TUGAS AKHIR – EE 184801

PERANCANGAN DIAGRAM *LADDER* DAN HMI UNTUK *AUTOMATIC CLIMATE CONTROL* PADA MOBIL OTONOM MENGGUNAKAN METODE HUFFMAN

Iqbal Muhammad Santoso
NRP 07111640000094

Dosen Pembimbing
Eka Iskandar, S.T., M.T.

DEPATERMEN TEKNIK ELEKTRO
Fakultas Teknologi Elektro
Institut Teknologi Sepuluh Nopember
Surabaya 2020

Halaman ini sengaja dikosongkan

TUGAS AKHIR – EE 184801

PERANCANGAN DIAGRAM *LADDER* DAN HMI UNTUK *AUTOMATIC CLIMATE CONTROL* PADA MOBIL OTONOM MENGGUNAKAN METODE HUFFMAN

Iqbal Muhammad Santoso
NRP 07111640000094

Dosen Pembimbing
Eka Iskandar, S.T., M.T.

DEPARTEMEN TEKNIK ELEKTRO
Fakultas Teknologi Elektro
Institut Teknologi Sepuluh Nopember
Surabaya 2020

Halaman ini sengaja dikosongkan

FINAL PROJECT – EE 184801

LADDER DIAGRAM AND HMI DESIGN FOR AUTOMATIC CLIMATE CONTROL ON AUTONOMOUS CAR USING HUFFMAN METODE

Iqbal Muhammad Santoso
NRP 07111640000094

Supervisor
Eka Iskandar, ST., MT.

ELECTRICAL ENGINEERING DEPARTMENT
Faculty of Electrical Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2020

Halaman ini sengaja dikosongkan

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul "**Perancangan Diagram *Ladder* untuk *Automatic Climate Control* pada Mobil Otonom Menggunakan Metode Huffman**" adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, Juli 2020

Iqbal Muhammad Santoso
NRP. 0711 16 4000 0094

Halaman ini sengaja dikosongkan

PERANCANGAN DIAGRAM LADDER DAN HMI UNTUK AUTOMATIC CLIMATE CONTROL PADA MOBIL OTONOM MENGGUNAKAN METODE HUFFMAN

TUGAS AKHIR

Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik

Pada

Bidang Studi Teknik Sistem Pengaturan
Departemen Teknik Elektro
Fakultas Teknologi Elektro
Institut Teknologi Sepuluh Nopember

Menyetujui :

Dosen Pembimbing

Eko Iskandar, ST., MT
NIP. 198005282008121001

SURABAYA
JULI, 2019

Halaman ini sengaja dikosongkan

PERANCANGAN DIAGRAM LADDER DAN HMI UNTUK AUTOMATIC CLIMATE CONTROL PADA MOBIL OTONOM MENGGUNAKAN METODE HUFFMAN

Iqbal Muhammad Santoso
07111640000094

Dosen Pembimbing : 1. Eka Iskandar, S.T., M.T.

ABSTRAK

Automatic Climate Control (ACC) merupakan bentuk perluasan dari *Air-conditioning systems* pada mobil. Sistem ini lebih maju, terkomputasi dan lebih otomatis. Sehingga sistem tersebut dapat mengontrol temperatur di dalam mobil secara efisien. ACC menggunakan *Programmable Logic Controller* (PLC) sebagai perangkat pemrosesannya dan menggunakan bahasa pemrograman diagram *ladder*. PLC merupakan perangkat pengontrol yang sering digunakan, karena PLC mudah untuk digunakan. Untuk konsep dan pembuatan program, akan digunakan metode Huffman. Metode Huffman merupakan salah satu metode klasik dalam perancangan sistem sekuensial yang lebih rumit. Dengan metode ini, desain yang dihasilkan memiliki jumlah relay yang lebih minimal.

Kata Kunci: *Automatic Climate Control, Human Machine Interface*, Metode Huffman, Diagram *Ladder*

Halaman ini sengaja dikosongkan

LADDER DIAGRAM AND HMI DESIGN FOR AUTOMATIC CLIMATE CONTROL ON AUTONOMOUS CAR USING HUFFMAN METHOD

Iqbal Muhammad Santoso
07111640000094

Supervisor : 1. Eka Iskandar, S.T., M.T.

ABSTRACT

Automatic Climate Control (ACC) is an extension of the air-conditioning system in cars. This sistem is more advanced, computational and more automatic. So that the system can control the temperature in the car more efficiently. ACC uses the Programmable Logic Controller (PLC) as Processing device and uses the ladder diagram as it's programming language. PLC is a control device that often used, because PLC is easy to use. For the concept and making of the program, the Huffman method will be used. The Huffman method is one of the classic method in designing a more complicated sequential system. With this method, the designs that is produced have a minimal number of relays.

Keyword: *Automatic Climate Control, Huffman Method, Human Machine Interface, Ladder Diagram.*

Halaman ini sengaja dikosongkan

KATA PENGANTAR

Segala Puji Syukur ke Hadirat Allah Subhanallahu Wa Ta'ala yang senantiasa memberikan ridha dan inayah-Nya sehingga penulis dapat menyelesaikan Tugas Akhir ini. Shalawat serta Salam kepada junjungan kita Nabi Muhammad Shallallahu Alaihi Wasallam serta para sahabat tabi' tabi'in, dengan ilmu pengetahuan yang telah diberikan sehingga penulis dapat menyelesaikan Tugas Akhir dengan judul "**Perancangan Diagram Ladder dan HMI untuk *Automatic Climate Control* pada Mobil Otonom Menggunakan Metode Huffman**".

Banyak pihak telah berperan dalam membantu selama proses studi maupun penelitian ini. Oleh karena itu, penulis menyampaikan terimakasih kepada:

1. Orang tua serta keluarga penulis yang telah memberi dukungan waktu, pembelajaran dan doa kepada penulis.
2. Bapak Eka Iskandar, S.T., M.T., yang dengan baik dan sabar dalam mengarahkan dan membimbing selama proses pengerjaan Tugas Akhir ini.
3. Tenaga Didik serta anggota Lab B105 yang telah memberikan izin pinjaman alat untuk menyelesaikan studi penulis.
4. Teman-teman Prodi Sistem Pengaturan yang selalu membantu dan mendukung untuk menyelesaikan Tugas Akhir ini.
5. Kuntoro Waskhito Adi dan Oriza Tsabitah Dhiazwanti yang memberikan semangat sehingga penulis tetap bersemangat dalam mengerjakan Tugas Akhir.
6. Kepada semua pihak, rekan, teman, para guru yang telah membantu penulis selama proses studi.

Penulis berharap Tugas Akhir ini dapat bermanfaat bagi pembaca. Saran dan kritik membangun selalu penulis harapkan untuk kelengkapan pengetahuan penulis terkait Tugas Akhir ini.

Surabaya, Juli
2020

Penulis

Halaman ini sengaja dikosongkan

DAFTAR ISI

HALAMAN JUDUL
PERNYATAAN KEASLIAN TUGAS AKHIR ..... **vii**
ABSTRAK ..... **i**
ABSTRACT ..... **iii**
KATA PENGANTAR ..... **v**
DAFTAR ISI ..... **vii**
DAFTAR GAMBAR ..... **ix**
DAFTAR TABEL ..... **x**
BAB 1 ..... **1**
1.1 Latar Belakang 1
1.2 Perumusan Masalah 2
1.3 Batasan Masalah 2
1.4 Tujuan 2
1.5 Metode Penelitian 2
1.6 Sistematika Penulisan 3
1.7 Relevansi 4
BAB 2 ..... **5**
2.1. Automatic Climate Control (ACC) 5
2.1.1. Konsep kerja *Air-Conditioner* 5
2.1.2. Komponen *Air-Conditioner* 6
2.2. *Programmable Logic Diagram* 7
2.2.1. Fungsi PLC 8
2.2.2. Komponen Dasar PLC 8
2.2.3. Bahasa Pemograman: *Ladder Diagram* 9
2.3. Metode Huffman 10
2.3.1. Langkah-langkah Metode Huffman 11
2.4. *Human Machine Interface* 11
2.5. TIA Portal 12
BAB 3 ..... **13**
3.1. Merancang Plant 13
3.1.1. Persamaan *Transfer Function* 14
3.1.2. TIA Portal Library 15
3.1.2.1. Function Block (FB) “SimMultizone” 15

3.1.2.1. Data Block (DB) “SimMultizone” 16
3.1.3. Merangkai Block Plant dan PID 17
3.1.4. PID_Temp 17
3.2. Mendesain Ladder dengan Metode Huffman 20
3.2.1. Menentukan State, Input, dan Output 20
3.2.2. Membuat Primitive flow table 22
3.2.3. Membuat Merged Flow Table.....23
3.2.4. Mencari Persamaan Relay dan Output 23
3.2.5. Merancang Diagram *Ladder* 27
3.3. Modifikasi Ladder Sesuai dengan Kebutuhan Sistem.....28
3.3.1. Mengatur Input Sensor Temp 28
3.3.2. Mengatur Range Temperatur 29
3.3.3. Mengatur tombol input temperatur.....30
3.3.4. Mengatur Tombol Input Mode.....30
3.3.5. Mengatur Batas Atas dan Bawah Input Temperatur....32
3.4. Merancang HMI 33
BAB 4 35
4.1. Menghubungkan TIA Portal dengan PLC-Simulator 35
4.2. Simulasi Sistem.....36
4.2.1. Menentukan Setpoint Temperatur 38
4.2.2. Tuning PID_Temp 38
4.2.3. Uji Coba dengan HMI 38
BAB 5 41
5.1. Kesimpulan 41
5.2. Saran.....41
DAFTAR PUSTAKA.....42
LAMPIRAN 43

DAFTAR GAMBAR

Gambar 2.1 Siklus dasar sistem *air-conditioning* 5
Gambar 3.1 Block Diagram Sistem Multizone 13
Gambar 3.2 Block Diagram Sistem yang digunakan 14
Gambar 3.3 Block PID_Temp 18
Gambar 3.4 Setting tipe kontroller PID_Temp 19
Gambar 3.5 Setting parameter input dan output 19
Gambar 3.6 Setting limit nilai proses PID_Temp 20
Gambar 3.7 State diagram sistem 22
Gambar 3.8 Ladder diagram untuk mengontrol relay Y 28
Gambar 3.9 Ladder diagram untuk output 'C' dan 'H' 28
Gambar 3.10 Ladder diagram untuk input sensor temp 29
Gambar 3.11 Diagram ladder untuk batas setpoint antar mode 29
Gambar 3.12 Block counter 30
Gambar 3.13 Block "Move" 30
Gambar 3.14 Ladder Diagram "Cool Mode" 31
Gambar 3.15 Ladder Diagram "Heat Mode" 31
Gambar 3.16 Diagram untuk mengatur batas maksimum dan minimum setpoint 32
Gambar 3.17 Tampilan default HMI 33
Gambar 3.18 Tampilan HMI yang telah didesain 34
Gambar 4.1 Pengaturan ethernet PLC 35
Gambar 4.2 Menghubungkan pada PLCSim 35
Gambar 4.3 Tampilan PLCSim 36
Gambar 4.4 Tampilan *project view* PLCSIM 37
Gambar 4.5 Tampilan HMI selama simulasi 39
Gambar 4.5 Tampilan HMI selama simulasi 39
Gambar 4.5 Tampilan HMI selama simulasi 39
Gambar 4.5 Tampilan HMI selama simulasi 39

DAFTAR TABEL

Tabel 3.1 FB "SimMultizone" 15
Tabel 3.2 Data Block Plant 16
Tabel 3.3 Data Block Zones 16
Tabel 3.4 State, input, dan output 21
Tabel 3.5 Primitive flow table 23
Tabel 3.6 Merge flow table 23
Tabel 3.7 Posisi nilai dalam k-map 24
Tabel 3.8 Tabel karakteristik SR flip-flop 24
Tabel 3.9 K-map set relay Y 24
Tabel 3.10 K-map reset relay Y 25
Tabel 3.11 K-map output 'C' 26
Tabel 3.12 K-map output 'H' 26
Tabel 3.13 K-map output 'C' setelah diberi 'Start' 27
Tabel 3.14 K-map output 'H' setelah diberi 'Start' 27
Tabel 4.1 Tags nilai integer 37

BAB 1
PENDAHULUAN

Pada bab ini akan dipaparkan tentang latar belakang penelitian, permasalahan yang akan diselesaikan, tujuan dari penelitian, metodologi yang digunakan, sistematika penulisan serta relevansi penelitian ini terhadap peelitian sejenis atau lebih rumit di masa mendatang.

1.1 Latar Belakang

Dengan keadaan pemanasan global dan iklim tropis di Indonesia, pengadaan *air conditioner* dalam suatu kendaraan merupakan suatu kebutuhan. Namun dengan semakin parahnya pemanasan global, sistem *air conditioner* biasa kurang cukup untuk memberikan suasana sejuk pada penumpang kendaraan. *Automatic climate control* (ACC) merupakan perluasan dari sistem *air conditioner*. ACC tidak hanya dapat memberikan udara dingin, tapi juga dapat menghangatkan pula. Selain itu pengaturan temperatur dilakukan berdasarkan pada keadaan diluar pula. Sehingga dapat memberikan kenyamanan pada pengguna kendaraan.

Dengan menggunakan ACC akan mempermudah pengguna untuk mengatur temperatur dalam kendaraan secara manual. Pada dasarnya, sistem hanya akan butuh mengetahui temperatur yang dinginkan pengguna didalam kendaraan, dan sistem akan bekerja sendiri secara otomatis, menggunakan komputer. Sistem juga akan mengatur temperatur berdasarkan keadaan diluar kendaraan. Sehingga saat temperatur diluar terlalu dingin, maka ACC akan mengurangi kecepatan kipas, dan begitu pula sebaliknya.

Programable Logic Controller (PLC) merupakan salah satu perangkat yang sering digunakan sebagai perangkat kontrol. Beberapa alasannya karena PLC sangat mudah diprogram dan perangkatnya mempunyai umur pakai yang panjang. Selain alasan tersebut, PLC juga dapat digunakan untuk skala proses yang kecil maupun besar, dan semua proses itu dapat saling berkomunikasi sehingga dapat menghasilkan proses yang diinginkan dan aman untuk digunakan.

Pemrograman pada PLC dinamakan dengan *ladder diagram*, ladder yang berarti anak tangga karena memang memiliki bentuk seperti anak tangga. Untuk pembuatan *ladder diagram* itu sendiri memiliki beberapa metode yang beberapa metodenya adalah *Huffman, Cascade, State diagram* dan masih ada beberapa metode lainnya. Secara singkat,

penggunaan metode digunakan untuk mempermudah pembuatan dan proses troubleshooting nantinya. Sehingga pembuat dan pengguna nantinya bisa memiliki pemahaman yang selaras tentang *ladder diagram* yang sudah dibuat.

1.2 Perumusan Masalah

Automatic Climate Control menggunakan sekuensial yang tidak 1 (satu) arah. Beberapa langkah dapat berulang atau berubah ke langkah yang lain. Sehingga sekuensial sistem dapat menjadi cukup kompleks. Dengan metode Huffman, kompleksitas sekuensial dari sistem akan dapat diminimumkan.

1.3 Batasan Masalah

Perlu diperhatikan Batasan dalam melakukan penelitian ini, antara lain sebagai berikut:

1. Sistem yang akan diatur hanyalah Pengaturan suhu
2. Mode Pengaturan air-conditioning hanya akan terdiri dari 2 (dua), yaitu mode cooling dan heating
3. Dikarenakan kendala *lockdown* selama pandemik covid-19, maka penelitian ini dilakukan sepenuhnya dengan simulasi.

1.4 Tujuan

Tujuan dari tugas akhir ini adalah membuat desain ladder diagram untuk pengaturan temperatur *Automatic Climate Control* dengan menggunakan metode Huffman.

1.5 Metode Penelitian

Pada Tugas Akhir ini digunakan suatu tahapan sistematik sehingga tujuan dari tugas akhit ini dapat tercapai. Tahapan yang akan diterapkan untuk penelitian tugas akhir kali ini adalah sebagai berikut:

1. Studi Literatur

   Pada tahap awal penelitian ini, dilakukan studi literatur untuk mempelajari dan mengkaji teori-teori serta mengumpulkan data penelitian yang relevan dan terpercaya untuk keabsahan Tugas Akhir ini. Literatur yang digunakan dalam penelitian ini memiliki batasan yaitu literatur yang digunakan harus bersumber dari paper, jurnal,

buku, maupun artikel yang berasal dari badan pemerintahan atau institusi akademik terpercaya.

2. Pemahaman urutan kinerja tiap-tiap perangkat *Automatic Climate Control*

   Dalam Kontrol Sekuensial menggunakan metode Huffman, urutan kinerja dari tiap-tiap perangkat sangatlah penting. Dalam tahap ini, akan dirancang skema urutan kinerja perangkat dari sistem berdasarkan dari kinerja perangkat pada *Automatic Climate Control* yang didapatkan pada study literatur.

3. Perancangan diagram *ladder* sesuai metode Huffman

   Dilakukan perancangan diagram ladder dengan metode Huffman berdasarkan dari skema dan urutan kinerja perangkat pada *Automatic Climate Control*, serta berdasarkan input output dari sistem.

4. Melakukan Uji Coba

   Pengujicoba akan dilakukan berdasarkan dari hasil ladder diagram yang telah didesain, dengan melihat apakah hasil output sesuai dengan yang diinginkan untuk memastikan apakah ladder yang telah dirangcang telah berjalan dengan benar.

5. Membuat *Human Machine Interface*

   Setelah uji coba berhasil, akan dirancang sebuah interface untuk mempermudah pengoperasian dan monitoring sistem. Interface akan menggunakan build in dari aplikasi TIA Portal dari Siemens.

6. Penyusunan Laporan Tugas Akhir

   Penyusunan laporan adalah tahap terakhir dalam penelitian ini, berisi tentang tugas akhir yang dikerjakan, meliputi pendahuluan, studi literatur, perancangan dan pembuatan sistem, pengujian, dan Analisa serta penutup.

1.6 Sistematika Penulisan

Sistematika penulisan yang diterapkan pada buku tugas akhir ini terbagi menjadi lima bab, yaitu:

BAB I PENDAHULUAN

Pada bab pendahuluan membahas tentang latar belakang, permasalahan yang ada, batasan masalah, tujuan pelaksanaan

tugas akhir, metodologi pelaksanaan, sistematika penulisan laporan tugas akhir, dan relevansi.

BAB II TEORI PENUNJANG

Menjelaskan tentang tinjauan pustaka, konsep dasar dari *Automatic Climate Control*, *Programmable Logic Controller*, Metode Huffman, dan *Human Machine Interface*.

BAB III PERANCANGAN SISTEM

Bab ini membahas perancangan desain diagram *ladder* berdasarkan dari kinerja perangkat pada ACC menggunakan metode Huffman dan pemograman *interface*.

BAB IV HASIL DAN ANALISA DATA

Bab ini berisi hasil simulasi dari diagram *ladder* yang telah didesain.

BAB V PENUTUP

Setelah melakukan pengujian dan analisa pada bab 4, didapatkan kesimpulan yang akan dijabarkan pada bab 5 yaitu kesimpulan dan saran serta pemberian saran untuk penelitian yang akan menggunakan topik serupa dengan Tugas Akhir ini.

1.7 Relevansi

Hasil dari Tugas Akhir ini diharapkan dapat diimplementasikan untuk menyelesaikan permasalahan untuk kontrol dari ACC dan diagram *ladder* khususnya dengan metode Huffman. Penelitian ini diharapkan menjadi referensi bagi peneliti lain yang akan meneliti baik ACC maupun ladder diagram dengan metode Huffman.

BAB 2
TEORI PENUNJANG

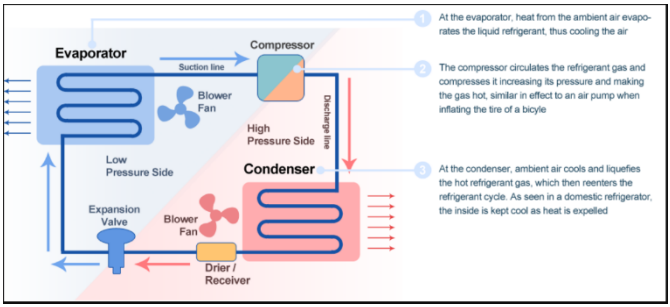
Pada bab ini, menjelaskan tentang dasar-dasar teori yang digunakan dalam melakukan penelitian ini, yaitu mencakup penjelasan secara umum tentang *Automatic Climate Control* (ACC), komponen dari ACC, konsep kerja dari ACC, diagram *ladder*, metode Huffman, serta *Human Machine Interface* (HMI).

2.1. Automatic Climate Control (ACC)

Automatic Climate Control merupakan sebuah Kembangan dari heating ventilating and air-conditioner (HVAC) secara digital. Berbeda dengan HVAC biasa, automatic climate control mengatur tidak hanya suhu ruangan, tetapi juga mengatur kelembapan kabin. Selain itu, automatic climate control juga memiliki fitur seperti, pengatuan tingkat karbondioksida dan kualitas udara.

2.1.1. Konsep kerja *Air-Conditioner*

Konsep cara kerja dari AC sama dengan konsep kinerja dari pendingin seperti kulkas. Kinerja sistem ini berupa sebuah siklus yang bekerja secara bergantian.



Gambar 2.1 Siklus dasar sistem *air-conditioning*

Sesuai dengan Gambar 2.1, pertama kompressor akan menerima refrigeran dalam bentuk gas bertekanan rendah dan bertemperatur rendah, yang kemudian akan dikompres menjadi gas bertekanan tinggi dan bertemperatur tinggi. Kemudian refrigeran akan dikirimkan dari kompressor menuju kondenser.

Pada kondenser, sebuah kipas akan meniupkan udara dingin pada

refrigeran dan mengubah refrigeran dari gas bertekanan tinggi dan bertemperatur tinggi, menjadi cairan bertekanan tinggi. Cairan refrigeran ini kemudian akan dikirimkan menuju *receiver/dryer*.

Pada *receiver/dryer* ini, refrigeran akan di-*filter* dari uap-uap yang dapat menkontaminasi refrigeran. Kemudia uap-uap tersebut akan diserap oleh receiver/dryer.

Setelah refrigeran telah difilter oleh receiver/dryer, refrigeran akan bergerak menuju *expansion valve*. Dimana refrigeran akan diubah dari cairan bertekanan tinggi menjadi cairan bertekanan rendah. Yang kemudian akan dikirimkan pada *evaporator*.

Pada evaporator refrigeran akan dididihkan dan diubah menjadi gas bertekanan rendah dan menyerah panas pada evaporator. Melalui proses itu, dinding pada evaporator akan mendingin dan kipas pada evaporator akan meniupkan udara pada evaporator yang telah menjadi dingin menuju kabin kendaraan.

Setelah melalui evaporator, refrigeran akan dikirimkan kembali pada kompressor, untuk memulai kembali siklus tersebut.

2.1.2. Komponen *Air-Conditioner*

Terdapat beberapa komponen dalam *air-conditioning system*. Yaitu:

a. *Compressor*

kompressor berfungsi untuk menghisap uap refrigeran dari evaporator dan menekan (mengkompres) sehingga tekanan dan temperatur refrigeran menjadi lebih tinggi. Kompressor juga berfungsi untuk mengalirkan refrigeran ke seluruh sistem pendingin.

b. *Condenser*

kondenser berfungsi mengubah atau mendinginkan gas bertekanan tinggi agar berubah menjadi cairan bertekanan tinggi.

c. *Receiver/Dryer*

Receiver/dryer berfungsi untuk memfilter dan menyerap. uap-uap yang dapat mengkontaminasi atau mengotori refrigeran.

d. *Expansion Valve*

Expansion valve berfungsi untuk mengatur jumlah dan mengubah refrigeran dari cairan bertekenan tinggi menjadi cairan bertekanan rendah.

e. Evaporator

Evaporator berfungsi untuk menampung uap digin yang dihasilkan selama proses pendinginan. Secara mendasar, fungsi dari evaporator adalah kebalikan dari kondenser. Bila kondenser berfungsi untuk menyerap panas dan dibuang ke udara, evaporator berfungsi untuk menyerap panas dari udara dan mengirimkan pada refrigeran. Evaporator akan menyerap panas yang dibawa oleh udara sehingga udara seletah melewati evaporator menjadi dingin. Yang kemudian udara dingin itu akan ditiupkan kedalam kabin kendaran.

2.2. *Programmable Logic Diagram*

Programable Logic Controller (PLC) adalah sistem berbasis mikroprosesor yang menggunakan memori yang dapat diprogram untuk menyimpan instruksi dan mengimplementasikan fungsi-fungsi seperti fungsi logika, pengurutan, perwaktuan pencacahan dan aritmarika untuk mengontrol mesin dan proses, serta dirancang untuk dioperasikan oleh insinyur yang mungkin memiliki kemampuan dan pengetahuan terbatas mengenai komputer dan bahasa pemograman.

PLC pada dasarnya adalah sistem komputer digital yang mampu melaksanakan proses kerja yang kompleks. Penggunaan PLC dilingkungan industri terhitung lebih praktis dan efisien daripada sistem kontrol konvensional. Saat terjadi perubahan proses, pengguna hanya perlu mengubah instruksi pada program yang dimasukan. Sedangkan pada kontrol konvensional, dirasa tidak praktis, kompleks, dan membutuhkan *rewiring*. Selain itu, dalam hal pengabelan PLC mudah untuk dimodifikasi. PLC memiliki kemampuan unutk mengontrol sistem proses mesin yang kompleks seperti *Human Machine Interface* (HMI) dengan kecepatan respon yang tinggi dan efisien.

PLC biasanya digunakan di industri, dan dapat bertahan dari kelembapan, debu, getaran, dan temperature ekstrem. Beberapa PLC bersifat modular, sehingga bila suatu bagian rusak, hanya perlu diganti bagiannya. Tetapi ini hanya berlaku untuk PLC jenis lama. PLC modern relative lebih kecil, dan disimpan didalam sebuah kotak. Sehingga lebih kecil resiko untuk menerima kerusakan dari kelembapan atau debu. Tetapi, bila terjadi kerusakan, maka seluruh bagian harus diganti.

2.2.1. Fungsi PLC

PLC dirancang untuk menggantikan relay sekuensial dalam suatu sistem kontrol. Sistem kendali tradisional yang masih menggunakan relay menjadi terlalu sulit untuk melakukan pemeliharaan dan perbaikan bila terjadi masalah, ataupun bila terdapat perubahan pada suatu sistem kontrolnya. Dikarenakan kompleksitas dari sistem kendali tradisional dengan relay, pemeliharaan atau perbaikan akan membuat sistem untuk dilucuti dan diatur ulang, sehingga memakan waktu dan biaya. Karena PLC yang dapat diprogram, modifikasi sistem dapat dilakukan dengan mudah.

Selain untuk menggantikan relay sekuensial dalam mengontrol suatu sistem, PLC juga memiliki fungsi sebagai supervisory (pemantauan). Dengan begitu, user atau operator dapat memantau kinerja dari sistem, dan dapat memantau bila sistem terdapat masalah atau terdapat kerusakan. Dengan begitu kerusakan ataupun masalah yang terjadi dalam sistem dapat diselesaikan sebelum masalah tersebut menjadi lebih besar.

2.2.2. Komponen Dasar PLC

PLC tersusun atas beberapa komponen dasar yang terdiri dari:

a. CPU (Central Processing Unit)

   CPU merupakan otak dari PLC yang mengerjakan berbagai operasi, antara lain mengerjakan berbagai operasi, antara lain mengeksekusi program, menyimpan dan mengambil data dari memori, membaca kondisi/nilai input serta mengatur nilai output, memeriksa adanya kerusakan, serta melakukan komunikasi dengan perangkat lain.

b. Input

   Input merupakan bagian PLC yang berhubungan dengan perangkat luar yang memberikan masukan kepada CPU. Perangkat luar input dapat berupa tombol, sensor atau perangkat lain.

c. Output

   Output merupakan bagian PLC yang berhubungan dengan perangkat luar yang memberikan keluaran dari CPU. Perangkat luar output dapat berupa lampu, katub (valve), motor, dan perangkat-perangkat lain.

d. Memori

Memori adalah tempat untuk menyimpan program dan data yang akan dijalankan dan diolah oleh CPU. Dalam pembahasaan PLC, memori sering disebut sebagai file. Dalam PLC memori terdiri dari memori program untuk menyimpan program yang akan dieksekusi, memori data untuk menyimpan nilai-nilai hasil operasi CPU, nilai timer dan counter, serta memori yang menyimpan nilai kondisi input dan output. Kebanyakan PLC sekarang memiliki satuan memori dalam word (16 bit).

e. Protokol Komunikasi

Protokol komunikasi adalah fasilitas yang akan membantu CPU untuk melakukan pertukaran data dengan perangkat lain, termasuk untuk berkomunikasi dengan komputer untuk proses pemograman dan pemantauan.

Pada dasarnya sinyal yang diterima atau dibangkitkan oleh unit input/output PLC berupa sinyal digital, yang bernilai biner 0 atau 1. Perangkat input/output yang memiliki sinyal analog memerlukan piranti ADC (Analog to Digital Converter) atau DAC (Digital to Analog Converter) agar dapat berhubungan ke PLC. Biasanya perangkat terdapat dalam modul analog yang diproduksi pabrik pembuat PLC. Sinyal analog yang biasanya digunakan dalam PLC mengikuti standar industri.

2.2.3. Bahasa Pemograman: *Ladder Diagram*

Terdapat banyak bahasa untuk mem-program PLC. Masing-masing bahasa tersebut memiliki keuntungan dan kerugian tersendiri. Salah satu dari bahasa pemograman yang paling umum digunakan adalah ladder diagram.

Ladder diagram merupakan sebuah diagram skematik yang digunakan pada otomasi industri yang mewakili rangkaian kontrol logika. Ladder diagram terdiri dari 2 *power rail* vertikal dan horizontal logic rungs sehingga terlihat seperti seperti tangga.

Ladder diagram digunakan untuk menggambarkan rangkaian listrik dan dimaksudkan untuk menunjukan urutan kejadian, bukan hubungan kabel antar komponen. pada *ladder diagram* memungkinkan elemen-elemen elektrik dihubungkan sedemikian rupa sehingga output tidak hanya terbatas pada ketergantungan terhadap input, tetapi juga terhadap logika.

Ladder diagram memuat beberapa blok yang dapat mempresentasikan

aliran program dan fungsi, yaitu:

a. Contact

Contract dapat berupa input (saklar, push button), kontak internal variabel (relay otomatis) dan lain-lain, ada 4 macam tipe kontak, yaitu:

- Normally Open (NO), merupaka kontak yang terdapat pada ladder diagram, diaman pada saat keadaan sistem belum bekerja, kondisi kontak dalam keadaan terbuka.
- Normally Close (NC), merupakan kontak yang terdapat pada ladder diagram dimana pada saat keadaan sistem belum bekerja, kondisi kontak dalam keadaan tertutup.
- Rising edge adalah kontak yang terdapat pada ladder diagram dimana pada saat pada saat keadaan sistem mulai bekerja kondisi kontak berubaha dari lagoika "0" menjadi logika "1".
- Falling edge adalah kontak yang terdapat pada ladder diagram dimana pada saat keadaan sistem mulai bekerja kondisi kontak berubah dari logika "1" menjadi logika "0".

b. Coil

Coil secara umum merupakan output. Terdapat beberapa tipe coil, yaitu:

- Coil
- Negatif coil
- SET coil
- RESET coil

2.3. Metode Huffman

Metode Huffman merupakan salah satu dari beberapa metode yang dapat digunakan untuk merancang diagram *ladder*. Metode ini merupakan salah satu mode klasik yang digunakan untuk merancang sistem sekuensial. Pada dasarnya, sistem ini lebih rumit dalam perancangan dan perhitungannya. Tetapi, dengan menggunakan metode Huffman, desain dihasilkan dapat memiliki relay yang lebih minimal. Tujuan utama dari penggunaan metode Huffman untuk merancang diagram *ladder* adalah untuk meminimalisir jumlah relay yang digunakan pada diagram *ladder*.

2.3.1. Langkah-langkah Metode Huffman

Terdapat 6 langkah dalam mendesain sebuah ladder diagram dengan menggunakan metode huffman, yaitu:

a. Menentukan state, input, dan output
b. Membuat primitive flow table
c. Membuat merge flow table
d. Menentukan fungsi eksitasi dan output
e. Menambahkan sinyal start (opsional)
f. Mendesain ladder diagram ladder berdasarkan fungsi eksitasi dan output.

2.4. *Human Machine Interface*

Human machine interface (HMI) adalah suatu komponen dari sebuah perangkat yang dapat berinteraksi dengan manusia. *Interface* terdiri dari perangkat kersa dan perangkat lunak yang dapat mengartikan input dari pengguna sebagai sinyal untuk mesin, dan sebagai gantinya, dapat memberikan hasil output pada user. Teknologi HMI dapat digunakan diberbagai industri seperti elektronik, militer, medikal, dan lain sebagainya. HMI membantu dalam integrasi manusia dengan sistem teknologi yang kompleks.

Pada HMI. Interaksi terdiri dari dua tipe, yaitu manusia pada mesin, dan mesin pada manusia. Dikarenakan teknologi HMI yang berada dimana-mana, interface dapat berada di sensor gerak, keyboard dan sejenisnya, pengenalan suara dan interaksi-interaksi lain yang dapat bertukar informasi menggunakan pengelihatan, suara, panas, dan kognitif dan fisikal mode yang lain yang dapat dijadikan bagian HMI.

Walau dianggap sebagai teknologi yang mandiri, teknologi HMI dapat digunakan sebagai adapter untuk teknologi lain. Dasar dari pembuatan HMI sangat bergantung pada pemahaman manusia secara fisik, perilaku, dan kemampuan mental. Dengan kata lain ergonomik membentuk dasar dari HMI. Selain mengembangkan pengalaman dan efisiensi pengguna, HMI dapat memberikan kesempatan yang unik untuk pengaplikasian, pembelajaran, dan rekreasi. Faktanya, HMI sangat membantu dalam perolehan kemampuan untuk pengguna. HMI yang baik dapat memberikan interaksi realistis dan natural dengan perangkat eksternal.

Keuntungan yang didapat dengan menggabungkan HMI adalah, pengurangan error, meningkatkan efisiensi sistem dan pengguna,

mengembangkan keandalan dan pemeliharaan, memberikan kenyamanan pada user, mengurangi pelatihan dan kemampuan yang dibutuhkan, mengurangi stress fisik dan mental pada pengguna, dan mengurangi saturasi tugas, mengingkatkan ekonomi dari produksi dan proktivitas, dan lain sebagainya.

Touchscreen dan saklar dapat dianggap juga sebagai HMI. Teknologi HMI juga digunakan secara luad di virtual dan flat display, pengenalan pola, akses internet dan personal computer data input untuk perangkat elektronik, dan penggabungan informasi.

Badan profesional seperti GEIA dan ISO memberikan standar dan pedoman yang berlaku untuk teknologi HMI.

2.5. TIA Portal

Totally Integrated Automation Portal (TIA Portal) merupakan software dari Siemens yang digunakan untuk membuat program untuk PLC dengan bahasa *Ladder Logic Programming Langguage* (LAD), *Statement List Programming Language* (STL), ataupun Function Block Diagram Programming Language (FBD). Selain itu konsep terintegrasi pada TIA Portal membuat programmer PLC dapat sekaligus membuat aplikasi HMI (*Human Machine Interface*) ataupun SCADA (*Supervisory Control and Data Acquisition*), dan fungsi-fungsi lain yang berkaitan dengan otomasi industri.

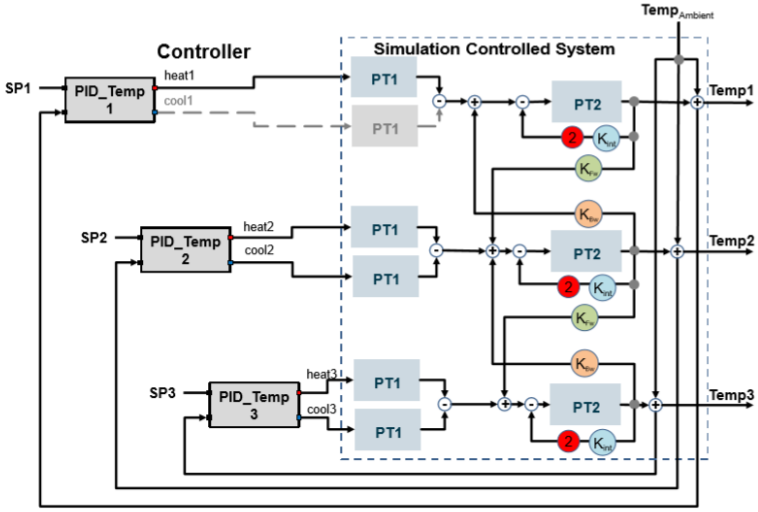
TIA Portal dijalankan pada komputer user. Diagram ladder yang telah selesai dibuat untuk dapat disimulasikan dengan aplikasi PLC-Simulator (PLCSim) maupun dapat langsung dapat diunduh pada PLC. TIA Portal yang berada pada PC atau laptop dapat dihubungkan pada PLC melalui kabel ethernet ataupun secara wireless.

BAB 3
PERANCANGAN SISTEM

Pada tugas akhir ini, akan digunakan sebuah plan sederhana, dikarenakan *lockdown* selama pandemic Covid-19, penelitian ini akan dilakukan sepenuhnya dengan simulasi.

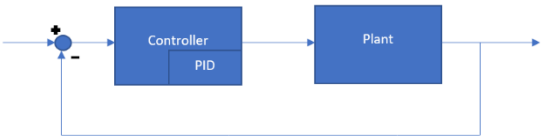
3.1. Merancang Plant

Plant pada penelitian ini akan didesain sepenuhnya dengan menggunakan TIA Portal. Plant ini berdasarkan dari sebuah rancangan plant untuk pengaturan temperatur "*Multi Zone*" dari *Siemens*. Multi zone merupakan sebuah pengaturan temperatur untuk berbagai ruangan yang tiap-tiap ruangan dapat diatur secara independen. Blok diagram dari sistem "*Multi Zone*" dapat dilihat pada Gambar 3.1.



Gambar 3.1 Block diagram sistem Multizone

Namun, dalam penelitian ini, plant dirancang untuk mengatur hanya satu ruangan (atau dalam kendaraan, satu kabin). Sehingga, rancangan yang digunakan hanya untuk zona 3 dimana, berdasarkan dari diagram blok tersebut, dapat mengontrol untuk memanaskan, dan juga mendinginkan. Dengan blok diagram seperti pada Gambar 3.2.



Gambar 3.2 Block diagram sistem yang digunakan

3.1.1. Persamaan *Transfer Function*

Untuk menentukan persamaan *transfer function*, akan didapatkan berdasarkan dari persamaan (1), yaitu:

$$T_Z = y_Z + T_{Ambient} \quad \text{.....(1)}$$

$T_{Ambient}$ = Temperatur Ruangan
T_Z = Temperatur Zone

y_Z didapatkan berdasarkan dari transfer function fungsi delay, berikut:

$$y_z = \frac{gain_{PT2}}{(tmLag1_{PT2} * p + 1) * (tmLag2_{PT2} * p + 1)} * u \quad \text{.....(2)}$$

u_z sendiri didapatkan dari kalkulasi persamaan transfer fuction input sebagai berikut:

$$u_z = kFW_{z-1} * y_{z-1} - 2 * kINT_z * y_z + kBW_{z+1} * y_{z+1} + ulnp_z \quad (3)$$

kFW = coupling factor forward
$kINT$ = coupling factor internal
kBW = coupling factor backward

$ulnp_z$ didapatkan dari penjumlahan dari variable delay yang telah dimanulasi dengan persamaan (2), yaitu:

$$ulnp_z = \frac{gain_{Heat}}{tmLag1_{Heat} * p + 1} * uHeat_z - \frac{gain_{Cool}}{tmLag1_{Cool} * p + 1} * uCool_z + dist_z \quad (4)$$

$uHeat$ = Variable *Heating* yang dapat diubah
$uCool$ = Variable Cooling yang dapat diubah
$dist$ = variable *disturbance*.

Dalam perancangannya, persamaan ini dapat digunakan untuk memodelkan zona sebanyak apapun. Pada persamaan (3), dapat diperhatikan bahwa pada zona periferal, zona sebelumnya dan zona sesudahnya dimatikan, sehingga pada keadaan tersebut, z-1 dan z+1 dapat disetel menjadi sama dengan 0 (nol).

3.1.2. TIA Portal Library

Untuk dapat menggunakan transfer function, TIA Portal memiliki fungsi berupa "LSim Library". Menggunakan fungsi tersebut, dapat dikirimkan sebuah *block* khusus yang telah diprogram secara langsung dari Siemens. Library tersebut dapat didownload pada website resmi Siemens.

Untuk mengimpor library dari luar TIA Portal, dapat dilakukan pada menu "Libraries" yang kemudian terdapat panel "Global Libraries". Apa bila Lsim Library yang diinginkan ada pada library tersebut, maka dapat langsung diimpor pada TIA Portal dengan cara "click" dua kali. Namun bila Lsim Library yang diinginkan belum terdapat pada library, dapat diimport dengan cara memilih "Open global library". Dan mengirimkan Lsim library yang telah didownload.

Dari library yang telah diimport. Akan didapatkan 8 block plant yang telah diprogram. Block tersebut terdiri dari 8 block dengan fungsi berbeda-beda.

3.1.2.1. Function Block (FB) "SimMultizone"

FB "SimMultizone" akan mensimulasikan kurva temperatur dari zona-zona dan berganti mode bergantung dari PID_Temp. berdasarkan dari persamaan yang telah dijabarkan pada subbab 3.1.1, didapatkan parameter sesuai dengan tabel 3.1

Tabel 3.1 FB "SimMultizone"

Nama	Tipe Port	Jenis Data
Velocity	Input	SInt
ambTemp	Input	Real
Reset	Input	Boolean
Cycle	Input	Real
Error	Output	Boolean
StatusID	Output	UInt

Nama	Tipe Port	Jenis Data
Status	Output	Word
maxReached	Output	Boolean
minReached	Output	Boolean
zones	Input dan Output	Array[1..#MAX] dari "typeZone"

3.1.2.1. Data Block (DB) "SimMultizone"

DB "SimMultizone" berisi data dan tags yang akan digunakan untuk mengisi parameter-parameter pada FB dan block PID_Temp. DB terdiri dari 2 tags, yaitu:

a. Plant

Pada data block ini, jenis data berupa "typeSimMultizoneln" yang didapatkan dari sistem multi zone dari Siemens. Pada DB ini, berisi data seperti pada Tabel 3.2.

Tabel 3.2 Data Block Plant

Nama Tags	Jenis Data	Deskripsi
velocity	Signed Integer	Kecepatan transport dari extender screw dalam persen (%)
ambTemp	Real	Temperatur ruangan
reset	Boolean	Me-*reset* semua parameter yang relevant

b. Zones

Tags ini didapatkan berdasarkan dari sistem multi zone dari Siemens. Tags ini berisi data dengan jenis data array. Data pada tabel 3.3.

Tabel 3.3 Data Block Zones

Nama Tags	Jenis Data	Deskripsi
pidTemp	"typePidTemp"	Jenis data PLC dengan parameter yang diperlukan pada PID_Temp

Nama	Jenis Data	Deskripsi
simMultizone	“typeSimMultizoneArray”	Jenis data PLC dengan parameter untuk simulasi temperatur
tuningMode	Unsigned Short Integer	Tipe spesifikasi energi untuk *fine tuning*.

3.1.3. Merangkai Block Plant dan PID

Setelah block untuk plant dan PID_Temp telah ditetapkan, maka langkah selanjutnya adalah merangkainya. Untuk itu akan ditambakan sebuah program block baru. Untuk menambahkan program block baru, dapat dilakukan pada menu “program block” dan memilih “add new block”.

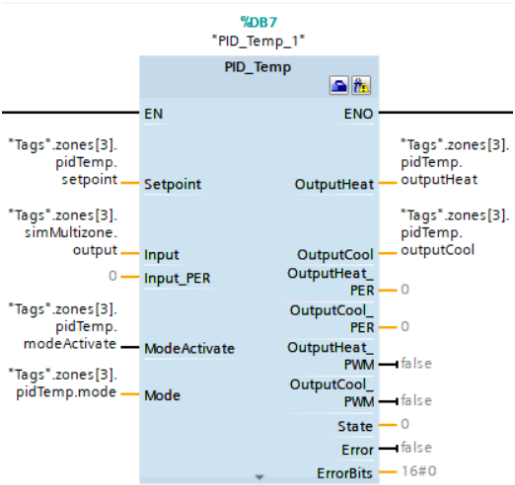
Block yang akan dipilih adalah sebuah *organization block*, dengan tipe block adalah ”*cyclic interrupt*”. Untuk memasukan block plant pada program block *cyclic interrupt*, cukup *drag and drop* dari *function block* yang telah didownload dari *Lsim Library*. Kemudia untuk menentukan parameter-parameter dari block tersebut dapat disesuaikan dari tags yang telah didesain pada DB.

PID_Temp terdapat pada menu *instruction*, pada folder “Technology”, kemudian dipilih folder “PID Control”. Pada folder PID Control, terdapat 3 folder baru, yaitu “Compact PID”, “PID Basic function”, dan “Auxilary function”. Block PID_Temp dapat diakses pada folder “Compact PID”. Untuk memasukan PID_Temp kedalam program block, cukup *drag and drop* block kedalam program block.

3.1.4. PID_Temp

Pada TIA Portal terdapat berbagai jenis PID untuk mengatur berbagai jenis plant. Pada penelitian ini, dikarenakan output plant berupa temperatur, maka akan digunakan *PID Temperature* (PID_Temp).

Pada PID ini terdapat perbedaan dengan PID_Compact. PID ini memiliki 2 jenis output, yaitu output cooling dan output heating. Dimana nilai tersebut akan berubah bergantung apa nilai apa yang ditetapkan pada setpoint. Tampilan dari Blok PID_Temp dapat dilihat pada Gambar 3.3.



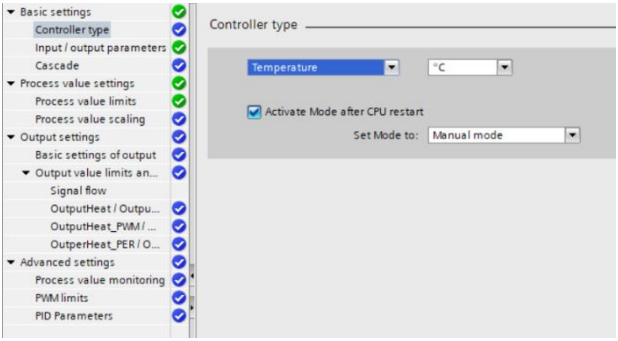
Gambar 3.3 Block PID_Temp

Pertama-tama, akan diatur terlebih dahulu PID_Temp tersebut. Untuk melihat *properties* dari PID_Temp, dapat dibuka dengan melalui menu di sebelah kanan-atas dari block PID_Temp.

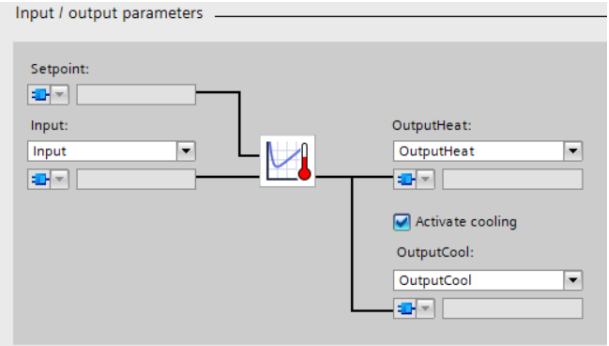
Pertama adalah mengatur pengaktifan PID_Temp. Pada Gambar 3.4, terlihat pada menu "*control type*", dapat diberi centang pada pilihan "*Activate Mode after CPU restart*". Sehingga pada saat simulasi, PID akan dapat langsung bekerja.

Pengaturan selanjutnya adalah untuk mengatur parameter *input/output*. Pada Gambar 3.5, terlihat pada nilai setpoint akan diatur secara langsung pada block nya, sehingga tidak perlu diatur. Pada setting input, akan dipilih input berupa "input". Terdapat 2 pilihan, yaitu "input" dan "Input

PER (Analog)". Input PER adalah input berupa persen atau rasio. Sehingga jenis input yang dipilih adalah "Input".



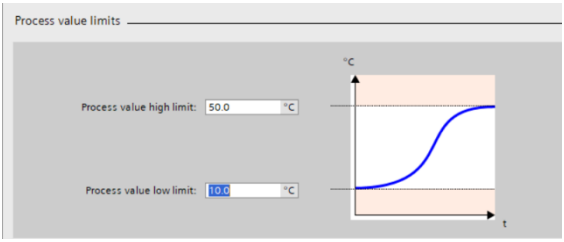
Gambar 3.4 setting tipe kontroller PID_Temp



Gambar 3.5 Setting parameter input dan output

Pada menu output, secara default, PID_Temp hanya akan mengaktifkan "OutputHeat". Sehingga bila perlu "OutputCool" akan diperlukan untuk mengaktifkan dengan cara memberi centang pada menu "Activate cooling". Untuk menu jenis Output, terdapat 3 pilihan jenis, yaitu "OutputHeat/Cool", "OutputHeat/Cool PWM", dan "OutputHeat/Cool PER(Analog)". Seperti pada input, "OutputHeat/Cool PER" adalah input dalam bentuk persen atau rasio. Sedangkan "OutputHeat/Cool PWM" merupakan output dalam bentuk *pulse width modulation*. Untuk plant ini, output yang digunakan adalah "OutputHeat/Cool".

Yang terakhir adalah mengatur limit dari nilai dan proses. Untuk mengatur limit dari nilai, dapat dilakukan pada menu "Process value limits". Pada Gambar 3.6, terlihat pada menu tersebut terdapat option "*Process value high limit*" dan "*Process value low limit*". Kedua option itu berfungsi untuk memberikan limit nilai atas dan bawah dari sistem. Karena pada sistem ini, setpoint dan output akan dibatasi pada range 15 derajat celsius hingga 45 derajat celsius, maka untuk memudahkan pemantauan, akan diberi batas atas sebesar 50 derajat celsius, dan batas bawah sebesar 10 derajat celsius.



Gambar 3.6 setting limit nilai proses PID_Temp

3.2. Mendesain Ladder dengan Metode Huffman

Setelah mendapatkan plant, langkah selanjutnya adalah memulai mendesain diagram *ladder*. Dalam mendesain ladder diagram dengan menggunakan metode huffman, terdapat 6 langkah, yaitu:

a. Menentukan state, input, dan output
b. Membuat primitive flow table
c. Membuat merge flow table
d. Menentukan fungsi eksitasi dan output
e. Menambahkan sinyal start (opsional)
f. Mendesain ladder diagram ladder berdasarkan fungsi eksitasi dan output.

3.2.1. Menentukan State, Input, dan Output

Langkah pertama adalah menentukan input dan output. Pada sistem ini, input dari sistem terdapat 3 (tiga), yaitu:

1. Input mode cooling
2. Input mode heating
3. Sensor suhu pada kabin

Sedangkan untuk *output*-nya, terdapat 2, yaitu:

1. Cooling mode aktif
2. Heating mode aktif

Setelah didapatkan *input* dan *output*-nya, langkah selanjutnya adalah untuk menentukan *state* dari setiap langkah pada proses. *State* dari sistem ini terbagi dalam 5 *state* yang terdiri dari:

1. Sistem dalam keadaan tidak terdapat input apapun yang telah dimasukan.
2. Diberikan input 'mode cooling' tetapi sensor suhu pada kabin kendaraan belum mencapai suhu yang sesuai dengan input yang diberikan
3. Diberikan input 'mode cooling' tetapi sensor suhu pada kabin kendaraan telah mencapai suhu yang sesuai dengan input yang diberikan
4. Diberikan input 'mode heating' tetapi sensor suhu pada kabin kendaraan belum mencapai suhu yang sesuai dengan input yang diberikan
5. Diberikan input 'mode heating' tetapi sensor suhu pada kabin kendaraan telah mencapai suhu yang sesuai dengan input yang diberikan.

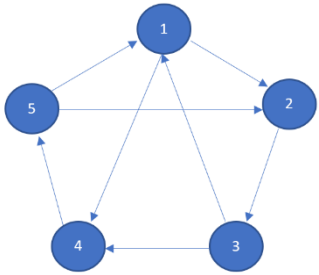
Dari input dan output, serta state yang telah didapatkan, didapatkan data seperti pada Tabel 3.4:

Tabel 3.4 State, input, dan output

State	Input			Output	
	c	h	SenTemp	C	H
1	0	0	0	0	0
2	1	0	0	0	0
3	1	0	1	1	0
4	0	1	0	0	0

State	Input			Output	
	c	h	SenTemp	C	H
5	0	1	1	0	1

Dari state yang telah didapatkan, selanjutnya adalah menghubungkan setiap state. Secara logika, saat sistem berada pada satet 1, belum menyala dan belum diberikan input sama sekali, maka pengguna akan memilih salah satu dari 2 (dua) input, yaitu 'mode cooling', atau 'mode heating'. Sehingga, dari state 1, sistem dapat bergerak menuju antara state 2, atau state 4. Sedangkan state 2 hanya dapat bergerak menuju state 3, begitu pula state 4, yang hanya dapat bergerak menuju state 5. Tetapi, saat sistem telah mencapai suhu yang diinginkan, pengguna dapat bergerak menjadi antara mematikan sistem (000), atau state 2 dan state 3. Dimana saat salah satu dari mode input telah tercapai, pengguna dapat mengubah inputnya. Sehingga, didapatkan sebuah diagram seperti pada Gambar 3.7.



Gambar 3.7 State diagram sistem

3.2.2. Membuat Primitive flow table

Setelah didapatkan hubungan antara state, selanjutnya adalah untuk membuat *primitive flow table*. Dari state, input, output, serta diagram hubungan antar state, didapatkan *primitive flow table* seperti pada Tabel 3.5.

Tabel 3.5 ***Primitive flow table***

State	Input					Output	
	000	100	101	010	011	C	H
1	**1**	2	-	4	-	0	0
2	-	**2**	3	-	-	0	0
3	1	-	**3**	4	-	1	0
4	-	-	-	**4**	5	0	0
5	1	2	-	-	**5**	0	1

3.2.3. Membuat Merged Flow Table

Langkah selanjutnya adalah untuk menyederhanakan *primitive flow table* tersebut menjadi *merged flow table*, untuk mempermudah penentuan input-output serta menentukan relay. Untuk menyederhanakan *primitive flow table* tersebut, pertama perlu diperhatikan diagram hubungan antar state tersebut. Dapat terlihat bahwa state 5 tidak terhubung pada state 3, state 2 tidak terhubung dengan state 4. Sehingga untuk membuat *merged flow table*, state 5 tidak boleh dipisahkan hanya bersama dengan state 3, dan state 2 tidak boleh dipisah hanya dengan state 4, karena kedua state tersebut tidak memiliki hubungan. Sehingga, kita memiliki pilihan me-*merge primitive flow table* tersebut antara:

1. {1,2,3};{4,5}
2. {1,3,4};{2,5}
3. {2,3,4};{1,5}
4. {1,2,5};{3,4}
5. {2,3,5};{1,4}
6. {1,4,5};{2,3}
7. {3,5.4};{1,2}
8. {2,4,5};{1,3}

Untuk sistem ini, merged flow table yang dipilih adalah konfigurasi no. 1, yaitu {1,2,3};{4,5}. Sehingga, didapatkan *merged flow table* seperti pada Tabel 3.6:

Tabel 3.6 ***Merge flow table***

State	Input					Relay
	000	100	101	010	011	
1,2,3	**1**	**2**	**3**	4	-	1
4,5	1	2	-	**4**	**5**	0

3.2.4. Mencari Persamaan Relay dan Output

Selanjutnya adalah untuk mencari persamaan untuk relay dan output.

Untuk mendapatkan persamaan tersebut, pertama-tama dibutuhkan untuk membuat k-map berdasarkan dari *merged flow table*. Didapatkan k-map seperti pada Tabel 3.7:

Tabel 3.7 Posisi nilai dalam k-map

	c,h				
t,Y		00	01	11	10
	00	1	**4**	-	2
	01	**1**	4	-	**2**
	11	-	**5**	-	**3**
	10	-	-	-	-

Setelah didapatkan k-map-nya, Langkah selanjutnya adalah menentukan persamaan relay. Sebenarnya, persamaan relay telah diketahui, yaitu:

$$Y = (S + Y).\bar{R} \quad \ldots\ldots\ldots\ldots\ldots(5)$$

Langkah ini adalah untuk menentukan set (S) dan reset (R), untuk melengkapi persamaan relay (Y). Untuk menentukan set dan reset, dibutuhkan untuk mengisi k-map berdasarkan dari table karakteristik dari flip-flop SR. Dimana Tabel karakteristik flip-flop dapat dilihat pada Tabel 3.8:

Tabel 3.8 Tabel karakteristik SR flip-flop

Y_k	Y_{k+1}	S	R
0	0	0	*
0	1	1	0
1	0	0	1
1	1	*	0

Berdasarkan dari *merged flow table* dan table karakteristik flip-flop SR, didapatkan k-map seperti table 3.9 dan table 3.10:

Tabel 3.9 K-map set relay Y

t,Y\c,h	01	01	11	10
00	0	x	x	0
01	0	1	x	0
11	x	x	x	0
10	x	x	x	x

Tabel 3.10 K-map reset relay Y

t,Y\c,h	00	01	11	10
00	1	0	x	1
01	x	0	x	x
11	x	x	x	x
10	x	0	x	x

Berdasarkan dari k-map tersebut, didapatkan persamaan sebagai berikut:

$$S = \bar{h} \quad \text{……..(6)}$$
$$R = h \quad \text{……..(7)}$$

Sebelum menentukan persamaan relay, pertama diperlukan sedikit modifikasi pada persamaan set, yaitu menambahkan tombol 'start'. Sehingga persamaan set menjadi:

$$S = \bar{h}.Start \quad \text{……..(8)}$$

Dengan mensubtitusi persamaan 2 dan 3 kepada persamaan 1, didapatkan persamaan relay sebagai berikut:

$$Y = \left(\bar{h}.start + Y\right).\overline{(h)} \quad \text{……..(9)}$$
$$Y = \left(\bar{h}.start + Y\right).\bar{h} \quad \text{……..(10)}$$

Berdasarkan sifat aljabar Boolean, yaitu sifat distributif, persamaan 5 dapat disederhanakan menjadi persamaan berikut:

$$Y = \left(\bar{h}.\bar{h}.start\right) + \left(Y.\bar{h}\right) \quad \text{……..(11)}$$

dan berdasarkan sifat dari aljabar Boolean, yaitu:

$$\bar{A}.\bar{A} = \bar{A} \quad \text{……(12)}$$

Didapatkan persamaan relay sebagai berikut:

$$Y = \bar{h}.(Start + Y) \quad \text{……..(13)}$$

Setelah didapatkan persamaan relay-nya, Langkah selanjutnya adalah untuk menentukan persamaan output. Yaitu persamaan C dan H. Berbeda dengan menentukan persamaan set reset, persamaan output didapatkan berdasarkan dari output yang telah ditetapkan pada *primitive flow table*,

yang kemudian akan dimasukan pada k-map seperti pada saat menentukan persamaan relay. Sehingga didapatkan k-map seperti pada Tabel 3.11 dan t\Tabel 3.12 sebagai berikut:

Tabel 3.11 K-map output 'C'

t,Y\c,h	00	01	11	10
00	x	0	x	x
01	0	x	x	0
11	x	1	x	0
10	x	x	x	x

Tabel 3.12 K-map output 'H'

t,Y\c,h	00	01	11	10
00	x	0	x	x
01	0	x	x	0
11	x	0	x	1
10	x	x	x	x

Sebelum bisa didapatkan persamaan kedua output 'C' dan 'H', terdapat modifikasi yang harus dilakukan pada k-map output. Dikarenakan persamaan set diberikan 'start', k-map output harus diberikan sedikit perubahan pada bagian state tidak stabil '1'. Dimana agar sistem dapat berhenti pada state tidak stabil '1' saat terdapat state stabil lain yang bergerak menuju state stabil '1', dibutuhkan output pada state tidak stabil menjadi 0 (nol). Sehingga didapatkan K-map baru, tabel 3.13 dan 3.14.

Tabel 3.13 K-map output 'C' setelah diberi 'Start'

t,Y\c,h	00	01	11	10
00	0	0	x	x
01	0	x	x	0
11	x	0	x	1
10	x	x	x	x

Tabel 3.14 K-map output 'H' setelah diberi 'Start'

t,Y\c,h	00	01	11	10
00	0	0	x	x
01	0	x	x	0
11	x	1	x	0
10	x	x	x	x

Sehingga, didapatkan persamaan output sebagai berikut:

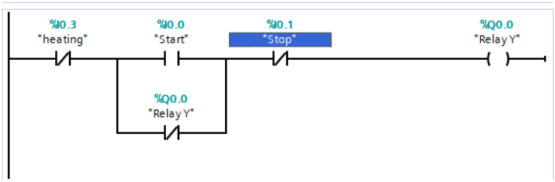
$$C = c.t \quad \ldots\ldots\ldots\ldots\ldots(14)$$
$$H = h.t \quad \ldots\ldots\ldots\ldots\ldots(15)$$

3.2.5. Merancang Diagram *Ladder*

Setelah mendapatkan persamaan output dan relay, proses selanjutnya adalah merancang diagram *ladder*-nya. Pada langkah ini, untuk menguji coba diagram ladder yang telah di desain akan digunakan simulator ladder diagram online melalui website plcfiddle.com, sebelum disimulasikan dengan menggunakan TIA Portal.

Langkah pertama dari merancang diagram ladder dengan metode huffman adalah untuk menentukan komponen menyesuaikan dari persamaan yang didapatkan pada subbab 3.5. Pertama adalah persamaan relay, yaitu persamaan 10, persamaan relay Y dihasilkan oleh start dengan keadaan *normally-closed* yang di paralelkan dengan relay Y itu sendiri, dengan keadaan *normally-closed* dan input *heating* dengan keadaan *normally-closed* pula. Sehingga didapatkan diagram *ladder* seperti pada

Gambar 3.8.



Gambar 3.8 Diagram ladder untuk relay Y

Langkah selanjutnya adalah untuk menentukan diagram *ladder* untuk persamaan output, yaitu persamaan 11 dan persamaan 12. Kedua persamaan tersebut merupakan persamaan yang sama, hanya saja untuk output *cooling* menggunakan input *cooling*, dan untuk output *heating* menggunakan input *heating*. Persamaan output terdiri dari input c atau input h dengan keadaan *normally-closed*. Dan diseri dengan input t dengan keadaan *normally-closed*. Sehingga untuk persamaan output, didapatkan diagram *ladder* seperti pada Gambar 3.9.

Gambar 3.9 Diagram ladder untuk output 'C' dan 'H'

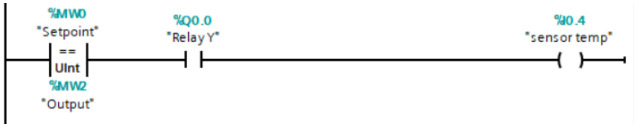
3.3. Modifikasi Ladder Sesuai dengan Kebutuhan Sistem

Dengan ladder diagram telah didapatkan, ladder diagram dengan metode Huffman telah selesai. Namun, diagram ladder tersebut tidak dapat diimplementasikan secara langsung. Dengan *automatic climate control*, dibutuhkan sedikit modifikasi pada bagian input.

3.3.1. Mengatur Input Sensor Temp

Pertama-tama, pada sistem *automatic climate control* input tidak hanya terdiri dari mode cooling atau mode heating. Tetapi juga terdapat input temperature yang diinginkan oleh pengguna sistem. Dimana input temperature akan dibaca dan akan dibandingkan dengan suhu pada kabin dengan menggunakan sensor temperature pada kabin.

Sehingga, perlu ditambahkan diagram ladder lagi untuk mengatur sistem tersebut. Dimana akan dipasangkan comparator yang akan membandingkan antara input dari pengguna sistem dengan output yang didapatkan dari pembacaan oleh sensor temperature pada kabin yang akan dihubungkan pada output plan pada Simulink. Sehingga didapatkan diagram ladder seperti pada Gambar 3.10.

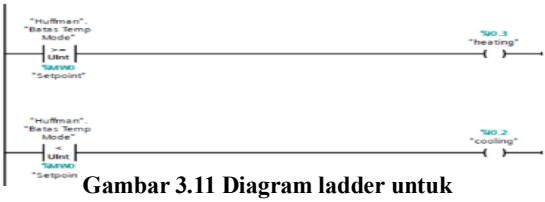


Gambar 3.10 Diagram ladder untuk sensor temp

3.3.2. Mengatur Range Temperatur

Akan dibutuhkan pengaturan untuk input cooling (c) dan input heating (h). Dimana saat pengguna sistem memilih salah satu mode, mode yang lain tidak boleh menyala dan harus tidak aktif. Selain itu, pada saat pengguna menaikan input temperature diatas batas yang pada sistem ini diatur pada temperatur 30 derajat, dimana bila pengguna menaikan temperature lebih dari sama dengan 30 derajat, maka sistem secara otomatis akan berubah dari mode cooling menjadi mode heating. Sedangkan bila temperature diturunkan dibawah 30 derajat, maka mode secara otomatis, mode akan berubah dari heating mode, menjadi mode cooling.

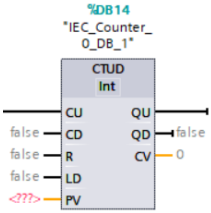
Sehingga pertama-tama dibutuhkan sistem set-reset antar input mode. Dan selain itu, juga dibutuhkan comparator. Dimana bila input temperature yang dimasukan oleh pengguna, lebih dari sama dengan 30 derajat, maka mode heating akan aktif. Dan untuk mode cooling, akan diberikan compactor yang akan membandingkan input oleh pengguna dengan batas. Bila temperature input berada di bawah temperature 30 derajat, maka mode cooling akan aktif. Sehingga didapatkan diagram pada Gambar 3.11.



Gambar 3.11 Diagram ladder untuk batas setpoint antar mode

3.3.3. Mengatur tombol input temperatur

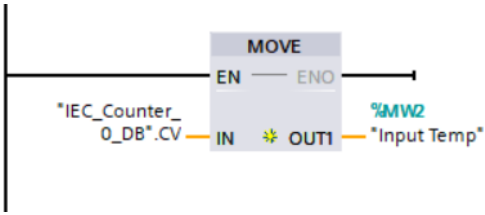
Selanjutnya adalah untuk mengatur tombol untuk mengisi input. Pada interface automatic climate control, saat tombol input up ditekan, maka input temperatur akan naik. Sedangkan saat tombol input down ditekan, maka input temperatur akan turun. Untuk mendapatkan fungsi tersebut, digunakan sebuah counter up and down.



Gambar 3.12 Block counter

Sesuai dengan Gambar 3.12, input CU diberikan contact normally open "tombol naik". Sedangkan input CD diberi contact normally open "tombol turun".

Untuk mengubah nilai counter menjadi nilai input temperatur, akan digunakan block "move" yang akan memindahkan nilai counter yang telah disimpan pada output "CV" pada input temperatur, sesuai dengan Gambar 3.13



Gambar 3.13 Block "Move"

3.3.4. Mengatur Tombol Input Mode

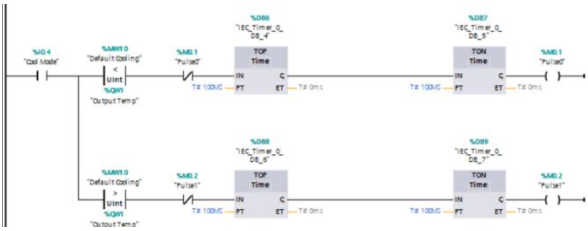
Bila pengguna tidak memilih memberikan input secara spesifik, tetapi langsung memilih mode cooling atau mode heating, input temperatur akan diatur secara otomatis. Pada saat mode cooling dipilih, maka input temperatur akan secara otomatis input temperatur akan berubah menjadi 20. Sedangkan bila dipilih mode heating dipilih, maka secara otomatis, input temperatur akan berubah menjadi 30.

Untuk dapat mengimplementasikan input tersebut, counter pada subbab 3.7.2 akan digunakan kembali. Untuk mengimplementasikan

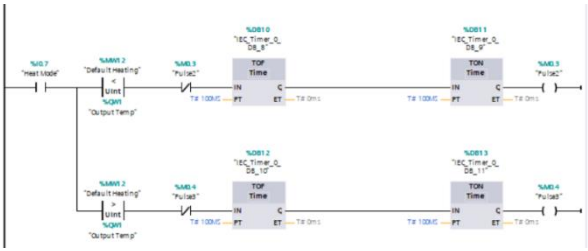
input secara otomatis, akan digunakan comparator yang membandingkan nilai temperatur mode cooling dan mode heating dengan output temperatur.

Terdapat 2 comparator yang akan digunakan. comparator pertama digunakan bila nilai temperatur mode cooling atau mode heating lebih besar dari output temperatur. Sedangkan comparator kedua digunakan bila nilai temperatur mode cooling dan mode heating lebih kecil dari output temperatur.

Disaat nilai temperatur lebih kecil dari output, maka input harus bertambah sampai dengan input temperatur sama dengan nilai temperatur tersebut. Untuk itu, akan digunakan sebuah timer off delay dan timer on delay sebagai sebuah pulse generator. Sehingga didapatkan *ladder* diagram seperti Gambar 3.14 dan Gambar 3.15.



Gambar 3.14 Ladder diagram "Cool Mode"



Gambar 3.15 Ladder diagram "Heat Mode"

Preset time pada timer akan diset dalam 0.1 second (atau dalam block-nya, 100 ms). Sebelum timer off delay, terdapat contact normally open, dan terdapat coil setelah timer on delay sebagai global memory bernama pulse. Disaat timer off delay menyala, timer on delay akan menyala setelah 0,1 sekon, mengaktifkan coil "pulse". Karena coil "pulse" aktif, contact normally closed akan terbuka, sehingga timer off delay akan

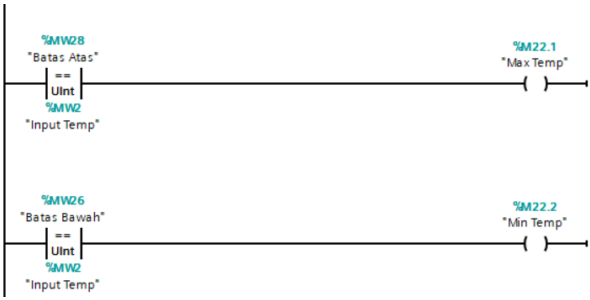
bekerja. Setelah 0,1 sekon, timer off delay akan non-aktif. Sehingga mematikan timer on delay. Dan kemudian kembali seperti awal lagi.

Coil “pulse” yang tersambung dengan comparator pertama terhubung dengan input CU dari counter up-down. Sehingga, bila output temperatur lebih kecil dari nilai temperatur cooling atau heating, maka counter akan secara otomatis mengurangi nilai input yang akan mengubah nilai output secara otomatis hingga nilai input temperatur sama dengan nilai temperatur cooling atau heating.

Sedangkan coil “pulse” yang tersambung dengan comparator kedua terhubung dengan input CD dari counter up-down. Sehingga saat nilai output temperatur lebih besar dari nilai temperatr cooling atau heating, maka secara otomatis counter akan menambah nilai input temperatur, sehingga nilai output berubah hingga nilai input temperatur sama dengan nilai temperatur cooling atau heating.

3.3.5. Mengatur Batas Atas dan Bawah Input Temperatur

Pada pengaturan suhu kendaraan, terdapat batas untuk temperatur input. Agar input temperatur tidak dapat bertambah atau berkurang lebih dari batas atas dan bawah input temperatur, akan dirancang sebuah diagram ladder tambahan seperti pada Gambar 3.16.



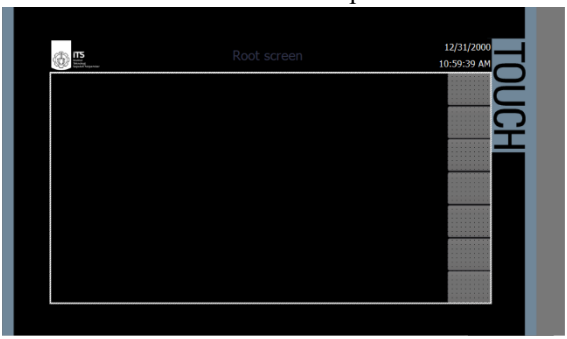
Gambar 3.16 Diagram untuk mengatur batas maksimum dan minimum setpoint

Pada diagram ladder ini, terdapat dua comparator, dengan comparator pertama adalah comparator “equal to” antara input temperatur dengan batas atas temperatur, yang terhubung pada coil “max temp”. Dimana coil tersebut terhubung dengan contact normally closed pada input CU counter up-down. Sehingga bila input temperatur telah menyentuh batas maksimal, input temperatur dapat ditekan, namun suhu tidak akan bertambah lebih dari batasnya.

Pada comparator kedua, terdapat comparator “equal to” antara input temperatur dengan batas bawah dan terhubung dengan coil “min temp”. dimana coil tersebut terhubung dengan contact normally closed pada input CD counter up-down. Sehingga bila input temperatur telah mencapai batas minimal, saat input temperatur telah menyentuh batas minimal temperatur, input temperatur dikecilkan, maka contact normally closed akan terbuka, sehingga saat input temperatur telah menyentuh batas minimum, input tidak akan berkurang menjadi lebih kecil.

3.4. Merancang HMI

Setelah desain ladder diagram telah selesai, langkah selanjutnya adalah merancang *Human Machine Interface* (HMI). HMI yang digunakan pada penelitian ini “Simatic Comfort Panel” dengan display berukuran 7 inchi (sekitar 17.78 cm), yaitu “TP700 Comfort”. Tampilan default dari HMI TP700 Comfort terlihat pada Gambar 3.17.



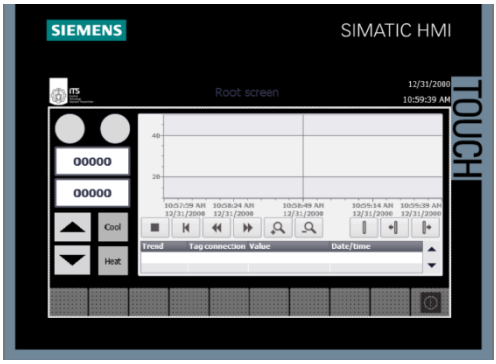
Gambar 3.17 Tampilan default HMI pada TIA Portal

Untuk membantu pengguna dapat mengatur suhu dan memilih mode, akan digunakan tombol (“Button”). Sedangkan untuk mengetahui nilai setpoint dan output dari temperatur sistem, maka akan diberikan display yang akan menunjukan nilai temperatur dengan “I/O Field”. Dan untuk membantu pengguna mengetahui bahwa sistem sedang dalam mode *heating* atau mode *cooling*, akan diberikan indikator yang akan diwakili dengan sepasang lingkaran, yang akan berubah warna saat mode yang dipilih telah terpenuhi.

Button “Temp Up”, digunakan untuk menambahkan nilai temperatur dan button “Temp Down” digunakan untuk mengurangi nilai temperatur. Sedangkan button “Cool” digunakan untuk memilih mode *cooling* dan button “Heat” digunakan untuk memilih mode *heating*.

Kedua display akan menunjukan nilai setpoint dan nilai output dari sistem. Sebenarnya, pada display sesungguhnya pada interface *automatic climate control* hanya perlu ditunjukan display input untuk membantu pengguna mengetahui berapakah setpoint suhu yang diberikan pada sistem. Namun untuk mempermudah display dengan tujuan penelitian ini, maka display setpoint dan output akan diberikan.

Terakhir, terdapat dua buah indikator berupa lingkaran. Kedua lingkaran itu berfungsi menginformasikan pengguna bila sistem memasuki mode *cooling* atau memasuki mode *heating*. Bila sistem memasuki mode *cooling*, maka indikator 1 akan berubah warna menjadi hijau. Sedangkan bila memasuki mode *heating*, maka indikator 2 yang akan menyala dan berubah warna menjadi oranye. Tampilan dari HMI yang telah di desain dapat terlihat pada Gambar 3.18

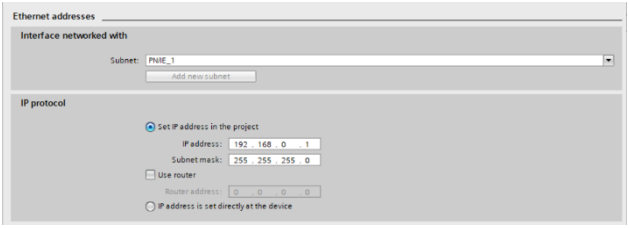


Gambar 3.18 Tampilan HMI yang didesain

BAB 4
HASIL DAN ANALISA DATA

4.1. Menghubungkan TIA Portal dengan PLC-Simulator

Sebelum memulai simulasi, pertama PLC perlu dihubungkan dengan PLC-Simulator (PLCSim). Untuk itu, pertama perlu diatur ethernet setting dari PLC yang digunakan. Pengaturan ethernet terlihat pada Gambar 4.1.

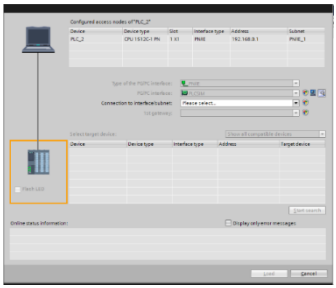


Gambar 4.1 Pengaturan Ethernet PLC

Pada Gambar 4.1 pilihan "interfaced networked with", akan terpilih secara otomatis pada subnet "PN/IE_1" saat kita memilih "add new subnet".

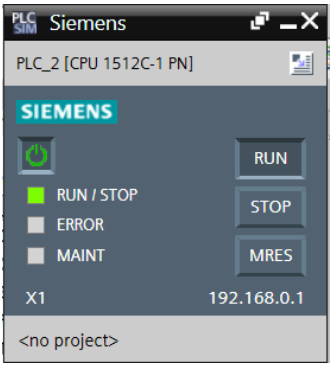
Selanjutnya adalah mengatur IP Adress dari PLC. IP Adress dari PLC tidaklah harus sama dengan laptop/PC yang digunakan untuk memprogram PLC tersebut. Secara *default* bila kita tidak mengubah IP Adress, maka IP Adress dari PLC akan diatur pada "192.168.0.1".

Langkah selanjutanya adalah mengaktifkan mode simulasi pada TIA Portal. Bila mode simulasi diaktifkan, maka akan muncul window baru pada TIA Portal seperti pada Gambar 4.2.



Gambar 4.2 menghubungkan pada PLCSim

Pada gambar 4.2, terlihat bahwa tidak diperlukan untuk memilih jenis interface PG/PC. Hal tersebut karena PLCSim telah aktif, dan TIA Portal secara otomatis akan menghubungkan pada PLCSim. Bila Laptop/PC dihubungkan secara langsung pada PLC, maka perlu dipilih jenis interface PG/PC. Selanjutnya, adalah memilih hubungan interface/subnet, sesuai dengan subnet yang telah dipilih saat mengatur pengaturan ethernet PLC, yaitu "PN/IE_1". Bila itu semua telah dilakukan, maka PLCSim telah terhubung dengan TIA Portal. Saat TIA Portal telah terhubung dengan PLCSIM, tampilan pada PLCSIM akan terlihat seperti pada Gambar 4.3.



Gambar 4.3 Tampilan PLCSim

Bila PLCSim telah terhubung dengan TIA Portal, sistem akan meng-*compile* dan mengunduh projek yang telah dihubungkan pada PLCSim secara otomatis.

4.2. Simulasi Sistem

Simulasi dari sistem ini cukup simple. Inti dari simulasi ini adalah untuk mengetahui apabila sistem telah bekerja sesuai dengan perintah atau tidak.

Untuk memulai simulasi, TIA Portal akan dimasukan pada mode monitoring. Secara langsung sistem akan bekerja dan HMI akan berjalan. Untuk memastikan apakah ladder telah bekerja atau tidak, dapat dilihat pada panel device pada TIA Portal.

Terlihat contact dan coil yang aktif akan berubah warna menjadi hijau, sedangkan contact dan coil yang tidak aktif akan berwarna biru. Sehingga dapat diketahui contact dan coil mana yang aktif berdasarkan dari

warnanya.

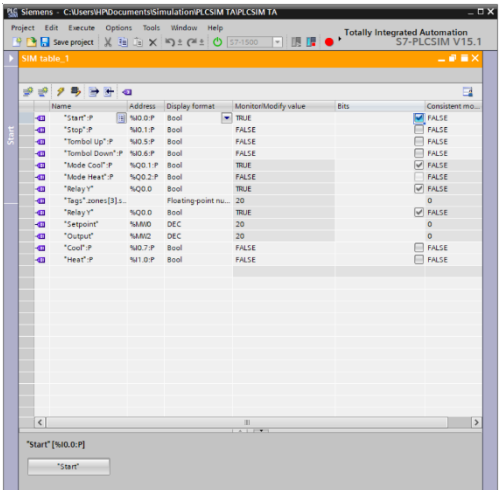
Pertama-tama, perlu diatur terlebih dahulu nilai-nilai integer memory yang akan digunakan. Sesuai dengan pembahasan pada bab III, terdapat 6 integer memory yang harus ditentukan terlebih dahulu nilainya. Nilai tersebut dapat terlihat pada Tabel 4.1.

Tabel 4.1 Tags nilai integer

No	Integer Memory	Nilai yang ditentukan
1	Batas Temp	30
2	Default Cool	20
3	Default Heat	30
4	Max Temp	45
5	Min Temp	15
6	Batas Reset	20

Dengan integer memory telah diatur, simulasi dapat dilakukan. untuk simulasi pertama adalah untuk mengetahui apakah sistem telah bekerja dengan baik. Untuk itu akan digunakan project view dari PLCSim.

Dapat dilihat Gambar 4.4, terdapat tiap-tiap tags yang telah ditentukan pada *project view*. Untuk mengaktifkan tags yang bersifat boolean, dapat dilakukan dengan memberikan "✓" (centang) pada kolom bits.



Gambar 4.4 Tampilan *project view* PLCSIM

4.2.1. Menentukan Setpoint Temperatur

Percobaan kedua adalah untuk menentukan setpoint temperatur. Untuk menetapkan nilai setpoint temperatur, akan digunakan melalui contact NO "Tombol Up" dan contact NO "Tombol Down" untuk menambahkan dan mengungai nilai setpoint dari PID. Terlihat pada contact NO "Tombol Up" diaktifkan maka nilai pada counter akan bertambah. Sedangkan bila contact NC "Tombol Down" diaktifkan, maka nilai pada counter akan berkurang.

Berdasarkan dari nilai-nilai yang disimpan pada counter, nilai tersebut akan diubah dari data berbentuk integer menjadi Unsigned Integer yang akan dibaca oleh HMI. Sedangkan untuk nilai yang akan dijadikan setpoint dari PID_Temp, akan diubah sifat datanya dari integer menjadi real, yang kemudian akan dikirmkan pada setpoint PID_Temp.

4.2.2. Tuning PID_Temp

Agar PID_Temp dapat bekerja dengan lebih lancar, pertama-tama diperlukan untuk melakukan tuning pada PID_Temp. Tuning dapat dilakukan melalui folder pada TIA Portal bernama, "Technology object". Pada folder tersebut terdapat PID yang digunakan, dan pada PID tersebut akan dipilih mode, "Commissioning".

Dengan memilih "Commissioning", akan terdapat window baru yang akan menunjukan grafik kerja PID_Temp. Untuk melakukan tuning, pertama adalah menentukan waktu *sampling*, pada penelitian ini waktu *sampling* ditentukan pada 0.3 s. setelah waktu *sampling* telah ditentukan, sampling dapat dilakukan dengan menekan tombol start.

Setelah sampling telah dilakukan, dapat dimulai tuning dengan menekan tombol "tuning" sistem akan secara otomatis memasuki mode "pre-tuning" dan "fine-tuning". Bila tuning telah selesai, secara otomatis sistem akan menentukan parameter-parameter PID-nya dan mengubah parameter PID tersebut.

4.2.3. Uji Coba dengan HMI

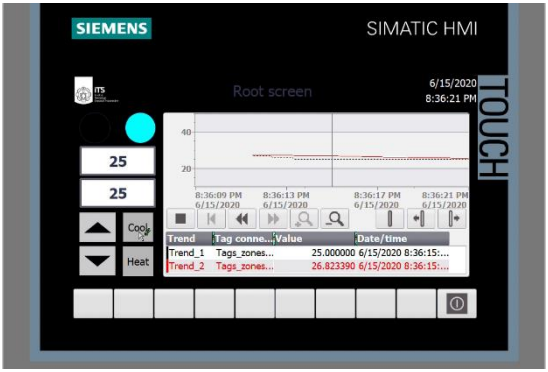
Dengan ladder telah bekerja dan PID telah di tune, maka selanjutnya adalah mengujicoba sistem dengan HMI. Untuk itu akan disimulasikan HMI dengan menggunakan SIMATIC WinCC.

Sesuai dengan penjelasan bab III telah dijelaskan fungsi dari tiap-tiap komponen pada HMI yang telah didesain. Pada display 1, akan ditunjukan nilai setpoint yang telah di tentukan. Dan untuk mengubah nilai setpoint, akan digunakan button "Temp Up" dan "Temp Down". Dimana masing-

masing tombol akan terhubung dengan coil NO “Tombol Up” dan coil NO “Tombol Down” yang akan menambahkan atau mengurangi nilai pada counter. Karena telah didesain sebuah ladder untuk membatasi setpoint, maka pada saat setpoint tersebut telah tercapai, secara otomatis counter akan terputus dari inputnya. Sehingga bila setpoint telah mencapai batas maksimal, button “Temp Up” tidak akan membuat nilai pada display setpoint bertambah. Begitu pula bila nilai minimum setpoint telah tercapai, maka button “Temp Down” tidak akan membuat nilai pada display setpoint berkurang lagi. Batas maksimal dan batas minimal dari setpoint telah diatur sesuai dengan pada Tabel 12. Dimana batas maksimalnya adalah 45, dan batas minimalnya adalah 15.

Tombol “Mode Cool” dan “Mode Heat” digunakan untuk mengaktifkan mode cooling atau mode heating secara default. Tombol “Mode Cool” dan “Mode Heat” terhubung dengan coil NO “Mode Cool” dan coil NO “Mode Heat”. Bila salah satu dari kedua button tersebut ditekan, maka secara otomatis mode akan berubah pada mode yang telah dipilih, dan setpoint temperatur akan berubah sesuai dengan nilai default mode yang telah ditentukan. Sesuai dengan tabel 12, nilai default setpoint mode cooling adalah 20 dan untuk default setpoint mode heating adalah 30.

Saat nilai output dari sistem dan setpoint telah sama, maka indikator akan menyala. Indikator akan mengeluarkan warna sesuai dengan modenya. Mode heating akan menyalakan indikator berwarna oranye bila output berada pada range 30 – 45. Sedangkan mode cooling akan menyalakan indikator berwarna biru muda bila output berada pada range 15 – 29. Tampilan HMI saat bekerja terlihat pada Gambar 4.5.



Gambar 4.5 Tampilan HMI selama simulasi

Halaman ini sengaja dikosongkan

BAB 5
PENUTUP

5.1. Kesimpulan

Berdasarkan dari hasil pengujian ini, dapat diketahui bahwa:

1. Dengan menggunakan metode Huffman, dihasilkan ladder diagram yang terdiri dari 1 relay, dan 2 output.
2. Berdasarkan dari uji coba dengan menggunakan plant pengaturan temperatur yang didownload pada website Siemens, terlihat bahwa temperatur memasuki titik *steady state* dalam waktu kurang lebih 1 menit.

5.2. Saran

Dalam mengembangkan sistem yang cukup besar seperti sebuah *automatic climate control* menggunakan PLC, dibutuhkan waktu, pemahaman, dan usaha, serta sumber daya yang cukup. Diharapkan kedepannya penelitian ini dapat digunakan sebagai pengembangan dan referensi bagi peneliti yang hendak meneliti PLC dengan metode Huffman, maupun untuk mengembangkan *automatic climate control* menggunakan PLC.

DAFTAR PUSTAKA

1. SIMATIC S7-1200/S7-1500 and STEP 7, “Multi-Zone Control with PID_Temp“, Siemens, 2019
2. STEP 7 V13 SP1, “Library for Controlled System Simulation with STEP 7 (TIA Portal)”, Siemens, 2017
3. D.Pessen, John Wiley & Sons, “Industrial Automation”, Technion – Israel Institute of Technology, Part A, 1989.
4. Hanon System, “What is Climate Control” <URL: https://www.hanonsystems.com/En/Technology/WhatisClimate>, 2019.
5. Bolton W. “Programmable Logic Controllers, 5th Edition”, London: Elsevier Newnes.

LAMPIRAN

```
1  //==================================================================
2  // SIEMENS AG
3  // (c)Copyright 2016 All Rights reserved
4  //------------------------------------------------------------------
5  // Tested with:   S7-1200 CPU FW >= V4.2 or S7-1500 CPU FW >= V2.0
6  // Engineering:   STEP7 V14 (TIA Portal)
7  // Restrictions:  OB cyclic interrupt
8  // Requirements:  SIMATIC Memory Card for using S7-1500 CPU
9  // Functionality: Simple simulation of coupled thermal zones
10 //------------------------------------------------------------------
11 // Change log table:
12 // Version  Date         Expert in charge    Changes applied
13 // 01.00.00 12.10.2016  DF CS SD CCC OS1    First released version
14 //==================================================================
15 // Inputs:
16 //              heating and cooling input (0..100%) for all zones
17 //              via #instPidTempX.OutputHeat and #instPidTempX.OutputCool
18 // Outputs:
19 //              the simulated temperatur for all zones
20 //              via #zones[X].simMultizone.output
21 // Error handling:
22 //              error source #statusID = z*10 + n*1; z =zone
23 //              n = 1: FB "LSim_PT1" (instHeatZ[z])
24 //              n = 2: FB "LSim_PT1" (instCoolZ[z])
25 //              n = 3: FB "LSim_PT2aper" (instPT2Z[z])
26 //
27 //              Example: #statusID = 21
28 //              ->  error source: FB "LSim_PT1" (instHeatZ[2])
29 //
30 //              error code #status
31 //              -> see comment #status of error source
32 //==================================================================
33
```

Gambar 1 Program Plant Dari Siemens 1

```
34 REGION PT1
35
36   FOR #tempIndex := 1 TO #MAX DO
37     // Simulation of Heating and Cooling actuator logic for each thermal zone
38
39     REGION heat
40       // Zone Heater dynamics (PT1) 0..100%, tmLag1 = 5sek
41       #instHeatZ[#tempIndex](input := #zones[#tempIndex].pidTemp.outputHeat,
42                              tmLag1 := #TMLAG_HEATER,
43                              cycle := #cycle,
44                              reset := #reset);
45
46       IF (#instHeatZ[#tempIndex].error = TRUE) THEN
47         #error := TRUE;
48         #statusID := 10 * DINT_TO_UINT(#tempIndex) + 1;
49         #status := #instHeatZ[#tempIndex].status;
50         RETURN;
51       END_IF;
52     END_REGION
53
54     REGION cool
55       // Zone Cooler dynamics (PT1) 0.100%, tmLag1 = 10sek
56       #instCoolZ[#tempIndex](input := #zones[#tempIndex].pidTemp.outputCool,
57                              tmLag1 := #TMLAG_COOLER,
58                              cycle := #cycle,
59                              reset := #reset);
60
61       IF (#instCoolZ[#tempIndex].error = TRUE) THEN
62         #error := TRUE;
63         #statusID := 10 * DINT_TO_UINT(#tempIndex) + 2;
64         #status := #instCoolZ[#tempIndex].status;
65         RETURN;
66       END_IF;
```

Gambar 2 Program Plant Dari Siemens 2

```
67      END_REGION
68      // generate inputvariable for thermal simulation area zone
69      #statZ[#tempIndex].uInp := #instHeatZ[#tempIndex].output
70      - #instCoolZ[#tempIndex].output
71      + #zones[#tempIndex].simMultizone.disturbance;
72
73    END_FOR;
74  END_REGION
75
76  REGION factors
77    //==========================================================================
78    // calculating the forward and backward coupling factors as a factor of
79    // fore example mass transport
80    //==========================================================================
81    #tempKFor := ((#K_FOR_MAX - #K_INT) / 100.0) * SINT_TO_REAL(#velocity) + #K_INT;
82    #tempKFor := LIMIT(MN := #K_BACK_MIN, IN := #tempKFor, MX := #K_FOR_MAX);
83
84    #tempKBack := ((#K_BACK_MIN - #K_INT) / 100.0) * SINT_TO_REAL(#velocity) + #K_INT;
85    #tempKBack := LIMIT(MN := #K_BACK_MIN, IN := #tempKBack, MX := #K_FOR_MAX);
86
87    FOR #tempIndex := 1 TO #MAX DO
88      #statZ[#tempIndex].kFor := #tempKFor;
89      #statZ[#tempIndex].kBack := #tempKBack;
90    END_FOR;
91  END_REGION
92
93  REGION coupling
94    //==========================================================================
95    // dynamic simulation of at least 3 thermal coupled zones
96    // Connection logic between thermal zones
97    //==========================================================================
98
99    //First zone: previous zone doesn't exist -> (#statZ[z-1].y * #statZ[z-1].kFor)= 0
```

Gambar 3 Program Plant Dari Siemens 3

```
100    #statZ[1].u := #statZ[1].uInp
101    - (2 * (#statZ[1].y * #statZ[1].kInt))
102    + (#statZ[2].y * #statZ[2].kBack);
103
104    FOR #tempIndex := 2 TO (#MAX - 1) DO
105      // 1 < zone < max: #statZ[z].u := #statZ[z].uInp
106      // + (#statZ[z-1].y * #statZ[z-1].kFor)
107      // - (2 * (#statZ[z].y * #statZ[z].kint))
108      // + (#statZ[z+1].y * #statZ[z+1].kBack);
109      #statZ[#tempIndex].u := #statZ[#tempIndex].uInp
110      + (#statZ[(#tempIndex - 1)].y * #statZ[(#tempIndex - 1)].kFor)
111      - (2 * (#statZ[#tempIndex].y * #statZ[#tempIndex].kInt))
112      + (#statZ[(#tempIndex + 1)].y * #statZ[(#tempIndex + 1)].kBack);
113    END_FOR;
114
115    //Last zone: next Zone doesn't exist -> (#statZ[z+1].y * #statZ[z+1].kBack);)= 0
116    #statZ[#MAX].u := #statZ[#MAX].uInp
117    + (#statZ[(#MAX - 1)].y * #statZ[(#MAX - 1)].kFor)
118    - (2 * (#statZ[#MAX].y * #statZ[#MAX].kInt));
119  END_REGION
120
121  REGION PT2
122    FOR #tempIndex := 1 TO #MAX DO
123      // dynamic Simulation of zone (PT2)
124      #instPT2Z[#tempIndex](input := #statZ[#tempIndex].u,
125                            tmLag1 := #TMLAG1_ZONE,
126                            tmLag2 := #TMLAG2_ZONE,
127                            gain := #GAIN_ZONE,
128                            cycle := #cycle,
129                            maxOut := #PT2_MAX_OUT,
130                            minOut := #PT2_MIN_OUT,
131                            reset := #reset,
132                            output => #statZ[#tempIndex].y);
```

Gambar 4 Program Plant Dari Siemens 4

```
133
134     IF (#instPT2Z[#tempIndex].error = TRUE) THEN
135       #error := TRUE;
136       #statusID := 10 * DINT_TO_UINT(#tempIndex) + 3;
137       #status := #instPT2Z[#tempIndex].status;
138       RETURN;
139     END_IF;
140
141     // Output of simulation model adding ambient temperature
142     #zones[#tempIndex].simMultizone.output := #statZ[#tempIndex].y + #ambTemp;
143
144   END_FOR;
145 END_REGION
146
147 FOR #tempIndex := 1 TO #MAX DO
148   //Reached handling (PT1 & PT2)
149   IF #instHeatZ[#tempIndex].maxReached OR #instCoolZ[#tempIndex].maxReached OR #instPT
150     #maxReached := TRUE;
151   END_IF;
152
153   IF #instHeatZ[#tempIndex].minReached OR #instCoolZ[#tempIndex].minReached OR #instPT
154     #minReached := TRUE;
155   END_IF;
156
157 END_FOR;
158
159 IF (#reset = TRUE) THEN
160   #maxReached := FALSE;
161   #minReached := FALSE;
162 END_IF;
```

Gambar 5 Program Plant Dari Siemens 5

RIWAYAT PENULIS

Penulis yang mempunyai nama lengkap Iqbal Muhammad Santoso. Merupakan mahasiswa Teknik Elektro Institut Teknologi Sepuluh Nopember Surabaya. Penulis merupakan anak kedua dari 3 bersaudara yang lahir di Montpellier, Perancis pada tanggal 23 April 1998. Pada bulan juli 2016 penulis diterima di S1 Teknik Elektro melalui jalur Seleksi Bersama Masuk Perguruan Tinggi (SBMPTN). Selama kuliah penulis aktif di organisasi Kendo ITS pada tahun 2017 sebagai salah satu staff divisi internal, yang kemudian menjabat sebagai Ketua Divisi Eksternal pada tahun 2018. Selain aktif di organisasi juga aktif sebagai asisten lab khususnya di Laboratorium Teknik Pengaturan AJ-104. Penulis berharap buku ini dapat dimanfaatkan sebagaimana mestinya. Dalam penulisan buku tugas akhir ini penulis mengharapkan kritik dan saran pembaca untuk perkembangan penelitian lebih lanjut bisa melalui email *iqbalms2304@gmail.com*.