



TUGAS AKHIR - TF 145565

RANCANG BANGUN SISTEM ALARM PADA SIMULATOR ALAT PENUKAR PANAS

WAHYU ADI KURNIAWAN
10 51 15 000 00 037

Dosen Pembimbing I
Dr. Ir. Totok Soehartanto, DEA.
NIP. 19650309 199002 1 001

Dosen Pembimbing II
Sefi Novendra Patrialova, S.Si., M.T.
NPP. 1991201712053

PROGRAM STUDI DIII TEKNOLOGI INSTRUMENTASI
DEPARTEMEN TEKNIK INSTRUMENTASI
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember
Surabaya 2018



TUGAS AKHIR - TF 145565

RANCANG BANGUN SISTEM ALARM PADA SIMULATOR ALAT PENUKAR PANAS

WAHYU ADI KURNIAWAN
10 51 15 000 00 037

Dosen Pembimbing I
Dr. Ir. Totok Soehartanto, DEA.
NIP. 19650309 199002 1 001

Dosen Pembimbing II
Sefi Novendra Patrialova, S.Si., M.T.
NPP. 1991201712053

PROGRAM STUDI DIII TEKNOLOGI INSTRUMENTASI
DEPARTEMEN TEKNIK INSTRUMENTASI
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember
Surabaya 2018



FINAL PROJECT - TF 145565

DESIGN OF THE ALARM SYSTEM ON THE HEAT EXCHANGER SIMULATOR

WAHYU ADI KURNIAWAN
10 51 15 000 00 037

Supervisor I
Dr. Ir. Totok Soehartanto, DEA.
NIP. 19650309 199002 1 001

Supervisor II
Sefi Novendra Patrialova, S.Si., M.T.
NPP. 1991201712053

DIII PROGRAM OF INSTRUMENTATION TECHNOLOGY
INSTRUMENTATION ENGINEERING DEPARTMENT
Faculty of Vocation
Institut Teknologi Sepuluh Nopember
Surabaya 2018

LEMBAR PENGESAHAN

RANCANG BANGUN SISTEM ALARM PADA SIMULATOR ALAT PENUKAR PANAS

TUGAS AKHIR

Oleh:

WAHYU ADI KURNIAWAN

NRP 10 51 15 000 00 037

Surabaya, 2 Agustus 2018
Mengetahui dan menyetujui,

Dosen Pembimbing I

Dr. Ir. Totok Soehartanto, DEA

NIP. 19650309 199002 1 001

Dosen Pembimbing II

Sefi Novendra Patrialova, S.Si., M.T.

NPP. 1991201712053

Kepala Departemen

Teknik Instrumenasi FV-ITS



Dr. Ir. Purwadi Agus Darwito, M.Sc.

NIP. 19620822 198803 1 001

LEMBER PERSETUJUAN

RANCANG BANGUN SISTEM ALARM PADA SIMULATOR ALAT PENUKAR PANAS

TUGAS AKHIR

Disusun untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Ahli Madya
pada
Departemen Teknik Instrumentasi
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember

Oleh:

WAHYU ADI KURNIAWAN

NRP 10 51 15 000 00 037

Disetujui Tim Penguji: Tanggal Ujian: 2 Agustus 2018
Periode Wisuda: September 2018

Dr. Ir. Totok Sochartanto, DEA



... (Pembimbing I)

Sefi Novendra Patrialova, S.Si., M.T.



... (Pembimbing II)

Dr. Ir. Purwadi Agus Darwito, M.Sc.



... (Penguji I)

Dwi Oktavianto Wahyu N., S.T., M.T.



... (Penguji II)

RANCANG BANGUN SISTEM ALARM PADA SIMULATOR ALAT PENUKAR PANAS

Nama : Wahyu Adi Kurniawan
NRP : 10 51 15 000 00 037
Departemen : Teknik Instrumenasi FV-ITS
Dosen Pembimbing I : Dr. Ir. Totok Soehartanto, DEA
Dosen Pembimbing II : Sefi Novendra Patrialova, S.Si., M.T.

Abstrak

Telah dirancang sistem alarm pada simulator alat penukar panas dengan aksi tombol reset yang ada pada *Human Machine Interface* sebagai aksi untuk menstabilkan proses pada simulator sehingga jalannya proses tetap berjalan normal. Selama proses dari simulator ditentukan *range operasional* dari tiap unit sensor pengendalian yang ada pada simulator sebagai batas *upper* dan *lower* alarm. *Range operasional* pada unit pemanas dengan menggunakan *heater* dengan batas *upper* 75 °C dan *lower* 23 °C, *range operasional* pada unit pendingin dengan menggunakan *refrigerant* dengan batas *upper* 35 °C dan *lower* 5 °C, serta *range operasional* pada level tangki dengan batas *upper* 45 cm dan *lower* 15 cm.

Kata Kunci: *Sistem Alarm, Range Operasional.*

DESIGN OF THE ALARM SYSTEM ON THE HEAT EXCHANGER SIMULATOR

Name	: Wahyu Adi Kurniawan
NRP	: 10 51 15 000 00 037
Department	: Instrumentation Engineering FV-ITS
Supervisor I	: Dr. Ir. Totok Soehartanto, DEA
Supervisor II	: Sefi Novendra Patrialova, S.Si., M.T.

Abstract

The alarm system has been designed on a simulator tool heat exchanger with the reset button on the Human Machine Interface as action to stabilize the process on the simulator so that the course of the process continue to run normally. During the process of the specified range operating simulator from each unit controlling censors on the simulator as the boundary of the upper and lower alarm. Range of operations on the unit heater using the heater with the upper limit of 75 °C and lower 23 °C, the cooling unit on the operational range by using the refrigerant with the upper 35 °C and lower 5 °C operating range, as well as on the level of the tank with the upper limit is 45 cm and lower 15 cm.

Keywords: *Alarm System, Range Operasional.*

KATA PENGANTAR

Segala puji dan syukur kehadirat Allah SWT atas limpahan rahmat, hidayah, dan kasih sayang-Nya serta shalawat dan salam atas junjungan Nabi Muhammad SAW atas selesainya Tugas Akhir dengan judul:

“RANCANG BANGUN SISTEM ALARM PADA SIMULATOR ALAT PENUKAR PANAS”

Sampai selesainya Tugas Akhir ini ada berbagai pihak yang telah membantu baik dukungan secara moril dan fisik, tidak lupa terima kasih penulis berikan pada:

1. Allah SWT yang telah memberikan Rahmat dan Hidayah-Nya sehingga laporan ini dapat terselesaikan tepat pada waktunya.
2. Kedua orang tua dan saudara yang senantiasa memberikan dukungan dan doa.
3. Adik saya, Khurmita Fatiha Sari yang memberikan motivasi untuk pengerjaan Tugas Akhir ini.
4. Sekar Puja Mahendrani yang selalu memberikan dukungan dan semangat serta motivasi.
5. Dr. Ir. Totok Soehartanto, DEA selaku Dosen Pembimbing I, atas segala bimbingannya, kerja sama, nasehat, dan bantuannya selama ini.
6. Sefi Novendra Patrialova, S.Si., M.T. selaku Dosen Pembimbing II, atas segala bimbingannya, kerja sama, nasehat, dan bantuannya selama ini.
7. Dr. Ir. Purwadi Agus Darwito, M.Sc. selaku Kepala Departemen Teknik Instrumentasi FV-ITS.
8. Bapak/Ibu dosen yang telah menjadi media transfer ilmu sehingga kami dapat merampungkan jenjang perkuliahan hingga terselesaikannya Tugas Akhir ini.
9. Seluruh staf pegawai Teknik Instrumentasi atas semua bentuk kerja samanya selama masa kuliah.

- 10.Mokhammad Hidayat, atas segala bantuan, ilmu, dan tenaga sehingga bisa terselesaikannya alat Tugas Akhir ini.
- 11.Seluruh sahabatku dari angkatan 2015 atas segala waktu, kebersamaan, diskusi, dan bantuannya.
- 12.Seluruh teman-teman dan segala pihak yang tidak dapat penulis sebut satu persatu terima kasih atas segala bantuannya.

Penulis menyadari bahwa penyusunan laporan ini masih jauh dari sempurna. Untuk itu saran serta kritik yang membangun sangat diharapkan. Semoga laporan ini bermanfaat bagi pembacanya dan dapat menjadi refrensi untuk pembacanya. Akhir kata penulis mohon maaf atas setiap kesalahan yang dilakukan selama pelaksanaan sampai penyusunan laporan ini.

Surabaya, 2 Agustus 2018

Penulis

DAFTAR ISI

LEMBAR PENGESAHAN	iii
LEMBER PERSETUJUAN	iv
ABSTRAK	v
<i>ABSTRACT</i>	vi
KATA PENGANTAR	vii
DAFTAR ISI	ix
DAFTAR GAMBAR	xi
DAFTAR TABEL	xiii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan	2
1.4 Ruang Lingkup Tugas Akhir	2
1.5 Manfaat	3
BAB II DASAR TEORI	5
2.1. Simulator Alat Penukar Panas	5
2.2. <i>Heat Exchanger</i>	7
2.2.1 <i>Shell and Tube Heat Exchanger</i>	7
2.3. Unit Pemanas	10
2.3.1 Elemen Panas pada <i>Water Heater</i>	11
2.3.2 Fungsi Elemen pada <i>Water Heater</i>	12
2.4. Unit Pendingin	13
2.4.1 Kompresor	15
2.4.2 Kondensor	15
2.4.3 Pipa Tembaga/Kuningan	16
2.4.4 Evaporator	17
2.4.5 <i>Thermostat</i>	18
2.5. Unit Penyimpanan	19
2.5.1 Sensor Kapasitansi	19
2.6. <i>Layer of Protection Analysis (LOPA)</i>	21
2.7. <i>Basic Process Control System</i>	27
2.7.1 Diagram Blok Unit Penyimpanan	28
2.7.2 Diagram Blok Unit Pemanas	28

2.7.3	Diagram Blok Unit Pendingin	29
2.8.	Visual Basic untuk <i>Alarm System</i>	29
BAB III METODOLOGI		31
3.1	Diagram Alir Pengerjaan Tugas Akhir	31
3.1.1	Diagram Alir Pengerjaan Perangkat Keras	32
3.1.2	Diagram Alir Pengerjaan Perangkat Lunak	34
3.2	Konsep Sistem Alarm	35
3.3	<i>Flowchart</i> Sistem Alarm	37
3.4	Deasin Mikro ke Komputer	39
3.5	Desain ke Aktuator	41
BAB IV HASIL DAN PEMBAHASAN		45
4.1.	Deskripsi Sistem Alarm.....	45
4.2.	Pengujian Alarm Dengan Data <i>Dummy</i>	47
4.3.	Hasil Uji Skenario Sistem Alarm.....	49
4.3.1	Pengujian Pada Unit Penyimpanan Air.....	49
4.3.2	Pengujian Pada Unit Pemanas	52
4.4.	Aksi Tombol Reset	56
4.5.	Alur Pengiriman dan Penerimaan Data.....	59
4.6.	Indikator Alarm Pada HMI	61
BAB V PENUTUP		63
5.1.	Kesimpulan.....	63
5.2.	Saran.....	63
DAFTAR PUSTAKA.....		65
LAMPIRAN		67
BIODATA PENULIS.....		127

DAFTAR GAMBAR

Gambar 2 1 P&ID Simulator Alat Penukar Panas	6
Gambar 2 2 Tampilan Luar <i>Heat Exchanger Shell and Tube</i>	8
Gambar 2 3 Tampilan Dalam <i>Heat Exchanger Shell and Tube</i>	8
Gambar 2 4 Jenis-Jenis <i>Shell</i>	9
Gambar 2 5 Pola <i>Tube</i>	10
Gambar 2 6 P&ID Unit Pemanas	11
Gambar 2 7 <i>Electrical Heater</i>	12
Gambar 2 8 P&ID Unit Pendingin	13
Gambar 2 9 Bagian-Bagian AC	14
Gambar 2 10 Siklus Refrigasi.....	15
Gambar 2 11 Kompresor	15
Gambar 2 12 Kondensor.....	16
Gambar 2 13 Pipa Tembaga/Kuningan.....	17
Gambar 2 14 Evaporator	17
Gambar 2 15 Thermostat	18
Gambar 2 16 P&ID Unit Penyimpanan	19
Gambar 2 17 Konsep Dasar Kapasitansi	20
Gambar 2 18 <i>Layer of Protection</i>	22
Gambar 2 19 Tingkat Bahaya Tiap <i>Layer</i>	27
Gambar 2 20 Diagram Blok Unit Penyimpanan	28
Gambar 2 21 Diagram Blok Unit Pemanas	28
Gambar 2 22 Diagram Blok Unit Pendingin.....	29
Gambar 2 23 Monitoring Data Sistem Alarm dengan HMI.....	30
Gambar 3 1 Diagram Alir Pengerjaan Sistem.....	31
Gambar 3 2 Diagram Alir Pengerjaan Perangkat Keras	33
Gambar 3 3 Diagram Alir Pengerjaan Perangkat Lunak.....	34
Gambar 3 4 Konsep Sistem Alarm	36
Gambar 3 5 Diagram Blok Sensor Thermocouple	36
Gambar 3 6 Diagram Blok Sensor Kapasitansi	37
Gambar 3 7 Diagram Alir Sistem Alarm	38
Gambar 3 9 Alur Data Mikro ke Komputer.....	39
Gambar 3 10 Tampilan HMI	40
Gambar 3 11 Alur Data ke Aktuator.....	42

Gambar 4 1 P&ID Simulator Alat Penukar Panas	46
Gambar 4 2 <i>Wiring</i> Simulasi Data Dummy	48
Gambar 4 3 <i>Virtual Serial Port Emulator</i>	49
Gambar 4 4 Indikator Alarm Unit Penyimpan Air.....	51
Gambar 4 5 Indikator Alarm Unit Pemanas	54
Gambar 4 6 Indikator Alarm Unit Pendingin	56
Gambar 4 7 Aksi Tombol Reset Unit Penyimpanan Air.....	57
Gambar 4 8 AksiTombol Reset Unit Pemanas	57
Gambar 4 9 Aksi Tombol Reset Unit Pendingin	58
Gambar 4 10 Program Komunikasi Data.....	59
Gambar 4 11 Indikator Alarm Pada HMI	61

DAFTAR TABEL

Tabel 2 1 Dielektrik Konstan Suatu Bahan	21
Tabel 2 2 <i>Safety Integrity Levels</i>	25
Tabel 2 3 <i>Target Mitigated Event Likelihood</i>	26
Tabel 2 4 <i>Range</i> Operasional	27
Tabel 3 1 <i>Range</i> Operasional Sensor.....	37
Tabel 3 2 <i>Read Address</i> Data	40
Tabel 3 3 <i>Write Address</i> Data.....	42
Tabel 4 1 Hasil Uji Alarm dengan Data <i>Dummy</i>	47
Tabel 4 2 <i>Range</i> Operasional Sensor.....	49
Tabel 4 3 Data Hasil Uji Alarm Pada Unit Penyimpanan Air.....	50
Tabel 4 4 Data Hasil Uji Alarm Pada Unit Pemanas	52
Tabel 4 5 Data Hasil Uji Alarm Pada Unit Pendingin	55
Tabel 4 6 Protokol Komunikasi Data	60

BAB I

PENDAHULUAN

1.1 Latar Belakang

Seiring dengan kemajuan ilmu teknologi dan bidang kontrol yang sangat cepat saat ini, kebutuhan akan keamanan suatu sistem bidang kontrol juga menjadi aspek penting bagi suatu plant di industri. Keselamatan dan keamanan pekerja sangat diutamakan dalam suatu proses produksi di industri. Dengan mengandalkan sistem K3 bagi pekerja dirasa masih belum cukup jika pada plant sistem keamanannya masih belum lengkap. Sehingga perlu dilakukannya peningkatan sistem keamanan pada plant di industri.

Alarm system merupakan salah satu sistem keamanan yang ada pada suatu plant. Suatu sistem alarm yang baik adalah yang dapat memberikan pengamanan terhadap pekerja dan peralatan-peralatan yang lain, tetapi juga memungkinkan hasil produksi yg maksimum. Instrumen-instrumen untuk keperluan alarm yang betul-betul terpisah dari instrumen-instrumen untuk keperluan keamanan akan memberikan pengamanan yang baik terhadap pekerja dan peralatan-peralatan, tetapi harganya menjadi mahal. Namun, bila sistem dirancang dengan baik, maka sistem ini tidak akan banyak menghambat kelangsungan produksi. Fungsi-fungsi alarm yang disatukan dalam satu instrumen yang sama akan memberikan harga yang murah. [1]

Sistem alarm yang bekerja secara *normally energizes*, sehingga sistem akan *fail-safe*, yang akan memberikan pengamanan yang optimal terhadap pekerja dan peralatan serta hanya berpengaruh sedikit terhadap harga.[2] Sistem yang bekerja secara *normally deenergized* adalah sistem yang lebih memprioritaskan kelangsungan operasi dari pada pengamanan terhadap pekerja dan peralatan. Sistem yang bekerja dengan menggunakan dua atau lebih peralatan yang dioperasikan untuk menghasilkan *normally deenergized* (sistem normalisasi) akan memberikan pengamanan yang optimal terhadap keselamatan pekerja, peralatan, dan kelangsungan produksi.

Untuk memenuhi kebutuhan sistem keamanan alarm pada suatu plant, maka penulis mencoba mengaplikasikan dua tipe alarm yang bekerja secara pararel melalui Tugas Akhir ini. Tugas Akhir ini, menjelaskan tentang sistem alarm yang mampu menormalkan suatu masalah yang terjadi pada suatu plant, sehingga ketika terjadi *error* maka alarm akan mendeteksi dan membunyikan alarm serta mencari *problem solving* yang tepat pada *error* tersebut sampai proses kembali berjalan secara normal tanpa perlu melakukan *interlocking/shutdown* dan menyimpan *error* tersebut pada *database* sehingga pekerja bisa melihat *log error* yang terjadi pada plant.

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, dapat disimpulkan beberapa rumusan masalah sebagai berikut:

1. Bagaimana merancang dan membangun sistem alarm pada simulator alat penukar panas?
2. Bagaimana mengetahui kinerja sistem alarm pada simulator alat penukar panas?

1.3 Tujuan

Adapun tujuan dari Tugas Akhir ini adalah sebagai berikut:

1. Untuk merancang dan membangun sistem alarm pada simulator alat penukar panas.
2. Untuk mengetahui kinerja sistem alarm pada simulator alat penukar panas.

1.4 Ruang Lingkup Tugas Akhir

Ruang lingkup Tugas Akhir ini terdiri dari:

1. Merancang konsep sistem alarm.
2. Data dari *field instrument* sampai ke *interface* komputer diperoleh dari TAwan yang lain.
3. Data operasional pada setiap unit proses diperoleh dari TAwan sebelumnya.

4. Sistem alarm dirancang berdasarkan respon dinamik plant pada *range* operasional unit proses (*upper limit* dan *lower limit*).
5. Sistem alarm akan bekerja jika BPCS tidak mampu menjaga stabilitas proses.
6. Sistem alarm dalam Tugas Akhir ini diasumsikan terjadi jika respon dinamis sistem menuju ke *upper limit/lower limit*.
7. Sistem alarm berfungsi untuk memberitahu operator tentang bahaya yang akan terjadi melalui suara alarm dan indikator pada unit proses yang berbahaya.
8. Pada alarm sistem yang dirancang dilengkapi dengan tombol untuk mereset agar respon dinamik unit proses menuju ke *set point* melalui aksi tombol yang dilakukan oleh operator.

1.5 Manfaat

Adapun manfaat dari Tugas Akhir ini adalah sebagai berikut:

1. Sebagai alat penunjang untuk praktikum Termodinamika, Sistem Pengukuran Aliran, Mikrokontroler dan Mikroprosesor, serta mata kuliah lain yang berhubungan dengan proses pada alat.
2. Sebagai sarana pengenalan dan pembelajaran tentang alat yang ada dalam dunia industri.

Halaman ini sengaja dikosongkan.

BAB II

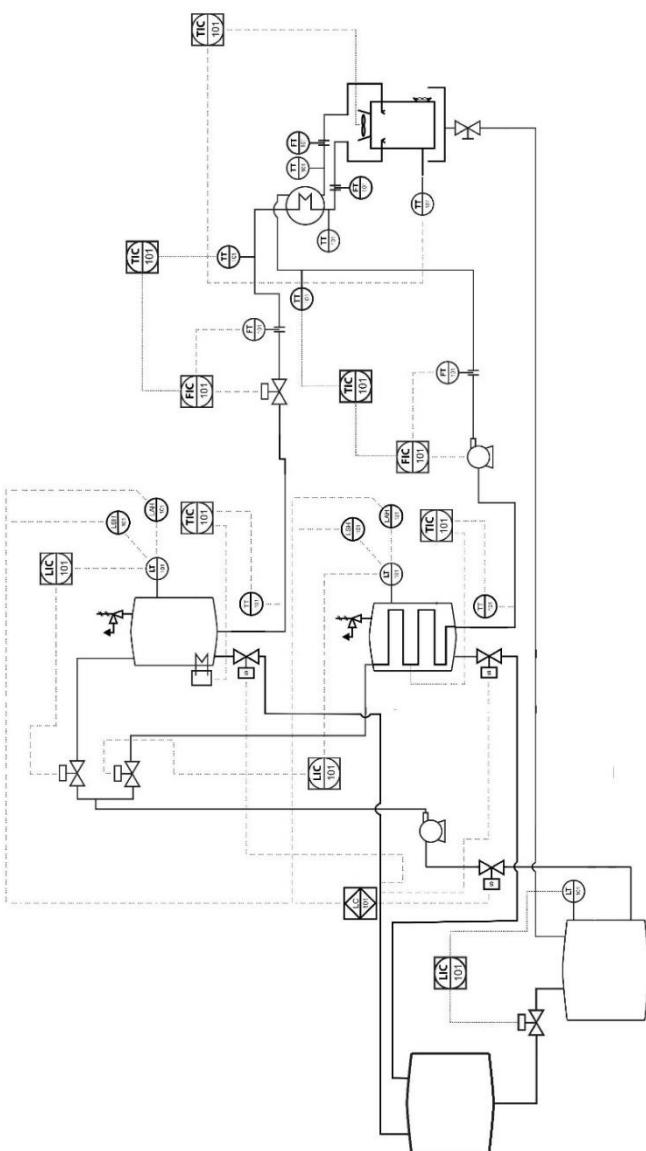
DASAR TEORI

2.1. Simulator Alat Penukar Panas

Simulator alat penukar panas memiliki sejumlah sistem proses diantaranya yaitu alat penukar panas yang berfungsi untuk mencampur air panas dan air dingin. Air panas diproduksi oleh unit pemanas dengan menggunakan heater, sedangkan air dingin diproduksi oleh unit pendingin dengan menggunakan refrigerant. Setelah dicampur, suhu air diatur kembali ke suhu normal dengan menggunakan cooling tower fan lalu dialirkan ke storage tank untuk digunakan lagi. Sistem proses yang terjadi di simulator alat penukar panas berjalan secara terus menerus (*continuous*). [1]

Semua unit yang ada pada simulator alat penukar panas dilengkapi dengan sistem pengendalian *Basic Process Control System* (BPCS) untuk menjaga stabilitas proses yang terjadi pada simulator. Pengendalian yang terjadi pada simulator alat penukar panas dikontrol melalui *Local Control Unit* (LCU). LCU mengendalikan beberapa unit yang ada di simulator alat penukar panas. LCU belum terhubung ke Control Center yang ada pada *Human Machine Interface* (HMI). Untuk menghubungkan antara LCU dengan HMI digunakan sistem komunikasi data, sehingga ada bagian yang mengatur mengenai komunikasi data dari LCU ke HMI dan mengatur HMI atau *Control Center*.

Pada Gambar 2.1, *heat exchanger* menjadi *main unit* dari simulator alat penukar panas yaitu digunakan untuk mencampur fluida panas dengan fluida dingin. Sebagai penghasil fluida panas terdapat unit pemanas yaitu *heater* yang digunakan untuk memanaskan fluida sesuai dengan *set point* yang telah ditetapkan. Dan sebagai penghasil fluida dingin terdapat unit pendingin yaitu *refrigerant* yang digunakan untuk mendinginkan fluida sesuai dengan *set point* yang telah ditetapkan. Serta terdapat unit penyimpanan yang digunakan untuk menyimpan fluida dan menyalurkan fluida.



Gambar 2 1 P&ID Simulator Alat Penukar Panas

2.2. Heat Exchanger

Heat exchanger adalah alat yang berfungsi untuk mengakomodasikan perpindahan panas dari fluida panas ke fluida dingin dengan adanya perbedaan temperatur, karena panas yang dipertukarkan terjadi dalam suatu sistem maka kehilangan panas dari suatu benda akan sama dengan panas yang diterima dari benda lain. *Heat exchanger* membantu proses perpindahan panas atau dingin suatu fluida dengan menggunakan elemen konduksi dengan tabung untuk memisahkan dua fluida. *Heat exchanger* selalu digunakan di industri bidang proses, power, perminyakan, *air conditionin*. Contoh umum penggunaan *heat exchanger* yang familiar adalah *condenser*, *evaporator*, dan *air cooler*. [2]

Untuk mengetahui besarnya panas yang dapat ditransfer dari fluida panas ke fluida dingin dengan *heat exchanger* dilakukan perhitungan dengan menggunakan persamaan laju perpindahan kalor berikut.

$$Q = \dot{m} \cdot C_p \cdot \Delta T \quad (2.1)$$

Dimana:

- Q = perpindahan kalor
- \dot{m} = kecepatan fluida (m^3/s)
- C_p = kapasitas panas
- ΔT = $T_2 - T_1$

Salah satu *heat exchanger* sederhana dimana pada aliran panas dan dinginnya bergerak dengan arah yang sama atau berlawanan. Pada aliran *parallel-flow* aliran fluida panas dan dingin untuk masuk dan kelaurnya dengan arah yang sama. Untuk *counter-flow* fluida yang masuk dan keluar memiliki perbedaan arah. Salah satu jenis *heat exchanger* yang dipakai dalam Tugas Akhir ini adalah *shell and tube heat exchanger*.

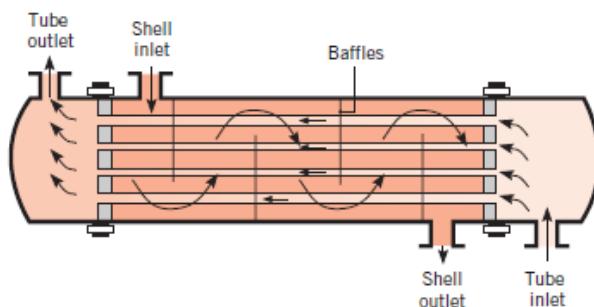
2.2.1 Shell and Tube Heat Exchanger

Jenis ini terdiri dari suatu tabung dengan diameter cukup besar yang di dalamnya berisi seberkas pipa dengan diameter

relatif kecil. Alat penukar panas ini terdiri atas suatu bundel pipa yang dihubungkan secara *parallel* dan ditempatkan dalam sebuah pipa mantel (cangkang). Fluida yang satu mengalir di dalam bundel pipa, sedangkan fluida yang lain mengalir di luar pipa pada arah yang sama, berlawanan, atau bersilangan. Untuk meningkatkan efisiensi pertukaran panas, biasanya pada alat penukar panas cangkang dan butuh dipasang sekat (*baffle*). Ini bertujuan untuk membuat turbulensi aliran fluida, namun pemasangan sekat akan memperbesar *pressure drop* operasi dan menambah beban kerja pompa, sehingga laju alir fluida yang dipertukarkan panasnya harus diatur. Berikut ini adalah gambar dari *shell and tube heat exchanger* yang digunakan pada Tugas Akhir ini. [3]



Gambar 2 2 Tampilan Luar *Heat Exchanger Shell and Tube*

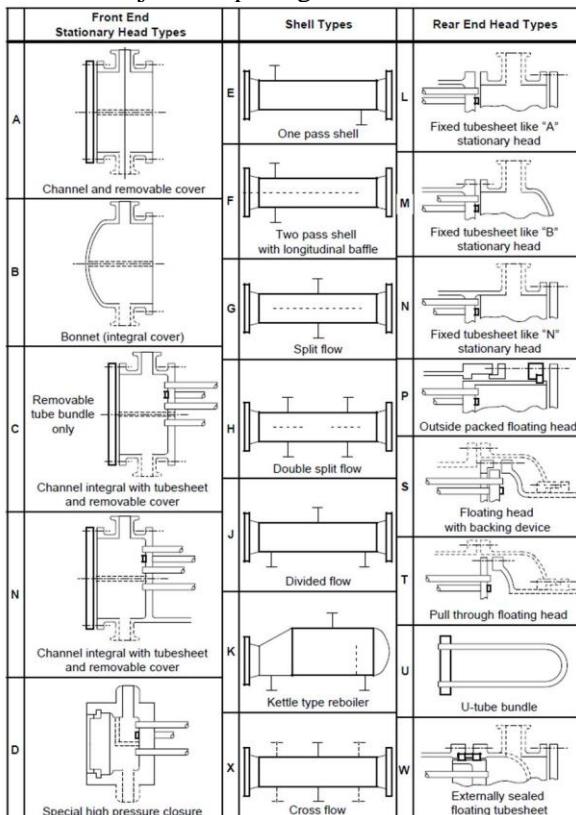


Gambar 2 3 Tampilan Dalam *Heat Exchanger Shell and Tube*

Sumber: orangorangerasing.wordpress.com, 2011

a. Shell

Konstruksi *shell* sangat ditentukan oleh keadaan *tubes* yang akan ditempatkan didalamnya. [4] *Shell* ini dapat dibuat dari pipa yang berukuran besar atau pelat logam yang dirol. *Shell* merupakan badan dari *heat exchanger*, dimana didalamnya terdapat *tube* dan *bundle*. Untuk temperatur yang sangat tinggi terkadang *shell* dibagi menjadi dua dan disambungkan dengan sambungan ekspansi. Bentuk-bentuk *shell* yang sering digunakan ditunjukkan pada gambar berikut ini.



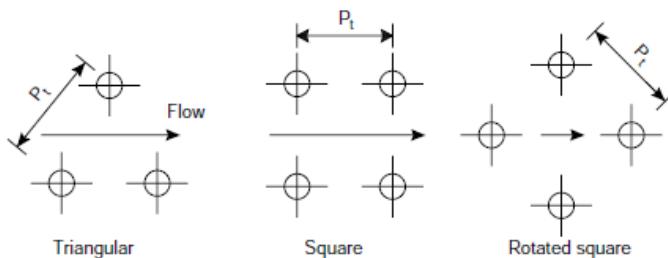
Gambar 2 4 Jenis-Jenis Shell

Sumber: surabaya.proxsisgroup.com, 2015

Tipe-tipe desain dari *shell* ditunjukkan pada Gambar 2.3 di atas. Tipe E adalah yang paling banyak digunakan karena desainnya yang sederhana serta harga yang relative murah. *Shell* tipe F memiliki nilai efisiensi perpindahan panas yang lebih tinggi dari tipe E, karena *shell* tipe F didesain untuk memiliki dua aliran (aliran U).

b. *Tube*

Tube adalah sebuah benda silindris yang memiliki lubang pada tengahnya untuk mengalirkan fluida. Berbeda dengan *pipe*, *tube* ukurannya relatif kecil dan tidak dibatasi oleh *spool* (ukuran panjang tertentu seperti pada pipa, umumnya 6 meteran). Di samping itu, *tube* lebih fleksibel dan mudah untuk dibentuk atau dibending dengan pipa. *Tube* ukurannya dibuat kecil namun banyak. Tujuannya untuk memperluas permukaan kontak dengan fluida lain, sehingga nantinya pertukaran panas lebih efektif. Biasanya *tube* ini akan disusun sedemikian rupa mengikuti pola tertentu, agar proses pemindahan panasnya efektif. Berikut ini contoh pola dari *tube*. [5]



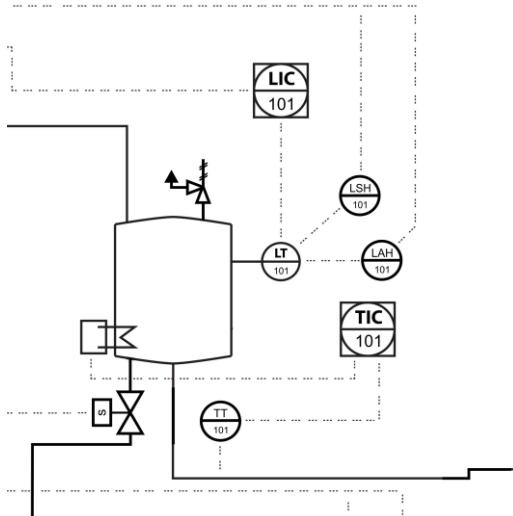
Gambar 2.5 Pola *Tube*

Sumber: artikel-teknologi.com, 2014

2.3. Unit Pemanas

Water heater adalah alat yang digunakan untuk memanaskan air yang menggunakan energi sebagai sumber pemanas. [6] Pada awalnya untuk mendapatkan air panas biasanya jika memasak air atau memanfaatkan air panas langsung dari alam. Cara kerja alat

ini sederhana, air ditempatkan dibagian atas wadah berupa tabung yang juga diisi jaringan kawat-kawat tipis sebagai pengantar panas, dimana bagian bawahnya diletakkan sebuah alat pemanas berbahan bakar gas. Lalu air panas mengalir ke bak mandi tanpa ada perantara. *Water heater* saat ini sudah semakin berkembang, dan sudah banyak sekali sistem yang dipakai dan bahan baku yang semakin baik dari hari ke hari. Dalam *water heater* terdapat alat yang bertindak untuk memanaskan air, alat tersebut kerap disebut elemen panas.



Gambar 2 6 P&ID Unit Pemanas

2.3.1 Elemen Panas pada Water Heater

Elemen panas listrik (*electrical heating element*) pada *water heater* yaitu suatu alat elektrik yang bisa memanaskan air dengan mudah serta cepat. Sumber panas elemen itu didapatkan dari kawat yang mempunyai tahan listrik tinggi (*resistance wire*). Niklin yaitu bahan umum yang digunakan pada elemen, lalu dilapisi oleh bahan isolasi yang bisa melanjutkan panas, jadi aman untuk dipakai. [7]



Gambar 2.7 Electrical Heater

Cepat atau lambat *water heater* dalam memanaskan air ditetapkan oleh besar kecilnya watt yang ada pada elemen. Tetapi, harus juga dicocokkan dengan tabung *water heater* berupa liter air yang akan dipanasi. Alat elektrik rumah tangga seperti strika, *magic com*, solder, panas pada dispenser, dan lain-lain juga memakai komponen *basic element*. Ukuran elemen yang digunakan *water heater* pada simulator alat penukar panas yaitu *water heater* besar yang membutuhkan daya 1200 watt – 2000 watt.

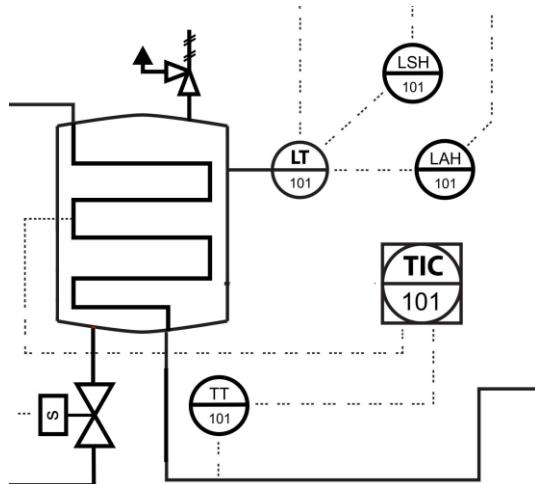
2.3.2 Fungsi Elemen pada Water Heater

Untuk memanaskan air yang ada pada tabung *water heater* dengan sumber panas dari kawat yang mempunyai tahanan listrik tinggi, memiliki bahan niklin serta dilapisi isolasi yang lalu dialiri arus listrik yaitu manfaat dari elemen pada *water heater*. Untuk menghindari agar tak berlangsung panas yang berkelebihan (*over*

heating). Umumnya *water heater* memakai satu alat yakni *thermostat*.

2.4. Unit Pendingin

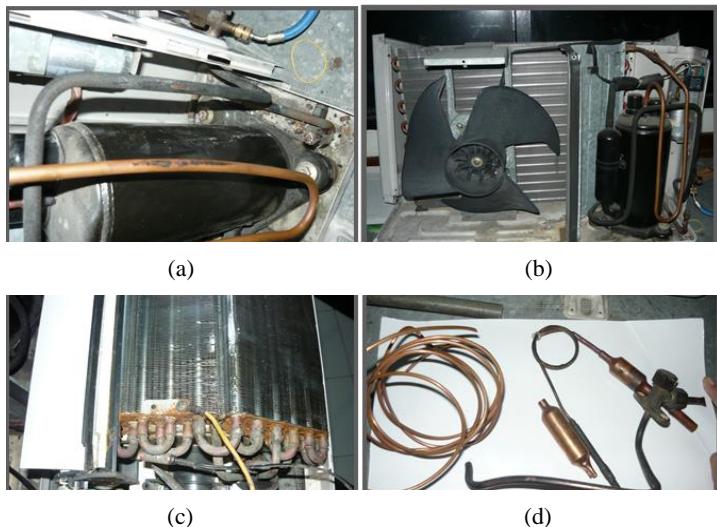
Unit pendingin yang digunakan pada simulator alat penukar panas adalah sistem otomatisasi pada *air conditioner* (AC) yang merupakan sebuah alat yang mampu mengkondisikan udara. Dengan kata lain, AC berfungsi sebagai penyejuk udara yang diinginkan (sejuk atau dingin) dan nyaman bagi tubuh. AC lebih banyak digunakan di wilayah yang beriklim tropis dengan kondisi temperatur udara yang relative tinggi (panas).



Gambar 2 8 P&ID Unit Pendingin

Refrigerant merupakan media pemindah kalor pada sistem refrigerasi, dimana *refrigerant* menyerap kalor pada tekanan rendah melalui evaporator dan melepaskan panas pada tekanan tinggi melalui kondensor. Evaporator menyerap panas dari ruangan yang dikondisikan sehingga temperatur ruangan menjadi dingin dan *refrigerant* bertekanan rendah di dalam evaporator mengalami

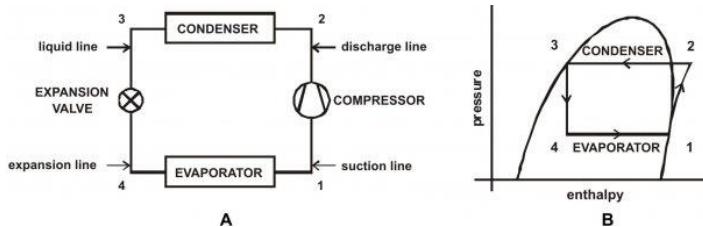
pendidihan. Uap refrigerant tersebut kemudian dikompresikan oleh kompresor bertekanan tinggi sehingga temperatur uap refrigerant tersebut juga mengalami kenaikan sehingga panas refrigerant tersebut dapat dilepaskan ke lingkungan melalui kondensor sedangkan refrigerant mengalami kondensasi sehingga refrigerant berubah fasa menjadi cairan pada tekanan tinggi. Cairan refrigerant tersebut kemudian diekspansi ke tekanan evaporator untuk siklus selanjutnya oleh alat ekspansi. Siklus refrigerasi dapat dilihat pada Gambar 3.3.



Gambar 2 9 Bagian-Bagian AC

(a) Kompresor, (b) Kondensor, (c) Evaporator, (d) Tembaga Kuningan

Komponen AC dikelompokan menjadi 4 bagian, yaitu komponen utama, komponen pendukung, kelistrikan, dan bahan pendingin (*refrigerant*) sesuai yang tertera pada Gambar 2.9.



Gambar 2 10 Siklus Refrigerasi
Sumber: www.researchgate.net, 2014

2.4.1 Kompresor

Kompresor adalah sebuah alat yang berfungsi untuk menyalurkan gas refrigerant ke seluruh sistem. Jika dianalogikan, cara kerja kompresor AC layaknya seperti jantung di tubuh manusia. Kompresor memiliki dua pipa, yaitu pipa hisap dan pipa tekan. Dan memiliki dua daerah tekanan, yaitu tekanan rendah dan tekanan tinggi. Ada tiga jenis kompresor, yaitu kompresor torak (*reciprocating*), kompresor sentrifugal, dan kompresor *rotary*. [8]



Gambar 2 11 Kompresor

2.4.2 Kondensor

Kondensor berfungsi sebagai alat penukar kalor, menurunkan temperatur refrigerant, dan mengubah wujud

refrigerant dari bentuk gas menjadi cair. Kondensor pada AC biasanya disimpan pada luar ruangan (*outdoor*). Kondensor biasanya didinginkan oleh kipas (*fan*), fan ini berfungsi menghembuskan panas yang dihasilkan oleh kondensor pada saat pelepasan kalor yang diserap oleh gas refrigerant. Agar proses pelepasan kalor bisa lebih cepat, pipa kondensor didesain berlukut dan dilengkapi dengan sirip.



Gambar 2 12 Kondensor

2.4.3 Pipa Tembaga/Kuningan

Pipa Tembaga/Kuningan merupakan komponen utama yang berfungsi menurunkan tekanan refrigeran dan mengatur aliran refrigeran menuju evaporator. Fungsi utama pipa kapiler ini sangat vital karena menghubungkan dua bagian tekanan berbeda, yaitu tekanan tinggi dan tekanan rendah. Refrigeran bertekanan tinggi sebelum melewati pipa kapiler akan diubah atau diturunkan tekanannya. Akibat dari penurunan tekanan refrigeran menyebabkan penurunan suhu. Pada bagian inilah (pipa kapiler) refrigeran mencapai suhu terendah (terdingin). Pipa kapiler terletak antara saringan (*filter*) dan evaporator. [9]



Gambar 2 13 Pipa Tembaga/Kuningan

Sumber: www.indotrading.com, 2018

2.4.4 Evaporator

Evaporator setelah refrigerant melewati katup ekspansi maka selanjutnya akan menuju ke evaporator. Evaporator pada sistem AC memiliki fungsi kebalikan dari komponen kondensor. Fungsi dari evaporator yaitu untuk menyerap panas, refrigerant yang melewati evaporator ini menyerap panas dari udara yang dihembuskan oleh komponen blower. Sehingga udara yang dihembuskan oleh blower setelah melewati evaporator bersuhu dingin. Karena refrigerant tadi menyerap panas dari udara maka bentuk refrigerant setelah keluar dari evaporator akan berubah dari yang berbentuk partikel-partikel kecil menjadi gas. [10]



Gambar 2 14 Evaporator

Sumber: www.hunker.com, 2012

Sedangkan untuk sistem pengendalian temperature pada AC (*Air Conditioner*) menggunakan thermostat yang mana fungsi thermostat adalah suatu rangkaian komponen yang berfungsi atau berguna untuk mengatur perubahan suhu (baik suhu panas ataupun suhu dingin) yang dapat diatur sesuai dengan yang dikehendaki ataupun berkerja secara independent (mengikuti perubahan suhu) yang peniterepannya terdapat pada macam-macam peralatan sistem pendingin dan sistem pemanas.

2.4.5 *Thermostat*

Peranan fungsi dari thermostat pada AC adalah mengatur tekanan suhu dingin yang dihasilkan pada evaporator AC (*cooling coil*), melalui *remote AC* yang telah disediakan yang dapat digunakan untuk mengatur temperatur yang diinginkan (sesuai yang dikehendaki), yaitu apabila sensor dingin pada thermostat telah mencapai titik yang di inginkan maka secara otomatis sistem kelistrikan yang ada pada thermostat akan mengirim perintah untuk mematikan atau mengistirahatkan kinerja dari komponen AC yaitu kompresor AC yang secara garis besar telah menghentikan sistem komponen AC secara keseluruhan. [11]



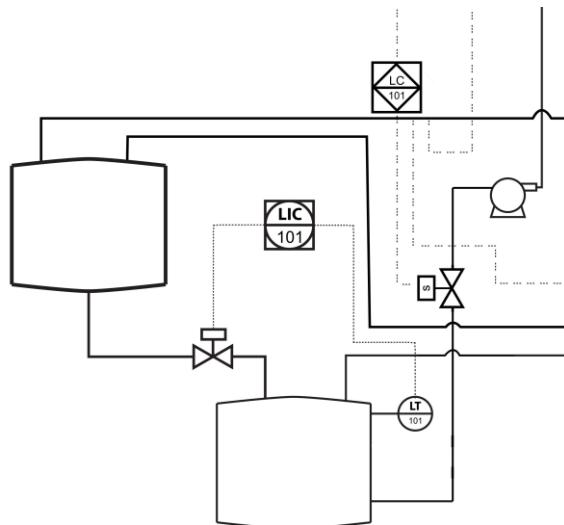
Gambar 2 15 Thermostat
Sumber: www.alibaba.com, 2017

Letak thermostat sendiri diletakan pada kisi-kisi evaporator (*cooling coil*) sehingga fungsi thermostat dapat berkerja secara

efektif dan maksimal untuk menerima dan mengatur suhu dingin yang dihasilkan pada sistem pendingin AC.

2.5. Unit Penyimpanan

Unit penyimpanan yang digunakan pada simulator alat penukar panas menggunakan sensor level kapasitansi sebagai elemen sensor untuk mendeteksi tinggi air pada tanki penyimpanan. Sedangkan untuk aktuatornya menggunakan pompa dan stepper.



Gambar 2 16 P&ID Unit Penyimpanan

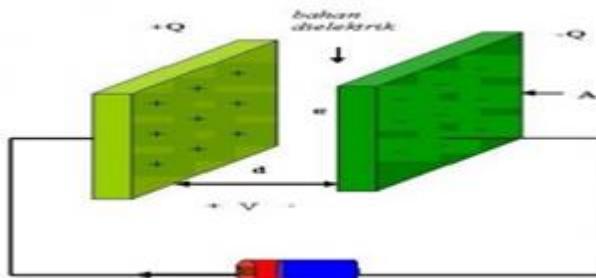
2.5.1 Sensor Kapasitansi

Kapasitor atau kapasitif adalah suatu komponen elektronika yang terdiri dari dua buah plat penghantar sejajar yang disekat satu sama lain dengan suatu bahan elektrik. Komponen – komponen ini sangat penting dalam elektronika karena memiliki sifat-sifat seperti berikut:

- Dapat menyimpan muatan listrik.

- b. Dapat menahan arus searah.
- c. Dapat melewatkannya bolak-balik.

Sensor kapasitif ini serupa dengan alat ukur regangan sebuah ukuran sel kapasitansi perubahan dalam karakteristik listrik. [12] Kapasitif adalah sebuah alat yang menyimpan muatan listrik, terdiri dari pelat logam yang dipisahkan oleh isolator listrik, pelat logam yang terhubung ke sebuah sirkuit listrik eksternal. Muatan listrik dapat ditransfer dari satu pelat logam yang lain. Kapasitansi sebuah kapasitif yaitu ukuran kemampuannya untuk menyimpan muatan. Kapasitansi – kapasitansi dari suatu kapasitor berbanding lurus dengan daerah pelat logam dan berbanding terbalik dengan jarak antar kedua pelat logam. Ini juga tergantung pada karakteristik dari bahan isolasi yang berada diantara pelat dan kapasitor. Karakteristik ini disebut permitivitas adalah ukuran seberapa baik bahan isolasi meningkatkan kemampuan kapasitor menyimpan muatan.



Gambar 2 17 Konsep Dasar Kapasitansi
Sumber: elektronika-dasar.web.id, 2016

Satuan kapasitansi adalah coulomb/volt (C/V) atau farad (F). Satu farad adalah jumlah muatan listrik sebesar satu coulomb yang disimpan di dalam elektrik (zat perantara) dengan beda potensial sebesar satu volt. Kapasitansi suatu kapasitor bergantung pada:

- a. Bahan dielektrik yang digunakan
- b. Luas dari masing – masing pelat
- c. Jarak antara kedua pelat

$$c = \epsilon \frac{A}{d} \quad (2.2)$$

Dimana:

- c = kapasitansi (F)
- A = luas pelat
- d = jarak pelat
- ϵ = permivitas

Berdasarkan rumus diatas dapat diketahui bahwa sensor kapasitif merupakan sensor elektronika yang bekerja berdasarkan konsep kapasitif. Konsep kapasitor yang digunakan dalam sensor kapasitif adalah proses menyimpan dan melepas energi listrik dalam bentuk muatan-muatan listrik pada kapasitor yang dipengaruhi oleh luas permukaan, jarak dan bahan dielektrum.

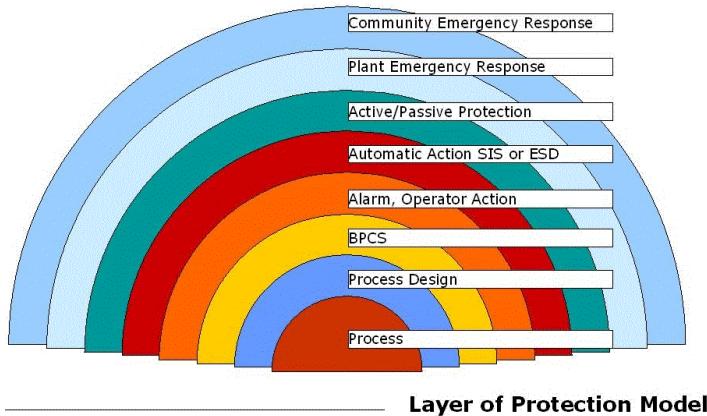
Tabel 2 1 Dielektrik Konstan Suatu Bahan

Material	Dielektrik Constant (ϵ)
Udara kering (68°F)	1,000536
Air	4 - 88
Kayu kering	2 - 6
Kertas minyak	4
Kaca	3,7 - 10
kertas	2,3

2.6. Layer of Protection Analysis (LOPA)

Layer of Protection Analysis (LOPA) adalah suatu metode yang digunakan untuk menilai tingkat keamanan suatu plant (*safety*) terhadap suatu ancaman risiko bahaya dengan menganalisa lapisan-lapisan pelindung yang telah diterapkan pada sistem atau komponen tersebut. [13]

Protection Layer (PL) adalah perangkat, sistem atau tindakan yang mampu mencegah penyimpangan proses dari kemungkinan dampak akhir yang lebih parah. Berikut adalah penjelasan untuk masing-masing PL:



Gambar 2 18 Layer of Protection

Sumber: Agung, 2017

a. ***Process Design***

Process Design merupakan layer protection pertama pada sebuah sistem atau komponen pada dunia Industri. Proses desain juga turut mereduksi bahaya proses dengan mengurangi kemungkinan adanya penyimpangan proses. Contoh dari proses desain adalah seperti ketebalan pipa, pemilihan bahan pada instrumen yang berhubungan dengan kondisi sistem.

b. ***Basic Process Control System (BPCS)***

Tahap pengamanan kedua setelah adanya desain untuk proses adalah BPCS. BPCS diberi kemampuan untuk menjaga agar kondisi proses tetap berada di daerah aman, yaitu di daerah *process set point*. BPCS biasanya terdiri dari *sensor transmitter*, *logic solver (indicated control)*, dan *final element* atau *actuator*. BPCS bisa berupa loop pengendalian *full-automatic* namun juga bisa berupa *semi-automatic*.

c. Alarms and Operator Actions

Alarm secara umum dapat didefinisikan sebagai bunyi peringatan atau pemberitahuan. Dalam istilah jaringan, alarm dapat juga didefinisikan sebagai pesan berisi pemberitahuan ketika terjadi penurunan atau kegagalan dalam penyampaian sinyal komunikasi data ataupun ada peralatan yang mengalami kerusakan (penurunan kinerja). Pesan ini digunakan untuk memperingatkan operator atau administrator mengenai adanya masalah (bahaya) pada jaringan. Alarm memberikan tanda bahaya berupa sinyal, bunyi, ataupun sinar. Fungsi alarm adalah memberitahukan apabila terjadi bahaya dan kerusakan ataupun kejadian yang tidak diharapkan pada jaringan melalui sinyal.

Sistem alarm merupakan level perlindungan ketiga pada layer LOPA. Sistem alarm ada yang tergabung dengan BPCS dan ada pula yang independen terhadap BPCS. Ketika BPCS tak mampu mengatasi permasalahan ini (kondisi bahaya proses terus meningkat), maka sistem pengaman selanjutnya akan aktif.

Alarm akan menyala apabila terjadi suatu penyimpangan proses dari keadaan yang diinginkan. Ketika kondisi ini terlampaui, maka sistem alarm akan aktif. Kondisi ini biasa dinamakan *high or low condition*. Alarm akan membutuhkan tindakan operator sesuai SOP sebagai suatu bentuk tindak lanjut indikasi kegagalan.

d. Automatic Actions (SIS)

SIS adalah kombinasi sensor, *logic solver*, dan *final element* dengan tingkat integritas tingkat tinggi sistem ini bisa berupa ESD (*Emergency ShutDown Valve*), yakni *valve* yang bekerja untuk mematikan plant secara mendadak dengan cara memutus aliran fluida proses. SIS merupakan layer terakhir yang berfungsi dalam penanganan sebelum kejadian (*prevention*).

e. Physical Protection (*Relief Valve, Rupture Disc, dst*)

Perlindungan fisik merupakan layer yang bertindak apabila dampak (*fire*) sudah terlanjur terjadi. Layer ini berfungsi untuk melindungi komponen dan instrumen mayor pada sebuah industri.

f. *Passive Protection (Dikes, Blast, Walls, etc)*

PL ini adalah alat pasif yang akan bekerja yang kegagalan dan kerusakan sudah terjadi, dan dibutuhkan tindakan penanggulangan untuk mencegah risiko kecelakaan meluas, seperti mencederai manusia, dan lingkungan sekitar. *Passive protections* biasa berupa *Dike* (galian) dan *Blast Wall* (tembok tahan api, biasanya dipasang di offshore).

g. *Plant Emergency Response*

Fitur ini mencakup pasukan pemadam kebakaran, sistem pemadam manual, fasilitas evakuasi, dll.

h. *Community Emergency Response*

Berbagai aset penting, properti bahkan SDM bisa saja menjadi terancam. Oleh karena itu tanggap darurat (*emergency response*) adalah hal yang wajib dikembangkan di perusahaan untuk mengantisipasi kerugian akibat bencana yang karena suatu hal dapat tidak terkendali. Untuk mengatasi kejadian-kejadian seperti meledaknya sebuah plant dan bencana alam diperlukan adanya sistem manajemen ERP yang tepat. *ERP (Emergency Response Plan/ Tanggap Darurat Bencana)* adalah sistem yang menggabungkan beberapa depertemen mencakup HRD, keamanan (*security*), kesehatan, termasuk K3 (keselamatan dan kesehatan kerja) itu sendiri untuk menanggulangi kejadian bencana tersebut.

i. *Safety Integrity Level*

Safety integrity level adalah suatu ukuran yang menyatakan besarnya PFD dari suatu sistem. Sedangkan PFD atau *probability failure on demand* merupakan kemungkinan kegagalan komponen atau sistem ketika ia dibutuhkan untuk bekerja. Dengan mengetahui nilai SIL yang bekerja pada sebuah sistem maka akan diketahui pula seberapa tingkat keamanan atau safety dari sistem tersebut.

Dalam praktiknya ada dua buah metode penentuan SIL, yakni *SIL determination* dan *SIL verification*. *SIL Determination* adalah

proses meninjau tingkat risiko yang terkait dengan peristiwa bahaya tertentu dan menilai kontribusi untuk pengurangan risiko yang diperlukan SIF agar kekurangan reduksi risiko dapat dicapai.

SIL *determination* termasuk di dalamnya adalah *Layer of Protection Analysis*, yakni bertujuan untuk menentukan rekomendasi tingkatan SIL yang sesuai untuk sebuah sistem *integrity function* agar dapat mengurangi kemungkinan bahaya atau risiko. Gambar berikut ini adalah pengklasifikasian kategori SIL:

Tabel 2.2 Safety Integrity Levels

DEMAND MODE OF OPERATION		
Safety Integrity Level (SIL)	Target Average Probability of Failure on Demand	Target Risk Reduction
4	$\geq 10^{-5}$ to $< 10^{-4}$	$> 10\ 000$ to $\leq 100\ 000$
3	$\geq 10^{-4}$ to $< 10^{-3}$	$> 1\ 000$ to $\leq 10\ 000$
2	$\geq 10^{-3}$ to $< 10^{-2}$	> 100 to $\leq 1\ 000$
1	$\geq 10^{-2}$ to $< 10^{-1}$	> 10 to ≤ 100

SIL 4 merupakan tingkatan nilai SIL dengan kategori tingkat bahaya paling tinggi dan dampak yang paling besar, oleh karenanya harus diterapkan SIL 4 sehingga harus dilakukan reduksi risiko keterjadinya kegagalan hingga 1 kejadian per 10000 jam. Berikut ini adalah rumusan untuk menentukan *Probability of Failure on Demand* (PFD) yang sesuai untuk sebuah SIF:

$$\text{SIF Integrity Level} = \frac{\text{TMEL}}{\text{IEL}} \quad (2.3)$$

Dimana:

TMEL = Target Mitigated Event Likelihood

IEL = Intermediate Event Likelihood

ICL = Initiating Cause Likelihood

$$\text{IEL} = \text{ICL} \times \text{PFD}_1 \times \text{PFD}_2 \times \text{PFD}_n \quad (2.4)$$

Dari Nilai PFD yang telah didapatkan bisa digunakan untuk merepresentasikan nilai SIL yang sesuai untuk sebuah *safety instrumented function* (SIF).

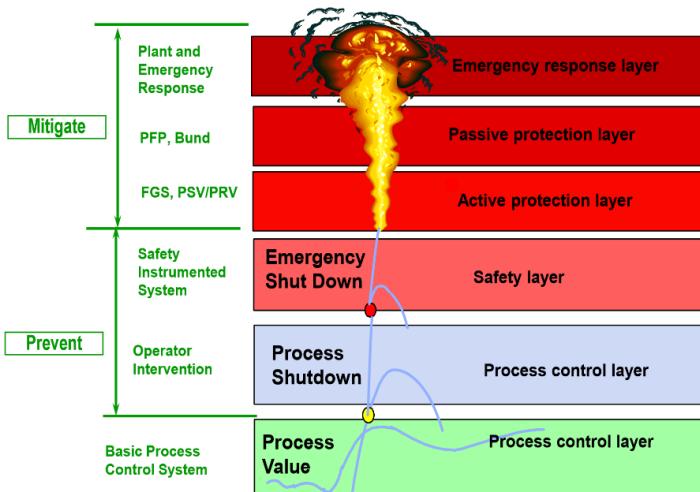
Penentuan TMEL disesuaikan dengan tingkat keparahan (*severity level*) dari bahaya yang mungkin terjadi. *Target Mitigated Event Likelihood* merupakan target suatu perusahaan yang ditujukan agar potensi/ probabilitas terjadinya suatu kegagalan atau bahaya pada plant dapat bernilai sekecil mungkin. Terdapat sebuah standar yang mewakili tingkat keparahan suatu bahaya berdasarkan konsekuensi yang ditimbulkannya yang juga telah diterjemahkan ke dalam angka-angka harapan TMEL, sebagaimana pada tabel berikut:

Tabel 2.3 Target Mitigated Event Likelihood

<i>Severity Level</i>	<i>Safety Consequence</i>	<i>Target Mitigated Event Likelihood</i>
C _A	<i>Singer first aid injury</i>	$3 \cdot 10^{-2}$ per year
C _B	<i>Multiple first aid injuries</i>	$3 \cdot 10^{-3}$ per year
C _C	<i>Single disabling injury or multiple serious injuries</i>	$3 \cdot 10^{-4}$ per year
C _D	<i>Single on-site fatality</i>	$3 \cdot 10^{-5}$ per year
C _E	<i>More than one and up to three on-site fatalities</i>	$3 \cdot 10^{-6}$ per year

Dibawah ini merupakan tingkatan bahaya yang dimungkinkan terjadi apabila setiap lapisan proteksi gagal menjalankan tugasnya. Terdapat pula pada Gambar 2.21 tersebut kinerja reduksi bahaya yang dilakukan oleh tiap-tiap *layer* serta pembagian *Layer of Protection* ke dalam 2 kategori, yakni *prevention* (pencegahan) dan *mitigation* (peringanakan insiden)

Dalam penerapan sistem alarm pada simulator alat penukar panas, ditetapkan *range* operasional (*upper limit/lower limit*) sebagai acuan identifikasi bahaya pada plant. Berikut *range* operasional dari beberapa unit kontrol pada simulator alat penukar panas:



Gambar 2 19 Tingkat Bahaya Tiap Layer

Sumber: www.flickr.com, 2011

Tabel 2 4 Range Operasional

No	Sensor	Unit	High	Low
1	Kapasitansi <i>level storage tank</i>	Penyimpanan	33 cm	13 cm
2	<i>Thermocouple heater</i>	Pemanas	75 °C	23 °C
3	<i>Thermocouple refrigerant</i>	Pendingin	30 °C	7 °C

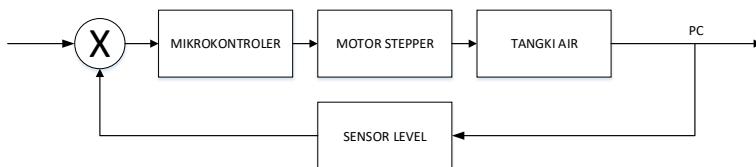
2.7. Basic Process Control System

BPCS merupakan salah satu *protection layer* yang bertujuan untuk memonitoring serta mengendalikan jalannya sebuah proses. Jalannya proses senantiasa dievaluasi melalui *set point* yang sudah ditetapkan perusahaan. BPCS merupakan salah satu *layer* yang paling signifikan dalam kaitannya meminimalisir kegagalan sistem. BPCS dikatakan terpasang apabila terdapat *element sensor*, *logic solver* dan *final element*. [14]

Dalam perancangan sistem alarm pada simulator alat penukar panas yang dirancang pada tiga unit kontrol yaitu unit pemanas, unit pendingin, dan unit penyimpanan. Ketiga unit tersebut dikontrol dengan menggunakan BPCS yang *range* operasionalnya telah ditetapkan sebelumnya pada Tabel 2.3. Berikut diagram blok BPCS sistem alarm pada masing-masing unit kontrol:

2.7.1 Diagram Blok Unit Penyimpanan

Pada pengendalian unit penyimpanan menggunakan sensor kapasitansi sebagai sensor elemen dan motor stepper sebagai actuator, sedangkan untuk kontrolernya menggunakan mikrokontroller ATMega 16. [14]

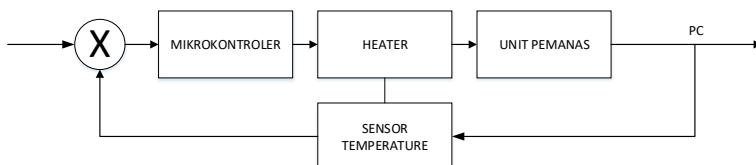


Gambar 2 20 Diagram Blok Unit Penyimpanan

Sumber: Hendrawan, 2014

2.7.2 Diagram Blok Unit Pemanas

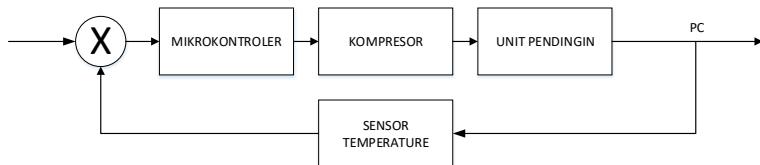
Pada pengendalian unit pemanas menggunakan sensor thermocouple sebagai sensor elemen dan heater sebagai actuator, sedangkan untuk kontrolernya menggunakan mikrokontroller ATMega 16.



Gambar 2 21 Diagram Blok Unit Pemanas

2.7.3 Diagram Blok Unit Pendingin

Pada pengendalian unit pendingin menggunakan sensor thermocouple sebagai sensor elemen dan kompresor sebagai actuator, sedangkan untuk kontrolernya menggunakan mikrokontroler ATMega 16.



Gambar 2 22 Diagram Blok Unit Pendingin

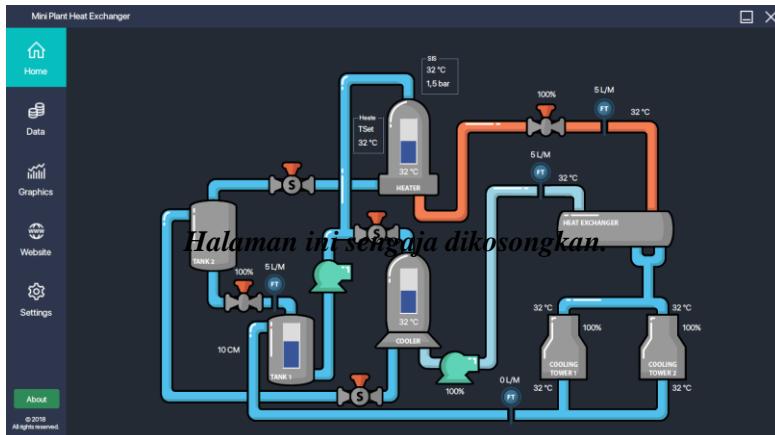
2.8. Visual Basic untuk *Alarm System*

Visual studio adalah salah satu aplikasi pemrograman *under Windows* yang berbasis visual atau grafis. Aplikasi ini dikeluarkan oleh Microsoft Corporation yang juga pemilik dari sistem operasi Microsoft Windows. [15]

Pada awalnya BASIC (*Beginner's All-purpose Symbolic Instruction Code*) adalah bahasa pemrograman yang merupakan awal dari bahasa pemrograman tingkat tinggi sesudahnya, yang berbasai DOS (*Diskette Operating System*). BASIC memiliki struktur bahasa yang sulit dan memiliki tampilan yang tidak menarik, dengan kemajuan teknologi maka diperlukan suatu aplikasi pemrograman yang bukan hanya cepat tapi juga menarik dan *user friendly* atau mudah digunakan. Maka Microsoft mengembangkan Visual Basic sebagai salah satu bahasa pemrograman tingkat tinggi berdasarkan dari bahasa pemrograman BASIC.

Visual Basic, membuat bahasa BASIC yang sudah diunakan menjadi lebih mudah dengan orientasi grafis dan objek atau OOP (*Objects Oriented Programming*). Yang lebih mudah digunakan, cepat dengan *wizard generator code*, dan memungkinkan mendisain *interface* yang menarik dan mudah untuk digunakan *user* nantinya.

Dalam penggunaannya sebagai bahasa pemrograman untuk sistem alarm pada simulator alat penukar panas, Visual Basic digunakan untuk memonitoring data dari mikrokontroller menggunakan *serial port* yang terhubung ke komputer. Data yang dibaca berupa data dari sensor yang ada di tiap unit proses yang ada pada simulator alat penukar panas. Dengan menggunakan Visual Basic pengiriman dan penerimaan data bisa berjalan secara *multitasking* yang membuat jalannya program bisa berjalan lebih ringan. Pembacaan menggunakan data *split* yang telah ditetapkan pada mikrokontroller.



Gambar 2 23 Monitoring Data Sistem Alarm dengan HMI

BAB III

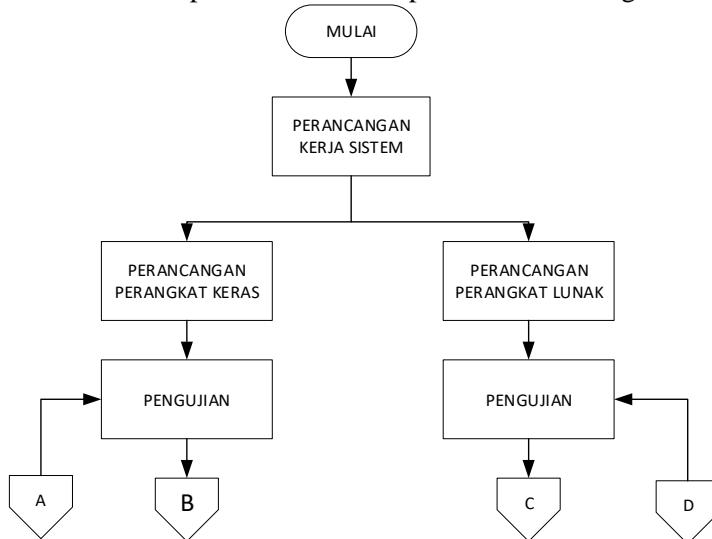
METODOLOGI

3.1 Diagram Alir Penggerjaan Tugas Akhir

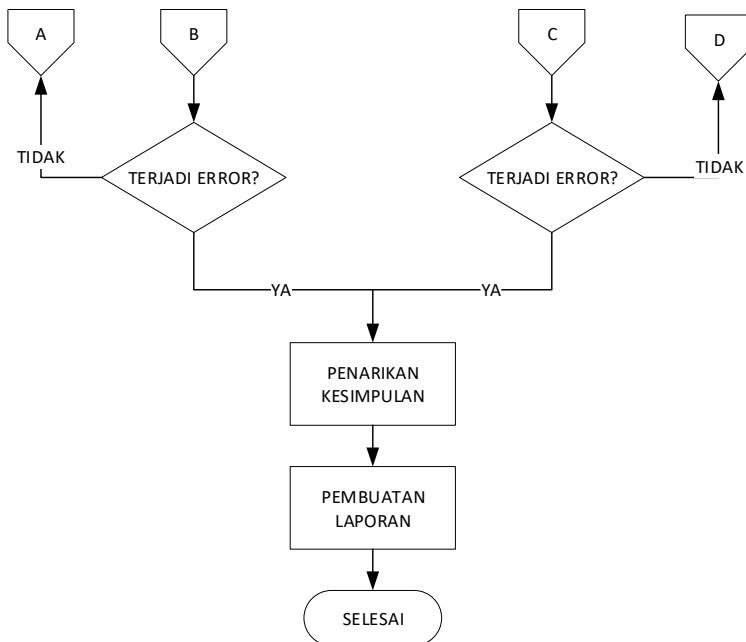
Untuk mencapai tujuan penyelesaian tugas akhir yang direncanakan, maka perlu dilakukan suatu langkah-langkah dalam menyelesaikan tugas akhir ini. Adapun langkah-langkahnya adalah sebagai berikut:

- a. Perancangan kerja sistem, dimulai dengan merancang skema mini plant heat exchanger lalu menambahkan sistem alarm pada skema tersebut sehingga menghasilkan sebuah desain (P&ID).
- b. Dibagi menjadi dua bagian, perancangan perangkat keras dan perangkat lunak yang dijelaskan pada sub-bab di bawah ini.
- c. Penarikan kesimpulan, ketika perancangan perangkat keras dan perangkat lunak telah selesai dan tidak terjadi *error* maka bisa ditarik sebuah kesimpulan dari hasil yang didapat.

Pembuatan laporan, membuat laporan hasil dari tugas akhir.



Gambar 3 1 Diagram Alir Penggerjaan Sistem



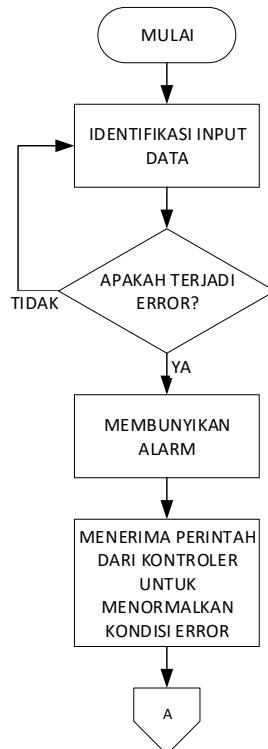
Gambar 3 1 (Lanjutan) Diagram Alir Penggerjaan Sistem

3.1.1 Diagram Alir Penggerjaan Perangkat Keras

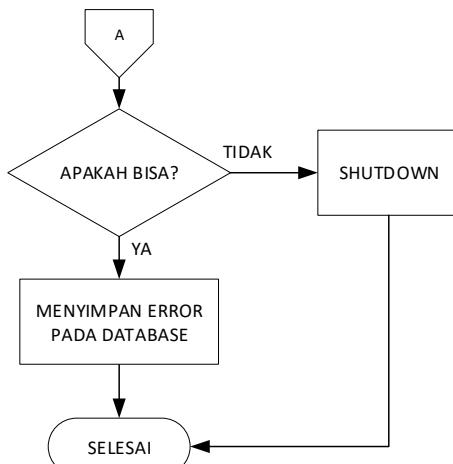
Dalam perencanaan kerja sistem dibagi menjadi dua bagian, yaitu perancangan perangkat keras dan perangkat lunak. Pada perancangan perangkat keras berikut adalah langkah-langkahnya:

- Identifikasi input data, sebelum memulai penggerjaan sistem alarm pada plant terlebih dahulu diidentifikasi input data yang digunakan dalam sistem alarm dari mulai identifikasi jumlah sensor yang digunakan dan *range* operasional serta *upper/lower limit* sensor. Hal ini berguna sebagai penentuan *upper limit* dan *lower limit* yang digunakan pada sistem alarm.
- Ketika input melebihi *range* operasional maka kontroler akan memberikan sinyal output untuk membunyikan alarm dan menampilkan *error* pada HMI.

- c. Pada saat terjadi *error* operator bisa menekan tombol *reset* untuk menormalkan kembali proses pada plant sesuai *range* operasionalnya dan mematikan alarm sehingga proses pada plant tetap berjalan.
- d. Jika aksi penekanan tombol *reset* berhasil tanpa terjadinya *error* lagi maka catatan *error* sebelumnya akan tersimpan di database sehingga teknisi bisa melihat *log error* yang pernah terjadi pada plant.
- e. Namun jika aksi penekanan tombol *reset* tidak berhasil dan plant tetap terjadi *error* maka perintah selanjutnya adalah mematikan seluruh proses plant (SIS).



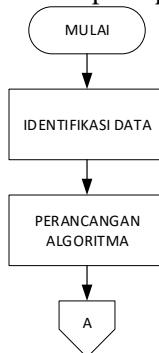
Gambar 3 2 Diagram Alir Pengerjaan Perangkat Keras



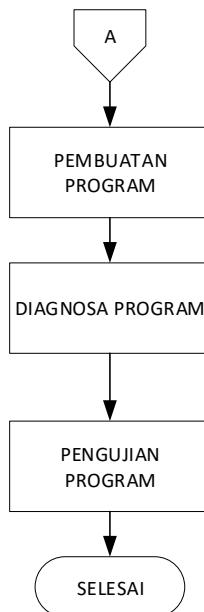
Gambar 3 2 (Lanjutan) Diagram Alir Penggerjaan Perangkat Keras

3.1.2 Diagram Alir Penggerjaan Perangkat Lunak

Dalam penggerjaan perangkat lunak yaitu untuk penyimpanan data *error* yang terjadi dan menampilkan notifikasi *error* pada pusat kontrol panel (HMI) dengan menggunakan mikrokontroler AVR ATMega16. Serta perancangan *logic solver* sehingga dapat menemukan elemen yang mengalami *error* untuk menormalkan kembali kondisi *error* pada plant.



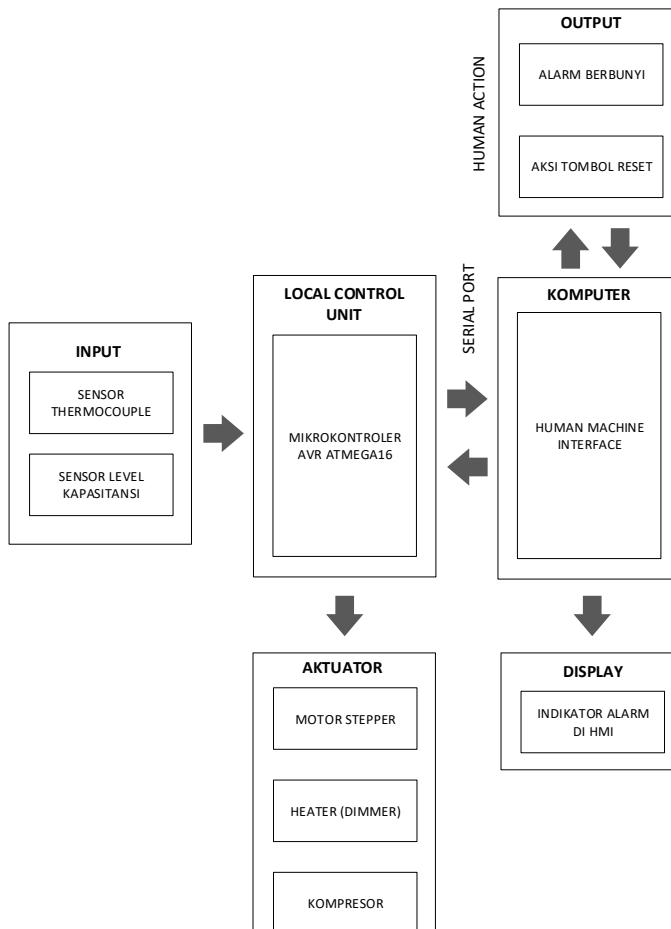
Gambar 3 3 Diagram Alir Penggerjaan Perangkat Lunak



Gambar 3 3 (Lanjutan) Diagram Alir Pengerjaan Perangkat Lunak

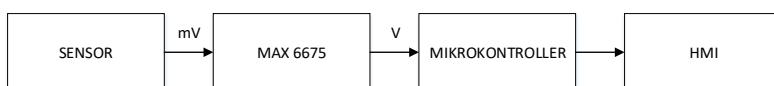
3.2 Konsep Sistem Alarm

Pada Gambar 3.1 akan dijelaskan mengenai konsep alur sistem alarm pada simulator alat penukar panas. Input berasal dari tiga unit proses yang telah terpasang sistem alarm, yaitu unit penyimpanan, unit pemanas, dan unit pendingin. Masing-masing memonitoring datar dari sensor tiap unit. Unit penyimpanan menggunakan sensor level kapasitansi, unit pemanas dan unit pendingin menggunakan sensor thermocouple.

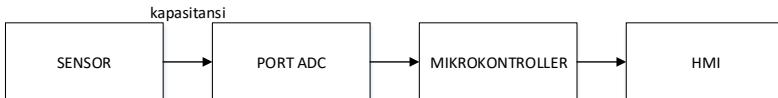


Gambar 3 4 Konsep Sistem Alarm

Berikut diagram blok dari sensor yang ada pada unit proses:



Gambar 3 5 Diagram Blok Sensor Thermocouple



Gambar 3 6 Diagram Blok Sensor Kapasitansi

Dengan *range* operasional dari masing-masing sensor sebagai berikut:

Tabel 3 1 Range Operasional Sensor

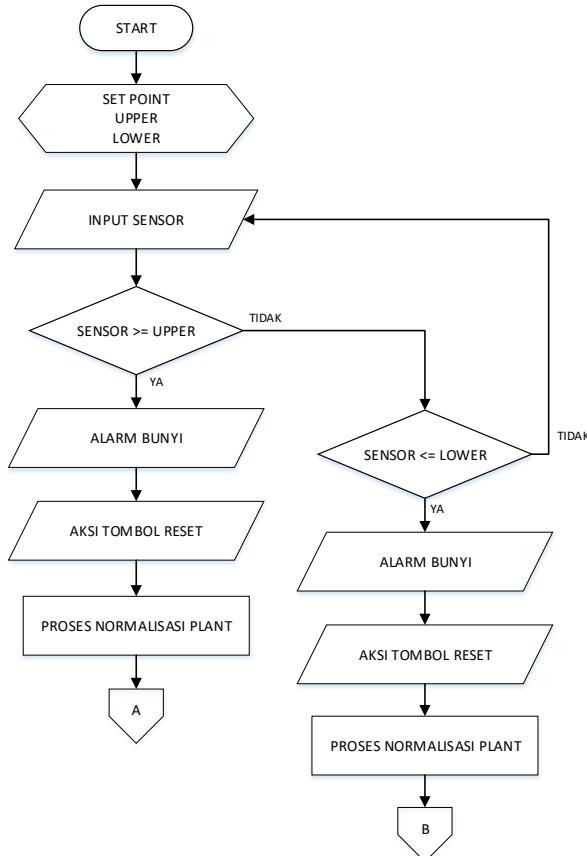
No	Sensor	Unit	SP	High	Low
1	Kapasitansi <i>level storage tank (LT)</i>	Penyimpanan	30 cm	33 cm	13 cm
2	<i>Thermocouple heater (TT)</i>	Pemanas	70 °C	75 °C	23 °C
3	<i>Thermocouple refrigerant (TT)</i>	Pendingin	10 °C	30°C	7 °C

3.3 Flowchart Sistem Alarm

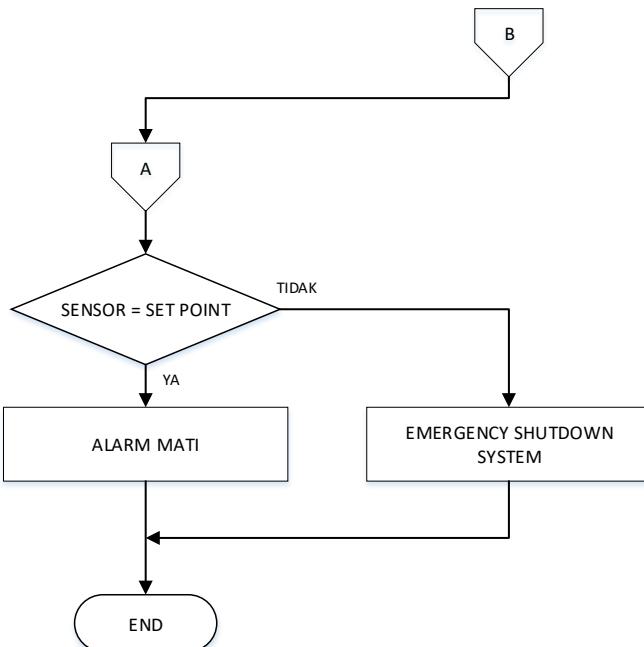
Alur sistem alarm yang ada pada simulator alat penukar panas pada ketiga *Local Control Unit* pemanas, pendingin, dan *storage tank* adalah sebagai berikut:

- Menetapkan terlebih dahulu nilai *set point*, *upper limit*, dan *lower limit* sesuai dengan kriteria pada unit alat simulasi.
- Mendeteksi input dari sensor masing-masing unit yang ada pada alat simulasi.
- Mengkondisikan nilai dari sensor yang telah dideteksi, melewati *upper limit* atau *lower limit*.
- Jika nilai dari sensor memenuhi pengkondisian, yaitu melewati *upper limit* atau *lower limit*, maka terjadi error dan alarm berbunyi.
- Terjadi aksi penekanan tombol reset oleh operator melalui HMI yang berfungsi untuk melakukan pernolmalisasian proses pada plant.

- f. Proses pernormalisasi plant secara otomatis sesuai dengan letak *error* unit alat simulasi.
- g. Jika pembacaan sensor kembali sesuai dengan nilai *set point*, maka alarm mati dan plant kembali berjalan sesuai dengan *normal operation*.
- h. Jika pembacaan sensor tidak kembali sesuai dengan nilai *set point* dengan waktu yang lama, maka terjadi *emergency shutdown system* demi keamanan dan keselamatan kerja.



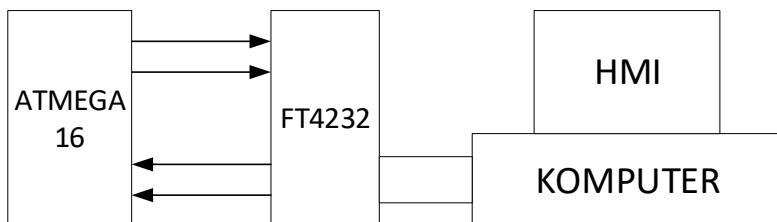
Gambar 3 7 Diagram Alir Sistem Alarm



Gambar 3 7 (Lanjutan) Diagram Alir Sistem Alarm

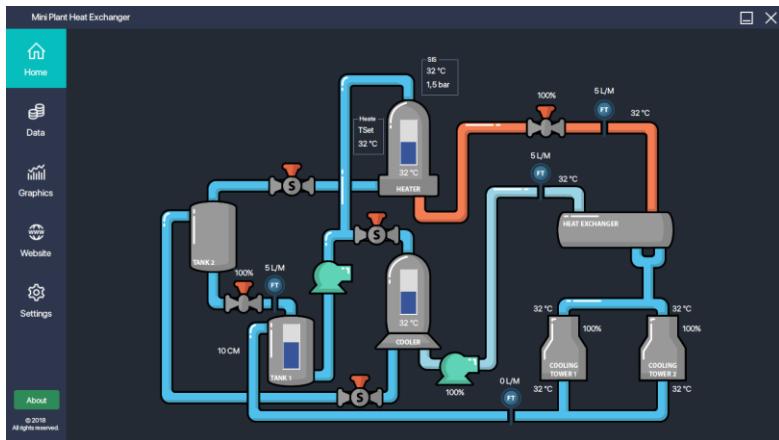
3.4 Deasin Mikro ke Komputer

Desain pembacaan dan pengiriman data dari mikrokotroller ke komputer menggunakan modul FT4232 yang terhubung melalui USB dan dikirim ke komputer berupa *serial port* dengan *address* yang tertera pada Tabel 3.2.



Gambar 3 8 Alur Data Mikro ke Komputer

Dengan tampilan HMI sebagai berikut:



Gambar 3 9 Tampilan HMI

Perintah yang dikirim dari komputer ke mikrokontroller menggunakan *address* yang tertera pada Tabel 3.3 di bawah ini.

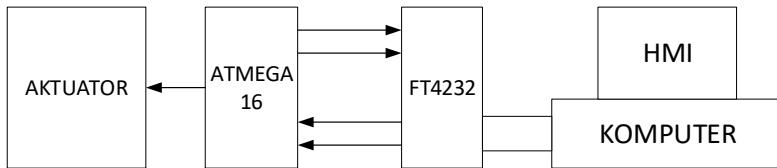
Tabel 3 2 Read Address Data

LCU	No	Plant	Read	SOH	Identifier	STX	Text			ETX
LCU 1 Tangki	1	Local Name	r	#	1	a	0	0	0	%
	2	Level	r	#	1	b	0	0	0	%
	3	Total Water	r	#	1	c	0	0	0	%
	4	Motor Stepper	r	#	1	d	0	0	0	%
	5	Pump	r	#	1	e	0	0	0	%
	6	Override	r	#	1	f	0	0	0	%
	7	On/Off Loop	r	#	1	g	0	0	0	%
	8	Alarm	r	#	1	h	0	0	0	%
LCU	No	Plant	Read	SOH	Identifier	STX	Text			ETX
	9	Local Name	r	#	2	a	0	0	0	%

LCU 2 Heater	10	Dry Level	r	#	2	b	0	0	0	%
LCU	No	Plant	Read	SOH	Identifier	STX	Text		ETX	
LCU 3 <i>Refrigerant</i>	18	Local Name	r	#	3	a	0	0	0	%
	19	Dry Level	r	#	3	b	0	0	0	%
	20	Temperature	r	#	3	c	0	0	0	%
	21	Refrigerant	r	#	3	d	0	0	0	%
	22	Temperatur Set	r	#	3	e	0	0	0	%
	23	Solenoid Valve	r	#	3	f	0	0	0	%
	24	Solenoid Valve	r	#	3	g	0	0	0	%
	25	Override	r	#	3	h	0	0	0	%
	26	On/Off Loop	r	#	3	i	0	0	0	%
	27	Alarm	r	#	3	j	0	0	0	%

3.5 Desain ke Aktuator

Desain pengiriman data dari HMI (komputer) ke aktuator melalui *serial port* yang terhubung ke modul FT4232 menggunakan kabel USB yang nantinya data perintah dari komputer akan dikirimkan melalui mikrokontroller menuju ke aktuator dengan *address* yang tertera pada Tabel 3.3.



Gambar 3 10 Alur Data ke Aktuator

Perintah yang dikirim dari komputer ke mikrokontroler menggunakan *address* yang tertera pada Tabel 3.3 di bawah ini.

Tabel 3 3 Write Address Data

LCU	No	Plant	Write	SOH	Identifier	STX	Text			ETX
LCU 1 Tangki	1	Local Name	W							
	2	Level	W							
	3	Total Water	W							
	4	Motor Stepper	W	#	1	m	0	0	0	%
	5	Pump	W	#	1	n	0	0	0	%
	6	Override	W	#	1	o	0	0	0	%
	7	On/Off Loop	W	#	1	p	0	0	0	%
	8	Alarm	W	#	1	q	0	0	0	%
LCU	No	Plant	Write	SOH	Identifier	STX	Text			ETX
LCU 2 Heater	9	Local Name	W							
	10	Dry Level	W							
	11	Temperature	W							
	12	Heater	W	#	2	l	0	0	0	%
	13	Temperature Set	W	#	2	m	0	0	0	%
	14	Solenoid Valve	W	#	2	n	0	0	0	%
	15	Override	W	#	2	o	0	0	0	%

	16	On/Off Loop	W	#	2	p	0	0	0	%
	17	Alarm	W	#	2	q	0	0	0	%
LCU	No	Plant	Write	SOH	Identifier	STX	Text			ETX
LCU 3 <i>Refrigerant</i>	18	Local Name	W							
	19	Dry Level	W							
	20	Temperature	W							
	21	Refrigerant	W	#	3	l	0	0	0	%
	22	Temperatur Set	W	#	3	m	0	0	0	%
	23	Solenoid Valve	W	#	3	n	0	0	0	%
	24	Solenoid Valve	W	#	3	o	0	0	0	%
	25	Override	W	#	3	p	0	0	0	%
	26	On/Off Loop	W	#	3	q	0	0	0	%
	27	Alarm	W	#	3	r	0	0	0	%

Halaman ini sengaja dikosongkan.

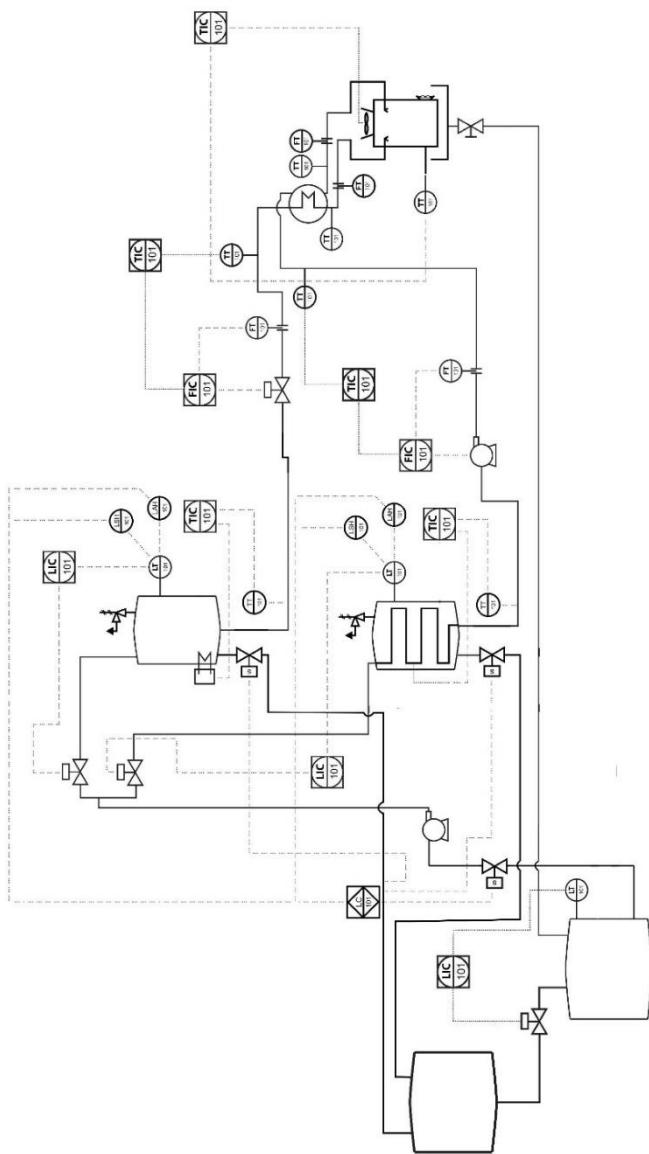
BAB IV

HASIL DAN PEMBAHASAN

4.1. Deskripsi Sistem Alarm

Simulasi alat penukar panas merupakan sebuah proses untuk mengubah temperatur dan fasa suatu jenis fluida. Proses tersebut terjadi dengan memanfaatkan proses perpindahan kalor dari fluida bersuhu tinggi menuju fluida bersuhu rendah. Sistem proses yang terjadi di simulator alat penukar panas berjalan secara terus menerus (*continuous*). Semua unit yang ada pada simulator alat penukar panas dilengkapi dengan sistem pengendalian *Basic Process Control System* (BPCS) untuk menjaga stabilitas proses yang terjadi pada simulator. Pengendalian yang terjadi pada simulator alat penukar panas dikontrol melalui *Local Control Unit* (LCU). Local Control Unit mengendalikan beberapa unit yang ada di simulator alat penukar panas. *Local Control Unit* belum terhubung ke *Control Center* yang ada pada *Human Machine Interface* (HMI). Untuk menghubungkan antara *Local Control Unit* dengan *Human Machine Interface* digunakan sistem komunikasi data, sehingga ada bagian yang mengatur mengenai komunikasi data dari *Local Control Unit* ke *Human Machine Interface* dan mengatur *Human Machine Interface* atau *Control Center*.

Demi keamanan pada simulator alat penukar panas maka pada Tugas Akhir ini dilakukan perancangan sistem alarm pada simulator alat penukar panas yang berguna jika terjadi malfungsi *error* pada sistem seperti *overheat* atau *overflow*. Sistem alarm berfungsi sebagai pemberitahuan atau notifikasi ketika terjadi *error* pada simulator dengan menetapkan *range operasional* dari tiga unit proses pengendalian (penyimpanan air, pendingin, dan pemanas). Notifikasi *error* berupa bunyi sirine yang dilengkapi dengan tombol reset pada HMI beserta indikator alarm yang muncul pada tampilan HMI supaya bisa menstabilkan kembali proses pada simulator.



Gambar 4 1 P&ID Simulator Alat Penukar Panas

4.2. Pengujian Alarm Dengan Data *Dummy*

Pada pengujian sistem alarm dilakukan simulasi menggunakan *software* Proteus dengan Arduino sebagai *local control unit* yang berfungsi untuk mengirimkan data acak antara 1-100. Untuk mengirim data dari proteus ke HMI digunakan *Virtual Serial Port*. pada HMI, nilai *upper limit* diatur sebesar 70 dan *lower limit* sebesar 30. Berikut hasil dari pengujian sistem alarm menggunakan data *dummy*:

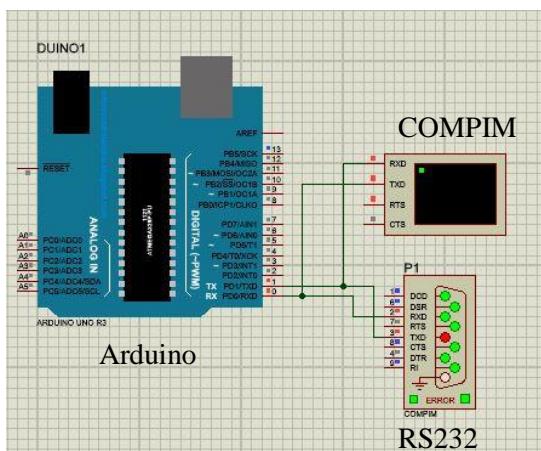
Tabel 4.1 Hasil Uji Alarm dengan Data *Dummy*

<i>Upper limit</i>	: 70		
<i>Lower limit</i>	: 30		
No	Waktu (s)	Pembacaan Data	Alarm (On/Off)
1	1	52	Off
2	2	74	On
3	3	2	On
4	4	83	On
5	4	23	On
6	5	86	On
7	6	13	On
8	7	95	On
9	8	2	On
10	8	46	Off
11	8	24	On
12	9	95	On
13	10	12	On
14	11	74	On
15	11	23	On
16	12	67	Off
17	12	1	On
18	13	46	Off
19	13	37	Off
20	14	18	On
21	14	31	Off
22	14	35	Off

Tabel 4 1 (Lanjutan)

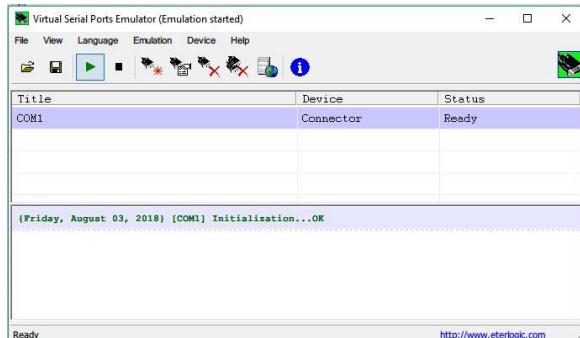
<i>Upper limit</i>	: 70		
<i>Lower limit</i>	: 30		
No	Waktu (s)	Pembacaan Data	Alarm (On/Off)
23	15	21	On
24	16	35	Off
25	16	57	Off
26	17	14	On

Pada Tabel 4.1 menerangkan bahwa ketika nilai pembacaan data mencapai lebih atau kurang dari nilai *upper/lower limit* maka alarm akan berbunyi. Berikut *wiring* simulasi yang dilakukan pada Proteus:

**Gambar 4 2 Wiring Simulasi Data Dummy**

Pada Gambar 4.2 adalah *wiring* simulasi sistem alarm dengan Proteus. Arduino digunakan sebagai *local control unit* yang berfungsi untuk mengeluarkan data acak antara 1-100. Data dari Arduino dikirim melalui *serial port* RS232 yang berfungsi untuk mengirimkan data ke HMI menggunakan *software Virtual Serial Port Emulator*. *Address port* yang diterapkan pada simulasi

menggunakan *address* COM1 seperti yang tertera pada Gambar 4.3.



Gambar 4 3 Virtual Serial Port Emulator

4.3. Hasil Uji Skenario Sistem Alarm

Untuk mengetahui kinerja sistem alarm dilakukan pengujian melalui pada masing-masing unit pengendalian dengan skenario perubahan nilai *set point* untuk mengetahui apakah sistem alarm bisa berfungsi dengan normal atau tidak.

Tabel 4 2 Range Operasional Sensor

No.	Unit	Sensor	Range Operasional	
			Upper	Lower
1	Penyimpanan Air	Kapasitansi	33 cm	13 cm
2	Pemanas	Thermocouple	75 °C	23 °C
3	Pendingin	Thermocouple	30 °C	7 °C

Untuk mengetahui kinerja alarm maka sistem harus mempunyai respon dinamik yang menuju ke *upper limit/lower limit* pada tiap unit proses.

4.3.1 Pengujian Pada Unit Penyimpanan Air

Berikut data hasil pengujian sistem alarm pada unit penyimpanan air simulator alat penukar panas:

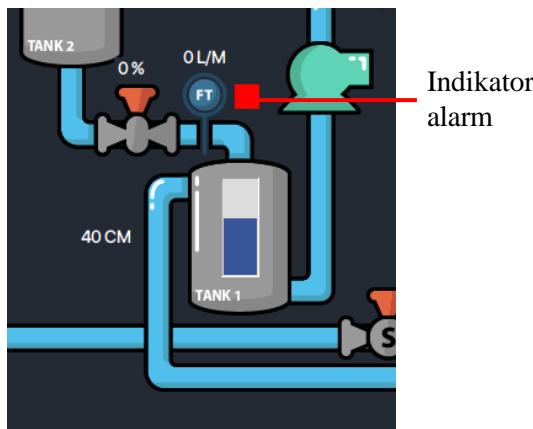
Tabel 4.3 Data Hasil Uji Alarm Pada Unit Penyimpanan Air

<i>Set Point</i>	: 30 cm				
<i>Upper limit</i>	: 40 cm				
<i>Lower limit</i>	: 20 cm				
No	Waktu (s)	Pembacaan Sensor Level (cm)	Bukaan Valve (%)	Pompa (On/Off)	Alarm (On/Off)
1	4	20	100	Off	On
2	8	21	100	Off	Off
3	12	22	100	Off	Off
4	17	23	100	Off	Off
5	21	24	100	Off	Off
6	25	25	100	Off	Off
7	30	26	100	Off	Off
8	34	27	100	Off	Off
9	38	28	100	Off	Off
10	42	29	100	Off	Off
11	46	30	100	Off	Off
12	51	31	100	Off	Off
13	55	32	100	Off	Off
14	59	33	100	Off	Off
15	64	34	100	Off	Off
16	68	35	100	Off	Off
17	72	36	100	Off	Off
18	77	37	100	Off	Off
19	81	38	100	Off	Off
20	85	39	100	Off	Off
21	90	40	100	Off	On
22	92	41	0	On	On
23	94	40	0	On	On
24	96	39	0	On	Off
25	98	38	0	On	Off
26	101	37	0	On	Off
27	103	36	0	On	Off

Tabel 4 3 (Lanjutan)

<i>Set Point</i>	: 30 cm				
<i>Upper limit</i>	: 40 cm				
<i>Lower limit</i>	: 20 cm				
No	Waktu (s)	Pembacaan Sensor Level (cm)	Bukaan Valve (%)	Pompa (On/Off)	Alarm (On/Off)
28	105	35	0	On	Off
29	108	34	0	On	Off
30	110	33	0	On	Off
31	113	32	0	On	Off
32	116	31	0	On	Off
33	118	30	0	Off	Off

Pada percobaan unit penyimpanan air aksi tombol reset yang dilakukan oleh operator untuk menormalkan kembali kondisi pada plant ketika terjadi *error* berfungsi untuk membuka *valve*/aliran air dan menyalakan atau mematikan pompa. Proses normalisasi plant akan berhenti jika nilai dari pembacaan sensor mencapai *set point* lagi.

**Gambar 4 4** Indikator Alarm Unit Penyimpanan Air

Dalam hal ini yang dilakukan pada percobaan pada unit penyimpanan air simulator alat penukar panas dilakukan skenario menaikkan level ketinggian air sampai mencapai *upper limit* dan menurunkan level ketinggian air sampai mencapai *lower limit* sehingga alarm akan berbunyi dan adanya aksi tombol reset untuk menormalkan kembali unit proses. Namun pada saat pembacaan sensor tidak mencapai nilai *upper/lower limit* maka alarm tidak berbunyi namun proses menormalkan kembali tetap berjalan sampai mencapa *set point*. Dalam percobaan scenario yang dilakukan yaitu mengurangi nilai *upper limit* menjadi 40 cm dan *lower limit* menjadi 20 cm dengan hasil uji tertera pada Tabel 4.2.

4.3.2 Pengujian Pada Unit Pemanas

Berikut data hasil pengujian sistem alarm pada unit pemanas simulator alat penukar panas:

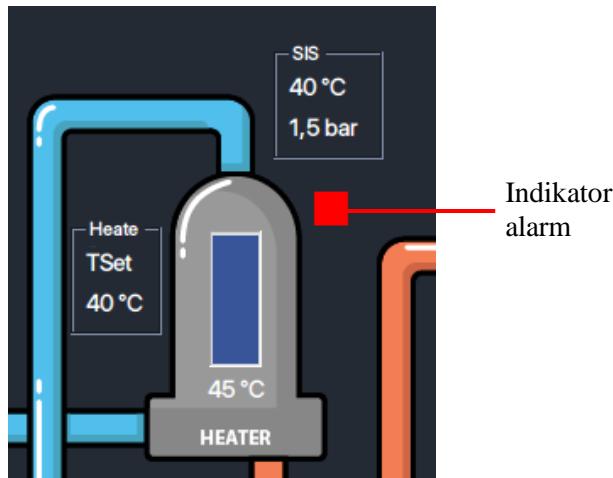
Tabel 4.4 Data Hasil Uji Alarm Pada Unit Pemanas

<i>Set Point</i>	: 40 °C					
<i>Upper limit</i>	: 45 °C					
<i>Lower limit</i>	: 20 °C					
No	Waktu (s)	Pembacaan Sensor Suhu (°C)	Float Switch (On/Off)	Bukaan Valve (%)	Pemanas (On/Off)	Alarm (On/Off)
1	20	25	On	0	On	Off
2	39	26	On	0	On	Off
3	59	27	On	0	On	Off
4	80	28	On	0	On	Off
5	100	29	On	0	On	Off
6	118	30	On	0	On	Off
7	138	31	On	0	On	Off
8	155	32	On	0	On	Off
9	174	33	On	0	On	Off
10	192	34	On	0	On	Off
11	210	35	On	0	On	Off
12	214	36	On	0	On	Off

Tabel 4.4 (Lanjutan)

<i>Set Point</i>	: 40 °C					
<i>Upper limit</i>	: 45 °C					
<i>Lower limit</i>	: 20 °C					
No	Waktu (s)	Pembacaan Sensor Suhu (°C)	Float Switch (On/Off)	Bukaan Valve (%)	Pemanas (On/Off)	Alarm (On/Off)
13	228	37	On	0	On	Off
14	232	38	On	0	On	Off
15	251	39	On	0	On	Off
16	271	40	On	0	On	Off
17	285	41	On	0	On	Off
18	299	42	On	0	On	Off
19	315	43	On	0	On	Off
20	334	44	On	0	On	Off
21	356	45	On	0	On	On
22	388	44	Off	100	Off	Off
23	419	43	Off	100	Off	Off
24	449	42	Off	100	Off	Off
25	480	41	Off	100	Off	Off
26	510	40	Off	0	Off	Off

Pada percobaan unit pemanas simulator alat penukar panas dilakukan aksi tombol reset oleh operator untuk menormalkan kembali kondisi pada plant ketika terjadi *error*. Aksi tersebut berfungsi untuk membuka *valve*/aliran air dari tangki pemanas keluar menuju *heat exchanger* dan mematikan pemanas sehingga membuat suhu air pada tangki pemanas menurun. Proses normalisasi plant akan berhenti jika nilai dari pembacaan sensor mencapai *set point* lagi dan sistem proses akan dikembalikan lagi ke BPCS dari unit pemanas.



Gambar 4 5 Indikator Alarm Unit Pemanas

Dalam hal ini yang dilakukan pada percobaan pada unit pemanas simulator alat penukar panas dilakukan skenario menaikkan suhu air sampai mencapai *upper limit* sehingga alarm akan berbunyi dan adanya aksi tombol reset untuk menormalkan kembali unit proses. Namun pada saat pembacaan sensor tidak mencapai nilai *upper/lower limit* maka alarm tidak berbunyi namun proses menormalkan kembali tetap berjalan sampai mencapa *set point*. Dalam percobaan scenario yang dilakukan yaitu mengurangi nilai *upper limit* menjadi 45 °C dengan hasil yang tertera pada Tabel 4.3.

4.3.3 Pengujian Pada Unit Pendingin

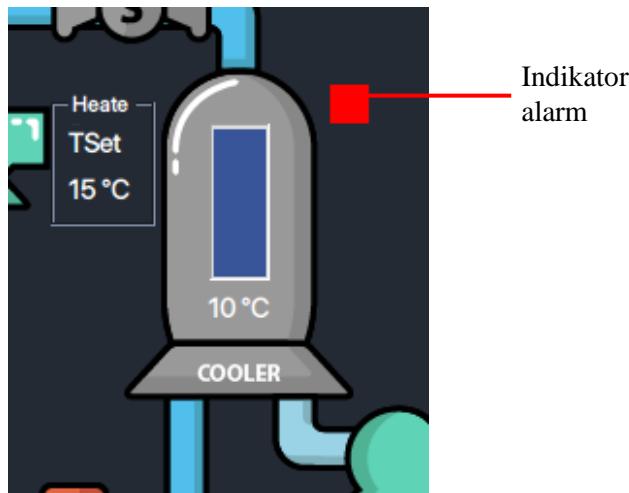
Berikut data hasil pengujian sistem alarm pada unit pendingin simulator alat penukar panas:

Tabel 4 5 Data Hasil Uji Alarm Pada Unit Pendingin

<i>Set Point</i>	: 15 °C					
<i>Upper limit</i>	: 30 °C					
<i>Lower limit</i>	: 10 °C					
No	Waktu (s)	Pembacaan Sensor Suhu (°C)	Float Switch (On/Off)	Bukaan Valve (%)	Kompresor (On/Off)	Alarm (On/Off)
1	35	25	On	0	On	Off
2	70	24	On	0	On	Off
3	105	23	On	0	On	Off
4	140	22	On	0	On	Off
5	175	21	On	0	On	Off
6	207	20	On	0	On	Off
7	240	19	On	0	On	Off
8	270	18	On	0	On	Off
9	300	17	On	0	On	Off
10	330	16	On	0	On	Off
11	360	15	On	0	On	Off
12	387	14	On	0	On	Off
13	414	13	On	0	On	Off
14	438	12	On	0	On	Off
15	466	11	On	0	On	Off
16	495	10	On	100	Off	On
17	550	11	Off	100	Off	Off
18	601	12	Off	100	Off	Off
19	653	13	Off	100	Off	Off
20	702	14	Off	100	Off	Off
21	751	15	Off	100	Off	Off

Pada percobaan unit pendingin simulator alat penukar panas dilakukan aksi tombol reset oleh operator untuk menormalkan kembali kondisi pada plant ketika terjadi *error*. Aksi tersebut berfungsi untuk membuka *valve/aliran air dari tangki pendingin keluar menuju heat exchanger* dan mematikan kompresor sehingga

membuat suhu air pada tangki pemanas meningkat. Proses normalisasi plant akan berhenti jika nilai dari pembacaan sensor mencapai *set point* lagi dan sistem proses akan dikembalikan lagi ke BPCS dari unit pendingin.



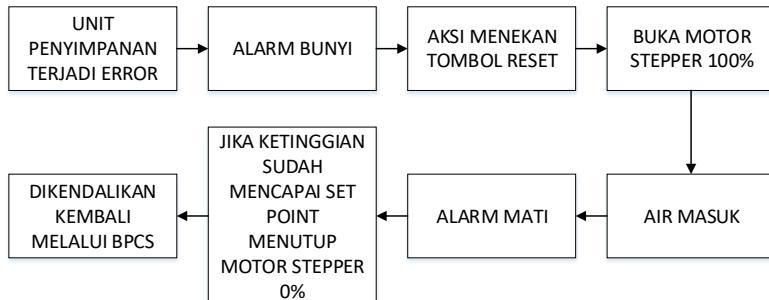
Gambar 4.6 Indikator Alarm Unit Pendingin

Dalam hal ini yang dilakukan pada percobaan ketiga dengan dilakukan skenario suhu air sampai mencapai *lower limit* sehingga alarm akan berbunyi dan adanya aksi tombol reset untuk menormalkan kembali unit proses. Namun pada saat pembacaan sensor tidak mencapai nilai *upper/lower limit* maka alarm tidak berbunyi namun proses menormalkan kembali tetap berjalan sampai mencapa *set point*. Dalam percobaan scenario yang dilakukan yaitu mengurangi nilai *lower limit* menjadi 10 °C dengan hasil yang tertera pada Tabel 4.4.

4.4. Aksi Tombol Reset

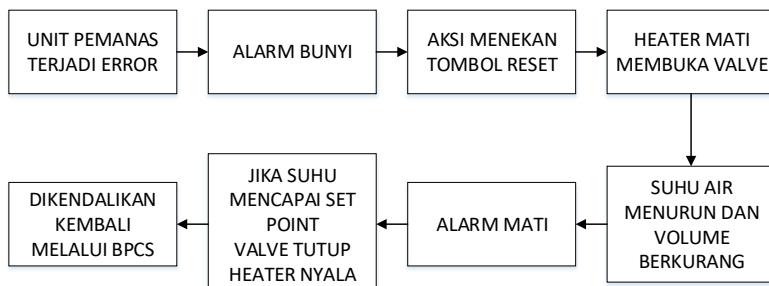
Berdasarkan pengujian sistem alarm pada simulator alat penukar panas yang telah dilakukan, aksi dari tombol reset untuk

menormalkan kembali plant yang terjadi *error* dijelaskan pada gambar di bawah ini.



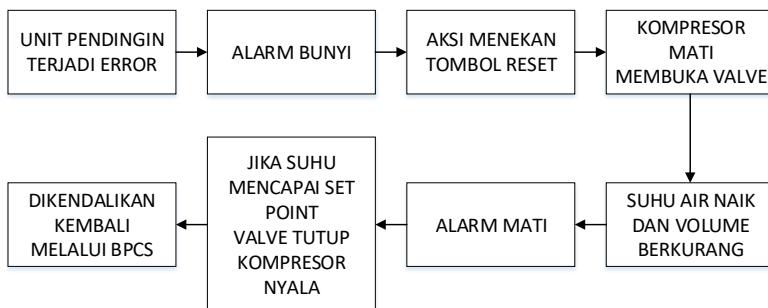
Gambar 4.7 Aksi Tombol Reset Unit Penyimpanan Air

Pada Gambar 4.7 dijelaskan bahwa aksi tombol reset pada sistem alarm unit penyimpanan air dimulai ketika unit tersebut terjadi *error* yaitu nilai dari pembacaan sensor yang mencapai *upper/lower limit* lalu alarm akan berbunyi dan operator akan menekan tombol reset untuk membuka motor stepper menjadi 100%, hal ini membuat air masuk dan ketika nilai pembacaan sensor tidak mencapai *upper/lower limit* maka alarm akan mati. Proses normalisasi plant akan berhenti ketika nilai pembacaan sensor sudah mencapai *set point* sehingga motor stepper tertutup menjadi 0% dan pengendalian dikembalikan lagi ke BPCS.



Gambar 4.8 Aksi Tombol Reset Unit Pemanas

Pada Gambar 4.8 dijelaskan bahwa aksi tombol reset pada sistem alarm unit pemanas dimulai ketika unit tersebut terjadi *error* yaitu nilai dari pembacaan sensor yang mencapai *upper/lower limit* lalu alarm akan berbunyi dan operator akan menekan tombol reset untuk membuka valve dan mematikan heater, hal ini membuat air keluar sehingga menurunkan temperatur fluida pada tanki heater dan ketika nilai pembacaan sensor tidak mencapai *upper/lower limit* maka alarm akan mati. Proses normalisasi plant akan berhenti ketika nilai pembacaan sensor sudah mencapai *set point* sehingga valve tertutup dan heater menyala serta pengendalian dikembalikan lagi ke BPCS.

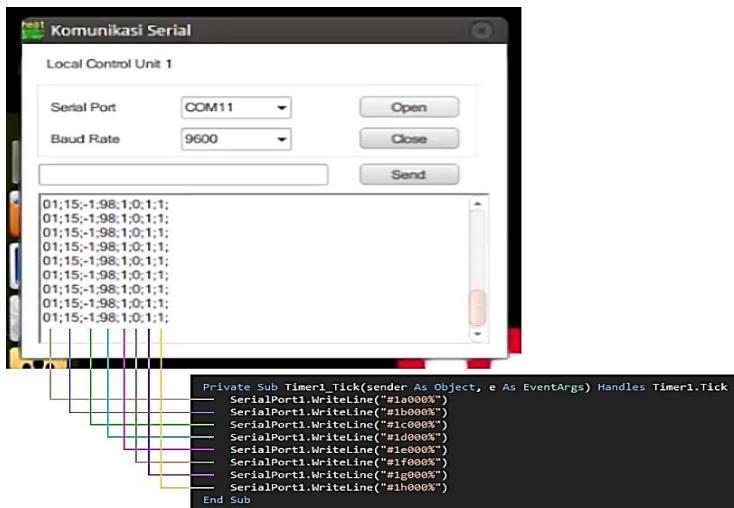


Gambar 4 9 Aksi Tombol Reset Unit Pendingin

Pada Gambar 4.9 dijelaskan bahwa aksi tombol reset pada sistem alarm unit pendingin dimulai ketika unit tersebut terjadi *error* yaitu nilai dari pembacaan sensor yang mencapai *upper/lower limit* lalu alarm akan berbunyi dan operator akan menekan tombol reset untuk membuka valve dan mematikan kompresor, hal ini membuat air keluar sehingga menaikkan temperatur fluida pada tanki refrigerant dan ketika nilai pembacaan sensor tidak mencapai *upper/lower limit* maka alarm akan mati. Proses normalisasi plant akan berhenti ketika nilai pembacaan sensor sudah mencapai *set point* sehingga valve tertutup dan kompresor menyala serta pengendalian dikembalikan lagi ke BPCS.

4.5. Alur Pengiriman dan Penerimaan Data

Jalur komunikasi data yang digunakan menggunakan sistem dengan menggunakan 2 modul F4232 menggunakan konfigurasi UART pada tiap mikrokontroler dengan komputer yang menjalankan HMI. Pada setiap mikrokontroler dihubungkan pin pada modul FT4232, agar dapat terkoneksi dengan program HMI di komputer, dengan begitu data yang masuk masih berupa data baris yang belum di klasifikasi berikut gambar data yang dihasilkan dari mikrokontroler.



Gambar 4 10 Program Komunikasi Data

Pada Gambar 4.10 adalah data serial yang dikirimkan ke komputer, setiap angka mempunyai arti sendiri yaitu data awal (STX a) berupa data *ID Local Control Unit* dan data selanjutnya (STX b-g) adalah data dari nilai *field instrument* yang ada di *plant*. Berikut salah satu contoh kode protokol komunikasi data dari *Local Control Unit ID 1*.

Tabel 4.6 Protokol Komunikasi Data

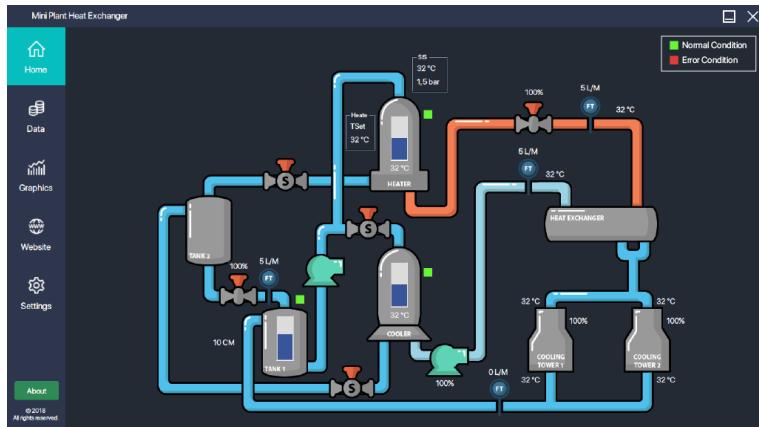
Plant	Read	SOH	Identifier	STX	Text			ETX
Local Name	r	#	1	a	0	0	0	%
Flow	r	#	1	b	0	0	0	%
Temperature	r	#	1	c	0	0	0	%
MOV	r	#	1	d	0	0	0	%
Override	r	#	1	e	0	0	0	%
On/Off Loop	r	#	1	f	0	0	0	%
Alarm	r	#	1	g	0	0	0	%

Pada Tabel 4.6 adalah contoh protokol komunikasi dari salah satu *Local Control Unit*, untuk mengetahui nilai variabel proses dari *field instrument* yang ada pada *plant*. Program dari HMI diharuskan mengirim kode-kode berupa data bit tersebut pada *Local Control Unit* supaya *Local Control Unit* memberi balasan berupa mengirimkan nilai proses variabel dari *field instrument* ke HMI. Contoh, untuk mengetahui nilai dari variabel *temperature* yang ada pada Tabel 4.6, program di HMI harus menuliskan kode (#1c0000%) dengan begitu *Local Control Unit* dapat mengidentifikasi kode data bit yang masuk dan *Local Control Unit* akan memberikan balasan berupa nilai *temperature* yang ada pada *loop plant* tersebut ke HMI. Struktur *frame data* dari *Local Control Unit* dibagi menjadi 5 bagian, yaitu:

1. *Start of Header* (SOH), merupakan data awal (*header*) dari komunikasi data.
2. *Identifier*, berupa angka ID dari masing-masing *Local Control Unit*. Contoh, pada Tabel 4.6 ID yang dipakai adalah ID *Local Control Unit* 1.
3. *Start of Text* (STX), merupakan kode awal untuk komunikasi data berupa angka, huruf, atau simbol (*string*) yang berisi nilai variabel dari *field instrument*.
4. *End of Text* (ETX), merupakan kode akhir (penutup) dari komunikasi data.

4.6. Indikator Alarm Pada HMI

Indikator alarm pada HMI dibuat supaya operator dapat dengan mudah mendeteksi unit proses yang terjadi *error*. Berikut indikator alalrm pada HMI:



Gambar 4 11 Indikator Alarm Pada HMI

Pada Gambar 4.11, indikator dibagi menjadi dua yaitu warna hijau adalah indikator ketika *plant* berjalan normal (*normal condition*) sesuai dengan *range operasional* dan warna merah adalah indikator ketika *plant* terjadi *error* (*error condition*) jika nilai dari pembacaan sensor mencapai *upper limit* atau *lower limit*.

Halaman ini sengaja dikosongkan.

BAB V

PENUTUP

5.1. Kesimpulan

Adapun kesimpulan dari Tugas Akhir ini adalah sebagai berikut:

1. Pada unit proses penyimpan air diperoleh *range operasional* sebesar 33 cm untuk *upper limit* dan 13 cm untuk *lower limit* dengan *set point* 30 cm. Dan pada unit proses pemanas diperoleh *range operasional* sebesar 75 °C untuk *upper limit* dan 23 °C untuk *lower limit*. Sedangkan pada unit proses pendingin diperoleh *range operasional* sebesar 30 °C untuk *upper limit* dan 7 °C untuk *lower limit*.
2. Pada pengujian unit proses penyimpan air ketika nilai sensor kapasitansi mencapai *upper limit* dengan aksi tombol reset membutuhkan waktu 23 detik untuk mencapai *set point*. Sedangkan pada unit pemanas ketika suhu mencapai *upper limit* dan dengan aksi tombol reset membutuhkan waktu 154 detik untuk mencapai *set point* lagi. Dan sedangkan pada unit pendingin ketika suhu mencapai *lower limit* dan dengan aksi tombol reset membutuhkan waktu 256 detik untuk mencapai *set point* lagi.

5.2. Saran

Selama proses pelaksanaan Tugas Akhir ini dari awal sampai akhir, adapun saran yang diperlukan untuk keberlangsungan Tugas Akhir ini apabila ada orang lain yang ingin mengembangkan Tugas Akhir ini di kemudian hari:

1. Perlu ditetapkannya *timeline* dari awal pelaksanaan sampai akhir dan kesepakatan dari masing-masing anggota kelompok mengenai *timeline* tersebut sehingga pelaksanaan simulasi alat penukar panas bisa berjalan sesuai dengan *timeline*.
2. Penetapan *range operasional* sensor bisa dilakukan sebelum semua alat sudah jadi, karena hal tersebut bisa dilakukan dengan menghitung spesifikasi dan desain dari masing-

masing alat yang berhubungan dengan sistem pengendalian sehingga penggeraan sistem alarm tidak terhambat dengan belum ditetapkannya tipe sensor apa saja yang dibutuhkan beserta *range operasionalnya*.

DAFTAR PUSTAKA

- [1] Anshori, N. Dan Mustajib, M. I., Sistem Perawatan Terpadu (Integrated Maintenance System), Edisi Pertama, Yogyakarta: Graha Ilmu, 2013.
- [2] Atindriyo K. Pamososuryo, Hafid S.N. Muzwar, *Studi Implementasi Alarm Management System pada Kolom Distilasi dengan Menggunakan Distributed Control System*, vol. Vol 6 (2), pp. 95-103, 2014.
- [3] Handi Winata, Eko Syamsuddin, Yohanes Calvinus, *Sistem Pengawasan dan Pengontrolan Cooling Tower*, vol. Vol 15, pp. 73-84, 2013.
- [4] Rangga Noviansyah Nuur Aziiz, Mochammad Hannats Hanafi Ichsan, Issa Arwani, *Implementasi Pengiriman Data Sensor LM35 untuk Perbandingan Waktu Insert pada Basis Data MySQL dan MongoDB*, vol. Vol 1, pp. 466-475, 2017.
- [5] U. A. Yogyakarta, Jurnal Ilmiah Data Manajemen dan Teknologi Informasi, Yogyakarta: Universitas Amikom Yogyakarta, 2017.
- [6] C. E. Ebeling, An Introduction to Reliability and Maintainability Engineering, 1st ed, Singapore: McGraw-Hill Companies, 1997.
- [7] A. A. Frederickson, The Layer of Protection Analysis (LOPA) Method, Safety Users Group Network, 2002.
- [8] Hatch, D. dan Stauffer, T., “Operators on Alert.,” 2009. [Online]. Available: <http://www.isa.org>. [Diakses 2018].
- [9] K. Indriawati, Penggunaan Teknik Rekonsiliasi Data dan Sistem Inferensi Fuzzy untuk Perbaikan Performansi Statistical Process Control, Institut Teknologi Sepuluh Nopember, 2007.
- [10] R. W. Johnson, ““Beyond-Compliance Uses HAZOP/ LOPA Studies”,” *Journal of Loss Prevention in the Process Industries*, vol. 23, pp. 727-733, 2010.

- [11] A. G. King, ““SIL Determination: Dealing with Unexpected”,” *AIDIC*, 2013.
- [12] C. A. Lassen, Layer of protection analysis (LOPA) for determination of safety integrity level (SIL), Norwegian: The Norwegian University of Science and Technology, 2008.
- [13] R. D. Leitch, Reliability Analysis for Engineers, New York: Oxford University Press, Inc., 1995.
- [14] Mitsubishi, Steam Turbine 105-JT 5MXL Model Datasheet, Japan: Mitsubishi Heavy Industry Manufacturer, 1994.
- [15] Montgomery, Introduction to Statistical Quality Control (Sixth Edition), USA: John Wiley & Sons, Inc, 2009.

LAMPIRAN

```
// Kode alarm pada HMI
Imports MySql.Data.MySqlClient

Public Class MainForm
    'MySql Connection
    Dim str As String = "server=samp.udah.pw; uid=yusuf;
    pwd=yusuf123456; database=kels; SslMode=None;
    Convert Zero Datetime=True"
    Dim con As New MySqlConnection(str)
    Public da As
        MySql.Data.MySqlClient.MySqlDataAdapter
        Public comBuilderDB As New
            MySql.Data.MySqlClient.MySqlCommandBuilder

    'Data View Declaration
    Dim dt1 As New DataTable
    Dim dt2 As New DataTable
    Dim dt3 As New DataTable
    Dim dt4 As New DataTable
    Dim dt5 As New DataTable
    Dim dt6 As New DataTable
    Dim dt7 As New DataTable
    Dim dt8 As New DataTable

    'Defines variables
    Dim OnSymbol As String =
        "D:\Documents\ITS\SEMESTER 6\TUGAS
        AKHIR\SOFTWARE\HMI\HMI\Resources\Icon\New
        Moon_32px.png"
    Dim OffSymbol As String =
        "D:\Documents\ITS\SEMESTER 6\TUGAS
        AKHIR\SOFTWARE\HMI\HMI\Resources\Icon\Full
        Moon_32px.png"
```

```
Dim ConSymbol As String =
"D:\Documents\ITS\SEMESTER 6\TUGAS
AKHIR\SOFTWARE\HMI\HMI\Resources\Icon\Connec
ted_32px.png"
Dim DisconSymbol As String =
"D:\Documents\ITS\SEMESTER 6\TUGAS
AKHIR\SOFTWARE\HMI\HMI\Resources\Icon\Discon
nected_32px.png"

'Sound
Dim MusicFile As String =
"C:\Users\wahyu\Downloads\Music\firebell.wav"

'COM Declaration
Dim Com1 As String = "COM13"
Dim Com2 As String = "COM10"
Dim Com3 As String = "COM15"
Dim Com4 As String = "COM11"
Dim Com5 As String = "COM14"
Dim Com6 As String = "COM12"
Dim Com7 As String = "COM17"
Dim Com8 As String = "COM16"

'Baudrate Setting
Dim BaudRate As String = "9600"

'Alarm
Dim Alarm As String = Nothing

'Read data variables
Dim data_masuk1 As String  'Data Fara
Dim data_masuk2 As String  'Data Jigong
Dim data_masuk3 As String  'Data Gogol
Dim data_masuk4 As String  'Data Naja
Dim data_masuk5 As String  'Data Enver
```

```
Dim data_masuk6 As String  'Data Sri
Dim data_masuk7 As String  'Data Vebby
Dim data_masuk8 As String  'Data Pakde

'Level Control (ID: 1)
Dim PumpLevel As String
Dim PumpFlow As String
Dim PumpStepper As String
Dim PumpTank As String
Dim PumpOverride As String
Dim PumpPlant As String
Dim PumpAlarm As String
Dim PumpLevelUpper As String = "25"
Dim PumpLevelLower As String = "15"
Dim PumpSetPoint As String = "20"

'Heater Tank (ID: 2)
Dim HeaterDryLevel As String
Dim HeaterTemperature As String
Dim HeaterValue As String
Dim HeaterTSet As String
Dim HeaterSolenoid As String
Dim HeaterOverride As String
Dim HeaterPlant As String
Dim HeaterAlarm As String
Dim HeaterTempUpper As String = "55"
Dim HeaterTempLower As String = "23"
Dim HeaterSetPoint As String = "50"

'Cooler Control (ID: 3)
Dim CoolerDryLevel As String
Dim CoolerTemperature As String
Dim CoolerTSet As String
Dim CoolerSelVal1 As String
Dim CoolerSelVal2 As String
```

```
Dim CoolerRefrigerant As String  
Dim CoolerOverride As String  
Dim CoolerPlant As String  
Dim CoolerAlarm As String  
Dim CoolerTempUpper As String = "30"  
Dim CoolerTempLower As String = "10"  
Dim CoolerSetPoint As String = "15"
```

```
'MOV Control (ID: 4)  
Dim MOVTemperature As String  
Dim MOVFlow As String  
Dim MOVValue As String  
Dim MOVOVERRIDE As String  
Dim MOVPlant As String  
Dim MOVALarm As String  
Dim MOVTemperatureUpper As Integer = 40  
Dim MOVTemperatureLower As Integer = 30
```

```
'VSD Control (ID: 5)  
Dim VSDFlow As String  
Dim VSDTemperature As String  
Dim VSDPump As String  
Dim VSDOverride As String  
Dim VSDPlant As String  
Dim VSDAlarm As String
```

```
'Cooling Tower 1 (ID: 6)  
Dim TowerTempIn1 As String  
Dim TowerTempOut1 As String  
Dim TowerFlow1 As String  
Dim TowerFan1 As String  
Dim TowerPlant1 As String  
Dim TowerOverride1 As String  
Dim TowerAlarm1 As String
```

```
'Cooling Tower 2 (ID: 7)
Dim TowerTempIn2 As String
Dim TowerTempOut2 As String
Dim TowerFlow2 As String
Dim TowerFan2 As String
Dim TowerPlant2 As String
Dim TowerOverride2 As String
Dim TowerAlarm2 As String

'SIS Control (ID:8)
Dim SISTemperature As String
Dim SISPressure As String
Dim SISContactor As String
Dim SISOverride As String
Dim SISBacklight As String
Dim SISSolenoidValve As String
Dim SISBuzzer As String

Private Sub MainForm_Load(sender As Object, e As
EventArgs) Handles MyBase.Load
    ButtonHome.color = Color.FromArgb(6, 190, 190)
    ButtonConnect.IconVisible = True
    ButtonDatabase.IconVisible = True
    Button1Plant.IconVisible = True
    Button1Override.IconVisible = True
    ButtonPump.IconVisible = True
    Button2Override.IconVisible = True
    Button2Plant.IconVisible = True
    Button2Valve.IconVisible = True
    Button3Override.IconVisible = True
    Button3Plant.IconVisible = True
    Button3ValveBot.IconVisible = True
    Button3ValveTop.IconVisible = True
    Button4Override.IconVisible = True
    Button4Plant.IconVisible = True
```

```
    Button5Override.IconVisible = True
    Button5Plant.IconVisible = True
    Button6Override.IconVisible = True
    Button7Override.IconVisible = True
    Button8Override.IconVisible = True
    Button8Contactor.IconVisible = True
End Sub

Private Sub ButtonClose_Click(sender As Object, e As EventArgs) Handles ButtonClose.Click
    Dim response As MsgBoxResult
    response = MsgBox("Are you sure want to exit?", _
        MsgBoxStyle.Question + MsgBoxStyle.YesNo, _
        "Confirmation")
    If response = MsgBoxResult.Yes Then
        Me.Dispose()
        SplashScreen.Dispose()
        About.Dispose()
    ElseIf response = MsgBoxResult.No Then
        Exit Sub
    End If
End Sub

Private Sub ButtonMinimize_Click(sender As Object, e As EventArgs) Handles ButtonMinimize.Click
    Me.WindowState =
        System.Windows.Forms.FormWindowState.Minimized
End Sub

Private Sub ButtonHome_Click(sender As Object, e As EventArgs) Handles ButtonHome.Click
    TabControl.SelectedTab = PageHome
    ButtonHome.color = Color.FromArgb(6, 190, 190)
    'Active
    ButtonData.color = Color.FromArgb(45, 54, 76)
```

```
    ButtonGraphics.color = Color.FromArgb(45, 54, 76)
    ButtonWeb.color = Color.FromArgb(45, 54, 76)
    ButtonSetting.color = Color.FromArgb(45, 54, 76)
End Sub
```

```
Private Sub ButtonData_Click(sender As Object, e As
EventArgs) Handles ButtonData.Click
    TabControl.SelectedTab = PageData
    ButtonHome.color = Color.FromArgb(45, 54, 76)
    ButtonData.color = Color.FromArgb(6, 190, 190)
'Active
    ButtonGraphics.color = Color.FromArgb(45, 54, 76)
    ButtonWeb.color = Color.FromArgb(45, 54, 76)
    ButtonSetting.color = Color.FromArgb(45, 54, 76)
End Sub
```

```
Private Sub ButtonGraphics_Click(sender As Object, e
As EventArgs) Handles ButtonGraphics.Click
    TabControl.SelectedTab = PageGraphics
    ButtonHome.color = Color.FromArgb(45, 54, 76)
    ButtonData.color = Color.FromArgb(45, 54, 76)
    ButtonGraphics.color = Color.FromArgb(6, 190,
190)'Active
    ButtonWeb.color = Color.FromArgb(45, 54, 76)
    ButtonSetting.color = Color.FromArgb(45, 54, 76)
End Sub
```

```
Private Sub ButtonWeb_Click(sender As Object, e As
EventArgs) Handles ButtonWeb.Click
    TabControl.SelectedTab = PageWeb
    ButtonHome.color = Color.FromArgb(45, 54, 76)
    ButtonData.color = Color.FromArgb(45, 54, 76)
    ButtonGraphics.color = Color.FromArgb(45, 54, 76)
    ButtonWeb.color = Color.FromArgb(6, 190, 190)
'Active
```

```
    ButtonSetting.color = Color.FromArgb(45, 54, 76)
End Sub

Private Sub ButtonSetting_Click(sender As Object, e
As EventArgs) Handles ButtonSetting.Click
    TabControl.SelectedTab = PageSettings
    ButtonHome.color = Color.FromArgb(45, 54, 76)
    ButtonData.color = Color.FromArgb(45, 54, 76)
    ButtonGraphics.color = Color.FromArgb(45, 54, 76)
    ButtonWeb.color = Color.FromArgb(45, 54, 76)
    ButtonSetting.color = Color.FromArgb(6, 190, 190)
'Active
End Sub

Private Sub ButtonDatabase_Click(sender As Object, e
As EventArgs) Handles ButtonDatabase.Click
    If ButtonDatabase.Text = "ON" Then
        TimerData.Enabled = True
        ButtonDatabase.Text = "OFF"
        ButtonDatabase.selected = True
        ButtonDatabase.TextAlign =
ContentAlignment.MiddleRight
        ButtonDatabase.BackColor =
Color.FromArgb(56, 85, 153)
        ButtonDatabase.Iconimage =
Image.FromFile(OffSymbol)
        LabelStatsDatValue.Text = "Online"
        LabelStatsDatValue.ForeColor =
Color.FromArgb(6, 190, 190)
    ElseIf ButtonDatabase.Text = "OFF" Then
        TimerData.Enabled = False
        ButtonDatabase.Text = "ON"
        ButtonDatabase.selected = False
        ButtonDatabase.TextAlign =
ContentAlignment.MiddleRight
```

```

        ButtonDatabase.BackColor =
Color.FromArgb(45, 54, 76)
        ButtonDatabase.Iconimage =
Image.FromFile(OnSymbol)
        LabelStatsDatValue.Text = "Offline"
        LabelStatsDatValue.ForeColor =
Color.FromArgb(234, 86, 97)
    End If
End Sub

Private Sub TimerData_Tick(sender As Object, e As
EventArgs) Handles TimerData.Tick
    Dim cmd As MySqlCommand
    con.Open()
    Try
        cmd = con.CreateCommand
        cmd.CommandText = "INSERT INTO
            data(`id`, `1_flow`, `1_level`,
`1 stepper`, `2_temperature`, `2_temperature_set`,
`3_temperature`, `3_temperature_set`, `4_flow`,
`4_temperature`, `4_mov`, `5_flow`, `5_temperature`,
`5_vsd`, `6_temperature_in`, `6_temperature_out`,
`6_flow`, `6_fan`, `7_temperature_in`,
`7_temperature_out`, `7_flow`, `7_fan`, `8_temperature`,
`8_pressure`, `timelog`)
            VALUES
            (", ?1_flow, ?1_level, ?1_stepper,
?2_temperature, ?2_temperature_set, ?3_temperature,
?3_temperature_set, ?4_flow, ?4_temperature, ?4_mov,
?5_flow, ?5_temperature, ?5_vsd, ?6_temperature_in,
?6_temperature_out, ?6_flow, ?6_fan, ?7_temperature_in,
?7_temperature_out, ?7_flow, ?7_fan, ?8_temperature,
?8_pressure, "" & Format(Date.Now, "yyyy-MM-dd
HH:mm:ss") & "")"
    End Try
End Sub

```

```
cmd.Parameters.AddWithValue("?1_flow",
PumpFlow)
    cmd.Parameters.AddWithValue("?1_level",
PumpLevel)
        cmd.Parameters.AddWithValue("?1_stepper",
PumpStepper)

cmd.Parameters.AddWithValue("?2_temperature",
HeaterTemperature)

cmd.Parameters.AddWithValue("?2_temperature_set",
HeaterTSet)

cmd.Parameters.AddWithValue("?3_temperature",
CoolerTemperature)

cmd.Parameters.AddWithValue("?3_temperature_set",
CoolerTSet)
    cmd.Parameters.AddWithValue("?4_flow",
MOVFlow)

cmd.Parameters.AddWithValue("?4_temperature",
MOVTemperature)
    cmd.Parameters.AddWithValue("?4_mov",
MOVValue)
        cmd.Parameters.AddWithValue("?5_flow",
VSDFlow)

cmd.Parameters.AddWithValue("?5_temperature",
VSDTemperature)
    cmd.Parameters.AddWithValue("?5_vsdp",
VSDPump)

cmd.Parameters.AddWithValue("?6_temperature_in",
TowerTempIn1)
```

```
cmd.Parameters.AddWithValue("?6_temperature_out",
TowerTempOut1)
    cmd.Parameters.AddWithValue("?6_flow",
TowerFlow1)
    cmd.Parameters.AddWithValue("?6_fan",
TowerFan1)

cmd.Parameters.AddWithValue("?7_temperature_in",
TowerTempIn2)

cmd.Parameters.AddWithValue("?7_temperature_out",
TowerTempOut2)
    cmd.Parameters.AddWithValue("?7_flow",
TowerFlow2)
    cmd.Parameters.AddWithValue("?7_fan",
TowerFan2)

cmd.Parameters.AddWithValue("?8_temperature",
SISTemperature)
    cmd.Parameters.AddWithValue("?8_pressure",
SISPressure)
        cmd.ExecuteNonQuery()
        LoadData1()
        LoadData2()
        LoadData3()
        LoadData4()
        LoadData5()
        LoadData6()
        LoadData7()
        LoadData8()
        con.Close()
Catch ex As Exception
    con.Close()
    MessageBox.Show(ex.ToString)
```

```
    End Try
End Sub

Private Sub ButtonConnect_Click(sender As Object, e
As EventArgs) Handles ButtonConnect.Click
    If ButtonConnect.Text = "Connect" Then
        'Port ID: 1 (Farah)
        SerialPort1.PortName = (Com1)
        SerialPort1.BaudRate = (BaudRate)
        'Port ID: 2 (Arian)
        SerialPort2.PortName = (Com2)
        SerialPort2.BaudRate = (BaudRate)
        'Port ID: 3 (Gogol)
        SerialPort3.PortName = (Com3)
        SerialPort3.BaudRate = (BaudRate)
        'Port ID: 4 (Naja)
        SerialPort4.PortName = (Com4)
        SerialPort4.BaudRate = (BaudRate)
        'Port ID: 5 (Enver)
        SerialPort5.PortName = (Com5)
        SerialPort5.BaudRate = (BaudRate)
        'Port ID: 6 (Sri)
        SerialPort6.PortName = (Com6)
        SerialPort6.BaudRate = (BaudRate)
        'Port ID: 7 (Vebby)
        SerialPort7.PortName = (Com7)
        SerialPort7.BaudRate = (BaudRate)
        'Port ID: 8 (Dicky)
        SerialPort8.PortName = (Com8)
        SerialPort8.BaudRate = (BaudRate)

    Try
        SerialPort1.Open()
        SerialPort2.Open()
        SerialPort3.Open()
```

```
SerialPort4.Open()
SerialPort5.Open()
SerialPort6.Open()
SerialPort7.Open()
SerialPort8.Open()

'TimerAlarm.Enabled = True

TimerSerial1.Enabled = True
TimerSerial2.Enabled = True
TimerSerial3.Enabled = True
TimerSerial4.Enabled = True
TimerSerial5.Enabled = True
TimerSerial6.Enabled = True
TimerSerial7.Enabled = True
TimerSerial8.Enabled = True

'Change stats
ButtonConnect.Text = "Disconnect"
ButtonConnect.selected = True
ButtonConnect.TextAlign =
ContentAlignment.MiddleRight
ButtonConnect.BackColor =
Color.FromArgb(56, 85, 153)
ButtonConnect.Iconimage =
Image.FromFile(DisconSymbol)
LabelStatsConValue.Text = "Connected"
LabelStatsConValue.ForeColor =
Color.FromArgb(6, 190, 190)
Catch ex As Exception
    MsgBox("Failed to connect!", 
    MsgBoxStyle.Critical, "Error")
        MessageBox.Show(ex.ToString)
End Try
```

```
ElseIf ButtonConnect.Text = "Disconnect" Then
    Try
        'TimerAlarm.Enabled = False

        TimerSerial1.Enabled = False
        TimerSerial2.Enabled = False
        TimerSerial3.Enabled = False
        TimerSerial4.Enabled = False
        TimerSerial5.Enabled = False
        TimerSerial6.Enabled = False
        TimerSerial7.Enabled = False
        TimerSerial8.Enabled = False

        SerialPort1.Close()
        SerialPort2.Close()
        SerialPort3.Close()
        SerialPort4.Close()
        SerialPort5.Close()
        SerialPort6.Close()
        SerialPort7.Close()
        SerialPort8.Close()

        ButtonConnect.Text = "Connect"
        ButtonConnect.selected = False
        ButtonConnect.TextAlign =
            ContentAlignment.MiddleRight
        ButtonConnect.BackColor =
            Color.FromArgb(45, 54, 76)
        ButtonConnect.Iconimage =
            Image.FromFile(ConSymbol)
        LabelStatsConValue.Text = "Disconnected"
        LabelStatsConValue.ForeColor =
            Color.FromArgb(234, 86, 97)
        Catch ex As Exception
```

```
        MsgBox("Failed to disconnect!",  
        MsgBoxStyle.Critical, "Error")  
    End Try  
End If  
End Sub  
  
Private Sub SerialPort1_DataReceived(sender As  
Object, e As IO.Ports.SerialDataReceivedEventArgs)  
Handles SerialPort1.DataReceived  
    data_masuk1 = SerialPort1.ReadLine  
End Sub  
  
Private Sub SerialPort2_DataReceived(sender As  
Object, e As IO.Ports.SerialDataReceivedEventArgs)  
Handles SerialPort2.DataReceived  
    data_masuk2 = SerialPort2.ReadLine  
End Sub  
  
Private Sub SerialPort3_DataReceived(sender As  
Object, e As IO.Ports.SerialDataReceivedEventArgs)  
Handles SerialPort3.DataReceived  
    data_masuk3 = SerialPort3.ReadLine  
End Sub  
  
Private Sub SerialPort4_DataReceived(sender As  
Object, e As IO.Ports.SerialDataReceivedEventArgs)  
Handles SerialPort4.DataReceived  
    data_masuk4 = SerialPort4.ReadLine  
End Sub  
  
Private Sub SerialPort5_DataReceived(sender As  
Object, e As IO.Ports.SerialDataReceivedEventArgs)  
Handles SerialPort5.DataReceived  
    data_masuk5 = SerialPort5.ReadLine  
End Sub
```

```
    Private Sub SerialPort6_DataReceived(sender As Object, e As IO.Ports.SerialDataReceivedEventArgs)
Handles SerialPort6.DataReceived
        data_masuk6 = SerialPort6.ReadLine
    End Sub
```

```
    Private Sub SerialPort7_DataReceived(sender As Object, e As IO.Ports.SerialDataReceivedEventArgs)
Handles SerialPort7.DataReceived
        data_masuk7 = SerialPort7.ReadLine
    End Sub
```

```
    Private Sub SerialPort8_DataReceived(sender As Object, e As IO.Ports.SerialDataReceivedEventArgs)
Handles SerialPort8.DataReceived
        data_masuk8 = SerialPort8.ReadLine
    End Sub
```

```
    Private Sub TimerSerial1_Tick(sender As Object, e As EventArgs) Handles TimerSerial1.Tick
        Try
            'data_masuk1 = SerialPort1.ReadLine

            SerialPort1.WriteLine("#1a000%") 'local name'
            SerialPort1.WriteLine("#1b000%") 'level'
            SerialPort1.WriteLine("#1c000%") 'total water'
            SerialPort1.WriteLine("#1d000%") 'motor
stepper'
            SerialPort1.WriteLine("#1e000%") 'pump'
            SerialPort1.WriteLine("#1f000%") 'override'
            SerialPort1.WriteLine("#1g000%") 'on/off loop'
            SerialPort1.WriteLine("#1h000%") 'alarm'
```

```
        Dim data1 As String() = data_masuk1.Split(",")
```

```
PumpLevel = data1(1)
PumpFlow = data1(2)
PumpStepper = data1(3)
PumpTank = data1(4)
PumpOverride = data1(5)
PumpPlant = data1(6)
PumpAlarm = data1(7)

LabelTankLevel.Text = PumpLevel + " CM"
LevelTank.Value = data1(1)
LabelTankFlow.Text = PumpFlow + " L/M"
LabelTankStepper.Text = PumpStepper + " %"

If PumpStepper = "0" Then
    StepperOff.Visible = True
    StepperOn.Visible = False
Else
    StepperOff.Visible = False
    StepperOn.Visible = True
End If

If PumpLevel < PumpLevelLower Then
    Alarm = "PumpLevelMin"
    SerialPort1.WriteLine("#1q001%") 'alarm'
    SerialPort8.WriteLine("#8q001%") 'buzzer'
    AlarmLevel.Visible = True
    ButtonReset.Visible = True
    'PlayLoopingBackgroundSoundFile()
ElseIf PumpLevel > PumpLevelUpper Then
    Alarm = "PumpLevelMax"
    SerialPort1.WriteLine("#1q001%") 'alarm'
    SerialPort8.WriteLine("#8q001%") 'buzzer'
    AlarmLevel.Visible = True
    ButtonReset.Visible = True
```

```

'PlayLoopingBackgroundSoundFile()
ElseIf PumpLevel = PumpSetPoint Then
    Alarm = ""
    SerialPort1.WriteLine("#1q000%") 'alarm
    SerialPort1.WriteLine("#1o000%") 'override
    SerialPort8.WriteLine("#8q000%") 'buzzer
    AlarmLevel.Visible = False
    ButtonReset.Visible = False
End If
Catch ex As Exception
    'MessageBox.Show(ex.ToString)
End Try
End Sub

Private Sub TimerSerial2_Tick(sender As Object, e As EventArgs) Handles TimerSerial2.Tick
    Try
        'data_masuk2 = SerialPort2.ReadLine

        SerialPort2.WriteLine("#2a000%") 'local name'
        SerialPort2.WriteLine("#2b000%") 'dry level'
        SerialPort2.WriteLine("#2c000%") 'temperature'
        SerialPort2.WriteLine("#2d000%") 'heater'
        SerialPort2.WriteLine("#2e000%") 'temperature
set'
        SerialPort2.WriteLine("#2f000%") 'solenoid
valve'
        SerialPort2.WriteLine("#2g000%") 'override'
        SerialPort2.WriteLine("#2h000%") 'on/off loop'
        SerialPort2.WriteLine("#2i000%") 'alarm

        Dim data2 As String() = data_masuk2.Split(";")
        HeaterDryLevel = data2(1)
        HeaterTemperature = data2(2)
    End Try
End Sub

```

```

HeaterValue = data2(3)
HeaterTSet = data2(4)
HeaterSolenoid = data2(5)
HeaterOverride = data2(6)
HeaterPlant = data2(7)
HeaterAlarm = data2(8)
'LabelHeaterTemperature.Text = data2(2)
LabelHeaterTemperature.Text =
HeaterTemperature + " °C"
LabelHeaterTSetValue.Text = HeaterTSet + " °C"

If HeaterDryLevel = "0" Then
    LevelHeater.Value = LevelHeater.Minimum
ElseIf HeaterDryLevel = "1" Then
    LevelHeater.Value = LevelHeater.Maximum
End If

If HeaterSolenoid = "0" Then
    SISHeaterOff.Visible = True
    SISHeaterOn.Visible = False
Else
    SISHeaterOff.Visible = False
    SISHeaterOn.Visible = True
End If

If HeaterTemperature < HeaterTempLower Then
    Alarm = "HeaterTempMin"
    SerialPort2.WriteLine("#2q001%") 'alarm'
    SerialPort8.WriteLine("#8q001%") 'buzzer
    AlarmHeater.Visible = True
    ButtonReset.Visible = True
    'PlayLoopingBackgroundSoundFile()
ElseIf HeaterTemperature > HeaterTempUpper
Then
    Alarm = "HeaterTempMax"

```

```
    SerialPort2.WriteLine("#2q001%") 'alarm'
    SerialPort8.WriteLine("#8q001%") 'buzzer
    AlarmHeater.Visible = True
    ButtonReset.Visible = True
    'PlayLoopingBackgroundSoundFile()
ElseIf HeaterTemperature = HeaterSetPoint Then
    Alarm = ""
    SerialPort2.WriteLine("#2q000%") 'alarm
    SerialPort4.WriteLine("#4n000%") 'override
naja
    SerialPort8.WriteLine("#8q000%") 'buzzer
    AlarmHeater.Visible = False
    ButtonReset.Visible = False
End If
Catch ex As Exception
```

```
End Try
```

```
End Sub
```

```
Private Sub TimerSerial3_Tick(sender As Object, e As
EventArgs) Handles TimerSerial3.Tick
Try
    'data_masuk3 = SerialPort3.ReadLine

    SerialPort3.WriteLine("#3a000%") 'local name'
    SerialPort3.WriteLine("#3b000%") 'dry level'
    SerialPort3.WriteLine("#3c000%") 'temperature'
    SerialPort3.WriteLine("#3d000%") 'refrigerant'
    SerialPort3.WriteLine("#3e000%") 'temperature
set'
    SerialPort3.WriteLine("#3f000%") 'solenoid
valve'
    SerialPort3.WriteLine("#3g000%") 'solenoid
valve'
    SerialPort3.WriteLine("#3h000%") 'override'
```

```
SerialPort3.WriteLine("#3i000%") 'on/off loop'
SerialPort3.WriteLine("#3j000%") 'alarm'

Dim data3 As String() = data_masuk3.Split(";")

CoolerDryLevel = data3(1)
CoolerTemperature = data3(2)
CoolerRefrigerant = data3(3)
CoolerTSet = data3(4)
CoolerSelVal1 = data3(5)
CoolerSelVal2 = data3(6)
CoolerOverride = data3(7)
CoolerPlant = data3(8)
CoolerAlarm = data3(9)

LabelCoolerTemperature.Text =
CoolerTemperature + " °C"

If CoolerSelVal1 = "0" Then
    InCoolerOff.Visible = True
    InCoolerOn.Visible = False
Else
    InCoolerOff.Visible = False
    InCoolerOn.Visible = True
End If

If CoolerSelVal2 = "0" Then
    SISCoolerOff.Visible = True
    SISCoolerOn.Visible = False
Else
    SISCoolerOff.Visible = False
    SISCoolerOn.Visible = True
End If

If CoolerDryLevel = "0" Then
```

```

        LevelCooler.Value = LevelCooler.Minimum
    ElseIf CoolerDryLevel = "1" Then
        LevelCooler.Value = LevelCooler.Maximum
    End If

    If CoolerTemperature < CoolerTempLower Then
        Alarm = "CoolerTempMin"
        SerialPort3.WriteLine("#3r001%") 'alarm'
        SerialPort8.WriteLine("#8q001%") 'buzzer
        AlarmCooler.Visible = True
        ButtonReset.Visible = True
        'PlayLoopingBackgroundSoundFile()
    ElseIf CoolerTemperature > CoolerTempUpper
    Then
        Alarm = "CoolerTempMax"
        SerialPort3.WriteLine("#3r001%") 'alarm'
        SerialPort8.WriteLine("#8q001%") 'buzzer
        AlarmCooler.Visible = True
        ButtonReset.Visible = True
        'PlayLoopingBackgroundSoundFile()
    ElseIf CoolerTemperature = CoolerSetPoint Then
        Alarm = ""
        SerialPort3.WriteLine("#3r000%") 'alarm
        SerialPort5.WriteLine("#5n000%") 'override
    enver
        SerialPort8.WriteLine("#8q000%") 'buzzer
        AlarmCooler.Visible = False
        ButtonReset.Visible = False
    End If
    Catch ex As Exception
        'MessageBox.Show(ex.ToString)
    End Try
End Sub

```

```
Private Sub TimerSerial4_Tick(sender As Object, e As  
EventArgs) Handles TimerSerial4.Tick  
    Try  
        'data_masuk4 = SerialPort4.ReadLine  
  
        SerialPort4.WriteLine("#4a000%") 'local name'  
        SerialPort4.WriteLine("#4b000%") 'flow'  
        SerialPort4.WriteLine("#4c000%") 'temperature'  
        SerialPort4.WriteLine("#4d000%") 'MOV'  
        SerialPort4.WriteLine("#4e000%") 'override'  
        SerialPort4.WriteLine("#4f000%") 'on/off loop'  
        SerialPort4.WriteLine("#4g000%") 'alarm'  
  
        Dim data4 As String() = data_masuk4.Split(";")  
  
        MOVFlow = data4(1)  
        MOTTemperature = data4(2)  
        MOVValue = data4(3)  
        MOVOVERRIDE = data4(4)  
        MOVPlant = data4(5)  
        MOVALarm = data4(6)  
  
        LabelMOVFlow.Text = MOVFlow + " LPM"  
        LabelMOTTemperature.Text =  
        MOTTemperature + " °C"  
        LabelMOVValue.Text = MOVValue + " %"  
  
        If MOVValue = "0" Then  
            MOVOff.Visible = True  
            MOVOn.Visible = False  
        Else  
            MOVOff.Visible = False  
            MOVOn.Visible = True  
        End If
```

```

'If MOVTemperature < MOVTemperatureLower
Then
    'PlayLoopingBackgroundSoundFile()
    'End If

'If MOVTemperature > MOVTemperatureUpper
Then
    'PlayLoopingBackgroundSoundFile()
    'End If
    Catch ex As Exception
        'MessageBox.Show(ex.ToString)
    End Try
End Sub

Private Sub TimerSerial5_Tick(sender As Object, e As
EventArgs) Handles TimerSerial5.Tick
    Try
        'data_masuk5 = SerialPort5.ReadLine

        SerialPort5.WriteLine("#5a000%") 'local name
        nia'
        SerialPort5.WriteLine("#5b000%") 'flow'
        SerialPort5.WriteLine("#5c000%") 'temperature'
        SerialPort5.WriteLine("#5d000%") 'vsd'
        SerialPort5.WriteLine("#5e000%") 'override'
        SerialPort5.WriteLine("#5f000%") 'on/off loop'
        SerialPort5.WriteLine("#5g000%") 'alarm'

        Dim data5 As String() = data_masuk5.Split(";")

        VSDFlow = data5(1)
        VSDTemperature = data5(2)
        VSDPump = data5(3)
        VSDOverride = data5(4)
        VSDPlant = data5(5)
    End Try
End Sub

```

```

VSDAlarm = data5(6)

LabelVSDFlow.Text = VSDFlow + " L/M"
LabelVSDTemperature.Text = VSDTemperature
+ " °C"
LabelVSDPump.Text = VSDPump + " %"
If VSDPump = "0" Then
    PumpCoolerOff.Visible = True
    PumpCoolerOn.Visible = False
Else
    PumpCoolerOff.Visible = False
    PumpCoolerOn.Visible = True
End If
Catch ex As Exception
    MessageBox.Show(ex.ToString)
End Try
End Sub

Private Sub TimerSerial6_Tick(sender As Object, e As
EventArgs) Handles TimerSerial6.Tick
Try
    'data_masuk6 = SerialPort6.ReadLine

    SerialPort6.WriteLine("#6a000%") 'local name
    nia'
    SerialPort6.WriteLine("#6b000%") 'temperature
    in'
    SerialPort6.WriteLine("#6c000%") 'temperature
    out'
    SerialPort6.WriteLine("#6d000%") 'flow'
    SerialPort6.WriteLine("#6e000%") 'fan'
    SerialPort6.WriteLine("#6f000%") 'override'
    SerialPort6.WriteLine("#6g000%") 'on/off loop'
    SerialPort6.WriteLine("#6h000%") 'alarm'

```

```
Dim data6 As String() = data_masuk6.Split(";")

TowerTempIn1 = data6(1)
TowerTempOut1 = data6(2)
TowerFlow1 = data6(3)
TowerFan1 = data6(4)
TowerOverride1 = data6(5)
TowerPlant1 = data6(6)
TowerAlarm1 = data6(7)

LabelTowerTempIn1.Text = TowerTempIn1 + "
°C"
LabelTowerTempOut1.Text = TowerTempOut1 +
" °C"
LabelTowerFlow.Text = TowerFlow1 + " L/M"
LabelTowerFan1.Text = TowerFan1 + " %"

If TowerFan1 = "0" Then
    FanTower1.Visible = False
Else
    FanTower1.Visible = True
End If
Catch ex As Exception
    'MessageBox.Show(ex.ToString)
End Try
End Sub

Private Sub TimerSerial7_Tick(sender As Object, e As
EventArgs) Handles TimerSerial7.Tick
Try
    'data_masuk7 = SerialPort7.ReadLine

    SerialPort7.WriteLine("#7a000%") 'local name
    nia'
```

```
SerialPort7.WriteLine("#7b000%") 'temperature
in'
SerialPort7.WriteLine("#7c000%") 'temperature
out'
SerialPort7.WriteLine("#7d000%") 'flow'
SerialPort7.WriteLine("#7e000%") 'fan'
SerialPort7.WriteLine("#7f000%") 'override'
SerialPort7.WriteLine("#7g000%") 'on/off loop'
SerialPort7.WriteLine("#7h000%") 'alarm'

Dim data7 As String() = data_masuk7.Split(";")

TowerTempIn2 = data7(1)
TowerTempOut2 = data7(2)
TowerFlow2 = data7(3)
TowerFan2 = data7(4)
TowerOverride2 = data7(5)
TowerPlant2 = data7(6)
TowerAlarm2 = data7(7)

LabelTowerTempIn2.Text = TowerTempIn2 + "
°C"
LabelTowerTempOut2.Text = TowerTempOut2 +
" °C"
LabelTowerFlow.Text = TowerFlow2 + " L/M"
LabelTowerFan2.Text = TowerFan2 + "%"

If TowerFan2 = "0" Then
    FanTower2.Visible = False
Else
    FanTower2.Visible = True
End If
Catch ex As Exception
    MessageBox.Show(ex.ToString)
End Try
```

```
End Sub

Private Sub TimerSerial8_Tick(sender As Object, e As
EventArgs) Handles TimerSerial8.Tick
    Try
        'data_masuk8 = SerialPort8.ReadLine

        SerialPort8.WriteLine("#8a000%") 'local name
        SerialPort8.WriteLine("#8b000%") 'temperature
        SerialPort8.WriteLine("#8c000%") 'pressure
        SerialPort8.WriteLine("#8d000%") 'contactor
        SerialPort8.WriteLine("#8e000%") 'override
        SerialPort8.WriteLine("#8f000%") 'alarm
        SerialPort8.WriteLine("#8g000%") 'solenoid
        valve
        SerialPort8.WriteLine("#8h000%") 'buzzer

        Dim data8 As String() = data_masuk8.Split(";")

        SISTemperature = data8(1)
        SISPressure = data8(2)
        SISContactor = data8(3)
        SISOverride = data8(4)
        SISBacklight = data8(5)
        SISSolenoidValve = data8(6)
        SISBuzzer = data8(7)

        LabelSISPressure.Text = SISPressure + " bar"
        LabelSISTemperature.Text = SISTemperature + "
        °C"

        Catch ex As Exception
            MessageBox.Show(ex.ToString)
        End Try
    End Sub
```

```
Private Sub Button1Plant_Click(sender As Object, e  
As EventArgs) Handles Button1Plant.Click  
    If Button1Plant.Text = "ON" Then  
        SerialPort1.WriteLine("#1p001%")  
  
        Button1Plant.Text = "OFF"  
        Button1Plant.selected = True  
        Button1Plant.TextAlign =  
ContentAlignment.MiddleRight  
        Button1Plant.BackColor = Color.FromArgb(56,  
85, 153)  
        Button1Plant.Iconimage =  
Image.FromFile(OffSymbol)  
        Label1PlantStatsValue.Text = "ON"  
        Label1PlantStatsValue.ForeColor =  
Color.FromArgb(6, 190, 190)  
    ElseIf Button1Plant.Text = "OFF" Then  
        SerialPort1.WriteLine("#1p000%")  
  
        Button1Plant.Text = "ON"  
        Button1Plant.selected = False  
        Button1Plant.TextAlign =  
ContentAlignment.MiddleRight  
        Button1Plant.BackColor = Color.FromArgb(45,  
54, 76)  
        Button1Plant.Iconimage =  
Image.FromFile(OnSymbol)  
        Label1PlantStatsValue.Text = "OFF"  
        Label1PlantStatsValue.ForeColor =  
Color.FromArgb(234, 86, 97)  
    End If  
End Sub  
  
Private Sub Button1Override_Click(sender As Object,  
e As EventArgs) Handles Button1Override.Click
```

```
If Button1Override.Text = "ON" Then
    SerialPort1.WriteLine("#1o001%")
    
    Button1Override.Text = "OFF"
    Button1Override.selected = True
    Button1Override.TextAlign =
ContentAlignment.MiddleRight
    Button1Override.BackColor =
Color.FromArgb(56, 85, 153)
    Button1Override.Iconimage =
Image.FromFile(OffSymbol)
    Label1OverrideStatsValue.Text = "ON"
    Label1OverrideStatsValue.ForeColor =
Color.FromArgb(6, 190, 190)
ElseIf Button1Override.Text = "OFF" Then
    SerialPort1.WriteLine("#1o000%")
    
    Button1Override.Text = "ON"
    Button1Override.selected = False
    Button1Override.TextAlign =
ContentAlignment.MiddleRight
    Button1Override.BackColor =
Color.FromArgb(45, 54, 76)
    Button1Override.Iconimage =
Image.FromFile(OnSymbol)
    Label1OverrideStatsValue.Text = "OFF"
    Label1OverrideStatsValue.ForeColor =
Color.FromArgb(234, 86, 97)
End If
End Sub
```

```
Private Sub ButtonPump_Click(sender As Object, e As
EventArgs) Handles ButtonPump.Click
    If ButtonPump.Text = "ON" Then
        SerialPort1.WriteLine("#1n001%")
```

```
    ButtonPump.Text = "OFF"
    ButtonPump.selected = True
    ButtonPump.TextAlign =
    ContentAlignment.MiddleRight
    ButtonPump.BackColor = Color.FromArgb(56,
85, 153)
    ButtonPump.Iconimage =
Image.FromFile(OffSymbol)
    Label1PumpStatsValue.Text = "ON"
    Label1PumpStatsValue.ForeColor =
Color.FromArgb(6, 190, 190)
ElseIf ButtonPump.Text = "OFF" Then
    SerialPort1.WriteLine("#1n000%")

    ButtonPump.Text = "ON"
    ButtonPump.selected = False
    ButtonPump.TextAlign =
    ContentAlignment.MiddleRight
    ButtonPump.BackColor = Color.FromArgb(45,
54, 76)
    ButtonPump.Iconimage =
Image.FromFile(OnSymbol)
    Label1PumpStatsValue.Text = "OFF"
    Label1PumpStatsValue.ForeColor =
Color.FromArgb(234, 86, 97)
End If
End Sub

Private Sub
Dropdown1Stepper_onItemSelected(sender As Object, e
As EventArgs) Handles
Dropdown1Stepper.onItemSelected
    Dim Stepper1Value As String =
Dropdown1Stepper.SelectedValue
```

```
If Stepper1Value = "0" Then Stepper1Value = "000"
If Stepper1Value = "10" Then Stepper1Value =
"010"
If Stepper1Value = "20" Then Stepper1Value =
"020"
If Stepper1Value = "30" Then Stepper1Value =
"030"
If Stepper1Value = "40" Then Stepper1Value =
"040"
If Stepper1Value = "50" Then Stepper1Value =
"050"
If Stepper1Value = "60" Then Stepper1Value =
"060"
If Stepper1Value = "70" Then Stepper1Value =
"070"
If Stepper1Value = "80" Then Stepper1Value =
"080"
If Stepper1Value = "90" Then Stepper1Value =
"090"
If Stepper1Value = "100" Then Stepper1Value =
"100"

    SerialPort1.WriteLine("#1m" + Stepper1Value +
    "%")
    Label1StepperStatsValue.Text =
    Dropdown1Stepper.selectedValue + "%"
End Sub

Private Sub Button2Plant_Click(sender As Object, e
As EventArgs) Handles Button2Plant.Click
    If Button2Plant.Text = "ON" Then
        SerialPort2.WriteLine("#2p001%")

        Button2Plant.Text = "OFF"
        Button2Plant.selected = True
```

```
    Button2Plant.TextAlign =
ContentAlignment.MiddleRight
    Button2Plant.BackColor = Color.FromArgb(56,
85, 153)
    Button2Plant.Iconimage =
Image.FromFile(OffSymbol)
    Label2PlantStatsValue.Text = "ON"
    Label2PlantStatsValue.ForeColor =
Color.FromArgb(6, 190, 190)
ElseIf Button2Plant.Text = "OFF" Then
    SerialPort2.WriteLine("#2p000% ")

    Button2Plant.Text = "ON"
    Button2Plant.selected = False
    Button2Plant.TextAlign =
ContentAlignment.MiddleRight
    Button2Plant.BackColor = Color.FromArgb(45,
54, 76)
    Button2Plant.Iconimage =
Image.FromFile(OnSymbol)
    Label2PlantStatsValue.Text = "OFF"
    Label2PlantStatsValue.ForeColor =
Color.FromArgb(234, 86, 97)
End If
End Sub

Private Sub Button2Override_Click(sender As Object,
e As EventArgs) Handles Button2Override.Click
If Button2Override.Text = "ON" Then
    SerialPort2.WriteLine("#2o001% ")

    Button2Override.Text = "OFF"
    Button2Override.selected = True
    Button2Override.TextAlign =
ContentAlignment.MiddleRight
```

```
        Button2Override.BackColor =
Color.FromArgb(56, 85, 153)
        Button2Override.Iconimage =
Image.FromFile(OffSymbol)
        Label2OverrideStatsValue.Text = "ON"
        Label2OverrideStatsValue.ForeColor =
Color.FromArgb(6, 190, 190)
    ElseIf Button2Override.Text = "OFF" Then
        SerialPort2.WriteLine("#2o000%")

        Button2Override.Text = "ON"
        Button2Override.selected = False
        Button2Override.TextAlign =
ContentAlignment.MiddleRight
        Button2Override.BackColor =
Color.FromArgb(45, 54, 76)
        Button2Override.Iconimage =
Image.FromFile(OnSymbol)
        Label2OverrideStatsValue.Text = "OFF"
        Label2OverrideStatsValue.ForeColor =
Color.FromArgb(234, 86, 97)
    End If
End Sub
```

```
Private Sub Button2Valve_Click(sender As Object, e
As EventArgs) Handles Button2Valve.Click
    If Button2Valve.Text = "ON" Then
        SerialPort2.WriteLine("#2n001%")

        Button2Valve.Text = "OFF"
        Button2Valve.selected = True
        Button2Valve.TextAlign =
ContentAlignment.MiddleRight
        Button2Valve.BackColor = Color.FromArgb(56,
85, 153)
```

```
        Button2Valve.Iconimage =
Image.FromFile(OffSymbol)
        Label2ValveStatsValue.Text = "ON"
        Label2ValveStatsValue.ForeColor =
Color.FromArgb(6, 190, 190)
    ElseIf Button2Valve.Text = "OFF" Then
        SerialPort2.WriteLine("#2n000%")

        Button2Valve.Text = "ON"
        Button2Valve.selected = False
        Button2Valve.TextAlign =
ContentAlignment.MiddleRight
        Button2Valve.BackColor = Color.FromArgb(45,
54, 76)
        Button2Valve.Iconimage =
Image.FromFile(OnSymbol)
        Label2ValveStatsValue.Text = "OFF"
        Label2ValveStatsValue.ForeColor =
Color.FromArgb(234, 86, 97)
    End If
End Sub
```

```
Private Sub Dropdown2TSet_onItemSelected(sender
As Object, e As EventArgs) Handles
Dropdown2TSet.onItemSelected
    Dim TSet2Value As String =
Dropdown2TSet.SelectedValue
    If TSet2Value = "0" Then TSet2Value = "000"
    If TSet2Value = "10" Then TSet2Value = "010"
    If TSet2Value = "20" Then TSet2Value = "020"
    If TSet2Value = "30" Then TSet2Value = "030"
    If TSet2Value = "40" Then TSet2Value = "040"
    If TSet2Value = "50" Then TSet2Value = "050"
    If TSet2Value = "60" Then TSet2Value = "060"
    If TSet2Value = "70" Then TSet2Value = "070"
```

```
If TSet2Value = "80" Then TSet2Value = "080"
If TSet2Value = "90" Then TSet2Value = "090"
If TSet2Value = "100" Then TSet2Value = "100"

    SerialPort2.WriteLine("#2m" + TSet2Value + "%")
    Label2TSetStatsValue.Text =
    Dropdown2TSet.selectedValue + "%"
End Sub

Private Sub Button3Plant_Click(sender As Object, e
As EventArgs) Handles Button3Plant.Click
    If Button3Plant.Text = "ON" Then
        SerialPort3.WriteLine("#3q001%")

        Button3Plant.Text = "OFF"
        Button3Plant.selected = True
        Button3Plant.TextAlign =
        ContentAlignment.MiddleRight
        Button3Plant.BackColor = Color.FromArgb(56,
85, 153)
        Button3Plant.Iconimage =
        Image.FromFile(OffSymbol)
        Label3PlantStatsValue.Text = "ON"
        Label3PlantStatsValue.ForeColor =
        Color.FromArgb(6, 190, 190)
    ElseIf Button3Plant.Text = "OFF" Then
        SerialPort3.WriteLine("#3q000%")

        Button3Plant.Text = "ON"
        Button3Plant.selected = False
        Button3Plant.TextAlign =
        ContentAlignment.MiddleRight
        Button3Plant.BackColor = Color.FromArgb(45,
54, 76)
```

```
        Button3Plant.Iconimage =
Image.FromFile(OnSymbol)
        Label3PlantStatsValue.Text = "OFF"
        Label3PlantStatsValue.ForeColor =
Color.FromArgb(234, 86, 97)
    End If
End Sub

Private Sub Button3Override_Click(sender As Object,
e As EventArgs) Handles Button3Override.Click
    If Button3Override.Text = "ON" Then
        SerialPort3.WriteLine("#3p001%")

        Button3Override.Text = "OFF"
        Button3Override.selected = True
        Button3Override.TextAlign =
ContentAlignment.MiddleRight
        Button3Override.BackColor =
Color.FromArgb(56, 85, 153)
        Button3Override.Iconimage =
Image.FromFile(OffSymbol)
        Label3OverrideStatsValue.Text = "ON"
        Label3OverrideStatsValue.ForeColor =
Color.FromArgb(6, 190, 190)
    ElseIf Button3Override.Text = "OFF" Then
        SerialPort3.WriteLine("#3p000%")

        Button3Override.Text = "ON"
        Button3Override.selected = False
        Button3Override.TextAlign =
ContentAlignment.MiddleRight
        Button3Override.BackColor =
Color.FromArgb(45, 54, 76)
        Button3Override.Iconimage =
Image.FromFile(OnSymbol)
```

```
    Label3OverrideStatsValue.Text = "OFF"
    Label3OverrideStatsValue.ForeColor =
Color.FromArgb(234, 86, 97)
End If
End Sub

Private Sub Button3ValveTop_Click(sender As Object,
e As EventArgs) Handles Button3ValveTop.Click
If Button3ValveTop.Text = "ON" Then
    SerialPort3.WriteLine("#3n001%")

    Button3ValveTop.Text = "OFF"
    Button3ValveTop.selected = True
    Button3ValveTop.TextAlign =
ContentAlignment.MiddleRight
    Button3ValveTop.BackColor =
Color.FromArgb(56, 85, 153)
    Button3ValveTop.Iconimage =
Image.FromFile(OffSymbol)
    Label3ValveTopStatsValue.Text = "ON"
    Label3ValveTopStatsValue.ForeColor =
Color.FromArgb(6, 190, 190)
ElseIf Button3ValveTop.Text = "OFF" Then
    SerialPort3.WriteLine("#3n000%")

    Button3ValveTop.Text = "ON"
    Button3ValveTop.selected = False
    Button3ValveTop.TextAlign =
ContentAlignment.MiddleRight
    Button3ValveTop.BackColor =
Color.FromArgb(45, 54, 76)
    Button3ValveTop.Iconimage =
Image.FromFile(OnSymbol)
    Label3ValveTopStatsValue.Text = "OFF"
```

```
    Label3ValveTopStatsValue.ForeColor =
Color.FromArgb(234, 86, 97)
End If
End Sub

Private Sub Button3ValveBot_Click(sender As Object,
e As EventArgs) Handles Button3ValveBot.Click
If Button3ValveBot.Text = "ON" Then
    SerialPort3.WriteLine("#3o001%")

    Button3ValveBot.Text = "OFF"
    Button3ValveBot.selected = True
    Button3ValveBot.TextAlign =
ContentAlignment.MiddleRight
    Button3ValveBot.BackColor =
Color.FromArgb(56, 85, 153)
    Button3ValveBot.Iconimage =
Image.FromFile(OffSymbol)
    Label3ValveBotStatsValue.Text = "ON"
    Label3ValveBotStatsValue.ForeColor =
Color.FromArgb(6, 190, 190)
ElseIf Button3ValveBot.Text = "OFF" Then
    SerialPort3.WriteLine("#3o000%")

    Button3ValveBot.Text = "ON"
    Button3ValveBot.selected = False
    Button3ValveBot.TextAlign =
ContentAlignment.MiddleRight
    Button3ValveBot.BackColor =
Color.FromArgb(45, 54, 76)
    Button3ValveBot.Iconimage =
Image.FromFile(OnSymbol)
    Label3ValveBotStatsValue.Text = "OFF"
    Label3ValveBotStatsValue.ForeColor =
Color.FromArgb(234, 86, 97)
```

```
    End If
End Sub

Private Sub Dropdown3TSet_onItemSelected(sender
As Object, e As EventArgs) Handles
Dropdown3TSet.onItemSelected
    Dim TSet3Value As String =
Dropdown3TSet.selectedValue
    If TSet3Value = "0" Then TSet3Value = "000"
    If TSet3Value = "10" Then TSet3Value = "010"
    If TSet3Value = "20" Then TSet3Value = "020"
    If TSet3Value = "30" Then TSet3Value = "030"
    If TSet3Value = "40" Then TSet3Value = "040"
    If TSet3Value = "50" Then TSet3Value = "050"
    If TSet3Value = "60" Then TSet3Value = "060"
    If TSet3Value = "70" Then TSet3Value = "070"
    If TSet3Value = "80" Then TSet3Value = "080"
    If TSet3Value = "90" Then TSet3Value = "090"
    If TSet3Value = "100" Then TSet3Value = "100"

    SerialPort3.WriteLine("#3m" + TSet3Value + "%")
    Label3TSetStatsValue.Text =
Dropdown3TSet.selectedValue + "%"
End Sub

Private Sub Button4Plant_Click(sender As Object, e
As EventArgs) Handles Button4Plant.Click
    If Button4Plant.Text = "ON" Then
        SerialPort4.WriteLine("#4o001%")

        Button4Plant.Text = "OFF"
        Button4Plant.selected = True
        Button4Plant.TextAlign =
ContentAlignment.MiddleRight
```

```
        Button4Plant.BackColor = Color.FromArgb(56,
85, 153)
        Button4Plant.Iconimage =
Image.FromFile(OffSymbol)
        Label4PlantStatsValue.Text = "ON"
        Label4PlantStatsValue.ForeColor =
Color.FromArgb(6, 190, 190)
    ElseIf Button4Plant.Text = "OFF" Then
        SerialPort4.WriteLine("#4o000%")

        Button4Plant.Text = "ON"
        Button4Plant.selected = False
        Button4Plant.TextAlign =
ContentAlignment.MiddleRight
        Button4Plant.BackColor = Color.FromArgb(45,
54, 76)
        Button4Plant.Iconimage =
Image.FromFile(OnSymbol)
        Label4PlantStatsValue.Text = "OFF"
        Label4PlantStatsValue.ForeColor =
Color.FromArgb(234, 86, 97)
    End If
End Sub

Private Sub Button4Override_Click(sender As Object,
e As EventArgs) Handles Button4Override.Click
    If Button4Override.Text = "ON" Then
        SerialPort4.WriteLine("#4n001%")

        Button4Override.Text = "OFF"
        Button4Override.selected = True
        Button4Override.TextAlign =
ContentAlignment.MiddleRight
        Button4Override.BackColor =
Color.FromArgb(56, 85, 153)
```

```
        Button4Override.Iconimage =
Image.FromFile(OffSymbol)
        Label4OverrideStatsValue.Text = "ON"
        Label4OverrideStatsValue.ForeColor =
Color.FromArgb(6, 190, 190)
    ElseIf Button4Override.Text = "OFF" Then
        SerialPort4.WriteLine("#4n000%")

        Button4Override.Text = "ON"
        Button4Override.selected = False
        Button4Override.TextAlign =
ContentAlignment.MiddleRight
        Button4Override.BackColor =
Color.FromArgb(45, 54, 76)
        Button4Override.Iconimage =
Image.FromFile(OnSymbol)
        Label4OverrideStatsValue.Text = "OFF"
        Label4OverrideStatsValue.ForeColor =
Color.FromArgb(234, 86, 97)
    End If
End Sub
```

```
Private Sub Dropdown4MOV_onItemSelected(sender
As Object, e As EventArgs) Handles
Dropdown4MOV.onItemSelected
    Dim MOV4Value As String =
Dropdown4MOV.SelectedValue
    If MOV4Value = "0" Then MOV4Value = "000"
    If MOV4Value = "10" Then MOV4Value = "010"
    If MOV4Value = "20" Then MOV4Value = "020"
    If MOV4Value = "30" Then MOV4Value = "030"
    If MOV4Value = "40" Then MOV4Value = "040"
    If MOV4Value = "50" Then MOV4Value = "050"
    If MOV4Value = "60" Then MOV4Value = "060"
    If MOV4Value = "70" Then MOV4Value = "070"
```

```
If MOV4Value = "80" Then MOV4Value = "080"
If MOV4Value = "90" Then MOV4Value = "090"
If MOV4Value = "100" Then MOV4Value = "100"

    SerialPort4.WriteLine("#4m" + MOV4Value + "%")
    Label4MOVStatsValue.Text =
    Dropdown4MOV.selectedValue + "%"
End Sub

Private Sub Button5Plant_Click(sender As Object, e
As EventArgs) Handles Button5Plant.Click
    If Button5Plant.Text = "ON" Then
        SerialPort5.WriteLine("#5o001%")

        Button5Plant.Text = "OFF"
        Button5Plant.selected = True
        Button5Plant.TextAlign =
        ContentAlignment.MiddleRight
        Button5Plant.BackColor = Color.FromArgb(56,
85, 153)
        Button5Plant.Iconimage =
        Image.FromFile(OffSymbol)
        Label5PlantStatsValue.Text = "ON"
        Label5PlantStatsValue.ForeColor =
        Color.FromArgb(6, 190, 190)
    ElseIf Button5Plant.Text = "OFF" Then
        SerialPort5.WriteLine("#5o000%")

        Button5Plant.Text = "ON"
        Button5Plant.selected = False
        Button5Plant.TextAlign =
        ContentAlignment.MiddleRight
        Button5Plant.BackColor = Color.FromArgb(45,
54, 76)
```

```
    Button5Plant.Iconimage =
Image.FromFile(OnSymbol)
    Label5PlantStatsValue.Text = "OFF"
    Label5PlantStatsValue.ForeColor =
Color.FromArgb(234, 86, 97)
End If
End Sub

Private Sub Button5Override_Click(sender As Object,
e As EventArgs) Handles Button5Override.Click
    If Button5Override.Text = "ON" Then
        SerialPort5.WriteLine("#5n001%")

        Button5Override.Text = "OFF"
        Button5Override.selected = True
        Button5Override.TextAlign =
ContentAlignment.MiddleRight
        Button5Override.BackColor =
Color.FromArgb(56, 85, 153)
        Button5Override.Iconimage =
Image.FromFile(OffSymbol)
        Label5OverrideStatsValue.Text = "ON"
        Label5OverrideStatsValue.ForeColor =
Color.FromArgb(6, 190, 190)
    ElseIf Button5Override.Text = "OFF" Then
        SerialPort5.WriteLine("#5n000%")

        Button5Override.Text = "ON"
        Button5Override.selected = False
        Button5Override.TextAlign =
ContentAlignment.MiddleRight
        Button5Override.BackColor =
Color.FromArgb(45, 54, 76)
        Button5Override.Iconimage =
Image.FromFile(OnSymbol)
```

```
Label5OverrideStatsValue.Text = "OFF"
Label5OverrideStatsValue.ForeColor =
Color.FromArgb(234, 86, 97)
End If
End Sub

Private Sub Dropdown5VSD_onItemSelected(sender
As Object, e As EventArgs) Handles
Dropdown5VSD.onItemSelected
    Dim VSDValue As String =
Dropdown5VSD.selectedValue
    If VSDValue = "0" Then VSDValue = "000"
    If VSDValue = "10" Then VSDValue = "010"
    If VSDValue = "20" Then VSDValue = "020"
    If VSDValue = "30" Then VSDValue = "030"
    If VSDValue = "40" Then VSDValue = "040"
    If VSDValue = "50" Then VSDValue = "050"
    If VSDValue = "60" Then VSDValue = "060"
    If VSDValue = "70" Then VSDValue = "070"
    If VSDValue = "80" Then VSDValue = "080"
    If VSDValue = "90" Then VSDValue = "090"
    If VSDValue = "100" Then VSDValue = "100"

    SerialPort5.WriteLine("#5m" + VSDValue + "%")
    Label5VSDStatsValue.Text =
Dropdown5VSD.selectedValue + "%"
End Sub

Private Sub Button6Override_Click(sender As Object,
e As EventArgs) Handles Button6Override.Click
    If Button6Override.Text = "ON" Then
        SerialPort6.WriteLine("#6n001%")

        Button6Override.Text = "OFF"
        Button6Override.selected = True
    End If
End Sub
```

```
    Button6Override.TextAlign =
ContentAlignment.MiddleRight
    Button6Override.BackColor =
Color.FromArgb(56, 85, 153)
    Button6Override.Iconimage =
Image.FromFile(OffSymbol)
    Label6OverrideStatsValue.Text = "ON"
    Label6OverrideStatsValue.ForeColor =
Color.FromArgb(6, 190, 190)
ElseIf Button6Override.Text = "OFF" Then
    SerialPort6.WriteLine("#6n000%")

    Button6Override.Text = "ON"
    Button6Override.selected = False
    Button6Override.TextAlign =
ContentAlignment.MiddleRight
    Button6Override.BackColor =
Color.FromArgb(45, 54, 76)
    Button6Override.Iconimage =
Image.FromFile(OnSymbol)
    Label6OverrideStatsValue.Text = "OFF"
    Label6OverrideStatsValue.ForeColor =
Color.FromArgb(234, 86, 97)
End If
End Sub
```

```
Private Sub Dropdown6Fan_onItemSelected(sender As
Object, e As EventArgs) Handles
Dropdown6Fan.onItemSelected
    Dim Fan1Value As String =
Dropdown6Fan.SelectedValue
    If Fan1Value = "0" Then Fan1Value = "000"
    If Fan1Value = "10" Then Fan1Value = "010"
    If Fan1Value = "20" Then Fan1Value = "020"
    If Fan1Value = "30" Then Fan1Value = "030"
```

```
If Fan1Value = "40" Then Fan1Value = "040"
If Fan1Value = "50" Then Fan1Value = "050"
If Fan1Value = "60" Then Fan1Value = "060"
If Fan1Value = "70" Then Fan1Value = "070"
If Fan1Value = "80" Then Fan1Value = "080"
If Fan1Value = "90" Then Fan1Value = "090"
If Fan1Value = "100" Then Fan1Value = "100"

    SerialPort6.WriteLine("#6m" + Fan1Value + "%")
    Label6FanStatsValue.Text =
    Dropdown6Fan.SelectedValue + "%"
End Sub

Private Sub Button7Override_Click(sender As Object,
e As EventArgs) Handles Button7Override.Click
    If Button7Override.Text = "ON" Then
        SerialPort7.WriteLine("#7n001%")

        Button7Override.Text = "OFF"
        Button7Override.selected = True
        Button7Override.TextAlign =
        ContentAlignment.MiddleRight
        Button7Override.BackColor =
        Color.FromArgb(56, 85, 153)
        Button7Override.IconImage =
        Image.FromFile(OffSymbol)
        Label7OverrideStatsValue.Text = "ON"
        Label7OverrideStatsValue.ForeColor =
        Color.FromArgb(6, 190, 190)
    ElseIf Button7Override.Text = "OFF" Then
        SerialPort7.WriteLine("#7n000%")

        Button7Override.Text = "ON"
        Button7Override.selected = False
```

```
    Button7Override.TextAlign =
ContentAlignment.MiddleRight
    Button7Override.BackColor =
Color.FromArgb(45, 54, 76)
    Button7Override.IconImage =
Image.FromFile(OnSymbol)
    Label7OverrideStatsValue.Text = "OFF"
    Label7OverrideStatsValue.ForeColor =
Color.FromArgb(234, 86, 97)
End If
End Sub

Private Sub Dropdown7Fan_onItemSelected(sender As
Object, e As EventArgs) Handles
Dropdown7Fan.onItemSelected
    Dim Fan2Value As String =
Dropdown7Fan.selectedValue
    If Fan2Value = "0" Then Fan2Value = "000"
    If Fan2Value = "10" Then Fan2Value = "010"
    If Fan2Value = "20" Then Fan2Value = "020"
    If Fan2Value = "30" Then Fan2Value = "030"
    If Fan2Value = "40" Then Fan2Value = "040"
    If Fan2Value = "50" Then Fan2Value = "050"
    If Fan2Value = "60" Then Fan2Value = "060"
    If Fan2Value = "70" Then Fan2Value = "070"
    If Fan2Value = "80" Then Fan2Value = "080"
    If Fan2Value = "90" Then Fan2Value = "090"
    If Fan2Value = "100" Then Fan2Value = "100"

    SerialPort7.WriteLine("#7m" + Fan2Value + "%")
    Label7FanStatsValue.Text =
Dropdown7Fan.selectedValue + "%"
End Sub
```

```
Private Sub Button8Override_Click(sender As Object,
e As EventArgs) Handles Button8Override.Click
    If Button8Override.Text = "ON" Then
        SerialPort8.WriteLine("#8n001% ")

        Button8Override.Text = "OFF"
        Button8Override.selected = True
        Button8Override.TextAlign =
ContentAlignment.MiddleRight
        Button8Override.BackColor =
Color.FromArgb(56, 85, 153)
        Button8Override.Iconimage =
Image.FromFile(OffSymbol)
        Label8OverrideStatsValue.Text = "ON"
        Label8OverrideStatsValue.ForeColor =
Color.FromArgb(6, 190, 190)
    ElseIf Button8Override.Text = "OFF" Then
        SerialPort8.WriteLine("#8n000% ")

        Button8Override.Text = "ON"
        Button8Override.selected = False
        Button8Override.TextAlign =
ContentAlignment.MiddleRight
        Button8Override.BackColor =
Color.FromArgb(45, 54, 76)
        Button8Override.Iconimage =
Image.FromFile(OnSymbol)
        Label8OverrideStatsValue.Text = "OFF"
        Label8OverrideStatsValue.ForeColor =
Color.FromArgb(234, 86, 97)
    End If
End Sub
```

```
Private Sub Button8Contactor_Click(sender As Object,
e As EventArgs) Handles Button8Contactor.Click
```

```
If Button8Contactor.Text = "ON" Then
    SerialPort8.WriteLine("#8m001%")

        Button8Contactor.Text = "OFF"
        Button8Contactor.selected = True
        Button8Contactor.TextAlign =
ContentAlignment.MiddleRight
        Button8Contactor.BackColor =
Color.FromArgb(56, 85, 153)
        Button8Contactor.Iconimage =
Image.FromFile(OffSymbol)
        Label8ContactorStatsValue.Text = "ON"
        Label8ContactorStatsValue.ForeColor =
Color.FromArgb(6, 190, 190)
    ElseIf Button8Contactor.Text = "OFF" Then
        SerialPort8.WriteLine("#8m000%")

            Button8Contactor.Text = "ON"
            Button8Contactor.selected = False
            Button8Contactor.TextAlign =
ContentAlignment.MiddleRight
            Button8Contactor.BackColor =
Color.FromArgb(45, 54, 76)
            Button8Contactor.Iconimage =
Image.FromFile(OnSymbol)
            Label8ContactorStatsValue.Text = "OFF"
            Label8ContactorStatsValue.ForeColor =
Color.FromArgb(234, 86, 97)
        End If
End Sub

Private Sub LoadData1()
    Try
        dt1 = New DataTable
```

```
da = New  
MySql.Data.MySqlClient.MySqlDataAdapter("SELECT  
1_flow, 1_level, 1 stepper FROM data", con)  
comBuilderDB = New  
MySql.Data.MySqlClient.MySqlCommandBuilder(da)  
da.Fill(dt1)  
DataViewID1.DataSource = dt1  
  
'Header text  
DataViewID1.Columns(0).HeaderText = "Flow"  
DataViewID1.Columns(1).HeaderText = "Level"  
DataViewID1.Columns(2).HeaderText =  
"Stepper"  
  
'Properties  
DataViewID1.Columns(0).ReadOnly = True  
DataViewID1.Columns(1).ReadOnly = True  
DataViewID1.Columns(2).ReadOnly = True  
  
Me.DataViewID1.AlternatingRowsDefaultCellStyle.Bac  
kColor = Color.AliceBlue  
  
Me.DataViewID1.AutoResizeColumns(DataGridViewAu  
toSizeColumnsMode.AllCells)  
  
Me.DataViewID1.FirstDisplayedScrollingRowIndex =  
Me.DataViewID1.RowCount - 1  
Catch ex As Exception  
  
End Try  
End Sub  
  
Private Sub LoadData2()  
Try
```

```
        dt2 = New DataTable
        da = New MySql.Data.MySqlClient.MySqlDataAdapter("SELECT
2_temperature, 2_temperature_set FROM data", con)
        comBuilderDB = New MySql.Data.MySqlClient.MySqlCommandBuilder(da)
        da.Fill(dt2)
        DataViewID2.DataSource = dt2

        'Header text
        DataViewID2.Columns(0).HeaderText =
    "Temperature"
        DataViewID2.Columns(1).HeaderText = "Temp
Set"

        'Properties
        DataViewID2.Columns(0).ReadOnly = True
        DataViewID2.Columns(1).ReadOnly = True

Me.DataViewID2.AlternatingRowsDefaultCellStyle.Bac
kColor = Color.AliceBlue

Me.DataViewID2.AutoResizeColumns(DataGridViewAu
toSizeColumnsMode.AllCells)

Me.DataViewID2.FirstDisplayedScrollingRowIndex =
Me.DataViewID2.RowCount - 1
    Catch ex As Exception

    End Try
End Sub

Private Sub LoadData3()
    Try
```

```
dt3 = New DataTable
da = New MySql.Data.MySqlClient.MySqlDataAdapter("SELECT
3_temperature, 3_temperature_set FROM data", con)
comBuilderDB = New MySql.Data.MySqlClient.MySqlCommandBuilder(da)
da.Fill(dt3)
DataViewID3.DataSource = dt3

'Header text
DataViewID3.Columns(0).HeaderText =
"Temperature"
DataViewID3.Columns(1).HeaderText = "Temp
Set"

'Properties
DataViewID3.Columns(0).ReadOnly = True
DataViewID3.Columns(1).ReadOnly = True

Me.DataViewID3.AlternatingRowsDefaultCellStyle.Bac
kColor = Color.AliceBlue

Me.DataViewID3.AutoResizeColumns(DataGridViewAu
toSizeColumnsMode.AllCells)

Me.DataViewID3.FirstDisplayedScrollingRowIndex =
Me.DataViewID3.RowCount - 1
Catch ex As Exception

End Try
End Sub

Private Sub LoadData4()
Try
```

```
dt4 = New DataTable
da = New MySql.Data.MySqlClient.MySqlDataAdapter("SELECT
4_flow, 4_temperature, 4_mov FROM data", con)
comBuilderDB = New MySql.Data.MySqlClient.MySqlCommandBuilder(da)
da.Fill(dt4)
DataViewID4.DataSource = dt4

'Header text
DataViewID4.Columns(0).HeaderText = "Flow"
DataViewID4.Columns(1).HeaderText =
"Temperature"
DataViewID4.Columns(2).HeaderText = "MOV"

'Properties
DataViewID4.Columns(0).ReadOnly = True
DataViewID4.Columns(1).ReadOnly = True
DataViewID4.Columns(2).ReadOnly = True

Me.DataViewID4.AlternatingRowsDefaultCellStyle.Bac
kColor = Color.AliceBlue

Me.DataViewID4.AutoResizeColumns(DataGridViewAu
toSizeColumnsMode.AllCells)

Me.DataViewID4.FirstDisplayedScrollingRowIndex =
Me.DataViewID4.RowCount - 1
Catch ex As Exception

End Try
End Sub

Private Sub LoadData5()
```

```
Try
    dt5 = New DataTable
    da = New MySql.Data.MySqlClient.MySqlDataAdapter("SELECT
5_flow, 5_temperature, 5_vsd FROM data", con)
    comBuilderDB = New MySql.Data.MySqlClient.MySqlCommandBuilder(da)
    da.Fill(dt5)
    DataViewID5.DataSource = dt5

    'Header text
    DataViewID5.Columns(0).HeaderText = "Flow"
    DataViewID5.Columns(1).HeaderText =
    "Temperature"
    DataViewID5.Columns(2).HeaderText = "VSD"

    'Properties
    DataViewID5.Columns(0).ReadOnly = True
    DataViewID5.Columns(1).ReadOnly = True
    DataViewID5.Columns(2).ReadOnly = True

Me.DataViewID5.AlternatingRowsDefaultCellStyle.Bac
kColor = Color.AliceBlue

Me.DataViewID5.AutoResizeColumns(DataGridViewAu
toSizeColumnsMode.AllCells)

Me.DataViewID5.FirstDisplayedScrollingRowIndex =
Me.DataViewID5.RowCount - 1
    Catch ex As Exception

        End Try
End Sub
```

```
Private Sub LoadData6()
    Try
        dt6 = New DataTable
        da = New MySql.Data.MySqlClient.MySqlDataAdapter("SELECT
6_temperature_in, 6_temperature_out, 6_flow, 6_fan
FROM data", con)
        comBuilderDB = New MySql.Data.MySqlClient.MySqlCommandBuilder(da)
        da.Fill(dt6)
        DataViewID6.DataSource = dt6

        'Header text
        DataViewID6.Columns(0).HeaderText = "Temp
In"
        DataViewID6.Columns(1).HeaderText = "Temp
Out"
        DataViewID6.Columns(2).HeaderText = "Flow"
        DataViewID6.Columns(3).HeaderText = "Fan"

        'Properties
        DataViewID6.Columns(0).ReadOnly = True
        DataViewID6.Columns(1).ReadOnly = True
        DataViewID6.Columns(2).ReadOnly = True
        DataViewID6.Columns(3).ReadOnly = True

Me.DataViewID6.AlternatingRowsDefaultCellStyle.Bac
kColor = Color.AliceBlue

Me.DataViewID6.AutoResizeColumns(DataGridViewAu
toSizeColumnsMode.AllCells)

Me.DataViewID6.FirstDisplayedScrollingRowIndex =
Me.DataViewID6.RowCount - 1
```

Catch ex As Exception

End Try

End Sub

Private Sub LoadData7()

Try

 dt7 = New DataTable

 da = New

 MySql.Data.MySqlClient.MySqlDataAdapter("SELECT
 7_temperature_in, 7_temperature_out, 7_fan FROM
 data", con)

 comBuilderDB = New

 MySql.Data.MySqlClient.MySqlCommandBuilder(da)

 da.Fill(dt7)

 DataViewID7.DataSource = dt7

'Header text

 DataViewID7.Columns(0).HeaderText = "Temp

In"

 DataViewID7.Columns(1).HeaderText = "Temp

Out"

 DataViewID7.Columns(2).HeaderText = "Fan"

'Properties

 DataViewID7.Columns(0).ReadOnly = True

 DataViewID7.Columns(1).ReadOnly = True

 DataViewID7.Columns(2).ReadOnly = True

Me.DataViewID7.AlternatingRowsDefaultCellStyle.Bac
kColor = Color.AliceBlue

Me.DataViewID7.AutoResizeColumns(DataGridViewAu
toSizeColumnsMode.AllCells)

```
Me.DataViewID7.FirstDisplayedScrollingRowIndex =
Me.DataViewID7.RowCount - 1
    Catch ex As Exception

End Try
End Sub

Private Sub LoadData8()
    Try
        dt8 = New DataTable
        da = New
MySql.Data.MySqlClient.MySqlDataAdapter("SELECT
8_temperature, 8_pressure FROM data", con)
        comBuilderDB = New
MySql.Data.MySqlClient.MySqlCommandBuilder(da)
        da.Fill(dt8)
        DataViewID8.DataSource = dt8

        'Header text
        DataViewID8.Columns(0).HeaderText =
"Temperature"
        DataViewID8.Columns(1).HeaderText =
"Pressure"

        'Properties
        DataViewID8.Columns(0).ReadOnly = True
        DataViewID8.Columns(1).ReadOnly = True

Me.DataViewID8.AlternatingRowsDefaultCellStyle.Bac
kColor = Color.AliceBlue

Me.DataViewID8.AutoResizeColumns(DataGridViewAu
toSizeColumnsMode.AllCells)
```

```
Me.DataViewID8.FirstDisplayedScrollingRowIndex =  
Me.DataViewID8.RowCount - 1  
Catch ex As Exception
```

```
End Try  
End Sub
```

```
Sub PlayLoopingBackgroundSoundFile()  
    My.Computer.Audio.Play(MusicFile,  
    AudioPlayMode.WaitToComplete)  
End Sub
```

```
Private Sub TimerAlarm_Tick(sender As Object, e As  
EventArgs) Handles TimerAlarm.Tick
```

```
End Sub
```

```
Private Sub ButtonReset_Click(sender As Object, e As  
EventArgs) Handles ButtonReset.Click  
If Alarm = "PumpLevelMin" Then  
    SerialPort1.WriteLine("#1o001%") 'Override  
    SerialPort1.WriteLine("#1n000%") 'Pump  
    SerialPort1.WriteLine("#1m100%") 'Stepper  
End If
```

```
If Alarm = "PumpLevelMax" Then  
    SerialPort1.WriteLine("#1o001%") 'Override  
    SerialPort1.WriteLine("#1n001%") 'Pump  
    SerialPort1.WriteLine("#1m000%") 'Stepper  
End If
```

```
If Alarm = "HeaterTempMin" Then  
    SerialPort4.WriteLine("#4n001%") 'override naja  
    SerialPort4.WriteLine("#4m000%") 'mov naja
```

```
End If

If Alarm = "HeaterTempMax" Then
    SerialPort4.WriteLine("#4n001%") 'override naja
    SerialPort4.WriteLine("#4m100%") 'mov
End If

If Alarm = "CoolerTempMin" Then
    SerialPort5.WriteLine("#5n001%") 'override
    enver
        SerialPort5.WriteLine("#5m000%") 'vsd
    End If

    If Alarm = "CoolerTempMax" Then
        SerialPort5.WriteLine("#5n001%") 'override
    enver
        SerialPort5.WriteLine("#5m100%") 'vsd
    End If
End Sub

Private Sub ButtonAbout_Click(sender As Object, e
As EventArgs) Handles ButtonAbout.Click
    About.Show()
End Sub

End Class
```

BIODATA PENULIS



Penulis yang bernama Wahyu Adi Kurniawan dilahirkan di Jombang pada tanggal 18 Mei 1997. Penulis merupakan anak pertama dari dua bersaudara dari orang tua Bapak Fuad dan Ibu Anik Yuliatin. Penulis menempuh pendidikan formal di RA Mambaul Ulum Corogo, SDN Alang-Alang Caruban II, SMPN 2 Jombang, dan SMAN Mojoagung. Kemudian penulis melanjutkan ke jenjang yang lebih tinggi yaitu di Institut Teknologi Sepuluh Nopember Surabaya, tepatnya di Departemen Teknik Instrumentasi

Fakultas Vokasi di tahun 2015 dan terdaftar dengan NRP 10 51 15 000 00 037.

Di Departemen Teknik Instrumentasi ini penulis aktif di beberapa organisasi kemahasiswaan seperti di BEM ITS Wahana Juang sebagai Staff Kementerian Hubungan Luar periode 2016/2017 dan di Himpunan Mahasiswa Teknik Instrumentasi (HIMATEKINS) FV ITS sebagai Ketua periode 2017/2018. Pada laporan ini, penulis telah melaksanakan Tugas Akhir Program Studi DIII Teknologi Instrumentasi Departemen Teknik Instrumentasi FV ITS.

Selesainya Tugas Akhir ini diajukan penulis untuk memenuhi syarat memperoleh gelar Ahli Madya di Departemen Teknik Instrumentasi Fakultas Vokasi Institut Teknologi Sepuluh Nopember Surabaya.

wahyuadikurniawan@live.com
facebook.com/ayungavis
twitter.com/ayungavis
instagram.com/ayungavis