungan diatas dapat disimpulkan bahwa nilai *zero point* adalah 810,6. Nilai ini akan menjadi variable pengurang saat PLC membaca nilai sensor. Sedangkan variabel kenaikan atau pengali nilai didapat dengan membagi nilai logika pembacaan kedua dari konversi nilai range 0-20 mA ke nilai logika 0-4053 dengan nilai pertama yaitu :

$$Kenaikan = 1013.25 / 810.6 \\ = 1.25$$

Pengecekan Nilai Logika PLC :

Dengan memasukkan variable pengurang dan pengali terhadap nilai logika PLC pada kolom “Hasil” dari table 4 sebagai berikut :

$$Nilai\ Logika = (Hasil - variabel\ pengurang) \times variabel\ pengali$$

Sebagai contoh perhitungan pada 5 °C dengan pembacaan aktual 972.72 sedangkan seharusnya 202.65, dengan menggunakan rumus diatas dapat dihitung sebagai :

$$Nilai\ Logika = (972.72 - 810.6) \times 1.25 \\ = 162.12 \times 1.25 \\ = 202.65$$

Tabel 5. Pengecekan scaling setelah memasukkan faktor pengurang dan pengali

Suhu (°C)	Kuat Arus (mA)	Nilai Logika Aktual	Dikurangi (810.6)	Dikali (1.25)	Nilai Logika Standar	Gap
5	4,8	972,72	162,12	202,65	202,65	0
12	5,91	1197,66	387,06	483,83	483,83	0
13	6,08	1232,11	421,51	526,89	526,89	0
14	6,25	1266,56	455,96	569,95	569,95	0
15	6,41	1298,99	488,39	610,48	610,48	0
16	6,55	1327,36	516,76	645,95	645,95	0
17	6,72	1361,81	551,21	689,01	689,01	0
18	6,89	1396,26	585,66	732,07	732,07	0
19	7,04	1426,66	616,06	770,07	770,07	0
20	7,22	1463,13	652,53	815,67	815,67	0

Dari tabel diatas dapat diambil kesimpulan bahwa hasil nilai logika setelah dimasukkan variable pengurang dan pengali bersesuaian dengan nilai logika seharusnya.

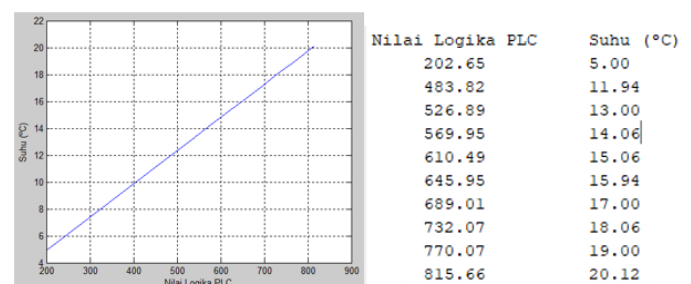
Scaling Hasil Nilai Logika PLC Ke Display Monitoring :

Karena hasil nilai logika PLC sudah sesuai dengan nilai logika yang seharusnya, maka nilai logika PLC 0-4053 ini dapat *discaling* agar menampilkan display monitoring 0-100 °C pada SCADA dengan menggunakan rumus *scaling* :

$$\frac{(Data\ logic - min.\ logic)}{Resolusi\ logic} \times resolusi\ display + min.\ display = Data\ logic$$

$$\frac{(202.65-0)}{4053} \times 100 + 0 = 5 \quad , \quad \frac{(483.8269-0)}{4053} \times 100 + 0 = 12$$

Dengan menggunakan Matlab, perhitungan diatas didapatkan hasil seperti berikut :



Gambar 39. Konversi nilai logika PLC ke 0-100 °C menggunakan matlab

Tabel 6. Konversi hasil nilai logika PLC 0-4053 menjadi 0-100 °C

Kuat Arus (mA)	Nilai Logika PLC	Hasil Konversi (°C)
4,8	202,65	5,00
5,91	483,83	11,94
6,08	526,89	13,00
6,25	569,95	14,06
6,41	610,48	15,06
6,55	645,95	15,94
6,72	689,01	17,00
6,89	732,07	18,06
7,04	770,07	19,00
7,22	815,67	20,12

Kemudian hasil konversi pada tabel 6 dibandingkan dengan hasil pengukuran dari alat ukur suhu yang sudah terkalibrasi.

Tabel 7. Perbandingan suhu hasil konversi vs suhu alat ukur

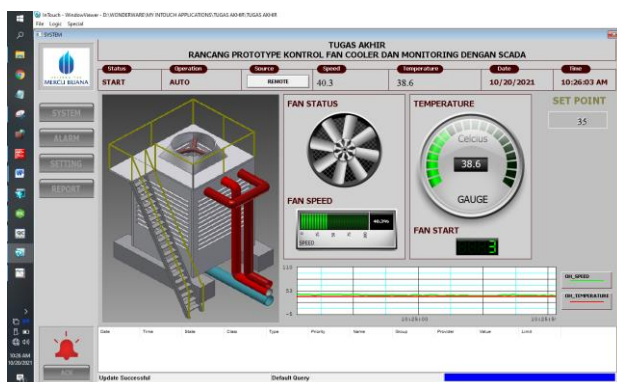
Suhu Hasil Konversi (°C)	Suhu alat ukur (°C)	Gap (°C)
5.00	5	0.00
11.94	12	0.06
13.00	13	0.00
14.06	14	0.06
15.06	15	0.06
15.94	16	0.06
17.00	17	0.00
18.06	18	0.06
19.00	19	0.00
20.12	20	0.12
Rata-rata		0,04

Dari tabel diatas dapat disimpulkan bahwa hasil pembacaan sensor RTD dengan hasil alat ukur memiliki gap rata-rata 0,04 °C dengan gap maksimal sebesar 0,12 °C

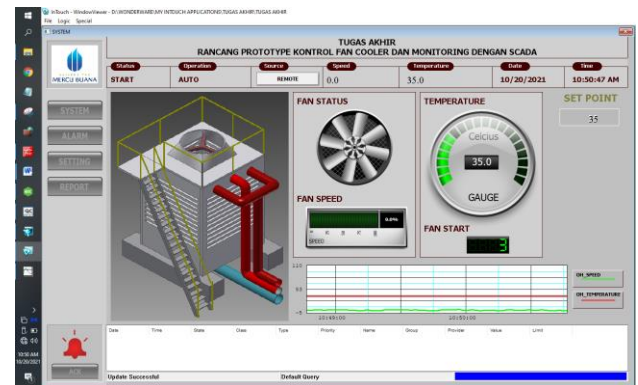
C. Hasil Monitoring Dengan SCADA Wonderware

Karena sebelumnya sudah dibuat design tampilan dan program, berikut hasil monitoringnya :

System window, dapat diamati bahwa pada kolom “Temperature” menunjukkan angka sesuai suhu aktual, “Status”, “Operation”, “Source” terisi sesuai kondisi PLC.



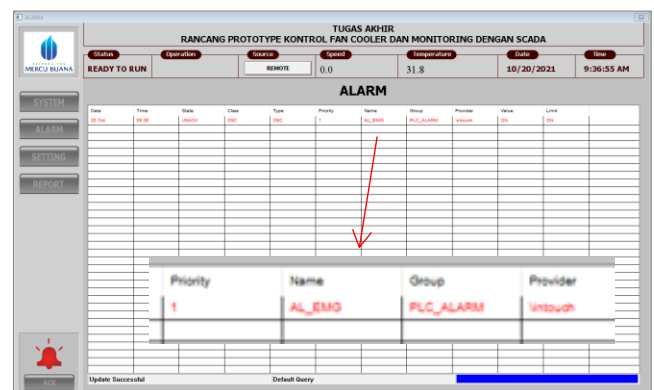
Gambar 40. System Window (1)



Gambar 41. System Window (2)

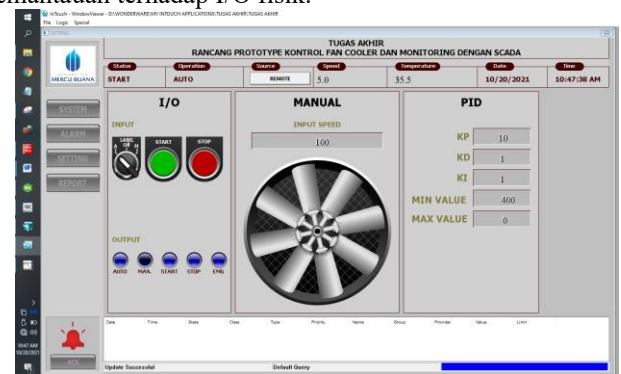
Dengan memasukkan , $K_P = 10$, $K_I = 1$, $K_D = 1$ yang didapat berdasarkan *trial and error* dipengaturan PID. Dapat diamati pada gambar 40 ketika suhu 38.6 °C kecepatan *fan* menjadi 40.3% kemudian pada dan pada gambar 41 ketika suhu 35 °C (sesuai *set point*) kecepatan *fan* menjadi 0%.

Alarm window, menu ini menampilkan daftar *alarm* untuk memberitahukan jika sistem mengalami eror atau penyebab sistem tidak berjalan normal seperti ditekannya tombol *emergency*.



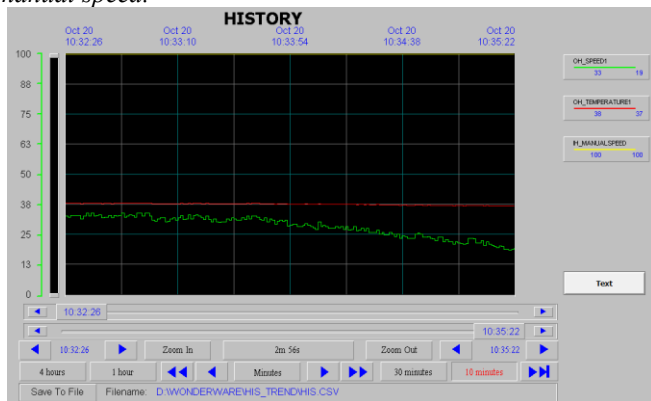
Gambar 42. Alarm Window

Setting window, pada menu ini menunjukkan *display* I/O SCADA berhasil dikomunikasikan dengan sistem kontrol *fan cooling tower*. Nyala dan matinya lampu *display* menandakan pemantauan terhadap I/O fisik.



Gambar 43. Setting Window

Report window, pada menu ini menampilkan grafik rekaman data berupa *fan speed*, *temperature*, dan inputan *manual speed*.



Gambar 44. Report Window

V. KESIMPULAN

A. Kesimpulan

Dari hasil pengujian yang telah dilakukan terhadap rancang bangun kontrol *fan cooling tower* dimonitoring oleh SCADA Wonderware dapat diperoleh kesimpulan sebagai berikut

Hasil akurasi sensor RTD yang digunakan pada penelitian ini rata-rata sebesar 0,11 % dengan penyimpangan maksimal 0,28%.

Nilai range sensor RTD 4-20 mA dapat *discaling* agar menampilkan display monitoring 0-100 °C pada SCADA Wonderware dengan gap rata-rata 0,04 °C dan gap maksimal sebesar 0,12 °C jika dibandingkan dengan alat ukur.

SCADA Wonderware mampu memonitoring sistem kontrol *fan cooling tower* yang ditunjukkan dengan tampilannya yang dapat menampilkan suhu, mode operasi, *source*, alarm, dan kecepatan *fan* yang dapat berubah sesuai keadaan suhu air *cooling tower* dengan melakukan pengaturan pada parameter PID secara *trial and error*.

B. Saran

Saran untuk pengembangan dari rancang bangun sistem kontrol *fan cooling tower* yang dimonitoring oleh SCADA

Untuk hasil pembacaan sensor yang lebih baik sebaiknya digunakan alat kalibrasi yang telah tersertifikasi.

Parameter PID harus diatur sedemikian rupa sampai sistem stabil pada penggunaan beban nyata

DAFTAR PUSTAKA

- [1] Abdullah, S. (2020). Rancang Bangun Sistem Auto Drain Pada Cooling Tower. *Jurnal Teknologi Elektro, Universitas Mercu Buana*, ISSN : 2086-9479.
- [2] Mujahid, I. U., Siregar, I. P., & Leksono, E. (Vol 9 (1), 2017). Studi Otomasi Dasar Pada Sistem Miniplant Boiler dan Heat Exchanger Saat Pengamanan Sistem Melalui HMI dan Database. *J.Oto.Ktrl.Inst (J.Auto.Ctrl.Inst)*, ISSN : 2085-2517.
- [3] Pranowo, I. D., Bagastama, Y. T., & Wibisono, T. A. (2020). Communication Between PLC Different Vendors Using OPC Server Improved With Application Device. *TELEKOMNIKA Telecommunication, Computing, Electronics and Control*, Vol 18, No 3, pp1491-1498, ISSN : 1693-6930.
- [4] Dwi Cahyo, B. F. (2021). Sistem Pelaporan Inspeksi dan Perawatan Ventex Explosion Isolation Valve Berbasis Android. *Jurnal Teknologi Elektro, Universitas Mercu Buana*.
- [5] Frenzel Jr, L. (2016). *Handbook Of Serial Communications Interfaces*. 225 Wyman Street, Waltham, MA 02451, USA: Elsevier Inc.
- [6] Manesis, S., & Nikolakopoulos, G. (2018). *Introduction To Industrial Automation*. Boca Raton: Taylor & Francis Group, LLC.
- [7] Eriyani, V., Triyanto, D., & Nirmala, I. (2018). Rancang Bangun Robot Pelayan Restoran Otomatis Berbasis Mikrokontroler Atmega16 Dengan Navigasi Line Follower. *Jurnal Coding Sistem Komputer Untan*, Vol 06, No 03, hal 66-74.
- [8] Munadhif, I., Rinato, N., & Afiqi, M. (2019). Pengendalian Otomatis Cooling Water System Pada Proses Pendinginan Turbin Gas. *Journal of Science and Technology*, Vol 12 (1), hal 36-42.
- [9] Irvawansyah, & Rahmansyah, A. A. (2018). Prototype Sistem Monitoring dan Pengontrolan Level Tangki Air Berbasis SCADA. *Jurnal Teknologi Terapan*, Vol 4, No 1.
- [10] Instruments, T. (2020). *The Engineer's Guide to Temperature Sensing*. Dallas, Texas: Texas Instruments Incorporated.
- [11] Usman, D. (2018). Sistem Komunikasi Sensor Jamak Pada Jaringan SCADA Berbasis RS-485. *Jurnal Rekayasa Teknologi dan Sains Terapan*, Vol. 1, No 2, Hal 9-17.
- [12] Ta'ali, & Eliza, F. (2020). Sistem Monitoring dan Kontrol Motor AC Berbasis SCADA. *JTEIN : Jurnal Teknik Elektro Indonesia*, Vol. 1, No. 1, Hal 15-20.
- [13] Controls, J. (2015). *Water-Cooled Screw Liquid Chillers*. Pennsylvania: Johnson Controls.
- [14] LOLETTE. (2021). "LE3U PLC", <https://www.lollette.com/le3u-plc-controller> diakses pada 4 November 2021 pukul 17.00 WIB.
- [15] Kho Dickson. (2020). "Pengertian Motor DC dan Prinsip Kerjanya", <https://teknikelektronika.com/pengertian-motor-dc-prinsip-kerja-dc-motor/> diakses pada 4 November 2021 pukul 18.00 WIB.
- [16] Miguel. (2015). "Analog Transmission and Scaling", <https://controlreal.com/en/analog-transmission-and-scaling/> diakses pada 28 Januari 2022 pukul 22.00 WIB.
- [17] Yuniartha, A. A. (2017). "Rancang Bangun Sistem Pengendalian Temperature Dalam Proses Degumming Pada Mini Plant Biodiesel Di Workshop Instrumentasi", Tugas Akhir. Teknik Instrumentasi. Fakultas Vokasi. Surabaya : Institut Teknologi Sepuluh Nopember.