TUGAS AKHIR – EE 184801

Desain *Ladder Diagram* Mengunakan Metode *State Diagram* untuk *Logic Sequence* Proses Pengolahan Batubara pada PLTU

Romadhona Ighfirliyan Namaskara
NRP 07111540000026

Dosen Pembimbing
Dr. Ir. Mochammad Rameli
Eka Iskandar, ST., MT.

DEPARTEMEN TEKNIK ELEKTRO
Fakultas Teknologi Elektro
Institut Teknologi Sepuluh Nopember
Surabaya 2019

FINAL PROJECT – EE 184801

Ladder Diagram Design using State Diagram Method for Logic Sequence of Coal Handling Proccess on Steam Power Plant

Romadhona Ighfirliyan Namaskara
NRP 07111540000026

Supervisor
Dr. Ir. Mochammad Rameli
Eka Iskandar, ST., MT.

ELECTRICAL ENGINEERING DEPARTMENT
Faculty of Electrical Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2019

PERNYATAAN KEASLIAN
TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul "***Desain Ladder Diagram Mengunakan Metode State Diagram untuk Logic Sequence Proses Pengolahan Batubara pada PLTU***" 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, Juni 2019

Romadhona Ighfirliyan Namaskara
NRP 07111540000026

[Halaman ini sengaja dikosongkan]

DESAIN *LADDER DIAGRAM* MENGUNAKAN METODE *STATE DIAGRAM* UNTUK *LOGIC SEQUENCE* PROSES PENGOLAHAN BATUBARA PADA PLTU

TUGAS AKHIR

Diajukan untuk Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik
Pada
Bidang Studi Teknik Sistem Pengaturan
Departemen Teknik Elektro
Institut Teknologi Sepuluh Nopember

Menyetujui :

Dosen Pembimbing I	Dosen Pembimbing II
Dr. Ir. Mochammad Rameli	Eka Iskandar, ST., MT.
NIP. 195412121981031002	NIP.198005282008121001

KEMENTERIAN RISET, TEKNOLOGI, DAN PENDIDIKAN TINGGI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
FAKULTAS TEKNOLOGI ELEKTRO
DEPARTEMEN TEKNIK ELEKTRO

SURABAYA
JUNI, 2019

[Halaman ini sengaja dikosongkan]

Desain *Ladder Diagram* Mengunakan Metode *State Diagram* untuk *Logic Sequence* Proses Pengolahan Batubara pada PLTU

Romadhona Ighfirliyan Namaskara – 07111540000026

Pembimbing : 1.Dr. Ir. Mochammad Rameli
2.Eka Iskandar, ST., MT.

ABSTRAK

Berdasarkan statistik ketenagalistrikan pada tahun 2017, Pembangkit Listrik Tenaga Uap (PLTU) dengan bahan bakar batubara masih akan menjadi tulang punggung kelistrikan di Indonesia dengan kapasitas produksi 30,208.23 MW dari total 60,789.98 MW atau sekitar 50% produksi listrik secara nasional. Berdasarkan data tersebut, maka diperlukan proses pengolahan batubara sebagai bahan bakar utama PLTU sehingga didapatkan efisiensi produksi serta meminimalisir jumlah polusi yang dihasilkan. Proses pengolahan batubara pada PLTU adalah proses yang kompleks dan berulang. Proses yang panjang dan kompleks tersebut dapat dirancang dalam bentuk *ladder diagram* dengan menggunakan metode *state diagram* yang akan diaplikasikan ke PLC sebagai kontroler. Dengan adanya *ladder diagram*, proses akan berjalan secara sekuensial. Desain *ladder diagram* dengan metode state diagram menghasilkan total 46 *rung ladder* dengan rincian 10 *rung relay* proses, 27 *rung* output dan 9 *rung relay* pendukung. Total kapasitas memori untuk file program yang dihasilkan sebesar 11 KB dengan rincian 3 KB file (.BAK), 3 KB file (.CXP), dan 5 KB file (.OPT) dengan waktu rata-rata sepuluh kali pengujian proses keseluruhan adalah 54,4 detik.

Kata kunci : *Coal Handling Process*, *State Diagram Method*, *Ladder Diagram*, PLC.

[Halaman ini sengaja dikosongkan]

LADDER DIAGRAM DESIGN USING STATE DIAGRAM METHOD FOR LOGIC SEQUENCE OF COAL HANDLING PROCCESS ON STEAM POWER PLANT

Romadhona Ighfirliyan Namaskara – 07111540000026

Supervisor : 1.Dr. Ir. Mochammad Rameli
2.Eka Iskandar, ST., MT.

ABSTRACT

Based on electricity statistics in 2017, the Coal fuel Steam Power Plant (PLTU) will still be the backbone of electricity in Indonesia with production capacity of 30,208.23 MW of a total of 60,789.98 MW or around 50% of national electricity production. Based on the data, it is necessary to process coal as the main fuel of the PLTU to obtain production efficiency and minimize the amount of pollution produced. The process of coal handling on PLTU is a complex and repetitive process. The long and complex process can be designed in the form of a ladder diagram using state diagram method that will be applied to the PLC as a controller. With the ladder diagram, the process will run sequentially. Design of ladder diagrams using state diaram method produces a total of 46 rung ladders with details of 10 process relay rung, 27 output rung and 9 supporting relay rung. The total memory capacity for the program files used is 11 KB with details of 3 KB (.BAK), 3 KB (.CXP), and 5 KB (. OPT) with an average time of ten times the overall process testing is 54,4 seconds.

Keywords : *Coal Handling Process, State Diagram Method, Ladder Diagram, PLC.*

[Halaman ini sengaja dikosongkan]

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Tuhan Yang Maha Esa karena berkat rahmat dan karunia-Nya penulis dapat menyelesaikan tugas akhir yang berjudul "***Desain Ladder Diagram Mengunakan Metode State Diagram untuk Logic Sequence Proses Pengolahan Batubara pada PLTU***" Pada kesempatan ini penulis menyampaikan terima kasih atas segala bantuan dan dukungannya yang telah diberikan selama proses pembuatan tugas akhir ini kepada :

1. Orang tua tercinta serta keluarga dan kerabat yang senantiasa memberikan doa serta dukungan.
2. Bapak Mochamad Rameli selaku dosen pembimbing yang selalu memberikan pengarahan, saran dan motivasi dalam kelancaran tugas akhir ini.
3. Bapak Eka Iskandar selaku dosen pembimbing yang telah banyak memberikan ilmu dan bimbingan selama penulis mengerjakan tugas akhir ini.
4. Seluruh dosen dan karyawan di Teknik Elektro Institut Teknologi Sepuluh Nopember.
5. Seluruh teman-teman e55 dan mahasiswa Lintas Jalur Teknik Sistem Pengaturan.
6. Semua pihak yang telah membantu penulis dalam menyelesaikan tugas akhir ini yang tidak dapat disebutkan satu per satu.

Penulis menyadari masih banyak kekurangan dalam tugas akhir ini. Kritik dan saran untuk perbaikan tugas akhir ini sangat diperlukan. Semoga tugas akhir ini dapat bermanfaat bagi semua pihak.

Surabaya, Juni 2019

Romadhona Ighfirliyan Namaskara
NRP 07111540000026

[Halaman ini sengaja dikosongkan]

DAFTAR ISI

PERNYATAAN KEASLIAN TUGAS AKHIR v
LEMBAR PENGESAHAN vii
ABSTRAK ix
ABSTRACT xi
KATA PENGANTAR xiii
DAFTAR ISI xv
DAFTAR GAMBAR xvii
DAFTAR TABEL xix
BAB I 1
1.1 Latar Belakang [1] 1
1.2 Permasalahan 2
1.3 Batasan Masalah 2
1.4 Tujuan 3
1.5 Metodologi 3
1.6 Sistematika 4
1.7 Relevansi dan Manfaat 4
BAB 2 5
2.1 Proses Pengolahan batubara PLTU [2] 5
2.1.1 *Coal transfer* 5
2.1.2 *Water Feeding* 8
2.1.3 *Burning Process* 9
2.1.4 *Bottom Ash Handling* [3] 11
2.1.5 *Fly Ash Handling* [5] 12
2.2 *Programmable Logic Controller* (PLC) [6] 13
2.2.1 Komponen Dasar PLC 14
2.2.2 Bahasa Pemrograman PLC [7] 15
2.2.4 *Addressing* dan Instruksi Dasar PLC Omron 18
2.2.5 PLC OMRON CP1E 20
2.3 *Human Machine Interface* 20
2.4 *State Diagram* [8] 21
2.4.1 Penentuan Urutan Proses 22
2.4.2 Pendiskripsian *Input* dan *Output* 23
2.4.3 Pembuatan *State Diagram I/O* 23

2.4.4 Pembuatan *Primitive Flow Table* 24
2.4.5 Penyusunan *Merged Flow Table* 25
2.4.6 Pembuatan *State Diagram R/O* 27
2.4.7 Konversi *Ladder Diagram* ke *Switching Function* 28
2.4.8 Pembuatan *Ladder Diagram* 30
BAB 3 31
3.1 Perancangan *Ladder Diagram* dengan Metode *State Diagram* 31
3.1.1 Pendeskripsian Urutan Kerja Proses 31
3.1.2 Pendeskripsian *Input* dan *Output* Sistem 36
3.1.3 Pembuatan *State Diagam* I/O 40
3.1.4 Penyusunan Primitive Flow Table 40
3.1.5 Penyusunan Merged Flow Table 41
3.1.6 Penyusunan *State Diagram* (R/O) 42
3.1.7 Perancangan *Switching Function* 43
3.1.8 Konversi ke Ladder Diagram 43
3.1.9 Initial System 44
3.2 Perancangan *Human Machine Interface* 45
3.3 Perancangan Integrasi Hardware 46
BAB 4 47
4.1 Pengujian Sistem 47
4.1.1 Unit Testing 47
4.1.2 *Operational Testing* 54
4.1.3 Pengujian Integrasi Sistem 56
4.2 Analisa Hasil 63
4.2.1 Waktu Proses 64
BAB 5 67
5.1 Kesimpulan 67
5.2 Saran 67
DAFTAR PUSTAKA 69
LAMPIRAN 71
RIWAYAT HIDUP 111

DAFTAR GAMBAR

Gambar 2.1 *Coalpile* 6
Gambar 2.2 Stacker/Reclaimer 6
Gambar 2.3 *Belt Conveyor* 7
Gambar 2.4 *Coal Silo* 8
Gambar 2.5 Diagram General Boiler 10
Gambar 2.6 Submerged Scraper Conveyor (SSC) 12
Gambar 2.7 Programmable Logic Control 13
Gambar 2.8 Komponen Dasar PLC 14
Gambar 2.9 Contoh Ladder Diagram 16
Gambar 2.10 Contoh *Function Block Diagram* 16
Gambar 2.11 Contoh Structured Text 17
Gambar 2.12 Contoh *Instruction List* 17
Gambar 2.13 *Contoh* Sequential Function Chart 18
Gambar 2.14 PLC CP1E 20
Gambar 2.15 CX Designer 21
Gambar 2.16 State Diagram 21
Gambar 2.17 Langkah Pembuatan *Ladder Diagram* dengan *State Diagram* 22
Gambar 2.18 Contoh *state diagr*am I/O 24
Gambar 2.19 Contoh state diagram R/O 28
Gambar 2.20 Contoh *ladder diagram* hasil *switching function* 30
Gambar 3.1 State Diagram input/output 40
Gambar 3.2 *State Diagram (R/O)* Sub-Proses 1 42
Gambar 3.3 Desain *Ladder Diagram* Sub-proses 1 44
Gambar 3.4 *Ladder Diagram Initial System* 45
Gambar 3.5 Tampilan menu Coal Transfer 46
Gambar 3.6 Integrasi PLC, *input* dan *output* 46
Gambar 4.1 Pengujian *Ladder Diagram* Langkah 1 pada sub-proses 1 49
Gambar 4.2 Pengujian *Ladder Diagram* Langkah 2 pada sub-proses 1 49
Gambar 4.3 Pengujian *Ladder Diagram* Langkah 3 pada sub-proses 1 50
Gambar 4.4 Pengujian *Ladder Diagram* Langkah 4 pada sub-proses 1 51
Gambar 4.5 Pengujian *Ladder Diagram* Langkah 5 pada sub-proses 1 51

Gambar 4.6 Pengujian *Ladder Diagram* Langkah 6 pada sub-proses 1 52
Gambar 4.7 Rancangan Sequence Chart sub-proses 1.....52
Gambar 4.8 Hasil pengujian dengan fluidsim.....53
Gambar 4.9 Langkah 1 sub proses 1.....59
Gambar 4.10 Langkah 2 sub proses 1.....59
Gambar 4.11 Langkah 3 sub proses 1.....60
Gambar 4.12 Langkah 4 sub proses 1.....60
Gambar 4.13 Langkah 5 sub proses 1.....61
Gambar 4.14 Langkah 6 sub proses 1.....61
Gambar 4.15 Grafik Pengujian Waktu Proses Keseluruhan 65

DAFTAR TABEL

Tabel 2.1 Lebar Alamat Pada PLC Omron 18
Tabel 2.2 Instruksi Dasar Pada PLC 19
Tabel 2.3 Penentuan Urutan Proses..... 23
Tabel 2.4 Pendeskripsian Input 23
Tabel 2.5 Contoh pengisian stabil, tidak stabil, dan *don't care* 25
Tabel 2.6 Contoh pengisian *output* 25
Tabel 2.7 Contoh penggabungan baris 26
Tabel 2.8 Contoh penentuan *relay* 27
Tabel 3.1 Deskripsi Urutan per sub-proses 34
Tabel 3.2 *Input* sistem 36
Tabel 3.3 *Output* Sistem..... 38
Tabel 3.4 Primitive Flow Table Sub-proses 1 41
Tabel 3.5 Tabel merged flow table sub-proses 1..... 42
Tabel 4.1 Hasil Pengujian *Board Toggle* 47
Tabel 4.2 Hasil pengujian Digital Input PLC..... 53
Tabel 4.3 Hasil pengujian Digital Output PLC 54
Tabel 4.4 Hasil Pengujian *Board Toggle* dan PLC 54
Tabel 4.5 Hasil Pengujian Output dari PLC ditampilkan ke HMI 55
Tabel 4.6 Hasil Pengujian Integrasi Board Toggle dengan HMI 56
Tabel 4.7 Hasil Pengujian Output dari PLC ditampilkan ke HMI 57
Tabel 4.8 Pengujian Sistem Keseluruhan..... 61
Tabel 4.9 Hasil pencatatan waktu pengujian implementasi pada PLC.. 64

[Halaman ini sengaja dikosongkan]

BAB I
PENDAHULUAN

Bab ini mengantarkan pembaca untuk dapat menjawab pertanyaan mengapa, apa yang diteliti, untuk apa suatu penelitian dilakukan. Jawaban pertanyaan tersebut akan diuraikan pada bab ini yang meliputi latar belakang, perumusan masalah, tujuan, metodologi, sistematika dan relevansi.

1.1 Latar Belakang [1]

Saat ini energi primer yang digunakan untuk pembangkit listrik di Indonesia masih didominasi oleh bahan bakar fosil seperti batubara, gas alam, dan minyak. Energi fosil digunakan oleh pembangkit listrik sebagai bahan bakar yang digunakan untuk memanaskan uap pada *boiler*. Uap tersebut kemudian dipakai untuk memutar turbin yang dikopel langsung dengan generator sinkron sehingga dapat menghasilkan energi listrik.

Pada tahun 2015 produksi energi listrik oleh pembangkit listrik di Indonesia masih didominasi energi fosil sebagai bahan bakar primernya dengan komposisi 54% batubara, kemudian disusul gas 21%, dan minyak 2% (RUPTL 2015- 2024). Di tahun 2024 nanti, proyeksi komposisi produksi energi listrik per jenis energi primer Indonesia akan menjadi 63,7% batubara, 19,2% gas alam, 9% panas bumi, 6,6% tenaga air, 1,5% minyak dan bahan bakar lainnya.

Berdasarkan statistik ketenagalistrikan pada tahun 2017, Pembangkit Listrik Tenaga Uap dengan bahan bakar batubara masih akan menjadi tulang punggung produksi listrik di Indonesia dengan kapasitas produksi 30,208.23 MW dari total 60,789.98 MW atau sekitar 50% produksi listrik secara nasional. Berdasarkan data tersebut, maka diperlukan proses pengolahan batubara sebagai bahan bakar PLTU sehingga didapatkan efisiensi produksi serta meminimalisir jumlah polusi yang dihasilkan.

Proses-proses pengolahan batubara PLTU secara umum terdiri dari proses transfer batubara (*coal transfer*), proses pengisian air ke dalam boiler untuk dipanaskan menjadi uap air yang menggerakkan turbin (*water feeding*), proses pembakaran bahan bakar minyak sebagai bahan bakar awal sebelum diganti dengan batubara (*burning process*), proses penanganan sisa batubara yang tidak ikut terbakar dan jatuh di dasar boiler (*bottom ash handling*), serta proses penanganan debu-debu sisa

pembakaran (*ash handling*) sebelum pada akhirnya dibuang ke atmosfer melalui *stack.* Semua proses otomasi ini akan berjalan secara sekuensial.

Dalam pengoperasian suatu *plant*, umumnya terdapat tiga jenis sistem pengoperasian yang diterapkan. Yang pertama adalah mode manual di mana proses pengoperasian dikontrol langsung oleh operator. Yang kedua adalah pengoperasian semi-otomatis di mana terdapat tahap-tahap tertentu yang harus dikontrol manual oleh operator. Yang ketiga adalah proses pengoperasian secara otomatis dengan operator yang melakukan pemantauan secara *realtime* di ruang kendali. Apabila terjadi masalah pada suatu proses, maka akan langsung terbaca oleh sistem kontrol sehingga titik masalah dapat lebih mudah ditemukan dan diperbaiki. Keseluruhan proses dapat dilakukan secara otomatis dengan menggunakan PLC sebagai kontroler. PLC sendiri merupakan sebuah alat yang dapat diprogram, dikontrol, dan dioperasikan sesuai dengan kebutuhan operasi. PLC diprogram dengan bahasa pemrograman *ladder diagram*. Dengan adanya *ladder diagram* ini, proses yang dilakukan dapat berjalan secara sekuensial.

Namun, dengan proses yang kompleks tentu akan menggunakan banyak relay sedangkan terbatasnya jumlah relay pada PLC yang memiliki spesifikasi rendah. Oleh karena itu, diperlukan suatu metode untuk meminimalkan penggunaan relay yang dapat digunakan dalam pemrograman PLC. Dengan meminimalkan jumlah penggunaan relay juga akan membuat program menjadi lebih singkat dan mudah dipahami.

1.2 Permasalahan

Permasalahan pada tugas akhir ini adalah bagaimana mendesain *ladder diagram* dengan menggunakan *State Diagram* pada proses pengolahan bahan bakar di PLTU

1.3 Batasan Masalah

Dari permasalahan tersebut, terdapat beberapa batasan permasalahan dalam Tugas Akhir ini, di antaranya adalah:

1. Proses pengolahan batubara PLTU ini tidak sedetail pada PLTU asli.
2. Proses kontruksi *ladder diagram* hanya menggunakan metode *state diagram*
3. *Controller* yang digunakan adalah PLC OMRON CP1E
4. *Human Machine Interace* yang digunakan adalah *CX-Designer*

1.4 Tujuan

Tujuan dari tugas akhir ini adalah mendesain *Ladder Diagram* dengan metode *State Diagram* untuk *logic sequence* pada proses pengolahan batubara di PLTU.

1.5 Metodologi

Pada pengerjaan Tugas Akhir, yang dilakukan pertama kali adalah mencari literatur mengenai Pembangkit Listrik Tenaga Uap (PLTU) khususnya proses pengolahan batubara pada PLTU. Literatur didapat dari data pembangkit listrik atau melalui paper dan buku bacaan. Selain itu juga dilakukan wawancara kepada *engineer* lapangan serta ke dosen pembimbing dalam bentuk bimbingan. Hasil studi akan digunakan untuk mengidentifikasi sistem sehingga didapatkan urutan proses serta *Input* dan *Output* yang digunakan pada proses pengolahan batubara PLTU. *Input* dan *Output* tersebut akan digunakan dalam perancangan *ladder diagram* dan diimplementasikan pada PLC. Selain itu dilakukan Studi Literatur mengenai perancangan *ladder diagram* menggunakan Metode *State Diagram* untuk menyelesaikan permasalahan perancangan ladder yang sistematis.

Selanjutnya adalah penerapan studi literatur yaitu membuat ladder diagram menggunakan metode *State Diagram* berdasarkan data dari urutan proses serta *Input* dan *Output* proses pengolahan batubara PLTU. Apabila telah didapatkan hasil *ladder diagram* yang sistematis maka dapat diimplementasikan pada PLC. Langkah selanjutnya adalah merancangan *Human Machine Interface* (HMI) sebagai visualisasi dari *Plant*. Setelah itu dilakukan proses implementasi dan pengujian dengan cara menghubungkan PLC yang telah diprogram dengan *laddder diagram* hasil perancangan dengan HMI.

Apabila hasil *Ladder Diagram* dan HMI yang telah dirancang sesuai dengan urutan proses pada saat studi literatur, maka dapat dilakukan pengambilan data. Namun apabila terdapat ketidaksesuaian dengan urutan proses maka dilakukan perancangan *ladder diagram* kembali hingga didapatkan hasil terbaik yang bisa menjawab rumusan permasalahan.

Tahap akhir pengerjaan adalah pembuatan laporan dari hasil pengujian yang telah sesuai dengan studi literatur. Penyusunan laporan

tugas akhir yang meliputi pendahuluan, tinjauan pustaka, perancangan sistem, simulasi dan analisa sistem dan penutup.

1.6 Sistematika

Penulis membagi laporan penelitian ini menjadi lima bab yang terhubung satu sama lain. Hal ini untuk menghindari kesalahan interpretasi terhadap isi yang terdapat isi yang terdapat di dalam laporan. Penjelasan tentang masing-masing bab dibuat dengan sistematika penulisan sebagai berikut :

BAB I : PENDAHULUAN

Bab ini meliputi latar belakang, perumusan masalah, batasan masalah, tujuan, metodologi, sistematika dan relevansi.

BAB II : TINJAUAN PUSTAKA

Bab ini membahas tinjauan pustaka yang membantu penelitian, di antaranya adalah teori *State Diagram*, penjelasan mengenai proses pengolahan bahan bakar PLTU, penjelasan mengenai dasar PLC, serta HMI.

BAB III : PERANCANGAN SISTEM

Bab ini membahas perancangan sistem yang meliputi perancangan *ladder diagram* untuk proses pengolahan batubara PLTU dengan metode *State Diagram* serta perancangan dari *Human Machine Interface* sehingga membantu pembaca dalam memahami tahapan dari setiap proses dalam sistem yang dirancang.

BAB IV : HASIL DAN ANALISA

Bab ini membahas pengujian sistem serta analisa data perancangan sistem yang berupa hasil desain *Ladder Diagram.*

BAB V : PENUTUP

Berisi penjelasan tentang kesimpulan yang didapat dari tugas akhir ini dan saran-saran dari hasil penelitian yang dilakukan.

1.7 Relevansi dan Manfaat

Hasil dari Tugas Akhir ini diharapkan dapat berkontribusi sebagai pengimpelementasian maupun referensi dalam pengembangan penelitian tentang desain *ladder diagram* dengan menggunakan metode S*tate Diagram* pada proses pengolahan batubara PLTU, baik untuk proses pembelajaran maupun untuk proses di Industri di masa yang akan datang.

BAB 2
TINJAUAN PUSTAKA

Pada bab ini akan dijelaskan hal-hal yang berkaitan dengan proses pengolahan batubara PLTU, metode *State Diagram*, *Programmable Logic Control,* serta HMI.

2.1 Proses Pengolahan batubara PLTU [2]

Proses pengolahan batubara PLTU ini adalah proses bagaimana batubara diolah sebagai bahan bakar utama pada PLTU hingga penanganan sisa pembakarannya agar polusi yang ditimbulkan dapat diminimalisir. Proses-proses pengolahan batubara PLTU sejatinya adalah proses yang sangat kompleks. Namun secara umum terdiri dari lima proses transfer batubara (*coal transfer*), proses pengisian air ke dalam boiler untuk dipanaskan menjadi uap air yang menggerakkan turbin (*water feeding*), proses pembakaran bahan bakar minyak sebagai bahan bakar awal sebelum diganti dengan batubara (*burning process*), proses penanganan sisa batubara yang tidak ikut terbakar dan jatuh di dasar boiler (*bottom ash handling*), serta proses penanganan debu-debu sisa pembakaran (*ash handling*) sebelum pada akhirnya dibuang ke atmosfer melalui *stack.* Semua proses otomasi ini akan berjalan secara sekuensial.

2.1.1 *Coal transfer*

Batubara merupakan bahan bakar utama pada PLTU. Batubara yang diambil dari pertambangan akan terus disuplai selama pengoperasian. Pengiriman batubara ke *plant* dilakukan dengan menggunakan kapal laut pengangkut batubara (tongkang) sebelum pada akhirnya ditimbun di *coalpile*. Proses transfer batubara pada PLTU dilakukan dari *coalpile* menuju ke *Silo* sebagai penampung batubara sebelum digunakan untuk pembakaran. Adapun komponen utama pada *coal transfer* ini adalah sebagai berikut :

2.1.1.1 *Coalpile*

Coalpile adalah tempat penampungan batubara yang memiliki kapasitas ratusan ribu ton. Di *Coalpile*, proses penimbunan dan pengambilan batubara dilakukan dengan alat yang disebut *Stacker/Reclaimer*.

Gambar 2.1 *Coalpile*

2.1.1.2 *Stacker/Reclaimer*

Stacker/Reclaimer merupakan sebuah konveyor yang kompleks dan terpasang pada sebuah struktur yang dapat bergerak. Di dalam proses penimbunan, *stacker* menyalurkan batubara melalui sebuah lengan yang dapat diatur agar selalu diam ditempat, sehingga batubara yang tumpah melalui lengan itu akan membentuk timbunan yang tinggi, apabila lengan bergerak maju mundur maka timbunan yang akan dihasilkan menjadi timbunan yang rapi dan memanjang. Pada saat pengambilan, *Reclaiming Bucket* akan berputar dan mengeruk batubara yang selanjutnya dituang ke *Belt Conveyor* untuk dibawa ke instalasi. Seperti halnya proses penimbunan, *Reclaiming Bucket* ini dapat juga diatur agar tetap diam di tempat atau maju mundur untuk mengeruk batubara.

Gambar 2.2 Stacker/Reclaimer

2.1.1.3 *Belt Conveyor*

Belt conveyor adalah ban berjalan yang digunakan untuk mentransfer batubara dari suatu tempat ke tempat lain secara kontinyu. Merupakan peralatan utama pada sistem *coal transfer*, digunakan untuk memindahkan batubara dalam jumlah besar, jarak yang jauh dan *rate* yang tinggi.

Gambar 2.3 *Belt Conveyor*

2.1.1.4 Crusher Building [5]

Batubara yang diangkut dengan menggunakan *belt conveyor* akan melewati *crusher building. Crusher Building* merupakan tempat penghancuran batubara yang menggumpal di *coalpile* menjadi lebih kecil dengan ukuran 300 mm sampai 20 mm. Hal ini bertujuan agar batubara lebih mudah dihaluskan pada saat di *pulverizer*. Pada *crusher building* juga dilengkapi dengan *magnetic separator* yang berfungsi untuk memisahkan logam yang masih terdapat pada batubara.

2.1.1.5 *Coal Silo*

Setelah batubara keluar dari *crusher building*, batubara kemudian ditransfer oleh *belt conveyor* menuju *coal silo*. *Coal silo* adalah tempat penyimpanan sementara batubara yang telah melewati proses transfer batubara yang telah berukuran seragam. Volume sebuah silo sebesar 600 Ton. Dari *silo*, batubara dimasukkan ke *Pulverizer* dengan menggunakan *Coal Feeder*, batubara dari *Pulverizer* inilah yang akan digunakan untuk pembakaran di Boiler.

Gambar 2.4 *Coal Silo*

2.1.1.6 *Pulverizer*

Bongkahan–bongkahan batubara yang seperti batu harus dihancurkan menjadi butiran-butiran halus agar batubara mudah tercampur dengan udara. *Pulverizer* adalah alat untuk menggiling batubara sehingga menjadi halus dan kemudian bersama dengan udara primer akan dialirkan ke *Furnace*. Fungsi lain dari *Pulverizer* adalah untuk mengeringkan batubara sehingga mudah dihaluskan dan dibakar, dan untuk mengklasifikasikan atau menyaring batubara untuk memastikan bahwa batubara yang masuk ke dalam Boiler benar-benar halus. Dalam penggunaan *Pulverizer* yang perlu diperhatikan adalah suhu dari udara *primer*, suhu yang terlalu tinggi dapat menyalakan batubara dari dalam *Pulverizer* dan menyebabkan ledakan. Jika suhu terlalu rendah, batubara tidak bisa kering benar dan sulit dihaluskan. Suhu idealnya kurang lebih 65°C. *Pulverizer* dilengkapi dengan *Feeder* (alat pengisi batubara) yang letaknya di atas *Pulverizer*, berfungsi untuk menyuplai sejumlah batubara sesuai dengan kebutuhaan. *Feeder* ini mendapat suplai batubara dari penampung batubara *Silo*.

2.1.2 *Water Feeding*

Untuk membangkitkan listrik pada PLTU, uap air diperlukan untuk menggerakan turbin yang terkopel dengan generator. Uap air berasal dari

air laut yang telah dimurnikan melalui proses desalinasi. Air hasil desalinasi ditampung dan diolah terlebih dahulu di Deaerator sebelum dipanaskan di Boiler. Adapun komponen utama pada *water feeding* ini adalah sebagai berikut :

2.1.2.1 Deaerator

Berfungsi untuk menyerap atau menghilangkan gas–gas yang terkandung pada air pengisi Boiler, terutama gas O2, karena gas ini akan menimbulkan korosi. Gas–gas lain yang cukup berbahaya adalah karbon dioksida (CO2). Gas O2 dan CO2 akan bereaksi dengan material Boiler dan menimbulkan korosi yang sangat merugikan.

2.1.2.2 Steam Drum

Pada proses *start up*, air terlebih dahulu dipompa ke *steam drum* melewati *economizer* untuk mendapatkan tambahan panas dari proses pembakaran di Boiler. Namun karena PLTU belum terjadi pembakaran, sehingga air yang melewati *economizer* tidak mendapatkan tambahan panas sehingga langsung menuju *steam drum* sebelum dipanaskan di boiler. Air dari *steam drum* dialirkan menuju *waterwall* untuk dilakukan proses pemanasan dengan bantuan *Boiler Water Circulating Pump*. *Steam drum* berfungsi untuk menyimpan air dalam volume yang besar dan untuk memisahkan uap dari air setelah proses pemanasan yang terjadi dalam boiler.

2.1.3 *Burning Process*

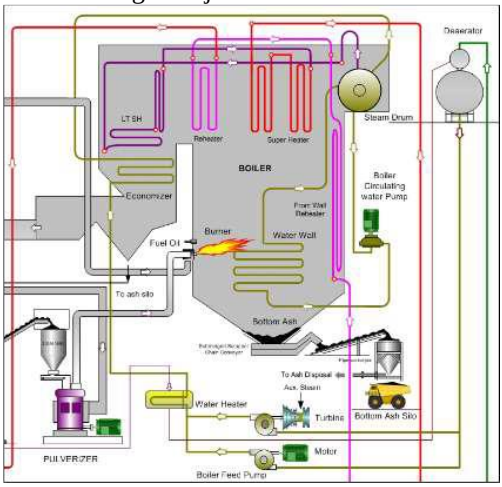
Ketika pertama kali *start up*, PLTU menggunakan bahan bakar berupa minyak (*HSD oil*) yang berasal dari HSD Tank. Setelah beban generator telah mencapai 30%, barulah terjadi pergantian pembakaran yang dilakukan oleh batubara yang telah menjadi serbuk halus di *Pulverizer*. Adapun komponen utama pada *burning process* ini adalah sebagai berikut :

2.1.3.1 *HSD Tank*

Pembangkitan listrik dengan turbin uap hanya menggunakan HSD *oil* untuk tujuan penyalaan awal sebelum digantikan oleh batubara gerusan, sehingga kapasitas penyimpanan (*main storage tank*) tidak terlalu besar. Sedangkan pada pembangkitan listrik dengan turbin gas, *HSD oil* merupakan bahan bakar utama selain gas alam sehingga kapasitas *main storage tank* cukup besar. *HSD oil* dipompa menuju *boiler* dengan menggunakan *HSD oil pump*. Setelah beban mencapai generator telah mencapai 30% barulah *HSD oil pump* mati.

2.1.3.2 *Boiler*

Salah satu komponen utama dari sistem pembangkit listrik tenaga uap adalah *boiler*. *Boiler* berfungsi untuk memanaskan air hingga menjadi uap yang kemudian akan memutar turbin dan generator agar dapat menghasilkan listrik. Dalam pembangkit listrik, uap yang dihasilkan boiler harus berupa uap kering, yaitu uap bersuhu tinggi yang tidak mudah kembali ke bentuk asalnya sebagai titik-titik air jika mengalami penurunan suhu. Air murni memiliki titik didih 100°C pada tekanan 1 atm. Uap yang dihasilkan dari proses sederhana tersebut adalah uap air basah, artinya dapat dengan mudah dan cepat berubah kembali menjadi air jika temperatur di lingkungan mengalami penurunan. Dalam bentuk uap kering, uap tersebut tidak berubah kembali menjadi air sebelum sampai di condenser. Siklus Rankine adalah siklus termodinamika mengenai konversi energi panas menjadi energi kinetik pada tekanan dan temperatur tertentu. Dalam pembangkit, siklus ini di aplikasikan pada perubahan air menjadi uap lalu kembali lagi menjadi air.



Gambar 2.5 Diagram General Boiler

2.1.3.3 *ID Fan, FD Fan* dan *PA Fan*

Udara pembakaran ada dua macam, yaitu *Primary Air* (udara primer) dan *Secondary Air* (udara sekunder). Udara primer dipasok oleh *Primary Air Fan* (PA Fan) yang dihembuskan menuju ke alat penggiling batubara (Pulverizer) kemudian bersama-sama dengan serbuk batubara dialirkan ke *Furnace* untuk dibakar (reaksi kimia). Bercampurnya

batubara dan udara dibantu oleh *Dumper* tetap yaitu pengatur pengaduk udara sehingga menimbulkan turbulensi yang memungkinkan terjadinya pembakaran yang efisien. Turbulensi mengacu pada gerakan udara di dalam Furnace, gerakan ini diperlukan karena dapat menyempurnakan pencampuran udara dan bahan bakar.

Udara primer tidak cukup untuk memenuhi kebutuhan turbulensi untuk melakukan pencampuran bahan bakar secara sempurna atau memenuhi kebutuhan akan oksigen untuk pembakaran sempurna. Untuk itulah diperlukan pasokan dari udara sekunder yang dihasilkan oleh *Forced Draft Fan* bersama *Induced Draft Fan*. *Boiler* yang bekerja dengan tekanan yang negatif atau dibawah tekanan atmosfir selalu dilengkapi dengan *Force Draft Fan (FD Fan)* dan *Induced Draft Fan (ID Fan)*.

2.1.3.2 *Ignitor*

Panas yang diperlukan untuk pembakaran disediakan oleh *Ignitor*. Begitu pembakaran dimulai, bahan bakar yang terbakar akan memasok panas yang cukup untuk menyalakan bahan bakar baru yang memasuki *Boiler* dan *Ignitor* dapat dimatikan.

2.1.4 *Bottom Ash Handling* [3]

Bottom Ash System adalah sistem *Ash Handling Plant* yang khusus menangani/menyalurkan abu sisa pembakaran yang jatuh ke bagian bawah ruang bakar. Adapun komponen utama pada *Bottom Ash Handling* ini adalah sebagai berikut :

2.1.4.1 *Submerged Scraper Conveyor (SSC)*

Submerged Scraper Conveyor (SSC) adalah peralatan yang berfungsi sebagai penampung dan penyaluran abu batubara sisa pembakaran yang berasal dari dalam ruang bakar. SSC menampung dan menyalurkan/memindahkan abu ke pembuangan terakhir melalui *Vibrating Screen* (penyaring), *Crusher* (penghancur) serta *conveyor.* SSC diisi air yang berasal dari *Discharge Water* dengan menggunakan *Circulating Water Pump* yang berfungsi sebagai pendingin abu yang jatuh dari ruang bakar. Agar terjaga level air di bak *SCC*, maka disuplai air secara terus menerus dan dilengkapi pompa-pompa yang mensirkulasikan airnya (*Cooling Water System*), sehingga level dan temperaturenya tetap terjaga (suhu tidak tinggi).

Prinsip kerja dari SCC adalah menampung abu yang berasal dari ruang bakar di bak SDCC, kemudian SDCC menyalurkan abu yang tertampung menuju Conveyor setelah melalui penyaringan (*Vibrating*

Screen) dan penggilingan (*Crusher*) sehingga abu tidak merusak *belt conveyor*. Selanjutnya *Conveyor* mengangkut abu menuju tempat penangan abu (*Ash Disposial*).

Gambar 2.6 Submerged Scraper Conveyor (SSC)

2.1.5 *Fly Ash Handling* [5]

Fly Ash Handling adalah proses dari PLTU berbahan bakar batubara untuk menangani debu sisa pebakaran batubara. Proses *Fly Ash Handling* pada PLTU dilakukan oleh *Fly Ash Handling Plant* yang berada dalam sistem aliran gas buang dan memiliki peralatan penangkap abu yang dibangun menyatu dengan aliran bahan bakar/gas buang. Adapun komponen utama pada *fly ash handling* ini adalah sebagai berikut :

2.1.5.1 *Electrostatic Precipitator*

Electrostatic Precipitator digunakan untuk memisahkan dan mengumpulkan partikel debu didalam flue gas yang dihasilkan dari pembakaran di dalam furnace. *Flue gas* memasuki EP melalui inlet *nozzle* dan disebar secara merata oleh *Gas Distribution Plate*. Partikel debu di dalam *flue gas* diberi muatan *negative corona discharge* yang disuplai dari *High Voltage Transformer-Rectifier Unit* melalui *Dischare Electrodes*, sehingga partikel tersebut akan menempel pada *Collecting Electrodes*. Partikel yang menempel pada *Collecting Electrodes* tersebut akan dijatuhkan dengan cara dihentak oleh hammering system. Debu yang jatuh tersebut nantinya akan dibawa menuju *Ash Silo* dengan bantuan *Vacuum Blower*.

2.1.5.2 *Flue Gas Desulfurization (FGD)*

Setelah melewati EP, flue gas akan kembali dialirkan menuju *Flue Gas Desulfurization (FGD)*. Di dalam FGD, dilakukan penyemprotan air laut terhadap *flue gas*, di mana air laut tersebut nantinya akan

mengikat kandungan SO2 dalam flue gas sebelum dilepaskan ke udara bebas. *Flue gas* yang sudah mengalami banyak kehilangan kandungan *fly ash* dan SO2 kemudian dialirkan menuju *stack* untuk segera dikeluarkan ke udara bebas. *Stack* yang digunakan memiliki ketinggian 220 meter untuk keamanan daerah sekitar.

2.2 *Programmable Logic Controller* (PLC) [6]

Programmable Logic Control (PLC) merupakan piranti elektronik yang berfungsi sebagai pengendali yang berbasis mikroprosesor. PLC memanfaatkan memori yang dapat diprogram untuk menyimpan instruksi-instruksi serta mengimplementasikan fungsi seperti logika, *sequence*, *timing*, *counting* dan operasi aritmatika yang berfungsi untuk mengendalikan mesin-mesin dan proses-proses. Pada PLC, terdapat sebuah program awal yang biasa dinamakan *preprogram* yang memungkinkan program-program kontrol dimasukkan menggunakan suatu bentuk bahasa pemrograman yang sederhana dan intuitif sehingga dapat dengan mudah diprogram oleh para operator yang hanya memiliki sedikit pengetahuan mengenai komputer dan bahasa pemrograman.

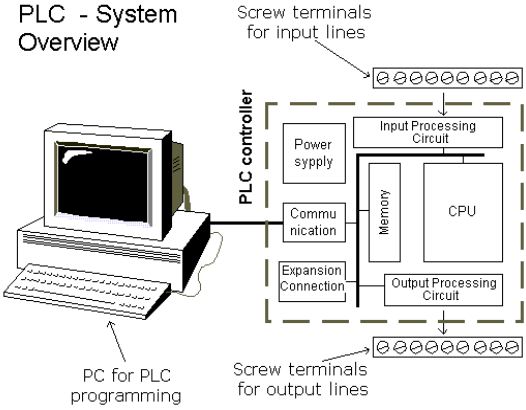
Gambar 2.7 Programmable Logic Control

PLC memiliki dua jenis, yaitu *compact* dan *modular*. PLC *compact* adalah jenis PLC yang seluruh komponennya telah terintegrasi sehingga tidak dapat ditambah lagi jumlah modul *input* dan *output*-nya maupun modul khusus. Sebaliknya, PLC *modular* adalah jenis PLC yang komponennya terpisah menjadi beberapa modul-modul, sehingga dapat ditambahkan lagi modul *Input* dan *output* maupun modul khusus. Secara umum, PLC mempunyai lima komponen dasar, di antaranya adalah *power supply*, CPU, modul Input-Output, dan modul komunikasi.

Pada tugas akhir ini, PLC digunakan sebagai kendali dari proses pengolahan Batubara PLTU. PLC yang digunakan adalah PLC Omron tipe *compact* dengan tipe perangkat CQM1 dan CPU 21.

2.2.1 Komponen Dasar PLC

Pada Gambar 2.8, diketahui bahwa PLC bisa terhubung dengan peralatan *input/output* melalui terminal *output* dan PC (untuk kebutuhan pemrograman).



Gambar 2.8 Komponen Dasar PLC

Secara umum PLC tersusun oleh beberapa komponen diantaranya :

1. *Power supply*

   Power supply mengubah suplai masukan listrik menjadi suplai listrik yang sesuai dengan CPU. Daya untuk PLC dapat berupa tegangan AC sebesar 120/240 VAC maupun tegangan DC sebesar 24 VDC. Selain itu PLC memiliki *power supply* internal (24 VDC) yang digunakan untuk menyediakan daya bagi peralatan I/O PLC.

2. *Central Processing Unit* (CPU)

   Bagian ini merupakan otak atau jantung dari PLC, karena bagian ini yang bertugas untuk membaca, mengolah dan mengeksekusi instruksi program yang tersimpan dalam PLC. CPU dapat mengerjakan tugas yang berhubungan dengan operasi logika dan aritmetika karena memiliki elemen kontrol ALU (*Arithmetic and Logic Unit*). Disamping itu CPU juga melakukan pengawasan atas semua operasional kerja PLC, transfer informasi melalui internal bus antar PLC, *memory* dan I/O. Umumnya memori terletak di dalam CPU (satu modul) atau disebut memori *internal*. Apabila terdapat memori eksternal maka itu merupakan memori tambahan.

3. Modul *Input/Output*

Modul ini merupakan bagian yang menerima sinyal elektrik dari *input* fisik (sensor, tombol, dsb) atau komponen lain dan sinyal itu akan dialirkan ke PLC untuk diproses yang lalu akan dialirkan ke *output* fisik (motor, lampu, pompa, dsb). Setiap *input/output* memiliki alamat khusus yang digunakan selama membuat program untuk memonitor aktifitas *input* dan *output* didalam program. Umumnya modul ini sudah terpasang secara *internal* di dalam PLC (ukuran *compact*). Untuk modul I/O yang terpisah dari CPU merupakan PLC modular.

4. Modul Komunikasi

   Koneksi antara CPU dan komputer (PC) diperlukan modul komunikasi agar dapat dilakukan pemrograman pada PLC. Selain itu juga untuk melakukan pemantauan (*monitoring*) maupun pertukaran data dengan perangkat lain.

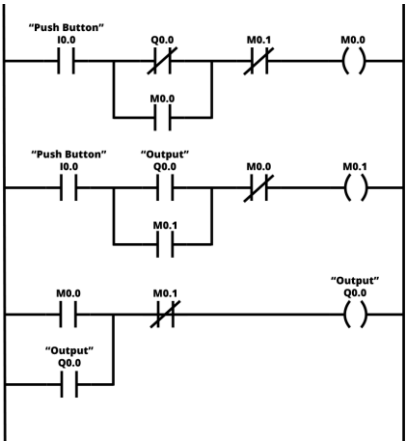
2.2.2 Bahasa Pemrograman PLC [7]

Berdasarkan standar yang telah ditentukan oleh IEC (*International Electrotechnical Commission*), badan standardisasi dunia dalam bidang teknik elektro, terdapat beberapa bahasa pemrograman PLC, yaitu:

1. *Ladder Diagram*

   Ladder diagram adalah bahasa pemrograman yang dibuat dari persamaan logika dan fungsi lain berupa pemrosesan data atau fungsi waktu dan pencacahan. *Ladder diagram* terdiri dari susunan kontak – kontak dalam grup perintah secara *horizontal* dari kiri ke kanan dan terdiri dari banyak grup perintah secara *vertical*. Contoh dari kontak *ladder diagram* adalah *normally open contact* (NO), *normally close contact* (NC), *output coil*.

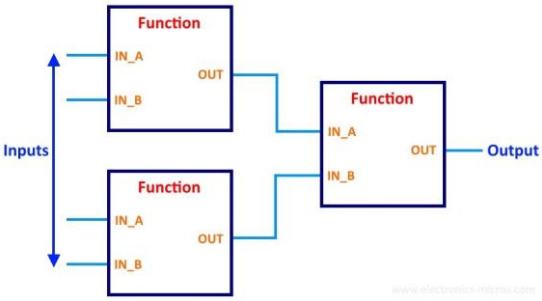
   Ladder diagram merupakan bahasa pemrograman untuk PLC yang paling populer. Seperti namanya yaitu diagram tangga, Gambar 2.9 menunjukkan bahwa bentuk bahasa ini mirip dengan tangga (*ladder*).



Gambar 2.9 Contoh Ladder Diagram

2. *Function Block Diagram* (FBD)

Function block diagram adalah suatu fungsi logika yang disederhanakan dalam gambar blok dan dapat dihubungkan dalam suatu fungsi atau digabungkan dengan fungsi blok lain. Program FBD muncul dalam bentuk kotak sebagai blok elemen yang dihubungkan bersama–sama dengan cara yang menyerupai diagram rangkaian. FBD berguna dalam aplikasi yang melibatkan tingkat tinggi informasi antara komponen control, seperti kontrol proses. *Block* seperti pada Gambar 2.10 menggambarkan hubungan antara variabel *input* dan *output* yang memiliki fungsi khusus seperti instruksi dan logika aritmetika.



Gambar 2.10 Contoh *Function Block Diagram*

3. *Structured Text* (ST)

Structured text merupakan bahasa pemrograman berbasis teks tingkat tinggi (PHP, Phyton dan C) yang dapat memproses system logika ataupun algoritma dan memungkinkan pemrosesan system lain. Perintah umumnya menggunakan IF, THEN, ELSE, WHILE DO, REPEAT, UNTIL, dll.

```
IF value < 7 THEN
    WHILE value < 8 DO
        value := value + 1;
    END_WHILE;
END_IF;
```

Gambar 2.11 Contoh Structured Text

Gambar 2.11 menunjukkan *syntax* dari ST dikembangkan mirip dengan *syntax* pada Bahasa pemrograman tingkat tinggi yang memiliki fungsi *loop*, variabel, kondisi dan operator.

4. *Instruction List*

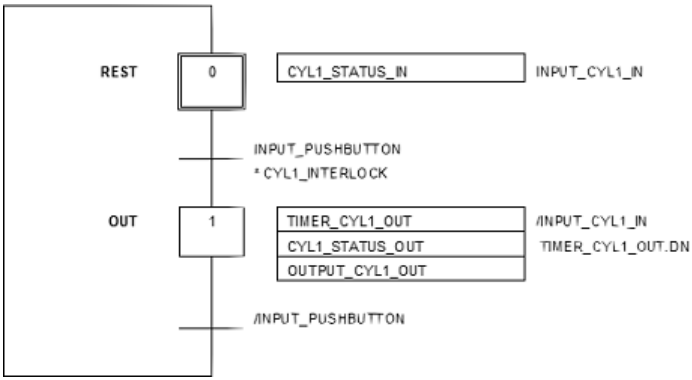
Instruction List adalah bahasa pemrograman jenis tingkat rendah yang menggunakan *statementvariable* (huruf) sebagai *input* dan sangat efektif untuk pengaplikasian secara kecil dimana terdapat perintah–perintah yang sudah baku. *Instruction List* bisa menyajikan banyak *input* dan *output*.

Step	Instruction	
0	LD	X400
1	OR	X402
2	LD	X401
3	OR	X403
4	ANB	
5	OUT	
		Y430

Gambar 2.12 Contoh *Instruction List*

5. *Sequential Function Chart* (SFC)

SFC adalah bahasa program yang dibuat dan disimpan dalam *chart*. Bagian–bagian *chart* memiliki fungsi urutan langkah, transisi, dan percabangan. Tiap *step* memiliki status proses dan bisa terdiri dari struktur yang berurutan.



Gambar 2.13 *Contoh* Sequential Function Chart

2.2.4 *Addressing* dan Instruksi Dasar PLC Omron

Addressing merupakan hal yang penting dari suatu pemrograman pada PLC, karena berfungsi sebagai identitas dari suatu komponen seperti *input* maupun *output*. Tiap jenis PLC memiliki aturan tersendiri dalam *addressing*, termasuk jenis PLC Omron yang akan dipakai pada tugas akhir kali ini. Pada Tabel 2.2 merupakan aturan dari *addressing* dari PLC jenis Omron.

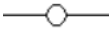
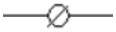
Tabel 2.1 Lebar Alamat Pada PLC Omron

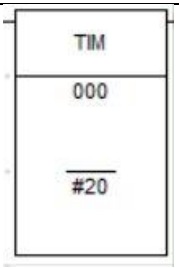
No	Fungsi	Alamat
1	*Input bits*	000.00 – 015.15
2	*Output bits*	100.00 – 115.15
3	*Special bits*	244.00 – 255.07
4	*Timer/Counter*	TIM/CNT 000 - 511

Penomoran alamat yang ada pada Tabel 2.1 akan ditempatkan pada komponen – komponen seperti *contact* dan *coil*. Sesuai pada informasi tabel, untuk *input bits* mempunyai lebar alamat antara 000.00 – 015.15, *output bits* mempunyai lebar alamat antara 100.00 – 115.15, *special bits* atau bisa disebut *memory* yang mempunyai lebar alamat dari 244.00–255.07, dan *timer/counter* yang mempunyai lebar alamat dari 000–511.

Instruksi atau perintah dasar pada PLC dirancang untuk melakukan pemrograman pada PLC. Perintah-perintah tersebut dibutuhkan untuk menangani sistem kontrol pada mesin industri. Tabel 2.2 menampilkan beberapa instruksi dasar yang nantinya akan dikonstruksikan ke *ladder diagram*.

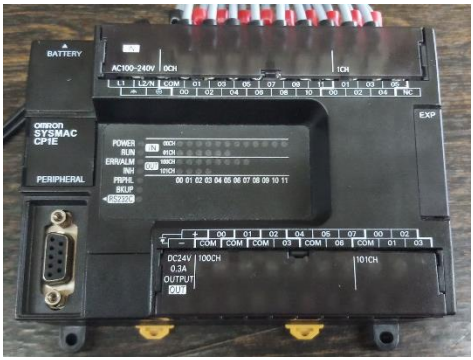
Tabel 2.2 Instruksi Dasar Pada PLC

Instruksi	Simbol	Keterangan
Load		Perintah ini digunakan jika kerja suatu sistem kontrol hanya membutuhkan satu keadaan logika, ketika *input* aktif, maka simbol juga akan aktif. *Load* merupakan kontak dari *Normally Open* (NO).
Load Not		Perintah ini digunakan jika kerja suatu sistem kontrol hanya membutuhkan satu keadaan logika. Cara kerjanya kenbalikan dari perintah *load*, jika *input* nonaktif maka simbol *load not* akan aktif. *Load Not* merupakan kontak dari *Normally Close* (NC).
Out		Mirip seperti *coil* pada *Normally Open*. Simbol akan aktif ketika mendapatkan sinyal dari *rung* di sebelah kiri *ladder*.
Out Not		Mirip seperti *coil* pada *Normally Close*. Simbol akan aktif ketika tidak mendapatkan sinyal dari *rung* di sebelah kiri *ladder*.

Instruksi	Simbol	Keterangan
Timer		Nilai pada *Timer* bisa bersifat mengitung mundur dari nilai awal yang ditetapkan atau bisa juga menghitung maju menuju nilai yang diinginkan, jika tercapai maka *contact timer* akan on.

2.2.5 PLC OMRON CP1E

Pengimplementasian *ladder diagram* dilakukan dengan menggunakan PLC merk OMRON CP1E yang merupakan PLC tipe *modular. Ladder diagram* diprogram di aplikasi bawaan dari OMRON CP1E ini yang bernama *CX-Programmer*. PLC ini memiliki pin I/O digital sebanyak 30 buah di mana 18 dapat digunakan sebagai pin input dan 12 dapat digunakan sebagai pin output. Adapun alamat Input pada CP1E yaitu 0.00 – 0.15 dan 1.00 – 1.05 sementara untuk alamat Output yaitu 100.00 – 100.07 dan 101.00 – 101.03.



Gambar 2.14 PLC CP1E

2.3 *Human Machine Interface*

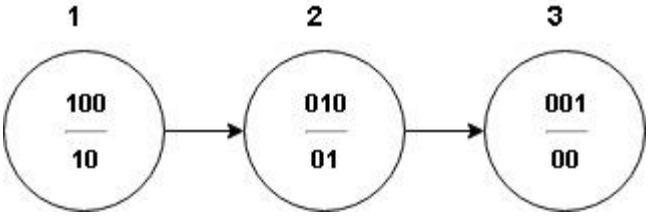
Human Machine Interface (HMI) merupakan sistem yang menghubungkan antara manusia dan mesin. HMI dapat mempermudah operator memantau status keadaan proses atau mesin dengan cara

memvisualisasi status dari proses dan kontroler secara *real time*. HMI yang baik dan handal akan membantu operator untuk mengamati, mengatur suatu proses pada *plant* secara efektif dan efisien. Pada tugas akhir ini, HMI digunakan untuk menyimulasikan pemantauan keadaan dari simulasi pengendalian proses melalui visualisasi komputer. Adapun HMI yang digunakan pada tugas akhir ini adalah CX-Designer yang merupakan program bawaan dari PLC Omron.

Gambar 2.15 CX Designer

2.4 *State Diagram* [8]

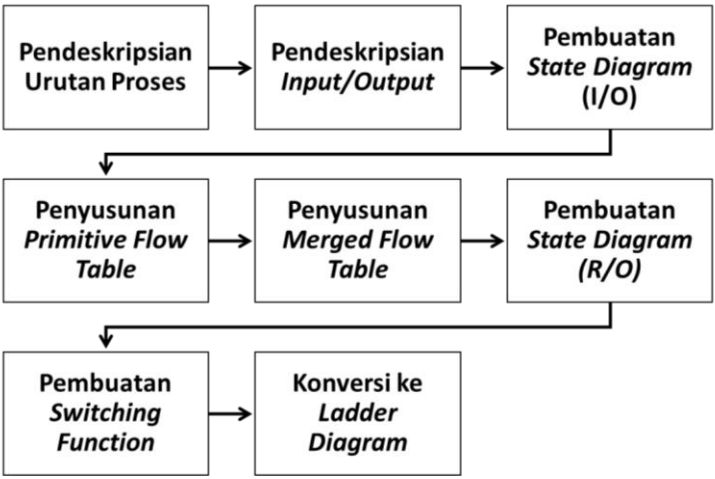
Metode *state diagram* atau *flow-table* merupakan suatu metode yang menggunakan grafik *state* untuk menggambarkan suatu keadaan sistem ke dalam bentuk lingkaran. Lingkaran inilah yang dimaksud dengan istilah *state*. Pada *state* tersebut berisi informasi data berupa *input* dan *output* dari sebuah sistem yang ditulis dalam bentuk digit biner seperti pada gambar 2.16.



Gambar 2.16 State Diagram

Pada Gambar 2.17 ditunjukkan langkah-langkah dalam membuat *state diagram*. Dimulai dengan pendeskripsian urutan kerja proses,

kemudian pendeskripisan *input* dan *output* yang digunakan. Setelah itu dilakukan pembuatan *state diagram* I/O. Setelah dilakukan pembuatan *state diagram*, kita bisa penyusunan *primitive flow table* dan *merged flow table*. Dari *Merged flow table* kemudian disusun *state diagram* (R/O) sehingga didapatkan *switching function* yang dikonversi sebagai *ladder diagram* dan diaplikasikan pada PLC.



Gambar 2.17 Langkah Pembuatan *Ladder Diagram* dengan *State Diagram*

2.4.1 Penentuan Urutan Proses

Dalam menggunakan metode *state diagam* ini, langkah pertama yang harus dilakukan adalah menentukan urutan proses dari *plant*. Hal yang harus diperhatikan dalam penentuan urutan proses ini adalah kesesuaian urutan langkah kerja sistem yang kita kehendaki. Pada langkah yang memiliki kondisi *input* dan *output* yang sama dengan langkah sebelumnya, maka akan dianggap sebagai satu *state* yang sama. Hal ini bertujuan untuk mempermudah kelanjutan perancangan *state diagram* pada suatu sistem.

Tabel 2.3 Penentuan Urutan Proses

State	Kondisi Input	Kondisi Output	Syarat Perlu
S	Start on	Stacker/reclaimer on	Kondisi diam
S+1	Input 1 on	Output 1 On	-
...	...	...	...
Sn	End on	Conveyor off	-

2.4.2 Pendiskripsian *Input* dan *Output*

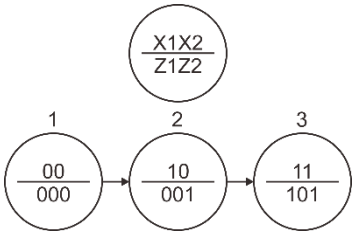
Selanjutnya pada langkah kedua, *input* dan *output* yang sudah ditentukan kemudian diberi penamaan dan deskripsi dari fungsinya masing-masing. Umumnya, *input* pada sistem diberi penamaan X1 hingga Xn sementara *output* diberi penamaan Z1 hingga Zn sesuai jumlah *input* dan *output* yang dipakai. Hal ini bertujuan untuk mempermudah penamaan pada metode *state diagram.*

Tabel 2.4 Pendeskripsian Input

No	Input	Simbol	Fungsi
N	Input N	Xn	Deskripsi fungsi input
N+1	Input N+1	Xn+1	...
...	...	...	...
M	Input M	Xm	...

2.4.3 Pembuatan *State Diagram I/O*

Setelah mendiskripsikan *input* dan *output* sistem, langkah ketiga adalah penyusunan *state diagram* dari sistem berdasarkan *input* dan *output* yang sebelumnya telah ditentukan dan dideskripsikan. Pada langkah ini dimulai dengan membuat *state-state* yang terdiri dari *input* dan *output* sistem sesuai dengan urutan proses yang telah ditentukan di tahap pertama. Penulisan *input* dan *output* yang yang digunakan adalah sistem bilangan biner yang berarti *state* akan bernilai "1" jika aktif dan sebaliknya akan bernilai "0" jika tidak aktif. Masing-masing *state* tersebut lalu dihubungkan dengan arah panah sebagai penanda arah transisi state.



Gambar 2.18 Contoh *state diagr*am I/O

2.4.4 Pembuatan *Primitive Flow Table*

Langkah selanjutnya adalah pembuatan *primitive flow table.* Pembuatan *Primitive flow table* bertujuan untuk menerjemahkan *state diagram* ke dalam bentuk tabel yang berisi jumlah *row, input,* dan *output* sistem. Mula-mula *row* diisi berdasarkan urutan penomoran *state* yang telah ditentukan jumlahnya pada *state diagram* I/O. Pada kolom *input* dibuat sebanyak kombinasi bit *input* yang ada. Misal pada *state diagram* I/O terdapat empat macam kombinasi *input* yaitu 1000, 0100, 0010 dan 0001, maka dibuat empat kolom yang masing-masing berisi kombinasi *input* tesebut. Apabila terdapat kombinasi *input state* yang sama dengan *state* yang lain maka cukup dibuat satu kolom kombinasi *input.* Selanjutnya adalah pengisian *state* stabil, *state* tidak stabil, dan *don't care* pada masing-masing *row*. *State* stabil adalah suatu kondisi *state* yang pada saat itu aktif atau memiliki kombinasi input yang sama dengan state diagram (I/O). State stabil biasanya ditandai dengan menggunakan huruf cetak tebal. *State* tidak stabil adalah *state* yang akan dituju oleh *state* aktif saat itu berdasarkan arah panah yang menghubungkan *state-state* tersebut. Misal state S memiliki arah panah ke state S+1 yang memiliki *input* 0100. Maka state S+1 ditulis sejajar di sel tabel yang sejajar dengan baris *row* S dan kolom 0100. Adapun kondisi *don't care* adalah kondisi di mana tidak ada pengaruh apapun dari *state* yang aktif saat itu dan biasa dinyatakan dengan tanda strip (-) pada kolom. Contoh pengisian dapat dilihat di tabel 2.5.

Tabel 2.5 Contoh pengisian stabil, tidak stabil, dan *don't care*

Row	Input				Output			
	1000	0100	0010	0010	Z_b	Z_{b+1}	....	Z_{b+n}
s	**s**	s+1	....	-	....	....	....	....
s+1	-	**s+1**	....	-	....	....	....	....
....	....	....	....	....	....	....	....	....
s+n	-	-	....	**s+n**	....	....	....	....

Selanjutnya adalah pengisian pada kolom *output*. Dalam penulisan output dinyatakan dengan huruf Z1 hingga Zn, sesuai dengan jumlah *output* yang sudah digunakan di awal proses. Sel-sel *output* tersebut diisi dengan bilangan biner yang menandakan "1" sebagai tanda aktif dan "0" sebagai tanda tidak aktif. Misal untuk state S dengan konfigurasi *output* Z_1 Z_2 Z_3 memiliki nilai *output* 101, maka pada sel-sel *output* yang sejajar dengan baris *row* S diisi nilai Z_1=1, Z_2=0 dan Z_3=1. Begitu selanjutnya hingga *output row* s+n

Tabel 2.6 Contoh pengisian *output*

Row	Input				Output		
	1000	0100	0010	0001	Z1	Z2	Z3
S	**S**	S+1	-	-	0	0	1
S+1	-	**S+1**	S+2	-	1	1	0
S+2	-	-	**S+2**	-	1	1	0

2.4.5 Penyusunan *Merged Flow Table*

Setelah pembuatan *Primitive flow table*, langkah selanjutnya adalah pembuatan *Merged flow table. Pada merged flow table* dilakukan penyederhanaan dengan cara menggabungkan kondisi *output* yang sama pada *primitive flow table.* Pada merged flow table juga dilakukan penentuan jumlah dan susunan *relay* yang akan digunakan sehingga *didapatkan* jumlah *relay* yang lebih efisien dibandingkan dengan tanpa menggunakan metode *state diagram*. Berikut adalah aturan yang harus dipenuhi pada saat menggabungkan baris *merged flow table*:

1. Apabila pada saat penggabungan terdapat state stabil, tidak stabil dan atau kondisi *don't care* di dalam sel yang sama, maka sel penggabungan diisi dengan state stabil.
2. Apabila pada saat penggabungan terdapat state tidak stabil dan kondisi *don't care* di dalam sel yang sama, maka sel penggabungan diisi dengan state tidak stabil.
3. Apabila pada saat penggabungkan hanya ada kondisi *don't care*, maka tetap diisi dengan *don't care*.

Misal pada *row* S+2 memiliki kondisi *output* yang sama dengan *row* S+1, maka kedua baris tersebut dapat digabungkan dengan mengikuti aturan penggabungan tersebut.

Tabel 2.7 Contoh penggabungan baris

Row	Input				Output		
	1000	0100	0010	0001	Z1	Z2	Z3
S	**S**	S+1	-	-	0	0	1
S+1, S+2	-	**S+1**	**S+2**	-	1	1	0

Dalam menentukan jumlah *relay* pada metode *state diagram*, digunakan teori RS *flip-flop*. Alasan penggunaan teori RS *flip flop* dikarenakan satu *relay* dapat menghasil dua buah kombinasi bit yaitu "1" dan "0", dan jika menggunakan dua *relay* dapat menghasilkan empat kombinasi bit. Dari teori RS *flip flop* tersebut bisa didapat rumusan untuk menentukan jumlah kombinasi bit dengan jumlah *relay* tertentu. Rumusan tersebut adalah 2^n dimana *n* adalah jumlah *relay*. Jika *m* adalah jumlah baris dari *row*, maka jumlah *relay* yang harus dipenuhi adalah $2^n \geq m$, sehingga dapat memenuhi kebutuhan jumlah *row*.

Tabel 2.8 Contoh penentuan *relay*

Row	Input				Output			Relay	
	1000	0100	0010	0001	Z1	Z2	Z3	Y1	Y2
S	**S**	S+1	-	-	1	0	1	1	0
S+1, S+2	-	**S+1**	**S+2**	S+3	1	1	0	1	1
S+3	-	**-**	S+4	**S+3**	1	0	0	0	1
S+4	-	**-**	**S+4**	-	0	0	0	0	0

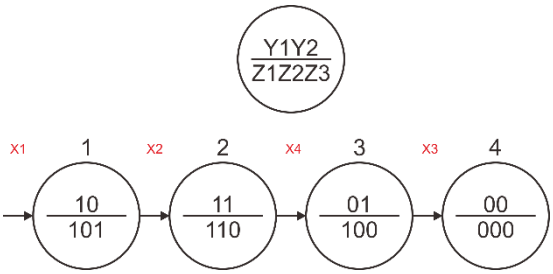
Dalam penentuan kombinasi *relay* harus mengikuti sebuah sistem yang bernama *reflected binary code* atau biasa dikenal *grey code*. Secara singkat, *grey code* adalah suatu sistem penomoran biner yang hanya boleh berubah satu bit tiap keadaannya. Tujuan dalam penggunaan *grey code* adalah untuk menghindari kesalahan dalam pembacaan *relay* pada *state diagram*. Misal suatu kombinasi *relay* 110 akan berpindah kondisi menjadi 101. Secara kasat mata mungkin akan tampak langsung berpindah dari 110 ke 101, namun jika waktu diperlambat tidak seperti itu. Ketika perpindahan 110 ke 101 akan ada kondisi dimana perubahannya dimulai dari 110, 100, baru kemudian menjadi 101, atau dimulai dari 110, 111, baru kemudian 101. Dengan menggunakan sistem *grey code* tersebut, kondisi state sistem harus berubah dari 110 ke 100 untuk mencegah kesalahan.

2.4.6 Pembuatan *State Diagram R/O*

Langkah selanjutnya adalah pembuatan *state diagram* berdasarkan *relay* dan *output* (R/O). perbedaan yang tampak antara state I/O dan R/O adalah jumlah statenya jika mengalami penggabungan pada *merged flow table*, dan posisi keterangan *input* yang terletak di panah penghubung *state* karena digantikan oleh keterangan *relay* sistem. Input yang digunakan tidak semuanya, hanya *input* yang memiliki pengaruh dalam perubahan state saja atau bisa disebut sebagai *trigger* dari perubahan *state*. Perubahan *state* I/O ke R/O mengartikan bahwa setiap perubahan *output* akan bergantung kepada perubahan kombinasi *relay*. Tujuan dari pemosisian *input* dicpanah yang menuju *state* adalah sebagai tanda bahwa perubahan kondisi *relay* akan dipengaruhi *input* yang menuju state tersebut.

Cara pembuatan state diagram R/O sangat mudah. Berikut tahapan pembuatan state diagram R/O:

1. Masukkan kombinasi *relay* dan kombinasi *output* ke state yang sama.
2. Penentuan arah panah dibuat sesuai dengan perubahan kondisi stabil ke tidak stabil yang ada di tabel *merged flow table*.
3. Memberikan keterangan *input* yang menjadi *trigger* dipanah yang menuju state yang akan di-*trigger*.



Gambar 2.19 Contoh state diagram R/O

2.4.7 Konversi *Ladder Diagram* ke *Switching Function*

Switching function adalah persamaan yang digunakan untuk membuat *ladder diagram*. Pembuatan *switching function* dibagi menjadi dua, yaitu *swicthing function* untuk *relay* dan untuk *output*. Untuk *input* biasanya akan digunakan huruf X1 hingga Xn, *relay* menggunakan huruf Y1 hingga Yn dan *output* akan menggunakan huruf Z1 hingga Zn, masing-masing tergantung berapa jumlah yang digunakan. *Relay* dan *output* memiliki dua kondisi yaitu *set* yang memiliki perubahan nilai dari "0" ke "1" dan *reset* memiliki perubahan nilai dari "1" ke "0".

Ada beberapa tahapan dalam pembuatan *switching function relay*, sebagai berikut:

1. Menentukan *input* yang mengubah *relay*-m menjadi kondisi *set*.
2. Menentukan *input* yang mengubah *relay*-m menjadi kondisi *reset*, kemudian mengganti kondisi *input*. Misal *input*-n pada sistem adalah X1, maka di ganti kondisi menjadi *not*-X1.
3. Menggabungkan semua *input* yang mengubah *relay*-m menjadi kondisi *set* dengan logika OR (jika ada).

4. Menambahkan *relay*-m ke gabungan *input* sebagai *safe holding relay*-m.
5. Menggabungkan semua *input* yang mengubah *relay*-m menjadi kondisi *set* (jika ada).

Saat penulisan *switching function*, logika AND ditulis dengan perkalian, dengankan untuk logika OR ditulis dengan penjumlahan. Dalam penggabungan *input* untuk kondisi *reset* memiliki syarat sebagai berikut:

1. Jika *input*-n yang me-*reset relay*-m aktif kembali atau tetap aktif ketika *relay*-m kembali ke kondisi *set* hingga di *reset* oleh *input* lainnya, maka penggabungan *input* menggunakan logika OR.
2. Jika *input*-n yang me-*reset relay*-m tidak aktif kembali ketika *relay*-m kembali ke kondisi *set* hingga di *reset* oleh *input* lainnya, maka penggabungan *input* menggunakan logika AND.

Berikut contoh *switching function relay*:

$$Y1 = (X1 + Y1) * \overline{X4}$$
$$Y2 = (X2 + Y2) * \overline{X3}$$

Ada kondisi di mana perlu ditambahkan suatu input yang menggunakan logikan OR dengan *input reset*. Kondisi tersebut adalah ketika *input*-n yang akan me-*reset relay*-m aktif dan tidak aktif saat kondisi *relay*-m belum seharusnya *reset*. Pada kondisi ini ditambahkan *input* lain yang sama-sama aktif dengan *relay*-m, namun sudah mati pada saat *input*-n akan me-*reset relay*-m. pada kondisi ini boleh menambahkan beberapa *input* tambahan dengan tujuan *relay*-m tetap aktif saat *relay*-m diharuskan aktif.

Setelah membuat *switching function relay*, kemudian dilanjutkan dengan pembuatan *switching function output*. Berikut tahapan pembuatan *switching function output*:

1. Menentukan state ke berapa saja saat *output*-n aktif.
2. Mencari tahu kombinasi *relay* apa saja yang ada di *state* yang telah ditentukan.
3. Kombinasi *relay* yang ada disatu state ditulis dengan logika AND.
4. Menggabungkan kombinasi *relay* yang memiliki *output*-n yang sama dengan logika OR.

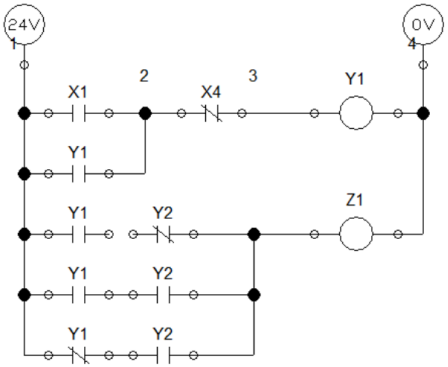
Berikut contoh *switching function output*:

$$Z1 = (Y1 * \overline{Y2}) + (Y1 * Y2) + (\overline{Y1} * Y2)$$

$$Z2 = (Y1 * \overline{Y2})$$
$$Z3 = (Y1 * Y2)$$

2.4.8 Pembuatan *Ladder Diagram*

Pembuatan *ladder diagram* merupakan bentuk pengaplikasian dari *switching function* yang telah dirumuskan pada tahap sebelumnya. Pada sebuah fungsi atau persamaan, di sebelah kiri tanda sama dengan (=) merupakan *output*, sedangkan disebelah kanan merupakan *input*. Begitu juga dengan *switching function*, dibagian sebelah kiri adalah *ouput* atau *relay*, sedangkan di sebelah kanan merupakan *input* atau kontak.



Gambar 2.20 Contoh *ladder diagram* hasil *switching function*

BAB 3
PERANCANGAN SISTEM

Pada Bab 3 dijelaskan tahapan dalam perancangan *ladder diagram* dengan metode *state diagram* serta tampilan dari rancangan HMI yang digunakan sebagai visualisasi.

3.1 Perancangan *Ladder Diagram* dengan Metode *State Diagram*

Untuk merancang sistem otomasi pengolahan batubara pada PLTU dengan menggunakan *State Diagram* tentu dibutuhkan beberapa tahap. Dimulai dengan pendeskripsian urutan kerja proses, kemudian pendeskripsian *input* dan *output* yang digunakan. Setelah itu dilakukan pembuatan *state diagram* I/O. Setelah dilakukan pembuatan *state diagram*, kita bisa penyusunan *primitive flow table* dan *merged flow table*. Dari *Merged flow table* kemudian disusun *state diagram* (R/O) sehingga didapatkan *switching function* yang dikonversi sebagai *ladder diagram*.

3.1.1 Pendeskripsian Urutan Kerja Proses

Dalam perancangan *ladder diagram* dengan menggunakan metode *state diagram*, pertama proses harus dideskripsikan agar pembaca dapat mengetahui alur kerja suatu proses. Untuk mempermudah pengunaan metode *state diagram* maka penulis membagi perancangan *ladder diagram* menjadi lima sesuai lima sub-proses pengolahan batubara PLTU.

3.1.1.1 Sub-Proses 1 (*Coal transfer*)

1. Kondisi *Input* dan *Output* dipastikan dalam keadaan non aktif
2. Ketika tombol *start* ditekan, *Stacker/Reclaimer* aktif mengeruk batubara dari *coalpile* dan jatuh ke conveyor.
3. Batubara yang jatuh mengaktifkan *loadcell sensor* pada *conveyor* sehingga *conveyor* aktif dan membawa batubara ke proses selanjutnya.
4. Batubara masuk ke dalam *crusher building* mengaktifkan *proximity sensor* pada *crusher building* sehingga *magnetic separator* aktif untuk memisahkan batubara dengan logam yang tersisa sementara *crusher* menghancurkan batubara menjadi lebih kecil.
5. Batubara kemudian dibawa oleh *conveyor* menuju ke *Silo* untuk ditampung sementara sebelum dihancurkan di pulverizer. Apabila Batubara telah memenuhi *Silo*, *Limit Switch High* (LSH)

Silo aktif sehingga proses transfer batubara berhenti agar tidak memenuhi *Silo*.

6. Ketika volume batubara pada Silo berkurang, maka LSH Silo akan non aktif sehingga seluruh aktuator pada sub-proses 1 akan kembali aktif.

3.1.1.2 Sub-Proses 2 (*Water Feeding*)

7. *Deaerator* yang sudah terisi air akan mengaktifkan sensor *Limit Switch High Deaerator*, *Boiler Feed Pump* kemudian aktif sehingga air dari *deaerator* dapat dipompa menuju *Steam Drum.*
8. Ketika air sudah memenuhi *Steam Drum*, *Limit Switch High Steam Drum* aktif. *Boiler Circulating Water Pump* kemudian aktif memompa air dari *Steam Drum* menuju ke waterwall, *Forced Draft Fan* lalu aktif untuk memasok udara dari atmosfer.

3.1.1.3 Sub-Proses 3 (*Burning Process*)

9. *HSD Tank* yang telah terisi *HSD Oil* membuat *Limit Switch High HSD* aktif sehingga *HSD Pump* aktif memompa *HSD oil* menuju ke Boiler.
10. Udara dan minyak yang mengalir ke *boiler* akan mengaktifkan *oil* dan *air detector*. *Ignitor* aktif sehingga mengakibatkan pembakaran pada *boiler*, air pun mengalami pemanasan.
11. Pembakaran yang terjadi di boiler mengakibatkan *Flame Detector* aktif, panas dari pembakaran di boiler kemudian mengaktifkan *Primary Air Heater* dan *Secondary Air Heater*.
12. Air yang dipanaskan di boiler berubah menjadi uap yang dapat menggerakkan turbin generator. Beban generator yang telah mencapai 30% mengaktifkan sensor beban generator. *HSD Pump* yang memompa minyak ke *boiler* non aktif, sementara *valve coalfeeder* aktif menyalurkan batubara ke pulverizer. *Crusher pulverizer* aktif sehingga batubara dapat tergiling menjadi serbuk halus. *Primary Air Heater fan* aktif membantu mengeringkan sekaligus menyalurkan batubara dari pulverizer menuju ke boiler sebagai bahan bakar utama menggantikan *HSD oil*.

3.1.1.4 Sub-Proses 4 (*Bottom Ash Handling*)

13. Batubara yang gagal mengalami pembakaran sempurna jatuh ke dasar *boiler* menjadi *bottom ash*, *Loadcell Hopper* aktif sehingga membuat *Transition Hopper Valve*, *Submerged Scrapper*

Conveyor (SSC), dan *Bottom Ash Conveyor* aktif pada sistem *bottom ash handling*.

14. Sensor suhu di *Hopper* mendeteksi adanya suhu tinggi dari *bottom ash* kemudian mengaktifkan *recirculation water pump* untuk menyalurkan air ke *bottom ash* agar suhunya turun.
15. *Bottom Ash* yang berada di *Submerged Scrapper Conveyor* kemudian terdeteksi oleh sensor proximity SSC sehingga mengaktifkan *clinker grinder* untuk menghancurkan *bottom ash* yang menggumpal. *Bottom Ash* kemudian jatuh dan dibawa oleh *Submerged Scrapper Conveyor* menuju ke *Ash Disposal.*

3.1.1.5 Sub-Proses 5 (*Fly Ash Handling*)

16. Debu hasil pembakaran batubara di *boiler (fly ash)* terbang menuju *Electrostatic Precipitator (EP)*. Sensor suhu di EP mendeteksi adanya *fly ash* yang masuk kemudian mengaktifkan *Induced Draft Fan 1* untuk aktif menyedot *fly ash* dari boiler menuju ke EP untuk diolah menjadi *flue gas*. *High Voltage Transformer Rectifier* aktif untuk memberikan tegangan DC ke plat-plat yang terdapat di EP sehingga partikel debu *fly ash* dapat menempel. *Induced Draft Fan 2* aktif menyedot *flue gas* dari EP menuju *Flue Gas Dezulfuration*.
17. *Fly Ash* yang masuk ke EP mengakibatkan *Timer Contact* aktif sehingga menggerakkan *rapper* secara periodik sebagai penghentak debu-debu yang menempel pada plat-plat EP sehingga *fly ash* menjadi *flue gas* untuk diolah di *Flue Gas Dezulfuration (FGD)*. *Vacuum Blower* kemudian membantu menyedot debu-debu yang jatuh ke dasar EP untuk menuju *Fly Ash Silo*.
18. *Flue gas* dari EP kemudian menuju FGD dengan bantuan *Induced Draft Fan 2*. Sensor gas FGD mendeteksi adanya *flue gas* yang masuk sehingga *Sea Water Scrubber pump* dapat aktif memompa air laut. *Water spray* aktif untuk menyemprotkan air laut sehingga mengurangi kandungan sulfur pada *flue gas*. *Induced Draft Fan* 3 kemudian menyedot flue gas dari FGD menuju ke stack untuk dibuang menuju ke atmosfer.

Deskripsi urutan proses pengolahan batubara pada PLTU juga diterjemahkan ke dalam bentuk tabel agar lebih memudahkan penyusunan

state diagram (I/O). Tabel 3.1 menunjukkan deskripsi urutan proses dalam bentuk tabel dengan pembagian per sub-proses

Tabel 3.1 Deskripsi Urutan per sub-proses

Sub Proses	No	Kondisi Input	Kondisi output
1	1	Seluruh input dalam keadaan diam	Seluruh output dalam keadaan diam
	2	Tombol Start on	*Stacker/reclaimer* on
	3	*Loadcell Sensor Conveyor* on	Conveyor on
	4	*Proximity Sensor CB on*	-*Magnetic Separator* on -*Crusher* on
	5	*LSH Silo* on	-*Stacker/reclaimer* off -Conveyor off -*Magnetic Separator* off -*Crusher* off
	6	*LSH Silo* off	-*Stacker/reclaimer* on -Conveyor on -*Magnetic Separator* on -*Crusher* on
2	1	Seluruh input off	Seluruh output off
	2	-Tombol Start on -*Limit Switch High* Deaerator on	*Boiler Feed Pump* on

	3	*Limit Switch High Steam Drum* on	-*Boiler Circulating Water Pump* on -*Forced Draft Fan* on -*Boiler Feed Pump* on
3	1	-*Limit Switch High HSD* on -*Limit Switch High Steam Drum* on	*HSD Pump* on
	2	-*Oil Detector* on -*Air Detector* on	*Ignitor* on
	3	*Flame detector* on	-*Primary Air Heater* on -*Secondary Air Heater* on
	4	*Sensor Beban Generator* aktif 30%	-HSD Pump off -*Valve Coal Feeder* on -Crusher Pulverizer on -*Primary Air Heater Fan* on
4	1	*Loadcell Hopper* on	-*Transition Hopper Valve* on -*Submerged Scrapper Conveyor (SSC)* on -*Bottom Ash Conveyor* on
	2	Sensor suhu *Hopper* on	*Recirculation Water Pump* on
	3	*Proximity Sensor SSC* on	*Clinker Grinder* on
5	1	*Sensor Suhu ESP* on	-*Induced Draft Fan* 1 on -*Induced Draft Fan* 2 on -*High Voltage Transformer Rectifier* on

	2	Timer Contact on	-Rapper aktif periodik -*Vacuum Blower* on
	3	*Gas Sensor FGD* on	-*Sea Water Scrubber Pump* on -*Water Spray* on -*Induced Draft Fan* 3 on

3.1.2 Pendeskripsian *Input* dan *Output* Sistem

Dalam perancangan sistem otomasi, dibutuhkan informasi mengenai *input* dan *output* yang akan digunakan. Berikut ini adalah daftar *input* dan *output* yang akan digunakan beserta fungsinya.

Tabel 3.2 *Input* sistem

No	Input	Simbol	Alamat	Fungsi
1	Tombol Start	X1	00.00	Untuk memulai Proses *Coal transfer*
2	*Loadcell Sensor Conveyor*	X2	00.01	Memberikan sinyal bahwa batubara telah berada di *conveyor*
3	*Proximity Sensor Crusher Building*	X3	00.02	Memberikan sinyal bahwa batubara telah berada di *crusher building*
4	*Limit Switch High Silo*	X4	00.03	Memberikan sinyal bahwa batubara telah memenuhi kapasitas Silo
5	*Limit Switch High* Deaerator	X5	00.04	Memberikan sinyal bahwa air deaerator telah memenuhi kapasitas Deaerator
6	*Limit Switch High Steam Drum*	X6	00.05	Memberikan sinyal bahwa air telah mencapai level maksimal di *Steam Drum*

7	*Limit Switch High HSD*	X7	00.06	Memberikan sinyal bahwa minyak telah memenuhi kapasitas *HSD Tank*
8	*Oil Detector*	X8	00.07	Memberikan sinyal bahwa minyak telah melewati saluran menuju *furnace*
9	*Air Detector*	X9	00.08	Memberikan sinyal bahwa udara telah melewati saluran menuju *furnace*
10	*Flame Detector*	X10	00.09	Memberikan sinyal bahwa telah terjadi pembakaran di *boiler*
11	*Sensor Beban Generator*	X11	00.10	Memberikan sinyal bahwa beban generator telah mencapai nilai yang telah diatur (30%)
12	*Loadcell Hopper*	X12	00.11	Memberikan sinyal bahwa bottom ash telah jatuh di dasar boiler
13	Sensor suhu *Hopper*	X13	00.12	Memberikan sinyal bahwa terdapat bottom ash dengan suhu tinggi di *Hopper*
14	*Proximity Sensor SSC*	X14	00.13	Mendeteksi adanya bottom ash telah jatuh di SSC
15	*Sensor Suhu Electrostatic Precipitator*	X15	00.14	Mendeteksi adanya suhu tinggi di Electrostatic Precipitator, sehingga dapat diketahui adanya Fly Ash yang masuk.
16	*Timer Contact*	X16	00.15	Memberikan sinyal agar rapper aktif

17	*Sensor Gas FGD*	X17	01.00	Mendeteksi adanya flue gas yang masuk ke dalam FGD
18	Emergency Stop	X20	04.00	Sebagai tombol berhenti darurat

Tabel 3.3 *Output* Sistem

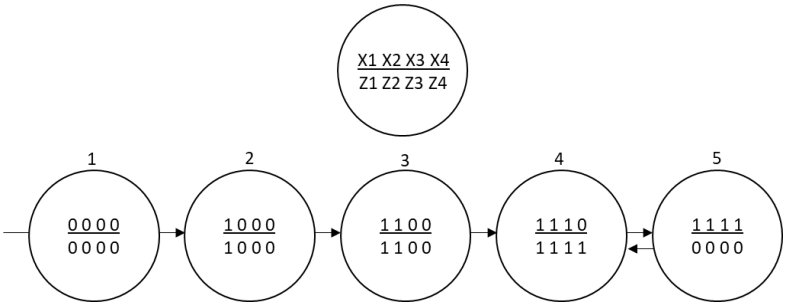
No	Output	Simbol		Fungsi
1	*Stacker/ reclaimer*	Z1	100.00	Mengeruk batubara dari *coalpile*
2	Conveyor	Z2	100.01	Menyalurkan batubara pada saat proses *coal transfer*
3	*Magnetic Separator*	Z3	100.02	Memisahkan logam dari batubara
4	*Crusher*	Z4	100.03	Menghancurkan batubara menjadi ukuran yang lebih kecil (300 mm-20 mm)
5	*Boiler Feed Pump*	Z5	100.04	Memompa air dari deaerator menuju *steam drum*
6	*Boiler Circulating Water Pump*	Z6	100.05	Memompa air dari *steam drum* menuju boiler
7	*Forced Draft Fan*	Z7	100.06	Memasok udara dari atmosfer sebagai sumber O2 ke Boiler
8	*HSD Pump*	Z8	100.07	Memompa minyak dari *HSD Tank*
9	*Ignitor*	Z9	100.08	Menyalakan api pembakaran di *furnace*
10	*Primary Air Heater*	Z10	100.09	Memberikan suhu panas ke *Air Heater Fan*
11	*Secondary Air Heater*	Z11	100.10	Memberikan pasokan udara panas tambahan ke *Forced Draft Fan*

12	*Valve Coal Feeder*	Z12	100.11	Mengatur keluaran batubara dari silo ke pulverizer
13	*Crusher Pulverizer*	Z13	100.12	Menghancurkan batubara menjadi serbuk halus
14	*Primary Air Heater Fan*	Z14	100.13	Memberikan udara panas ke batubara dan menyalurkannya ke *furnace*
15	*Transition Hopper Valve*	Z15	100.14	Mengatur keluaran *bottom ash* dari dasar boiler
16	*Submerged Scrapper Conveyor (SSC)*	Z16	100.15	Menyalurkan *bottom ash* ke *clinker grinder*
17	*Bottom ash conveyor*	Z17	200.00	Menyalurkan bottom ash ke ash disposal
18	*Recirculation Water Pump*	Z17	200.01	Mendinginkan *bottom ash* sisa pembakaran
19	*Clinker Grinder*	Z18	200.03	Menghancurkan *Bottom ash* agar lebih mudah dibawa ke *Ash Disposal*
20	*Induced Draft Fan* 1	Z19	200.04	Menyedot *fly ash* dari boiler ke EP
21	*Induced Draft Fan* 2	Z20	200.05	Menyedot *flue gas* dari EP ke FGD
22	*High Voltage Transformer Rectifier*	Z21	200.06	Memberikan tegangan DC ke plat-plat di EP
23	Rapper	Z22	200.07	Menghentakan plat-plat yang tertempel debu agar rontok
24	*Vacuum Blower*	Z23	200.08	Menyedot debu-debu dari EP melewati *cyclone separator*

25	*Sea Water Scrubber Pump*	Z24	200.09	Menyedot air yang digunakan untuk *water spray*
26	*Water Spray*	Z25	200.10	Menyemprotkan air laut ke *flue gas* untuk mengurangi kandungan sulfur
27	*Induced Draft Fan 3*	Z26	200.11	Menyedot *flue gas* dari FGD ke Stack

3.1.3 Pembuatan *State Diagam* I/O

Setelah diketahui deskripsi urutan proses serta *input* dan *output*-nya, langkah selanjutnya adalah pembuatan *State diagram input/output* (I/O) berdasarkan jumlah kombinasi bit kondisi *input output* yang berbeda pada tiap sub proses. Arah panah pada *state diagram (I/O)* berdasarkan deskripsi urutan proses yang ada. Gambar 3.1 adalah contoh dari pembuatan *state diagram* I/O sub proses 1 (*Coal Transfer*). Untuk *state diagram* (I/O) dari keseluruhan sub-proses dapat dilihat pada lampiran.



Gambar 3.1 State Diagram input/output

3.1.4 Penyusunan Primitive Flow Table

Informasi dari *state diagram (I/O)* secara keseluruhan pada setiap *sub-proses* digunakan untuk menyusun *primitive flow table*. Tabel 3.2 menampilkan *primitive flow table* dari *sub-proses* 1. Untuk menyusun *primitive flow table*, pertama kita harus menentukan komponen penyusunnya yaitu *row*, kolom *input* dan kolom *output*. Disusun 5 row berdasarkan jumlah *state diagram I/O*, sementara terdapat 5 bit kondisi

input maka disusunlah 5 kolom bit kondisi *input* juga terdapat 4 kolom *output* sejumlah *output* yang digunakan. Contoh untuk row 1 yang berarti *state* 1, pada *state* 1 memiliki keterangan *input* bernilai 0000 maka ditulis *state* 1 stabil yang disimbolkan dengan angka cetak tebal pada kolom bit kondisi *input* 0000. Pada *state diagram I/O*, *state* 1 akan menuju ke *state* 2 yang merupakan *state* tidak stabilnya yang ditunjukkan dengan arah panah, maka di kolom bit kondisi *input* 1000 ditulis *state* 2 tidak stabil. *State* tidak stabil di bit kondisi *input* 1000 lalu menuju ke *row* 2 pada bit kondisi *input* 1000 yang berisi state 2 stabil. Begitu seterusnya hingga row 5 dalam pembuatan *primitive flow table sub proses 1*.

Tabel 3.4 Primitive Flow Table Sub-proses 1

Row	Inputs X1 X2 X3 X4					Outputs Z1 Z2 Z3 Z4			
	0000	1000	1100	1110	1111	Z1	Z2	Z3	Z4
1	**1**	2	-	-	-	0	0	0	0
2	-	**2**	3	-	-	1	0	0	0
3	-	-	**3**	4	-	1	1	0	0
4	-	-	-	**4**	5	1	1	1	1
5	-	-	-	4	**5**	0	0	0	0

3.1.5 Penyusunan Merged Flow Table

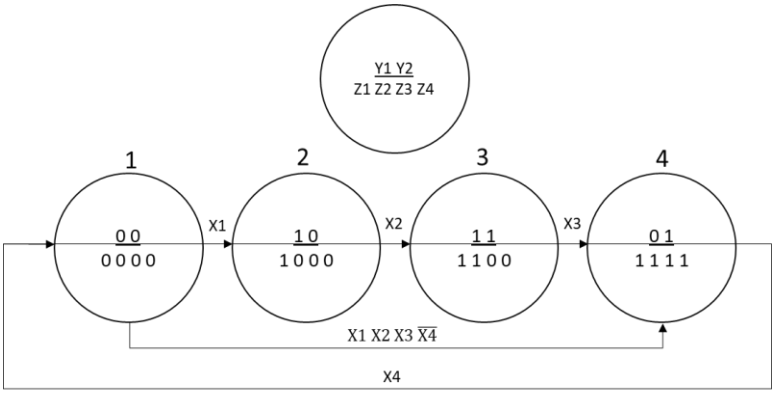
Dalam penyusunan *merged flow table* sub-proses 1, *row* 1 dan *row* 5 memiliki *output* yang sama sehingga dapat dilakukan penggabungan antara *row* 1 dan *row* 5, sehingga *state* 5 stabil dan *state* 4 tidak stabil menempati kondisi *don't care* pada row 1. Selanjutnya ditentukan jumlah kombinasi relay pada sistem. Karena pada *merged flow table* memiliki 4 row maka digunakan 2 relay Y1 dan Y2 yang memenuhi jumlah kombinasi bit relay yang dibutuhkan yaitu 4 kombinasi. Konsep tersebut diambil dari teori RS Flip-flop di mana 1 bit memiliki 2 kondisi (1 dan 0) atau dalam rumus $2^n \geq m$, dengan *n* adalah jumlah relay yang dibutuhkan, 2^n adalah jumlah kombinasi bit relay yang harus dibuat, sedangkan *m* adalah jumlah row pada *merged flow table*. Adapun penentuan urutan kombinasi bit relay berdasarkan sistem Greycode yang mana perubahan bit relay hanya boleh 1 bit tiap perubahan keadaan. Hasil *merged flow table* sub-proses 1 ditunjukkan oleh Tabel 3.3.

Tabel 3.5 Tabel merged flow table sub-proses 1

Row	Inputs X1 X2 X3 X4					Outputs Z1 Z2 Z3 Z4				Relay	
	0000	1000	1100	1110	1111	Z1	Z2	Z3	Z4	Y1	Y2
1, 5	**1**	2	-	4	**5**	0	0	0	0	0	0
2	-	**2**	3	-	-	1	0	0	0	1	0
3	-	-	**3**	4	-	1	1	0	0	1	1
4	-	-	-	**4**	5	1	1	1	1	0	1

3.1.6 Penyusunan *State Diagram* (R/O)

Pada langkah sebelumnya telah didapatkan *merged flow table*. Informasi di dalam tabel akan digunakan untuk menyusun state diagram (R/O) yang memiliki susunan relay/output pada setiap state. Gambar 3.2 menunjukkan hasil penyusunan state diagram (R/O) untuk sub-proses 1. Dibuat 4 state berdasarkan 4 *row* yang ada. Panah menunjukkan transisi dari state stabil menuju state tidak stabil yang disebabkan perubahan nilai inputnya. Contoh untuk transisi state 1 menuju ke state 2 diakibatkan oleh perubahan input X1 sehingga X1 berperan sebagai *input* perubah dari state 1 ke state 2.



Gambar 3.2 *State Diagram (R/O)* Sub-Proses 1

3.1.7 Perancangan *Switching Function*

Dari State Diagram (R/O) sub proses 1, maka *switching function* yang harus dicari adalah untuk *relay* Y1 Y2 dan *output* Z1 Z2 Z3 Z4. Perbedaan dari pembuatan *switching function relay* dan *output*, yaitu ketika untuk *switching function relay* akan dicari sinyal *set* dan *reset*, tapi untuk *output* hanya dicari sinyal *set*.

Contoh pembuatan *Switching Function* Relay Y1, Relay Y1 akan aktif dari *state* 1 hingga *state* 2 kemudian mati di *state* 3. Y1 akan aktif saat transisi state 1 ke state 2, maka ditetapkan lah *input* X1 yang ada pada arah panah 1 ke 2 sebagai *set* dari Y1. Selanjutnya adalah menentukan *reset* dari Y1. Sesuai dengan kondisi Y1 mati ketika transisi dari state 2 ke state 3. Maka *input* yang ada di panah 2 ke 3 merupakan *reset* dari Y1. Langkah terakhir dari *switching function* Y1 adalah menambahkan *safe holding* oleh Y1 itu sendiri dibagian *set*. Tahap yang sama dilakukan untuk menentukan relay dari Y2 iu sendiri. Y2 aktif pada saat transisi dari *state* 2 ke *state* 3 atau dari state 1 menuju ke *state* 4, sehingga input-input pengubahnya dilogika OR. Selanjutnya adalah menambahkan *safe holding* oleh Y2 itu sendiri dibagian *set. S*ementara Y2 mati pada saat transisi state 4 menuju state 1 sehingga *input* yang ada di panah 4 ke 1 merupakan *reset* dari Y2.

Selanjutnya untuk pembuatan *output*, Z1 akan aktif di state 1 *state* 2 dan state 3. Maka kondisi *relay* Y1 dan Y2 tiap state tersebut merupakan *set* dari Z1. Begitu tahap selanjutnya hingga semua *swtiching function* selesai. Berikut *switching function* untuk sub proses 1.

$Y1 = (X1+Y1)*(\overline{X3})$

$Y2 = [X2+(X1*X2*X3*\overline{X4})+Y2]*(\overline{X4})$

$Z1 = (Y1*\overline{Y2})+(Y1*Y2)+(\overline{Y1}*Y2)$

$Z2 = (Y1*Y2)+(\overline{Y1}*Y2)$

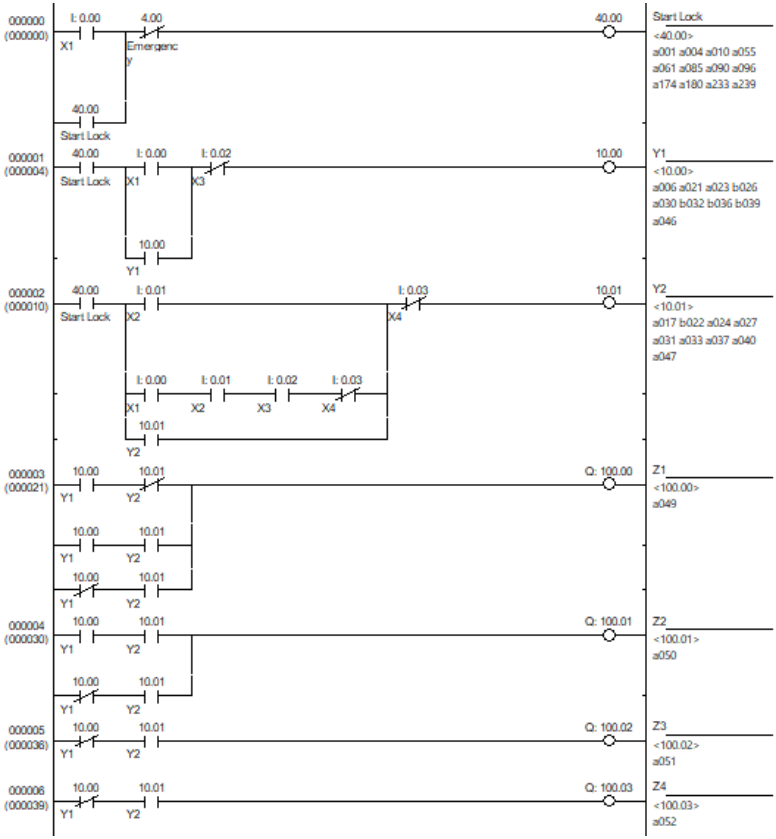
$Z3 = (\overline{Y1}*Y2)$

$Z4 = (\overline{Y1}*Y2)$

3.1.8 Konversi ke Ladder Diagram

Langkah terakhir adalah pembuatan *ladder diagram*. *Switching function* sub proses 1 yang sudah dibuat, dapat langsung diterjemahkan menjadi *ladder diagram*. Apabila logika AND maka disusun seri sementara untuk logika OR maka disusun paralel. Gambar 3.5 adalah

ladder diagram dari sub proses 1. Hasil *ladder diagram* untuk sub proses yang lain dapat dilihat pada lampiran.



Gambar 3.3 Desain *Ladder Diagram* Sub-proses 1

3.1.9 Initial System

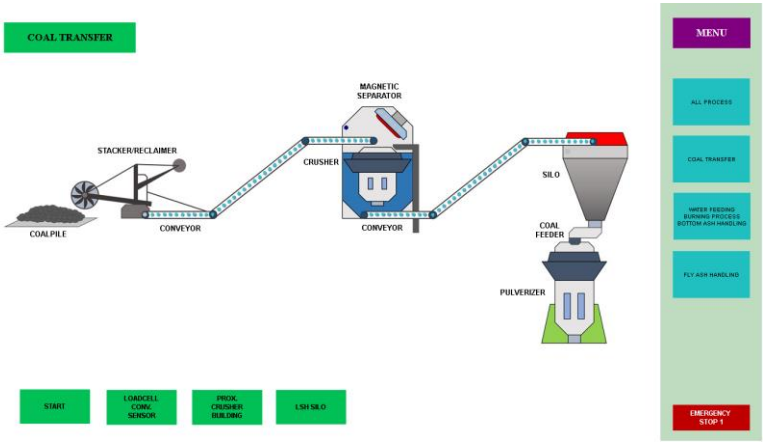
Pada perancangan *ladder diagram* ditambahkan eksternal sistem yaitu *Initial system* yang berfungsi sebagai syarat awal dari sistem berjalan dan sebagai sistem yang dapat menghentikan sistem keseluruhan ketika saat darurat. Ada dua buah tombol yaitu *start* dan *emergency*. Tombol *start* berfungsi sebagai syarat awal semua sistem berjalan, *emergency* berfungsi sebagai tombol untuk menghentikan sistem. Gambar 3.4. merupakan ladder dari *initial system* yang jika *start* ditekan

akan mengaktifkan *relay start lock*. *Start lock* tersebut digunakan untuk setiap awalan dari semua sub sistem yang ada. Contoh untuk diagram *ladder relay* sub sistem 1 akan di tunjukkan di gambar 3.3.

Gambar 3.4 *Ladder Diagram Initial System*

3.2 Perancangan *Human Machine Interface*

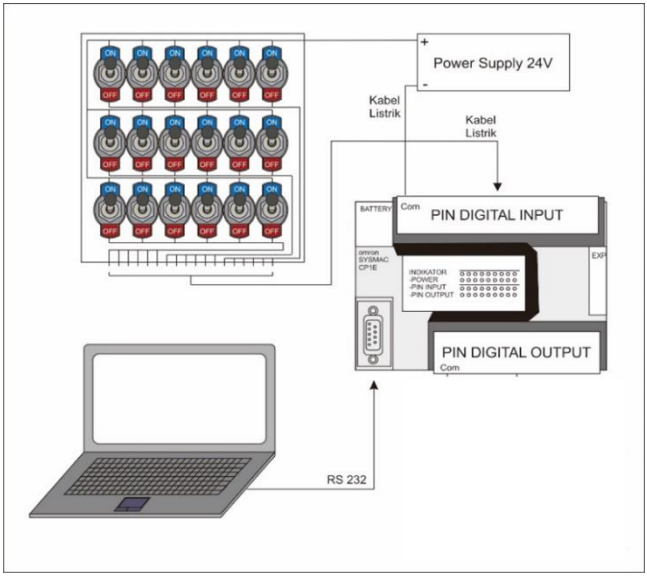
Setelah berhasil melakukan konversi *ladder diagram*, tahap selanjutnya adalah melakukan perancangan *Human Machine Interface* (HMI) menggunakan CX-Designer untuk memberi gambaran berupa visualisasi dari proses pengolahan batubara PLTU. Visualisasi HMI ini juga diperlukan untuk melihat keadaan *(state)* sistem saat melakukan pengujian simulasi sistem. Tampilan dari HMI yang dibuat memiliki 4 menu, yaitu menu "*All Process*" sebagai menu utama yang menunjukkan gambaran keseluruhan proses, kemudian menu *"Coal Transfer",* setelah itu *"Water feeding, Burning Process*, dan *Bottom Ash Handling"* yang digabung menjadi satu menu, dan yang terakhir adalah "*Fly Ash Handling".* Gambar 3.5 adalah contoh tampilan menu coal transfer, untuk menu lain dengan dengan tampilan yang lebih jelas dapat dilihat pada lampiran.



Gambar 3.5 Tampilan menu Coal Transfer

3.3 Perancangan Integrasi Hardware

Selanjutnya adalah perancangan Integrasi *Hardware*. Dibuat *toggle* akrilik yang berperan sebagai modul input, sementara HMI ditampilkan pada layar komputer. Pada Integrasi *Hardware*, *toggle* akrilik dihubungkan ke pin digital input pada PLC Omron. Digunakan *power suppy* dengan tegangan 24V sebagai sumber tegangan. Komputer berisi program *ladder diagram* yang diprogram dengan *CX-Programmer* dan HMI lalu dihubungkan dengan kabel RS 232 ke pin digital output PLC sehingga seluruh komponen dapat terintegrasikan. Gambar 3.6 menunjukkan integrasi antara PLC, input, dan output.



Gambar 3.6 Integrasi PLC, *input* dan *output*

BAB 4
PENGUJIAN DAN ANALISA

Ladder diagram yang telah dirancang dengan menggunakan metode *state diagram* kemudian dilakukan pengujian. Hal ini bertujuan untuk mempermudah proses analisa sistem.

4.1 Pengujian Sistem

Pengujian dilakukan untuk mengetahui tingkat keandalan dari sistem. Berdasarkan tingkatannya, proses pengujian terbagi atas 3 macam, yakni *unit testing*, *integraty testing*, dan *operational testing*. *Unit testing* adalah pengujian setiap komponen yang dilakukan secara terpisah. *Integration testing* adalah gabungan sejumlah dua komponen atau lebih yang dijalankan secara bersamaan. Sementara *operational testing* adalah pengujian sistem secara keseluruhan. Pada proses pengujian secara keseluruhan, yang menjadi parameter utama adalah waktu rata-rata dan serta kesesuaian langkah kerja.

4.1.1 Unit Testing

1. *Board Toggle*

   Pengujian dilakukan dengan cara memastikan Toogle dapat mengalirkan listrik dari Power Supply ketika Toogle aktif. Hasil dari pengujian ini dikatakan sesuai apabila keluaran Toogle berupa listrik dengan tegangan 24 Volt sesuai dengan tegangan yang dibutuhkan oleh *Digital Input* PLC. Pada Tabel 4.1 merupakan hasil pengujian Board Toggle.

Tabel 4.1 Hasil Pengujian *Board Toggle*

Nama Toogle	Tegangan Pin Output *Toogle* ketika diaktifkan
Tombol Start	24 Volt
Loadcell Sensor Conveyor	24 Volt
Proximity Sensor Crusher Building	24 Volt
LSH Silo	24 Volt
LSH Deaerator	24 Volt
LSH Steam Drum	24 Volt
LSH HSD	24 Volt

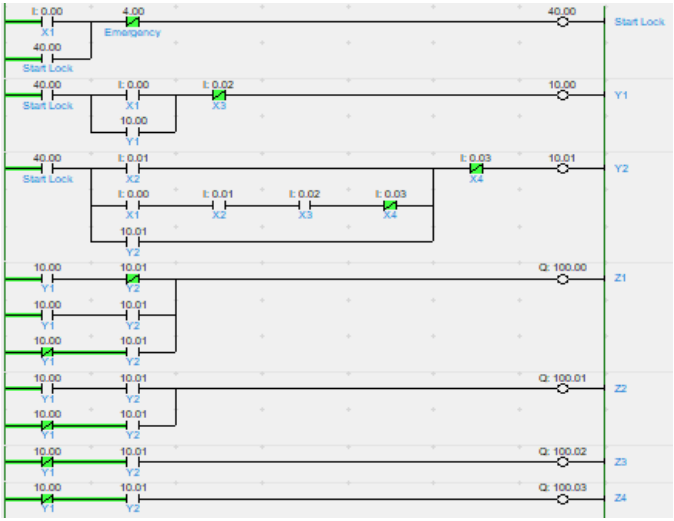
Oil Detector	24 Volt
Air Detector	24 Volt
Flame Detector	24 Volt
Sensor Beban Generator	24 Volt
Loadcell Hopper	24 Volt
Sensor suhu *Hopper*	24 Volt
Proximity Sensor SSC	24 Volt
Sensor Suhu EP	24 Volt
Timer Contact	24 Volt
Sensor Gas FGD	24 Volt
Tombol Start	24 Volt

2. Pengujian *Ladder Diagram*

Pengujian dilakukan dengan cara memastikan bahwa hasil konversi urutan proses menjadi *ladder diagram* dengan menggunakan metode *State Diagram* telah sesuai dengan sistem yang diinginkan. Pengujian dilakukan dengan cara melakukan *compile program* dan menguji tiap satu langkah deskripsi urutan proses yang telah dideskripsikan pada bab 3. Berikut adalah hasil pengujian *ladder diagram* untuk sub-proses 1. Adapun hasil pengujian *ladder diagram* yang lain dapat dilihat pada lampiran.

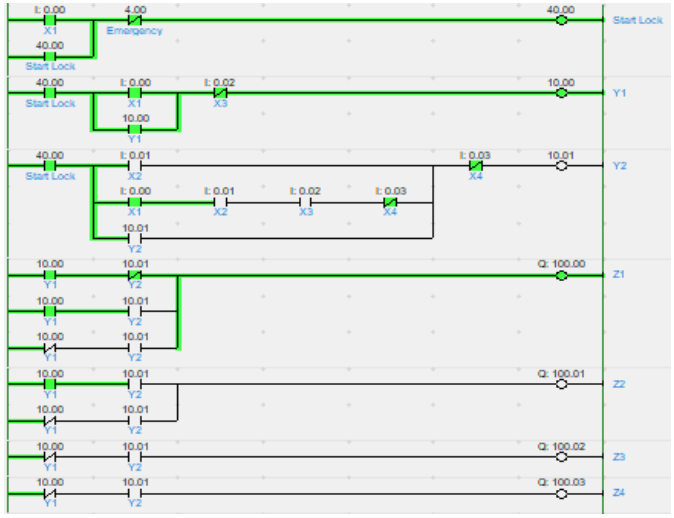
Sub-Proses 1:

1. Kondisi di mana sistem dalam keadaan mati, sehingga input maupun output tidak aktif



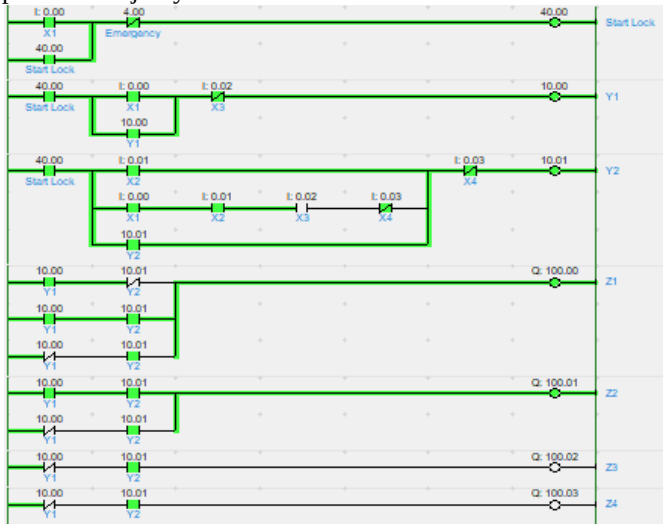
Gambar 4.1 Pengujian *Ladder Diagram* Langkah 1 pada sub-proses 1

2. Ketika tombol *start* ditekan, *Stacker/Reclaimer* aktif mengeruk batubara dari *coalpile* dan jatuh ke conveyor.



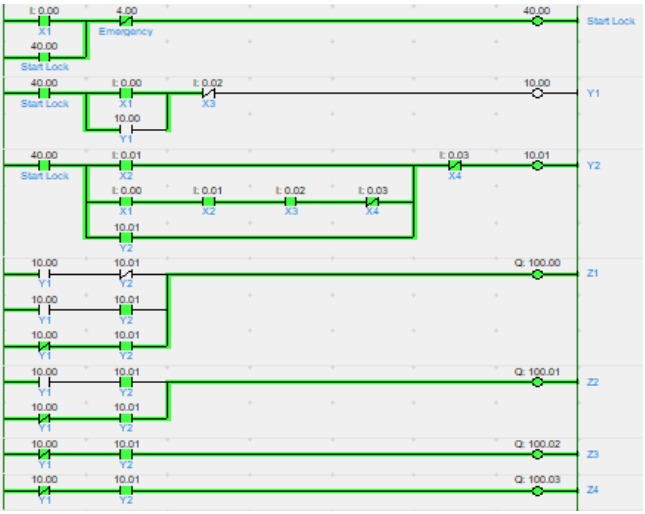
Gambar 4.2 Pengujian *Ladder Diagram* Langkah 2 pada sub-proses 1

3. Batubara yang jatuh mengaktifkan *loadcell sensor* pada conveyor sehingga conveyor aktif dan membawa batubara ke proses selanjutnya.



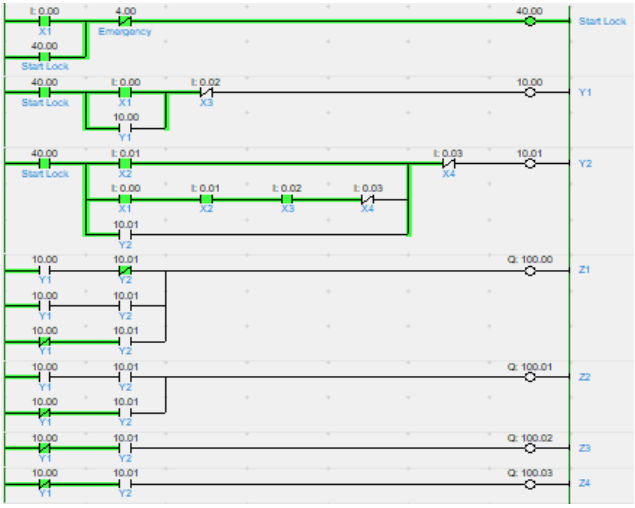
Gambar 4.3 Pengujian *Ladder Diagram* Langkah 3 pada sub-proses 1

4. Batubara masuk ke dalam crusher building melewati *proximity sensor* pada *crusher building* yang kemudian mengaktifkan *magnetic separator* sehingga memisahkan batubara dengan logam, sementara *crusher* menghancurkan batubara menjadi ukuran yang lebih kecil.



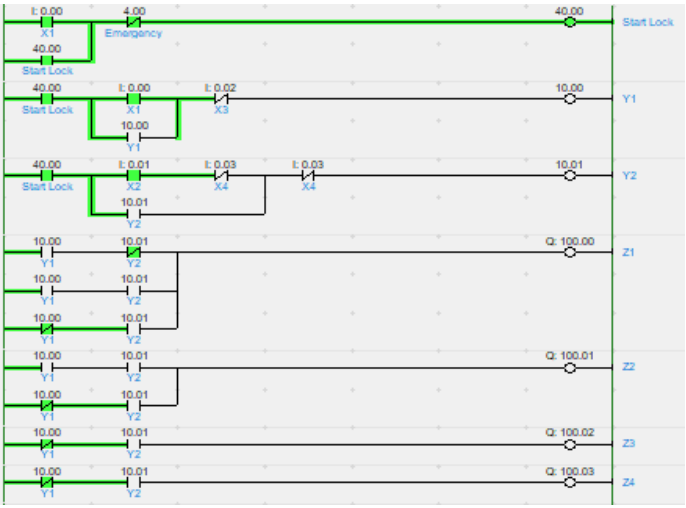
Gambar 4.4 Pengujian *Ladder Diagram* Langkah 4 pada sub-proses 1

5. Batubara selanjutnya dibawa oleh *conveyor* menuju ke *Silo*. Apabila Batubara telah mencapai *LSH Silo*, maka semua aktuator mati sehingga proses transfer batubara berhenti.



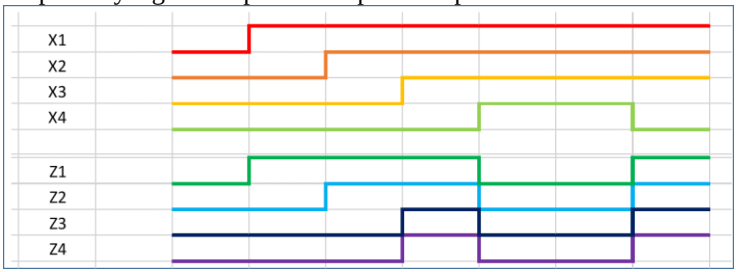
Gambar 4.5 Pengujian *Ladder Diagram* Langkah 5 pada sub-proses 1

6. Ketika volume batubara pada Silo berkurang karena pemakaian, maka LSH Silo akan non aktif sehingga seluruh aktuator pada sub-proses 1 akan kembali aktif.

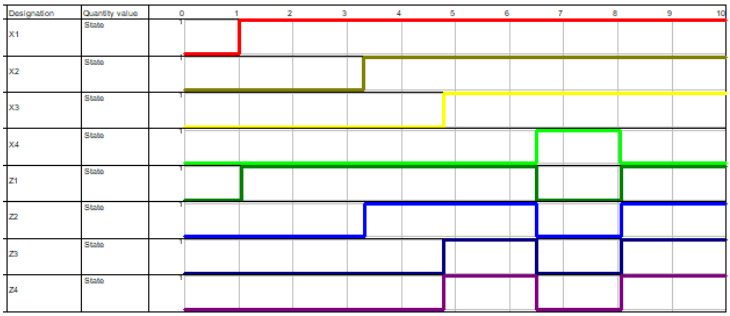


Gambar 4.6 Pengujian *Ladder Diagram* Langkah 6 pada sub-proses 1

Selanjutnya dilakukan validasi untuk melihat kesesuaian *sequence chart* hasil dari perancangan *ladder diagram* menggunakan metode *state diagram* dengan alur sistem hasil dari studi literatur. Validasi dilakukan dengan menggunakan *software Fluidsim*. Gambar 4.7 adalah rancangan sequence chart untuk sub-proses 1, sementara gambar 4.8 adalah hasil dari pengujian *ladder diagram* dengan menggunakan *fluidsim.* Adapun untuk hasil pengujian *ladder diagram* sub-proses yang lain dapat dilihat pada lampiran.



Gambar 4.7 Rancangan Sequence Chart sub-proses 1



Gambar 4.8 Hasil pengujian dengan fluidsim

3. PLC

Pengujian dilakukan dengan cara memastikan PLC mampu membaca sinyal input yang masuk ke pin Digital Input PLC dan mampu mengeluarkan sinyal Output melalui pin Digital Output PLC.

Tabel 4.2 Hasil pengujian Digital Input PLC

Alamat PLC	Kondisi Indikator Digital Input PLC ketika diaktifkan
0.00	Menyala
0.01	Menyala
0.02	Menyala
0.03	Menyala
0.04	Menyala
0.05	Menyala
0.06	Menyala
0.07	Menyala
0.08	Menyala
0.09	Menyala
0.10	Menyala
0.11	Menyala
1.00	Menyala
1.01	Menyala
1.02	Menyala
1.03	Menyala
1.04	Menyala

Tabel 4.3 Hasil pengujian Digital Output PLC

Alamat PLC	Tegangan Digital Output PLC ketika diaktifkan
100.00	Menyala
100.01	Menyala
100.02	Menyala
100.03	Menyala
100.04	Menyala
100.05	Menyala
100.06	Menyala
100.07	Menyala
100.08	Menyala
100.09	Menyala
100.10	Menyala
100.11	Menyala
100.12	Menyala
100.13	Menyala
100.14	Menyala
100.15	Menyala

4.1.2 *Operational Testing*

1. Board Toggle dan PLC

   Proses pengujian dilakukan dengan cara menghubungkan keluaran dari *Toogle* dengan pin digital input PLC. Keberhasilan dari pengujian ini ditunjukkan dengan menyalanya lampu indikator Digital Input PLC ketika toogle diaktifkan.

Tabel 4.4 Hasil Pengujian *Board Toggle* dan PLC

Nama Toogle	Alamat Input PLC	Kondisi Indikator PLC ketika diaktifkan
Tombol Start	0.00	Menyala
Loadcell Sensor Conveyor	0.01	Menyala
Proximity Sensor Crusher Building	0.02	Menyala
LSH Silo	0.03	Menyala
LSH Deaerator	0.04	Menyala
LSH Steam Drum	0.05	Menyala
LSH HSD	0.06	Menyala
Oil Detector	0.07	Menyala

Air Detector	0.08	Menyala
Flame Detector	0.09	Menyala
Sensor Beban Generator	0.10	Menyala
Loadcell Hopper	0.11	Menyala
Sensor suhu *Hopper*	1.00	Menyala
Proximity Sensor SSC	1.01	Menyala
Sensor Suhu EP	1.02	Menyala
Timer Contact	1.03	Menyala
Sensor Gas FGD	1.04	Menyala

2. Output data dari PLC ditampilkan ke HMI

Proses pengujian interation Testing antara PLC dengan HMI dilakukan dengan cara : Pin Digital Output PLC diaktifkan menggunakan sebuah Toggle. Ketika Toggle aktif maka PLC memproses sehingga pin Digital Output yang diuji aktif. Ketika Pin Digital Output PLC aktif, HMI akan mengaktifkan indikator Aktuator yang sesuai dengan Pin Digital Output yang diuji. Pengujian ini berhasil ketika Pin Digital Output PLC yang aktif juga diikuti dengan aktifnya Indikator Akuator pada HMI.

Tabel 4.5 Hasil Pengujian Output dari PLC ditampilkan ke HMI

Nama Pin	Alamat Output PLC	Tegangan Digital Output PLC ketika diaktifkan
Stacker/reclaimer	100.00	Menyala
Conveyor	100.01	Menyala
Magnetic Separator	100.02	Menyala
Crusher	100.03	Menyala
Boiler Feed Pump	100.04	Menyala
Boiler Circulating Water Pump	100.05	Menyala
Forced Draft Fan	100.06	Menyala
HSD Pump	100.07	Menyala
Ignitor	100.08	Menyala
Primary Air Heater	100.09	Menyala
Secondary Air Heater	100.10	Menyala
Valve Coal Feeder	100.11	Menyala
Crusher Pulverizer	100.12	Menyala
Primary Air Heater Fan	100.13	Menyala

Transition Hopper Valve	100.14	Menyala
Submerged Scrapper Conveyor (SSC)	100.15	Menyala
Bottom ash conveyor	200.00	Menyala
Recirculation Water Pump	200.01	Menyala
Clinker Grinder	200.03	Menyala
Induced Draft Fan 1	200.04	Menyala
Induced Draft Fan 2	200.05	Menyala
High Voltage Transformer Rectifier	200.06	Menyala
Rapper	200.07	Menyala
Vacuum Blower	200.08	Menyala
Sea Water Scrubber Pump	200.09	Menyala
Water Spray	200.10	Menyala
Induced Draft Fan 3	200.11	Menyala

4.1.3 Pengujian Integrasi Sistem

1. Input data dari Board Toggle dengan HMI

Proses pengujian integrasi antara Board *Toogle* dengan HMI dilakukan dengan cara : *Toggle* pada *Board Toggle* akan diaktifkan satu persatu. Ketika *Toggle* aktif maka mengirimkan sinyal digital ke pin Digital Input PLC. PLC akan menyimpan sinyal tersebut pada memori PLC sekaligus mengirimkan sinyal untuk mengaktifkan indikator HMI yang sesuai dengan alamat Digital Input PLC. Pengujian ini berhasil apabila setiap *Toggle* ketika diaktifkan menyebabkan indikator pada HMI juga ikut aktif, ditunjukkan dengan menyalanya lampu indikator HMI.

Tabel 4.6 Hasil Pengujian Integrasi Board Toggle dengan HMI

Nama Toogle Input	Alamat PLC	Kondisi Indikator Sensor pada HMI ketika diaktifkan
Tombol Start	0.00	Menyala
Loadcell Sensor Conveyor	0.01	Menyala
Proximity Sensor Crusher Building	0.02	Menyala
LSH Silo	0.03	Menyala
LSH Deaerator	0.04	Menyala

LSH Steam Drum	0.05	Menyala
LSH HSD	0.06	Menyala
Oil Detector	0.07	Menyala
Air Detector	0.08	Menyala
Flame Detector	0.09	Menyala
Sensor Beban Generator	0.10	Menyala
Loadcell Hopper	0.11	Menyala
Sensor suhu *Hopper*	1.00	Menyala
Proximity Sensor SSC	1.01	Menyala
Sensor Suhu EP	1.02	Menyala
Timer Contact	1.03	Menyala
Sensor Gas FGD	1.04	Menyala

2. Output data dari PLC ditampilkan ke HMI

Proses pengujian integrasi antara PLC dengan HMI dilakukan dengan cara : Pin Digital Output PLC diaktifkan menggunakan sebuah *Toggle*. Ketika *Toggle* diaktfikan maka PLC memproses sehingga pin Digital Output yang diuji aktif. Ketika Pin Digital Output PLC aktif, HMI akan mengaktifkan indikator Aktuator yang sesuai dengan Pin Digital Output yang diuji. Pengujian ini berhasil ketika Pin Digital Output PLC yang aktif juga diikuti dengan aktifnya Indikator Akuator pada HMI.

Tabel 4.7 Hasil Pengujian Output dari PLC ditampilkan ke HMI

Nama Output	Alamat PLC	Kondisi Indikator Aktuator pada HMI ketika diaktifkan
Stacker/reclaimer	100.00	Menyala
Conveyor	100.01	Menyala
Magnetic Separator	100.02	Menyala
Crusher	100.03	Menyala
Boiler Feed Pump	100.04	Menyala
Boiler Circulating Water Pump	100.05	Menyala
Forced Draft Fan	100.06	Menyala
HSD Pump	100.07	Menyala
Ignitor	100.08	Menyala

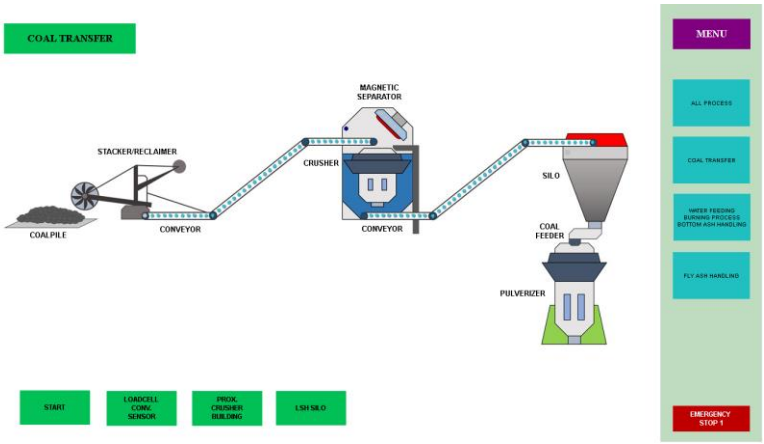
Primary Air Heater	100.09	Menyala
Secondary Air Heater	100.10	Menyala
Valve Coal Feeder	100.11	Menyala
Crusher Pulverizer	100.12	Menyala
Primary Air Heater Fan	100.13	Menyala
Transition Hopper Valve	100.14	Menyala
Submerged Scrapper Conveyor (SSC)	100.15	Menyala
Bottom ash conveyor	200.00	Menyala
Recirculation Water Pump	200.01	Menyala
Clinker Grinder	200.03	Menyala
Induced Draft Fan 1	200.04	Menyala
Induced Draft Fan 2	200.05	Menyala
High Voltage Transformer Rectifier	200.06	Menyala
Rapper	200.07	Menyala
Vacuum Blower	200.08	Menyala
Sea Water Scrubber Pump	200.09	Menyala
Water Spray	200.10	Menyala
Induced Draft Fan 3	200.11	Menyala

3. Pengujian Keseluruhan Sistem Berdasarkan *Sequence Chart*

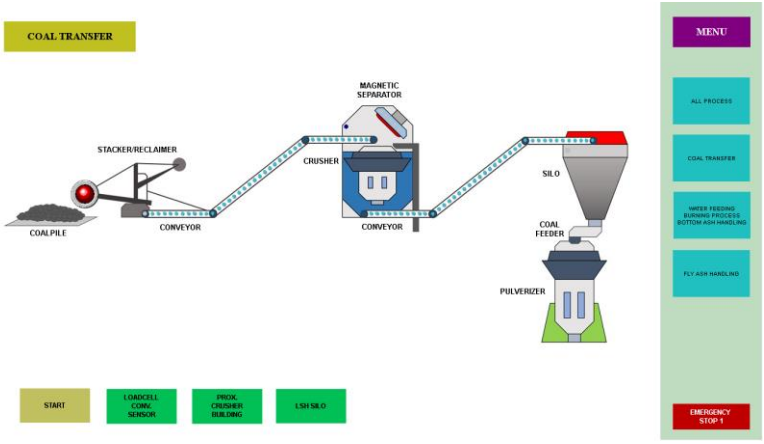
Pengujian keseluruhan sistem merupakan pengujian akhir yang dilakukan untuk memastikan setiap perangkat yang dirancang telah bekerja sesuai dengan fungsinya dan mampu berintegrasi dengan perangkat lain sehingga mampu menghasilkan sistem yang sesuai dengan tujuan perancangan. Tahapan pengujian Keseluruhan Sistem yaitu :

a. *Wiring* Setiap perangkat sesuai dengan perancangan Integrasi pada Bab 3
b. *Transfer* l*adder diagram* menggunakan metode *state diagram* yang telah dirancang ke CPU PLC
c. Buka Aplikasi HMI dan lakukan proses komunikasi antara PLC dan HMI
d. Lakukan pengujian sesuai *Sequence Chart* yang telah ditentukan dengan cara mengubah kondisi Input pada Board Toogle

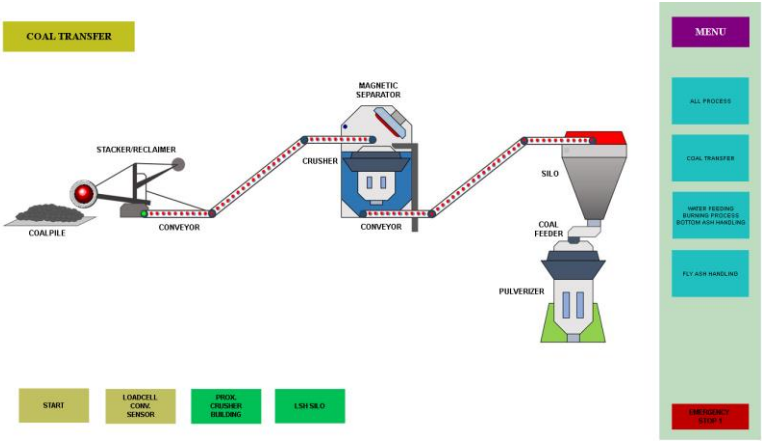
Dari tahapan pengujian tersebut didapatkan tampilan hasil HMI proses pengujian berdasarkan *sequence chart* sebagai berikut untuk sub proses 1, adapun untuk sub proses yang lain dapat dilihat pada lampiran.



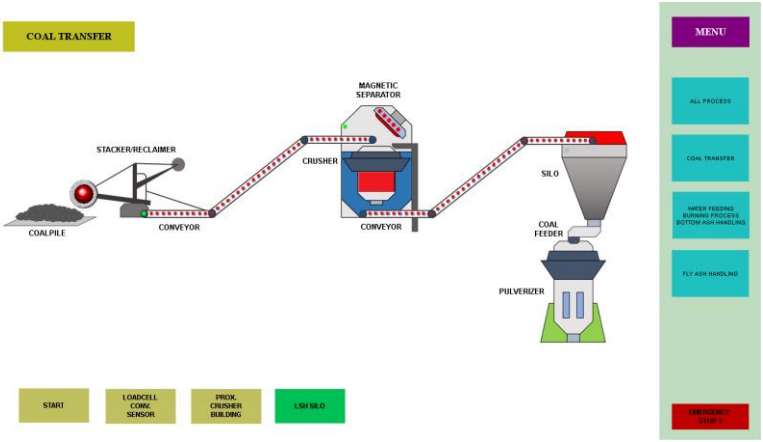
Gambar 4.9 Langkah 1 sub proses 1



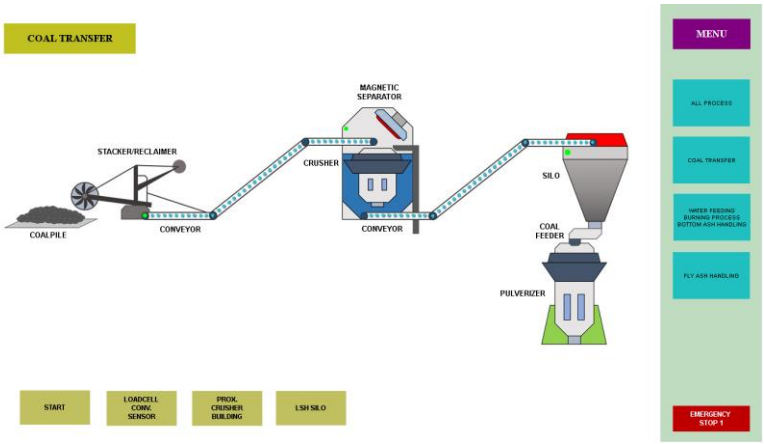
Gambar 4.10 Langkah 2 sub proses 1



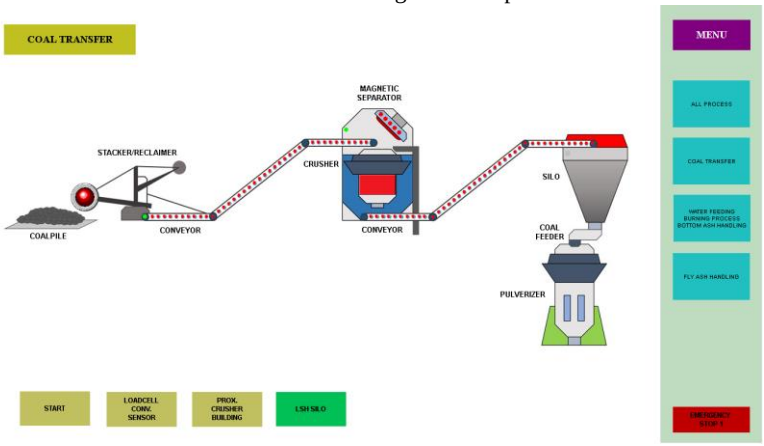
Gambar 4.11 Langkah 3 sub proses 1



Gambar 4.12 Langkah 4 sub proses 1



Gambar 4.13 Langkah 5 sub proses 1



Gambar 4.14 Langkah 6 sub proses 1

Tabel 4.8 Pengujian Sistem Keseluruhan

Sub Proses	No	Kondisi Input	Kondisi Output	Normal
1	1	Input off	Output off	√
	2	Tombol Start on	*Stacker/reclaimer* on	√

	3	*Loadcell Sensor Conveyor* on	Conveyor on	√
	4	*Proximity Sensor CB on*	-*Magnetic Separator* on -*Crusher* on	√
	5	*LSH Silo* on	-*Stacker/reclaimer* off -Conveyor off -*Magnetic Separator* off -*Crusher* off	√
	6	*LSH Silo* off	-*Stacker/reclaimer* on -Conveyor on -*Magnetic Separator* on -*Crusher* on	√
2	1	Seluruh input off	Seluruh output off	√
	2	-Tombol Start on -*Limit Switch High* Deaerator on	*Boiler Feed Pump* on	√
	3	*Limit Switch High Steam Drum* on	-*Boiler Circulating Water Pump* on -*Forced Draft Fan* on -*Boiler Feed Pump* on	√
3	1	-*Limit Switch High HSD* on -*Limit Switch High Steam Drum* on	*HSD Pump* on	√
	2	-*Oil Detector* on -*Air Detector* on	*Ignitor* on	√
	3	*Flame detector* on	-*Primary Air Heater* on -*Secondary Air Heater* on	√
	4	*Sensor Beban Generator* aktif 30%	-HSD Pump off -*Valve Coal Feeder* on -Crusher Pulverizer on -*Primary Air Heater Fan* on	√

4	1	*Loadcell Hopper* on	-*Transition Hopper Valve* on -*Submerged Scrapper Conveyor (SSC)* on -*Bottom Ash Conveyor* on	√
	2	Sensor suhu *Hopper* on	*Recirculation Water Pump* on	√
	3	*Proximity Sensor SSC* on	*Clinker Grinder* on	√
5	1	*Sensor Suhu ESP* on	-*Induced Draft Fan* 1 on -*Induced Draft Fan* 2 on -*High Voltage Transformer Rectifier* on	√
	2	Timer Contact on	-Rapper aktif periodik -*Vacuum Blower* on	√
	3	*Gas Sensor FGD* on	-*Sea Water Scrubber Pump* on -*Water Spray* on -*Induced Draft Fan* 3 on	√

Tabel 4.8 merupakan tabel hasil pengujian sistem keseluruhan. Hasil pengujian menunjukkan sistem berjalan sesuai dengan *sequence chart* yang telah ditentukan dan tidak terdapat masalah pada proses komunikasi antara PLC dengan HMI. Dapat disimpulkan bahwa perancagan *ladder diagram* menggunakan metode *state diagram* berjalan lancar dan dapat diimplementasikan pada sistem.

4.2 Analisa Hasil

Implementasi dari perancangan *ladder diagram* dengan menggunakan metode *state diagram* pada pengolahan batubara PLTU telah sesuai dengan perancangan awal. Desain *ladder diagram* menghasilkan total 46 *rung ladder* dengan rincian 10 *rung relay* proses dan 27 *rung output* (aktuator), serta penunjang sistem keseluruhan sebanyak 5 rung sistem eksternal sebagai indikator pada *ladder diagram*, 1 *rung* memori *timer*, 2 *rung timer*, 1 *rung* sistem *emergency stop*. Total memori untuk file program yang dihasilkan sebesar 11 KB dengan rincian 3 KB file (.BAK), 3 KB file (.CXP), dan 5 KB file (.OPT). Adapun hasil *overview* HMI memiliki 4 tampilan menu, yaitu menu "*All Process*" sebagai menu utama yang menunjukkan gambaran keseluruhan proses, kemudian menu *"Coal Transfer",* setelah itu *"Water feeding, Burning Process*, dan *Bottom Ash Handling"* yang digabung menjadi satu menu,

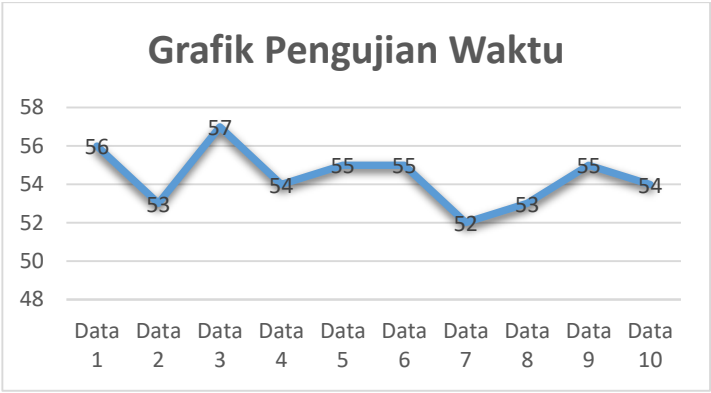
dan yang terakhir adalah "*Fly Ash Handling".* Adapun kapasitas memori yang digunakan pada HMI sebesar 2,98 Mb.

4.2.1 Waktu Proses

Keseluruhan proses pada HMI dari *CX Designer* dijalankan sebanyak 10 kali. Hal ini bertujuan untuk menguji lama waktu rata-rata yang dibutuhkan keseluruhan sistem untuk menyala. Berikut hasil tabel pencatatan waktu sistem ketika dijalankan sebanyak 10 kali.

Tabel 4.9 Hasil pencatatan waktu pengujian implementasi pada PLC

Data	Waktu (detik)
1	56
2	53
3	57
4	54
5	55
6	55
7	52
8	53
9	55
10	54



Gambar 4.15 Grafik Pengujian Waktu Proses Keseluruhan

Berdasarkan tabel 4.9, rata-rata waktu yang dibutuhkan untuk menjalankan proses dari awal hingga akhir sistem sekitar 54,4 detik. Selisih waktu pada setiap pengujian dapat terjadi karena proses pengubahan kondisi *Toggle* masih manual. Apabila diterapkan pada sebuah sistem yang menggunakan sensor maka respon PLC akan lebih cepat dan tidak terdapat selisih waktu.

[Halaman ini sengaja dikosongkan]

BAB 5
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dari hasil desain *ladder diagram* menggunakan metode *state diagram* untuk proses pengolahan batubara pada PLTU mempunyai hasil akhir sebagai berikut :

1. Perancangan *Ladder Diagram* terdiri dari 18 *sequences* dengan pembagian 5 sub-proses.
2. Terdapat 18 *input* dan 27 *output* yang digunakan pada sistem.
3. Desain *ladder diagram* dengan metode s*tate diagram* untuk Proses Pengolahan Batubara PLTU menghasilkan total 46 *rung ladder* dengan rincian 10 *rung relay* proses, 27 *rung output* dan 5 *rung* sistem eksternal, 1 *rung* memori *timer*, 2 *rung timer*, 1 *rung* sistem *emergency stop* yang berfungsi sebagai *relay* pendukung.
4. Kapasitas memori untuk file program yang dihasilkan sebesar 11 KB dengan rincian 3 KB file (.BAK), 3 KB file (.CXP), dan 5 KB file (.OPT).
5. Dilakukan uji coba sebanyak 10 kali dengan hasil waktu rata-rata 54,4 detik.

5.2 Saran

Dalam pembuatan dan perancangan *ladder diagram*, terdapat banyak metode dengan keunggulan masing-masing yang bisa digunakan. Sehingga untuk mengetahui metode yang paling efisien dalam mendesain *ladder diagram* perlu untuk dilakukan perbandingan.

[Halaman ini sengaja dikosongkan]

DAFTAR PUSTAKA

[1] Triboesono, Agoes, “Statistik Ketenagalistrikan 2017”, Jakarta: Direktorat Jenderal Ketenagalistrikan. 2017.

[2] Azis, Fathoni, dkk, “Deskripsi Sistem Dan Fungsi General Pada Pembangkit Listrik Tenaga Uap Unit 7 Dan 8 PT. IPMOMI Paiton Probolinggo”, Malang: Universitas Brawijaya, 2015.

[3] Endurra Consulting, “Penanganan Batubara untuk Pembangkit Listrik-Coal Handling”, <URL: http://www.endurra.co.id/artikel/penanganan-batu-bara-untuk-pembangkit-listrik-coal-handling/>, diakses 29 April, 2019.

[4] Zainal, Ahmad Rizky dan Tejo Sukmadi, “Ash Handling System PT. Pembangkitan Jawa Bali Unit Bisnis Jasa Operasi Dan Maintenance (Ubjom) PLTU Rembang”, Semarang: Universitas Diponegoro, 2015.

[5] Lodhi, Girja, “Operation And Maintenance Of Crusher House For Coal Handling In Thermal Power Plant”, International Journal of Mechanical Engineering & Robotics Research 2013.

[6] K.-H. John and M. Tiegelkamp, IEC61131-3: Programming Industrial Automation Systems, Berlin Heidelberg: Springer-Verlag, 2001.

[7] Nobel, Mahardika, “Konstruksi Diagram Ladder Menggunakan Metode Flow-Table/State Diagram Untuk Crude Palm Oil Process”, Surabaya: Institut Teknologi Sepuluh Nopember 2018.

[8] Iskandar, Eka, Mochammad Rameli dan Nicco, “Ladder Diagram based on State Diagram for Selection and Assembling Part on Dual Conveyor”, Surabaya: Sepuluh Nopember Institute of Tecnologi, 2017.

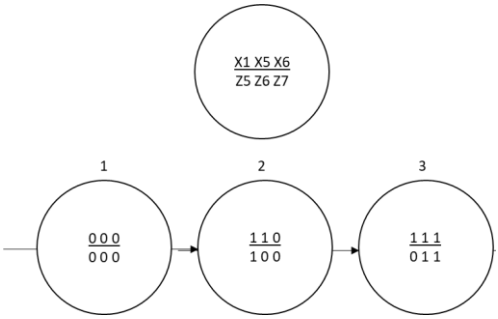
[Halaman ini sengaja dikosongkan]

LAMPIRAN

Perancangan State Diagram

Sub proses 2

State Diagram I/O



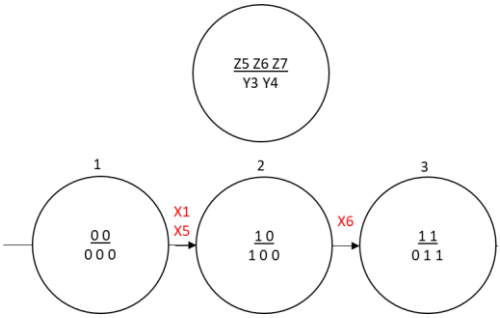
Primitive Flow Table

Row	Input X1 X5 X6			Output Z5 Z6 Z7		
	000	110	111	Z5	Z6	Z7
1	**1**	2	-	0	0	0
2	-	**2**	3	1	0	0
3	-	-	**3**	0	1	1

Merged Flow Table

Row	Input X1 X5 X6			Output Z5 Z6 Z7			Relay	
	000	110	111	Z5	Z6	Z7	Y3	Y4
1	**1**	2	-	0	0	0	0	0
2	-	**2**	3	1	0	0	1	0
3	-	-	**3**	0	1	1	1	1

State Diagram R/O



Switching Function

Y3 = (X1*X5+Y3)

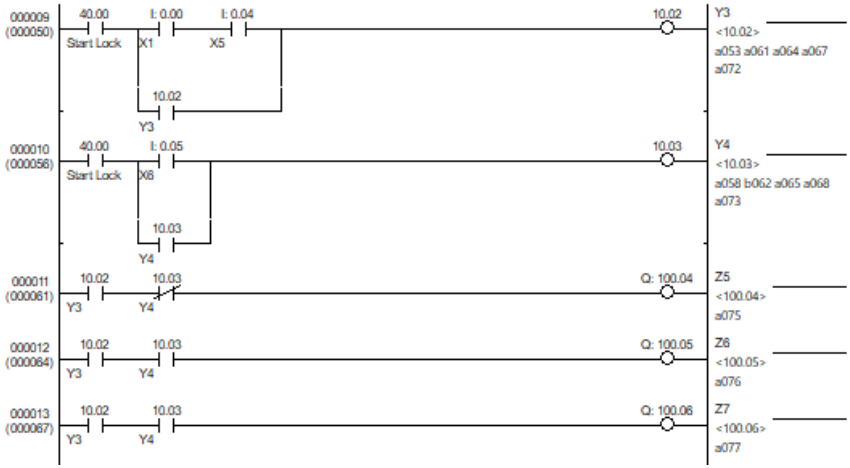
Y4 = (X6+Y4)

Z5 = (Y3*$\overline{Y4}$)

Z6 = (Y3*Y4)

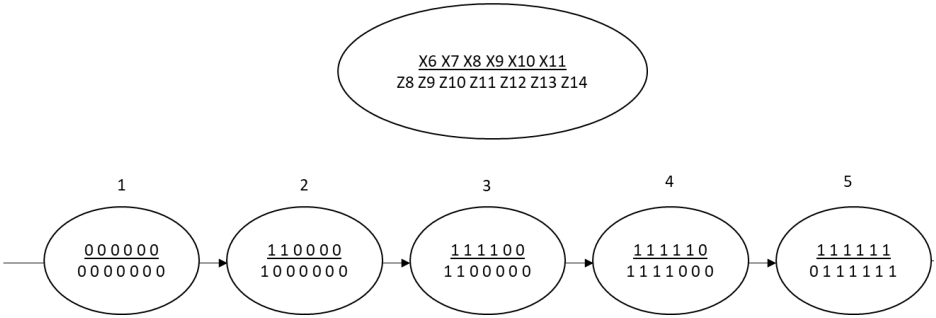
Z7 = (Y3*Y4)

Ladder Diagram



Sub Proses 3

State Diagram I/O



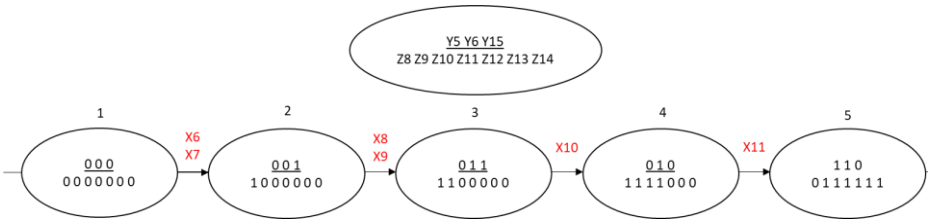
Primitive Flow Table

Row	Input X6 X7 X8 X9 X10 X11					Output Z8 Z9 Z10 Z11 Z12 Z13 Z14						
	000000	110000	111100	111110	111111	Z8	Z9	Z10	Z11	Z12	Z13	Z14
1	**1**	2	-	-	-	0	0	0	0	0	0	0
2	-	**2**	3	-	-	1	0	0	0	0	0	0
3	-	-	**3**	4	-	1	1	0	0	0	0	0
4	-	-	-	**4**	5	1	1	1	1	0	0	0
5	-	-	-	-	**5**	0	1	1	1	1	1	1

Merged Flow Table

Row	Input X6 X7 X8 X9 X10 X11					Output Z8 Z9 Z10 Z11 Z12 Z13 Z14							Relay		
	000000	110000	111100	111110	111111	Z8	Z9	Z10	Z11	Z12	Z13	Z14	Y5	Y6	Y15
1	**1**	2	-	-	-	0	0	0	0	0	0	0	0	0	0
2	-	**2**	3	-	-	1	0	0	0	0	0	0	0	0	1
3	-	-	**3**	4	-	1	1	0	0	0	0	0	0	1	1
4	-	-	-	**4**	5	1	1	1	1	0	0	0	0	1	0
5	-	-	-	-	**5**	0	1	1	1	1	1	1	1	1	0

State Diagram R/O



Switching Function

Y5 = (X11+Y5)

Y6 = (X8*X9+Y6)

Y15 = (X6*X7+Y15)* ($\overline{X10}$)

Z8 = ($\overline{Y5}$ *$\overline{Y6}$ *Y15)+($\overline{Y5}$ *Y6*Y15)+($\overline{Y5}$*Y6* $\overline{Y15}$)

Z9 = ($\overline{Y5}$*Y6*Y15)+($\overline{Y5}$*X6* $\overline{Y15}$)+(Y5*Y6* $\overline{Y15}$)

Z10 = ($\overline{Y5}$*Y6* $\overline{Y15}$)+(Y5*Y6* $\overline{Y15}$)

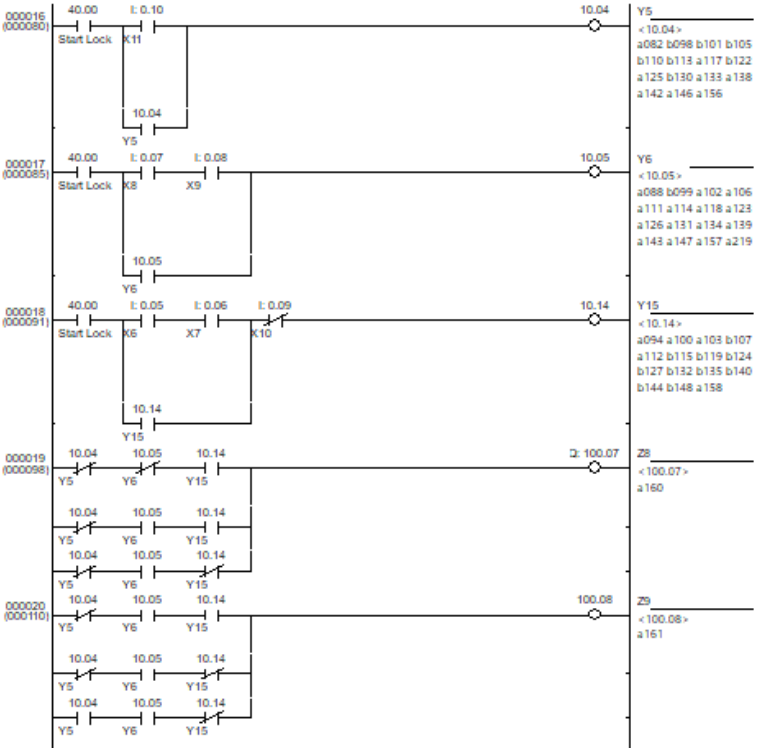
$Z11 = (\overline{Y5}*Y6*\overline{Y15})+(Y5*Y6*\overline{Y15})$

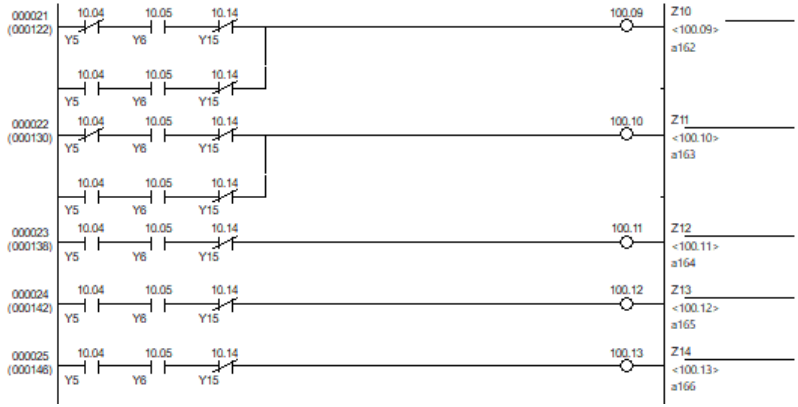
$Z12 = (Y5*Y6*\overline{Y15})$

$Z13 = (Y5*Y6*\overline{Y15})$

$Z14 = (Y5*Y6*\overline{Y15})$

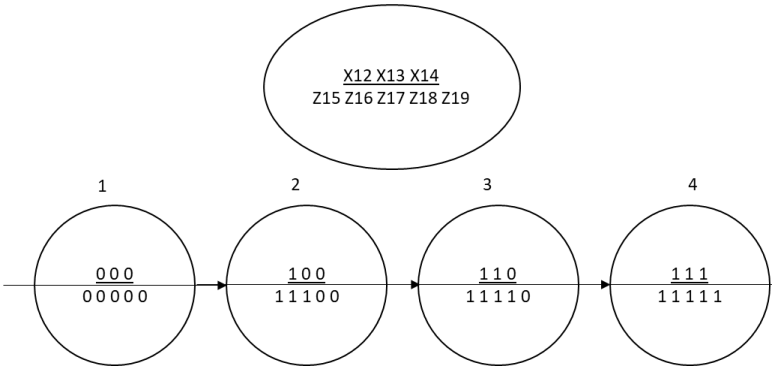
Ladder Diagram





Sub P**roses** 4

State Diagram I/O



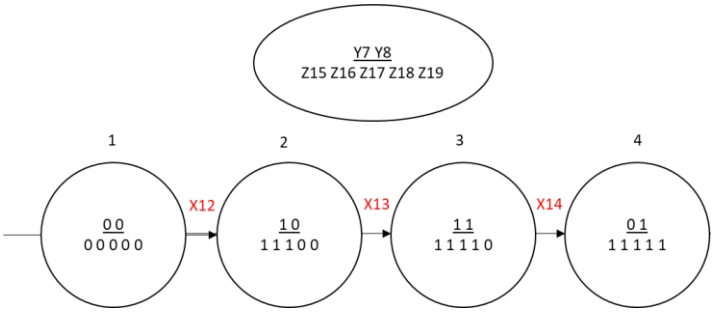
Primitive Flow Table

Row	Inputs X12 X13 X14				Outputs Z15 Z16 Z17 Z18 Z19				
	0 0 0	1 0 0	1 1 0	1 1 1	Z15	Z16	Z17	Z18	Z19
1	**1**	2	-	-	0	0	0	0	0
2	-	**2**	3	-	1	1	1	0	0
3	-	-	**3**	4	1	1	1	1	0
4	-	-	-	**4**	1	1	1	1	1

Merged Flow Table

Row	Inputs X12 X13 X14				Outputs Z15 Z16 Z17 Z18 Z19					Relay	
	0 0 0	1 0 0	1 1 0	1 1 1	Z15	Z16	Z17	Z18	Z19	Y7	Y8
1	**1**	2	-	-	0	0	0	0	0	0	0
2	-	**2**	3	-	1	1	1	0	0	1	0
3	-	-	**3**	4	1	1	1	1	0	1	1
4	-	-	-	**4**	1	1	1	1	1	0	1

State Diagram R/O



Switching Function

$Y7 = (X12+Y7)*(\overline{X14})$

$Y8 = (X13+Y8)$

$Z15 = (Y7*\overline{Y8})+(Y7*Y8)+(\overline{Y7}*Y8)$

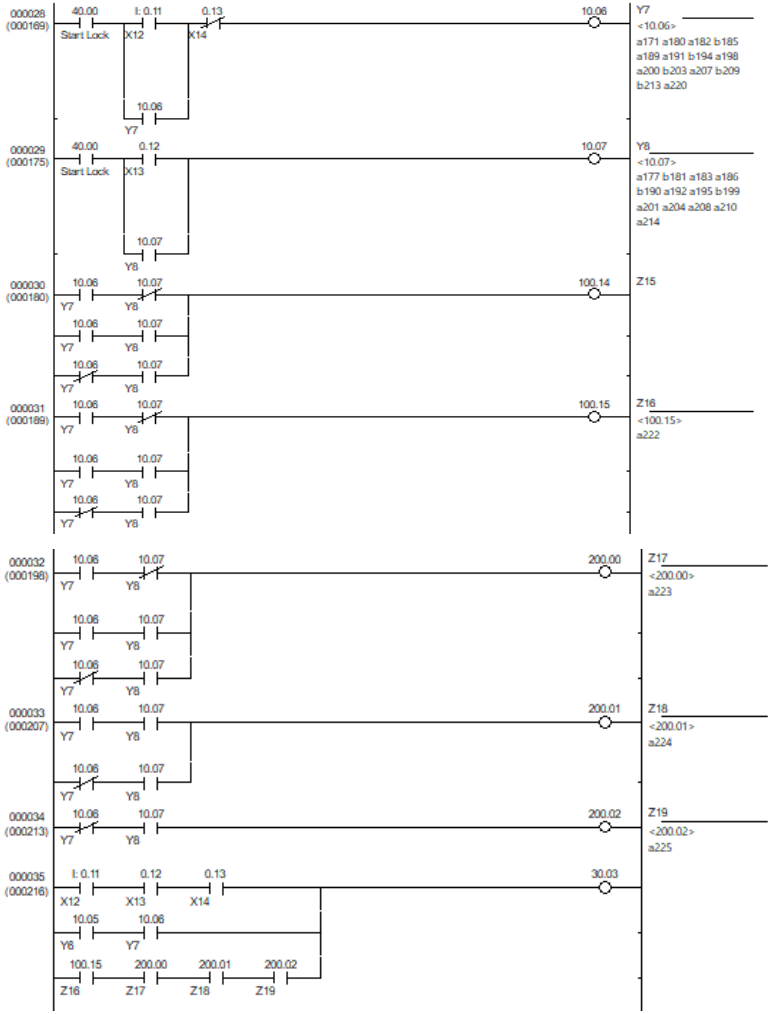
$Z16 = (Y7*\overline{Y8})+(Y7*Y8)+(\overline{Y7}*Y8)$

$Z17 = (Y7*\overline{Y8})+(Y7*Y8)+(\overline{Y7}*Y8)$

$Z18 = (Y7*Y8)+(\overline{Y7}*Y8)$

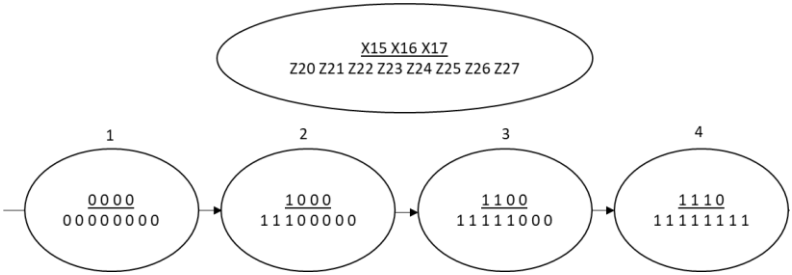
$Z19 = (\overline{Y7}*Y8)$

Ladder Diagram



Sub Proses 5

State Diagram I/O



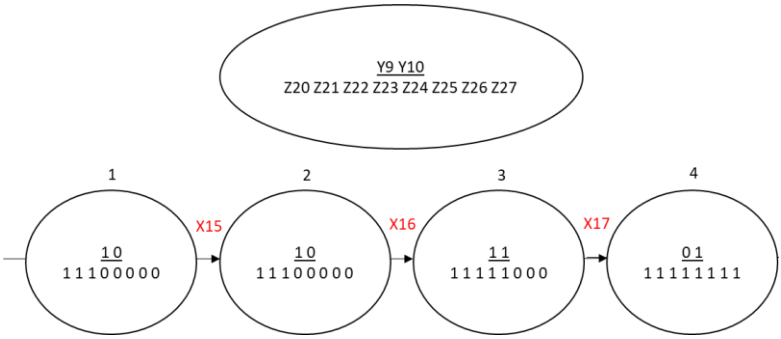
Primitive Flow Table

Row	Input X15 X16 X17				Output Z20 Z21 Z22 Z23 Z24 Z25 Z26 Z27							
	0000	1000	1100	1110	Z20	Z21	Z22	Z23	Z24	Z25	Z26	Z27
1	**1**	2	-	-	0	0	0	0	0	0	0	0
2	-	**2**	3	-	1	1	1	0	0	0	0	0
3	-	-	**3**	4	1	1	1	1	1	0	0	0
4	-	-	-	**4**	1	1	1	1	1	1	1	1

Merged Flow Table

Row	Input X15 X16 X17				Output Z20 Z21 Z22 Z23 Z24 Z25 Z26 Z27								Relay	
	0000	1000	1100	1110	Z20	Z21	Z22	Z23	Z24	Z25	Z26	Z27	Y9	Y10
1	**1**	2	-	-	0	0	0	0	0	0	0	0	0	0
2	-	**2**	3	-	1	1	1	0	0	0	0	0	1	0
3	-	-	**3**	4	1	1	1	1	1	0	0	0	1	1
4	-	-	-	**4**	1	1	1	1	1	1	1	1	0	1

State Diagram R/O



Switching Function

$Y9 = (X15+Y9)*(\overline{X17})$

$Y10 = (X16+Y10)$

$Z20 = (Y9*\overline{Y10})+ (Y9*Y10)+ (\overline{Y9}*Y10)$

$Z21 = (Y9*\overline{Y10})+ (Y9*Y10)+ (\overline{Y9}*Y10)$

$Z22 = (Y9*\overline{Y10})+ (Y9*Y10)+ (\overline{Y9}*Y10)$

$Z23 = (Y9*Y10)+ (\overline{Y9}*Y10)$

$Z24 = (Y9*Y10)+ (\overline{Y9}*Y10)$

$Z25 = (\overline{Y9}*Y10)$

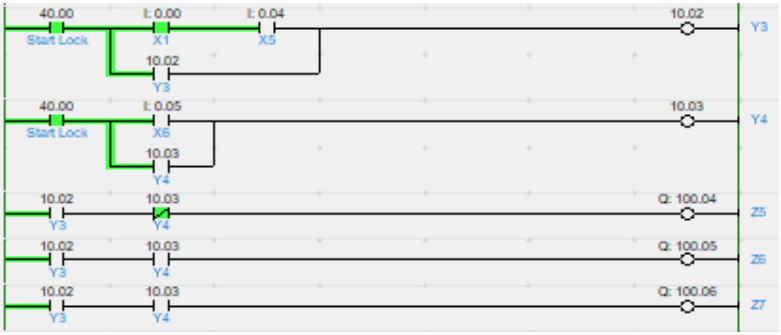
$Z26 = (\overline{Y9}*Y10)$

$Z27 = (\overline{Y9}*Y10)$

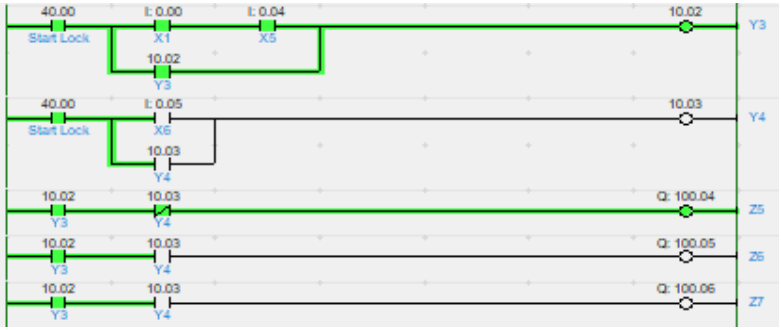
Pengujian Ladder Diagram

Sub proses 2

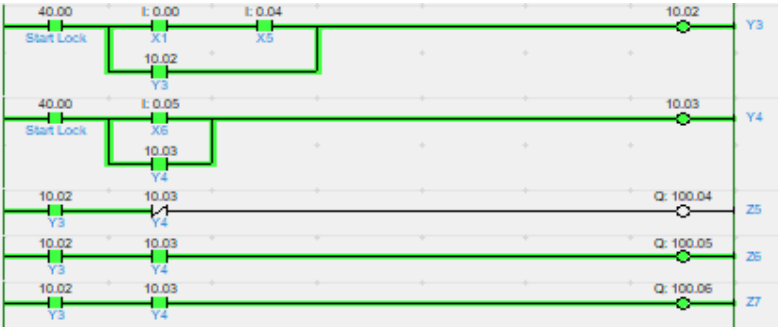
Langkah 1



Langkah 2

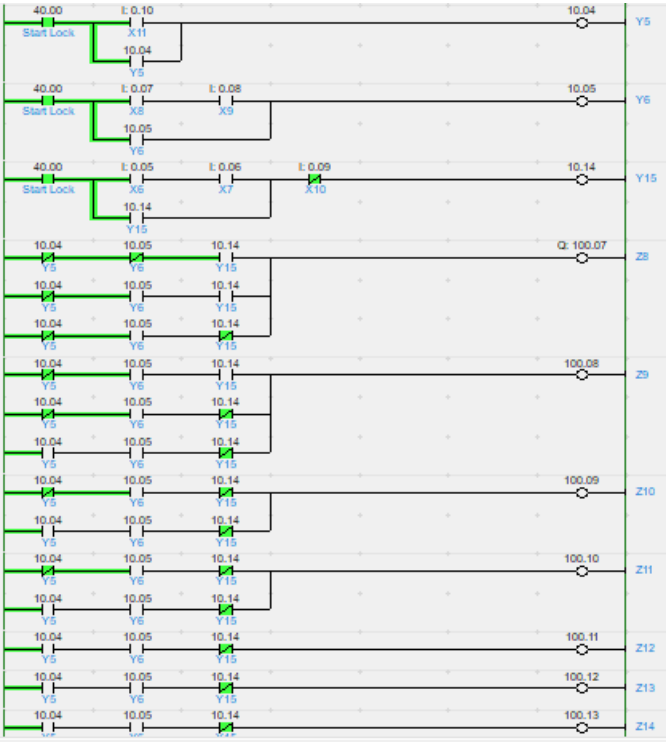


Langkah 3

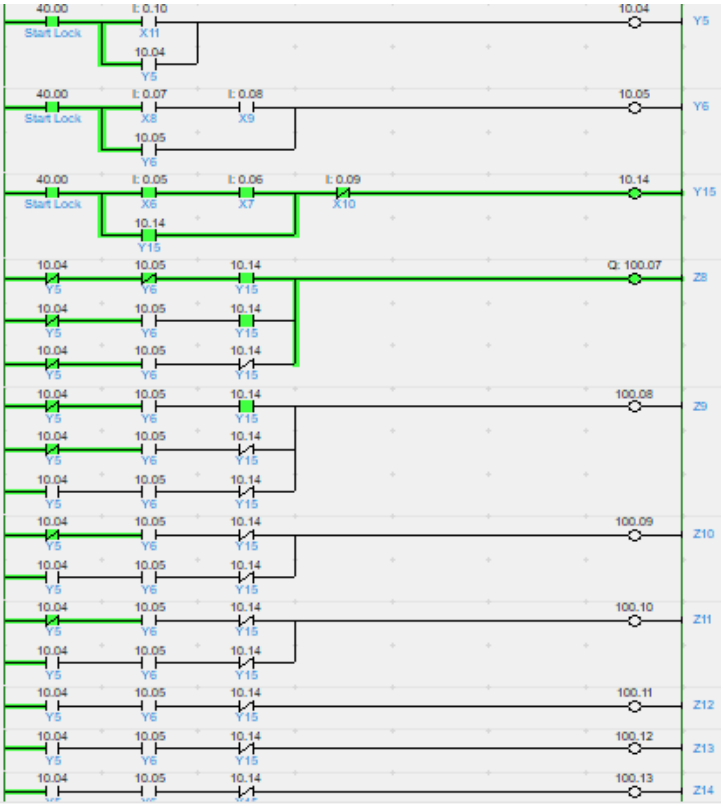

40.00
Start Lock
I: 0.00
X1
I: 0.04
X5
10.02
Y3
10.02
Y3
40.00
Start Lock
I: 0.05
X6
10.03
Y4
10.03
Y4
10.02
Y3
10.03
Y4
Q: 100.04
Z5
10.02
Y3
10.03
Y4
Q: 100.05
Z6
10.02
Y3
10.03
Y4
Q: 100.06
Z7

Sub proses 3

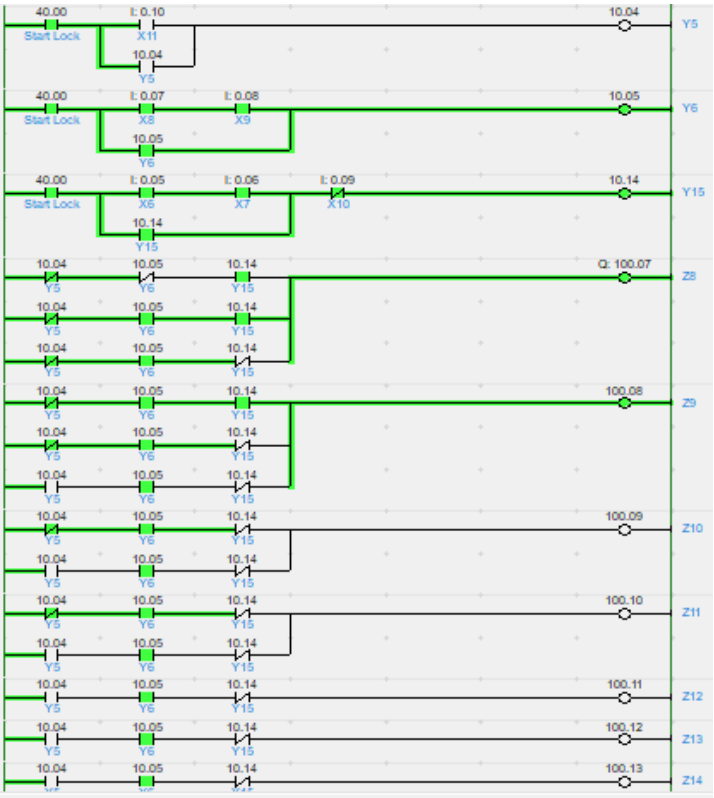
Langkah 1



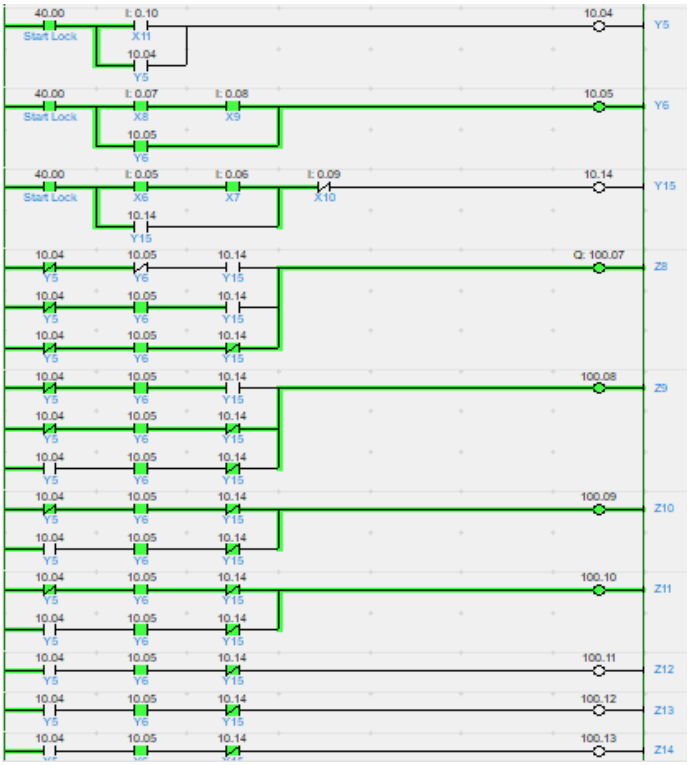
Langkah 2


40.00
Start Lock
I: 0.10
X11
10.04
Y5
10.04
I: 0.07
X8
I: 0.08
X9
10.05
Y6
10.05
I: 0.05
X6
I: 0.06
X7
I: 0.09
X10
10.14
Y15
10.14
Q: 100.07
Z8
100.08
Z9
100.09
Z10
100.10
Z11
100.11
Z12
100.12
Z13
100.13
Z14

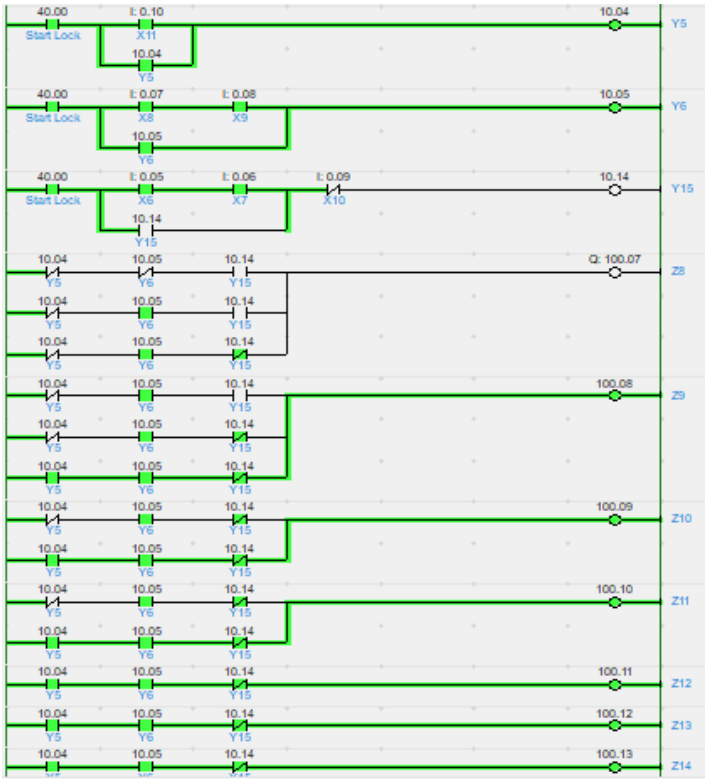
Langkah 3


40.00
Start Lock
I: 0.10
X11
10.04
Y5
10.04
I: 0.07
X8
I: 0.08
X9
10.05
Y6
10.05
I: 0.05
X6
I: 0.06
X7
I: 0.09
X10
10.14
Y15
10.14
Q: 100.07
Z8
100.08
Z9
100.09
Z10
100.10
Z11
100.11
Z12
100.12
Z13
100.13
Z14

Langkah 4

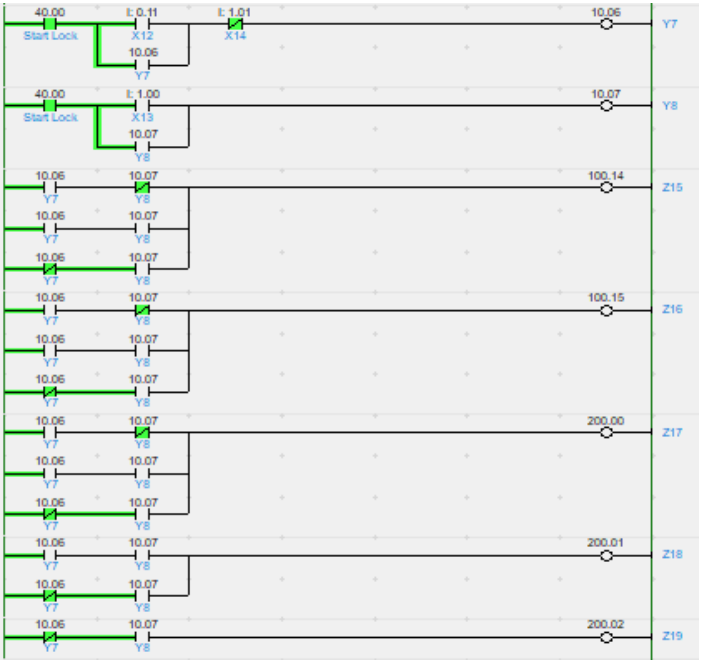

40.00
Start Lock
I: 0.10
X11
10.04
Y5
10.04
Y5
I: 0.07
X8
I: 0.08
X9
10.05
Y6
10.05
Y6
I: 0.05
X6
I: 0.06
X7
I: 0.09
X10
10.14
Y15
10.14
Y15
Q: 100.07
Z8
100.08
Z9
100.09
Z10
100.10
Z11
100.11
Z12
100.12
Z13
100.13
Z14

Langkah 5

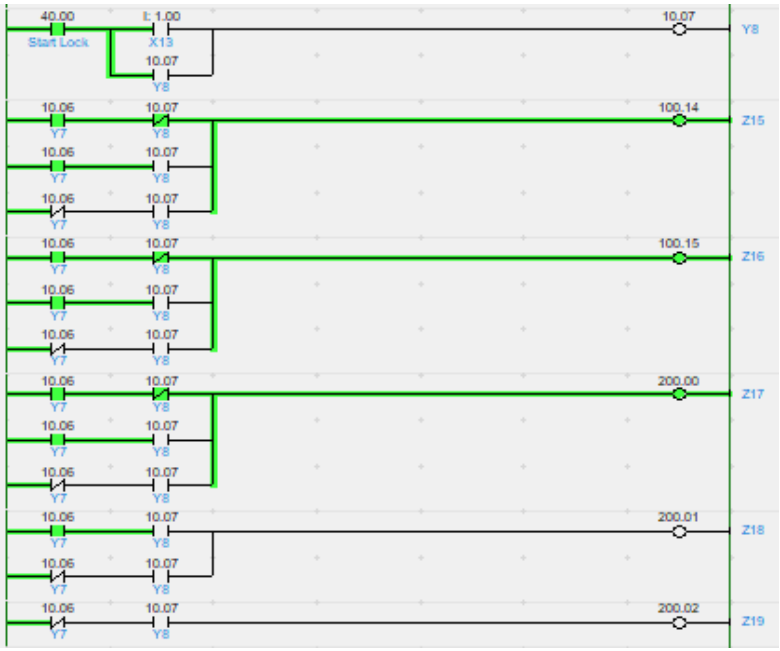

40.00
Start Lock
I: 0.10
X11
10.04
Y5
10.04
Y5
I: 0.07
X8
I: 0.08
X9
10.05
Y6
10.05
Y6
I: 0.05
X6
I: 0.06
X7
I: 0.09
X10
10.14
Y15
10.14
Y15
10.04
Y5
10.05
Y6
10.14
Y15
Q: 100.07
Z8
100.08
Z9
100.09
Z10
100.10
Z11
100.11
Z12
100.12
Z13
100.13
Z14

Sub proses 4

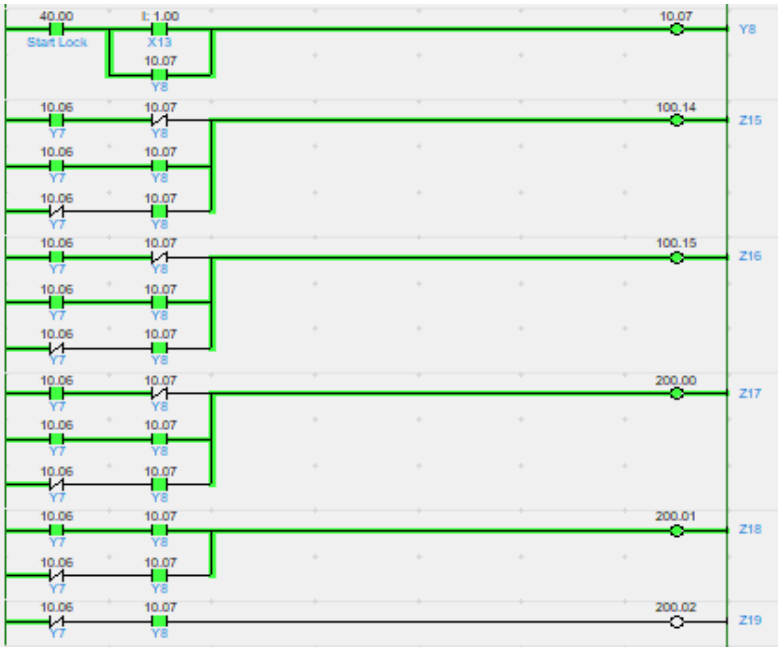
Langkah 1



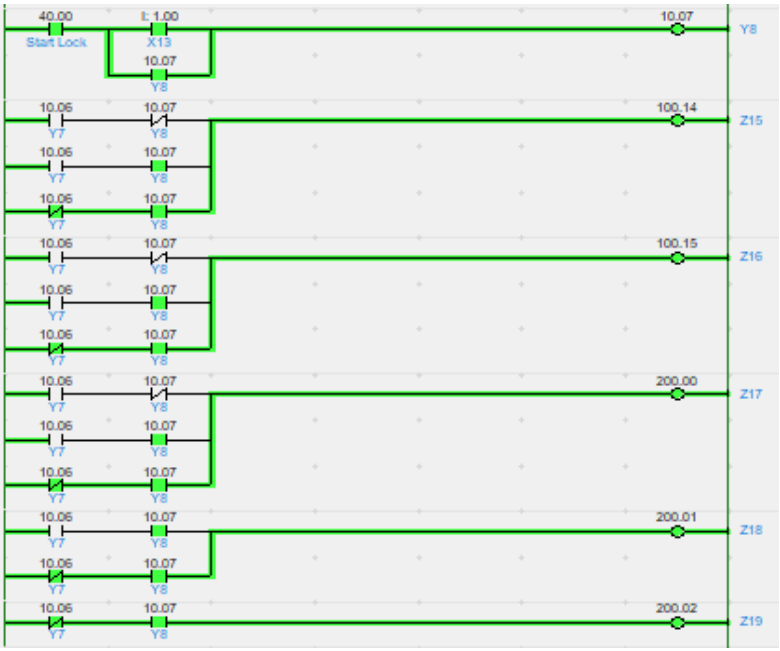
Langkah 2


40.00
Start Lock
I: 1.00
X13
10.07
Y8
10.07
Y8
10.06
Y7
10.07
Y8
100.14
Z15
100.15
Z16
200.00
Z17
200.01
Z18
200.02
Z19

Langkah 3


40.00
Start Lock
I: 1.00
X13
10.07
Y8
10.07
Y8
10.06
Y7
10.07
Y8
100.14
Z15
100.15
Z16
200.00
Z17
200.01
Z18
200.02
Z19

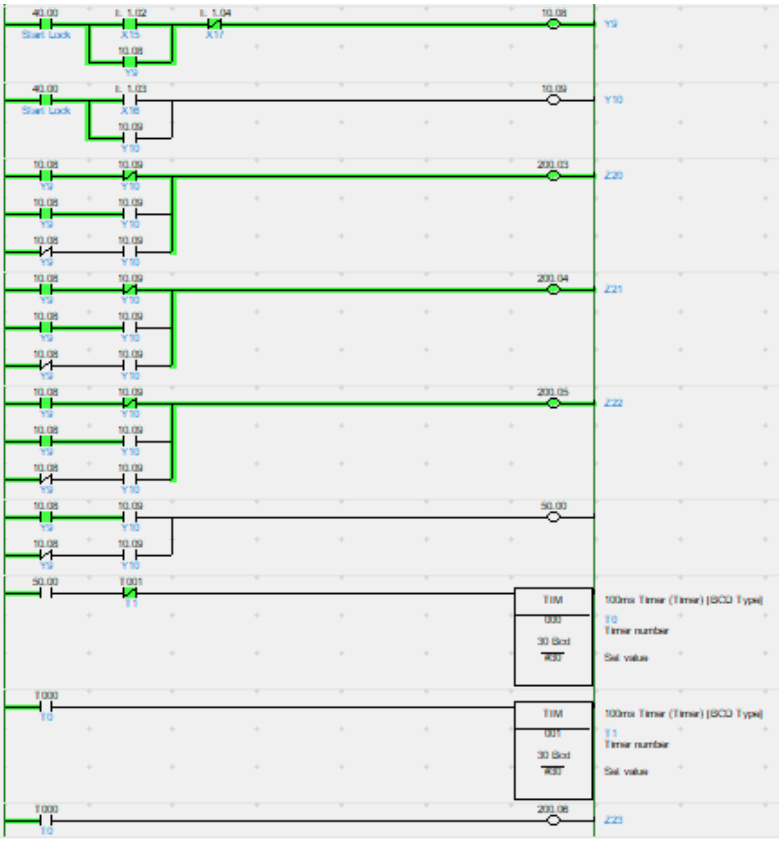
Langkah 4

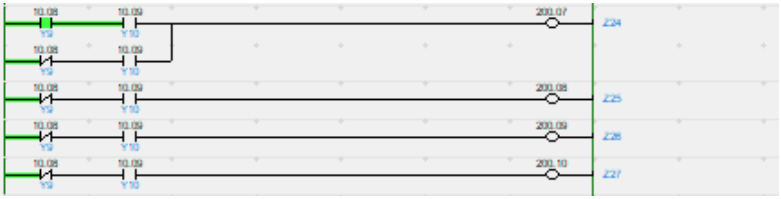

40.00
Start Lock
I: 1.00
X13
10.07
Y8
10.07
Y8
10.06
Y7
10.07
Y8
100.14
Z15
100.15
Z16
200.00
Z17
200.01
Z18
200.02
Z19

Sub proses 5

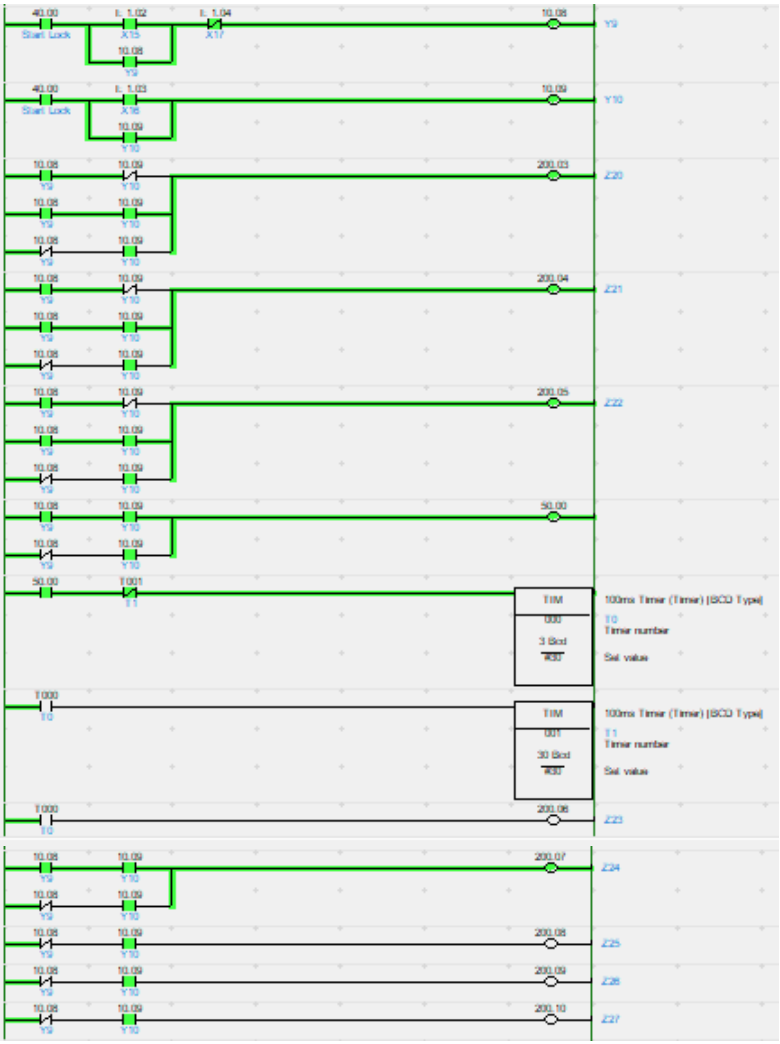
Langkah 1

Langkah 2

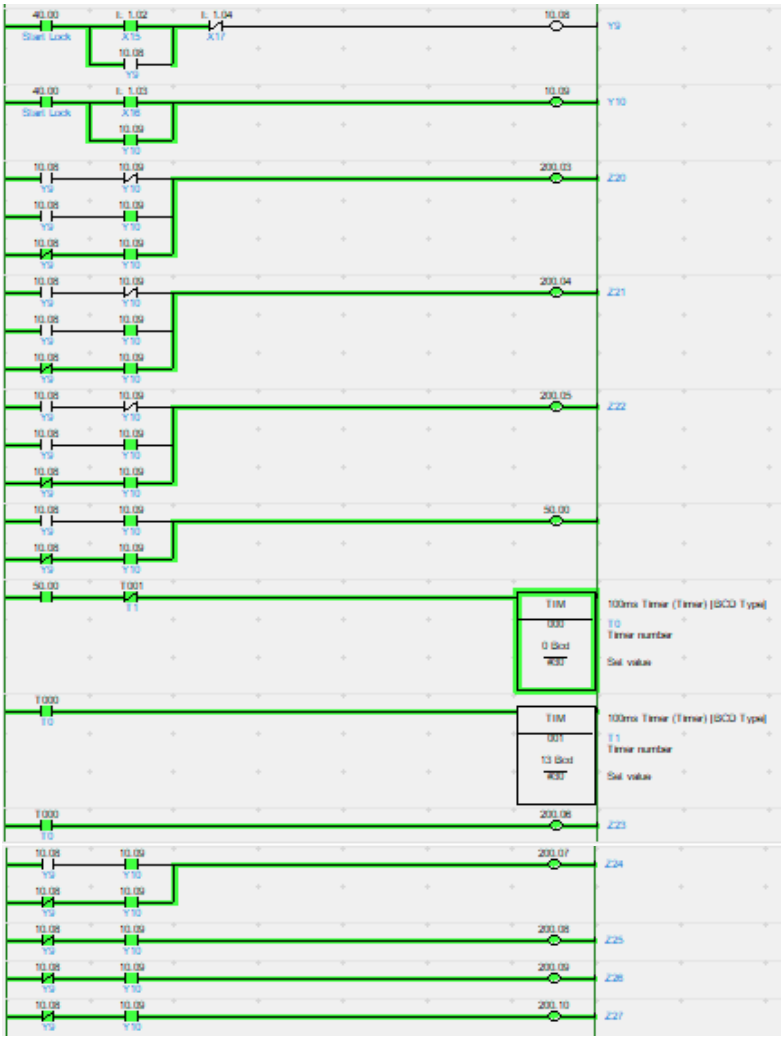

Start Lock
Y9
Y10
Z20
Z21
Z22
TIM
100ms Timer (Timer) (BCD Type)
Timer number
Set value
Z23


Z24
Z25
Z26
Z27

Langkah 3


40.00
Start Lock
I: 1.02
X15
I: 1.04
X17
10.08
Y9
10.09
Y10
I: 1.03
X16
200.03
Z20
200.04
Z21
200.05
Z22
50.00
T001
T
TIM
000
3 Bcd
#30
100ms Timer (Timer) [BCD Type]
T0
Timer number
Sel value
T000
T0
TIM
001
30 Bcd
#30
100ms Timer (Timer) [BCD Type]
T1
Timer number
Sel value
200.06
Z23
200.07
Z24
200.08
Z25
200.09
Z26
200.10
Z27

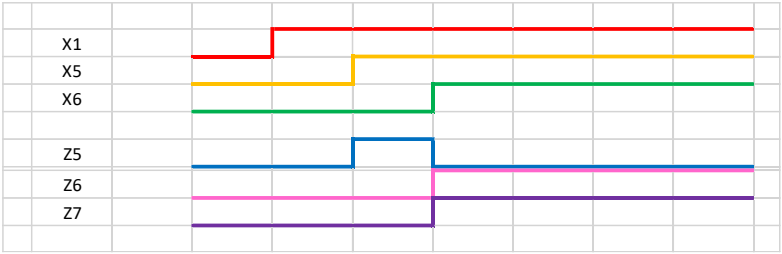
Langkah 4


40.00
Start Lock
I: 1.02
X15
I: 1.04
X17
10.08
Y9
10.09
Y10
200.03
Z20
200.04
Z21
200.05
Z22
50.00
T000
T1
TIM
000
0 Bcd
#30
100ms Timer (Timer) [BCD Type]
T0
Timer number
Set value
TIM
001
13 Bcd
#30
100ms Timer (Timer) [BCD Type]
T1
Timer number
Set value
200.06
Z23
200.07
Z24
200.08
Z25
200.09
Z26
200.10
Z27

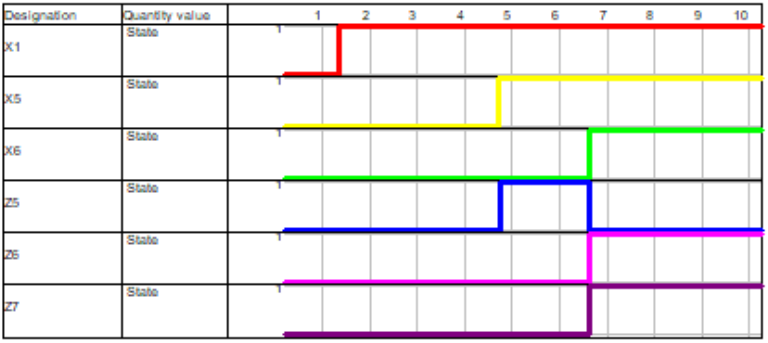
Pengujian Validasi Sistem

Sub proses 2

Perencanaan

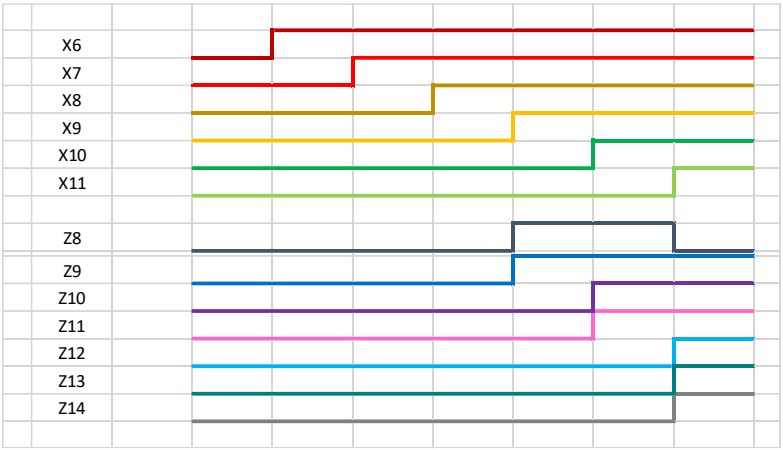


Hasil Ladder diagram

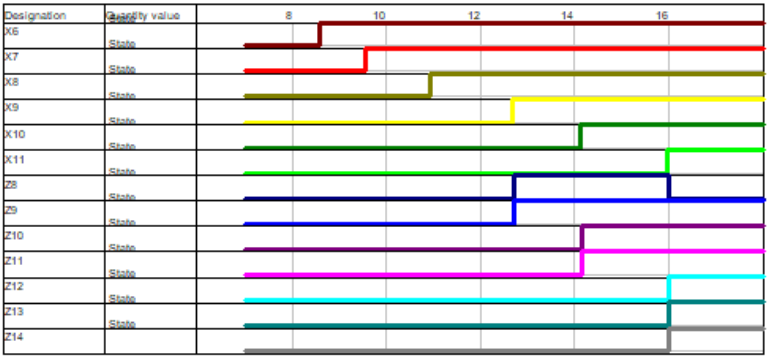


Sub proses 3

Perencanaan

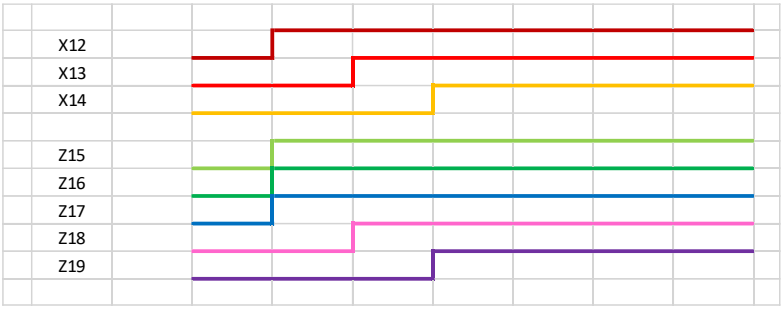


Hasil Ladder Diagram

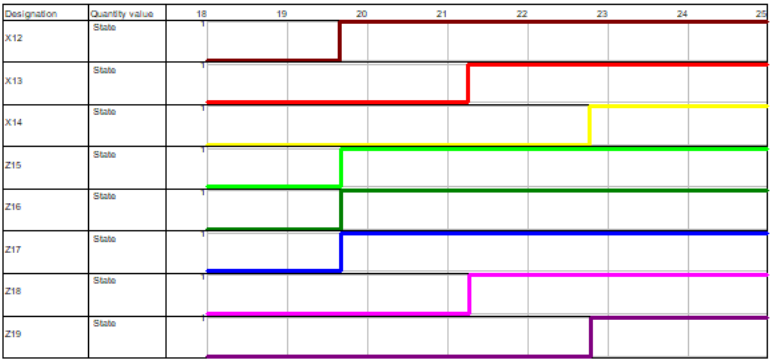


Sub Proses 4

Perencanaan

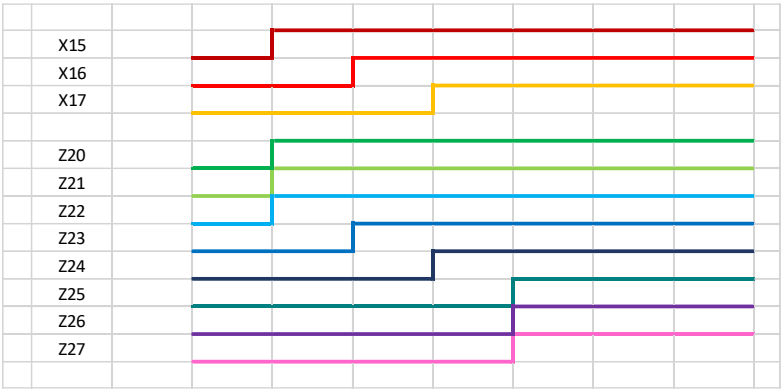


Hasil Ladder Diagram

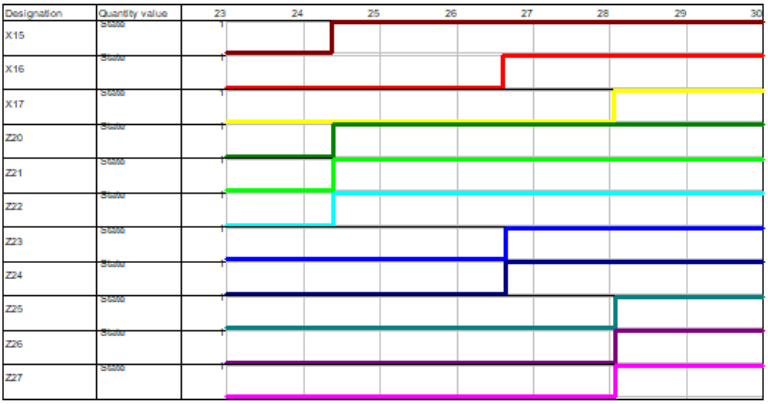


Sub proses 5

Perencanaan

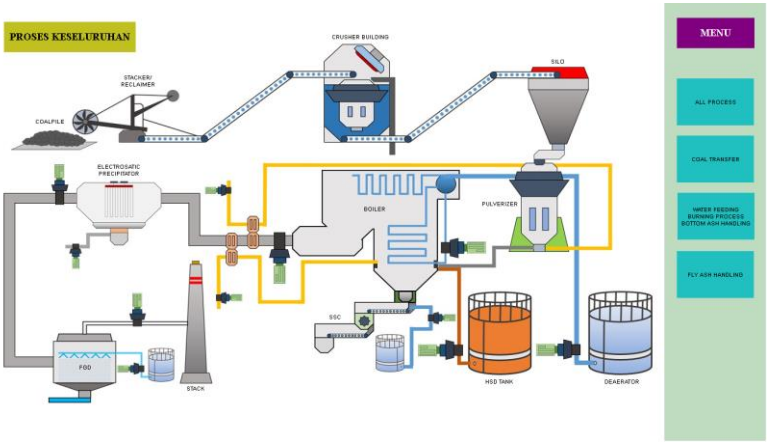


Hasil Ladder Diagram

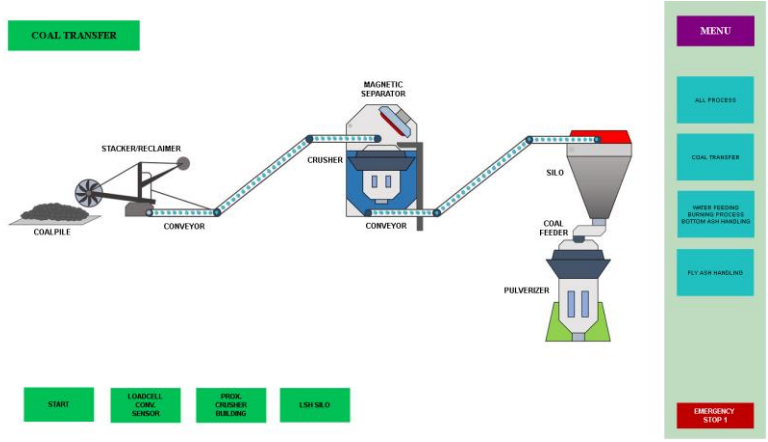


Pengujian HMI

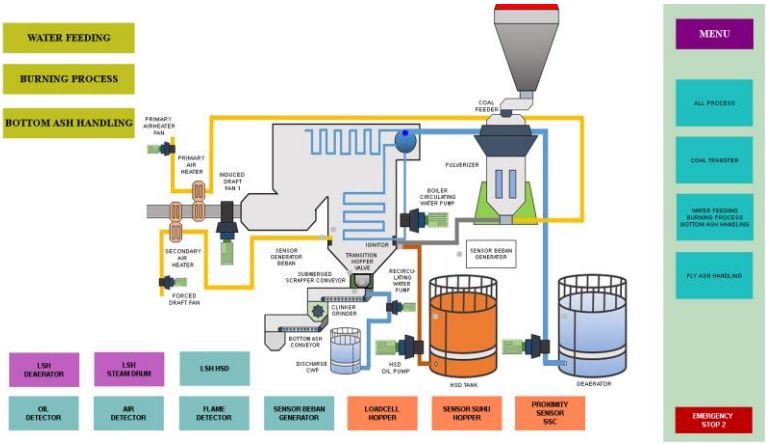
Tampilan menu utama



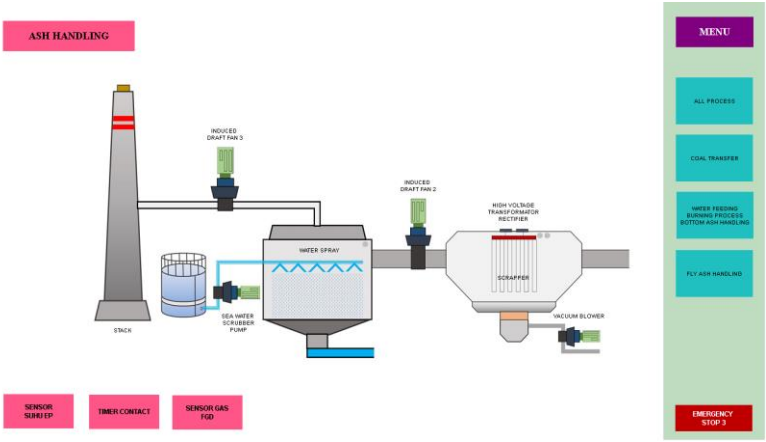
Tampilan menu coal transfer



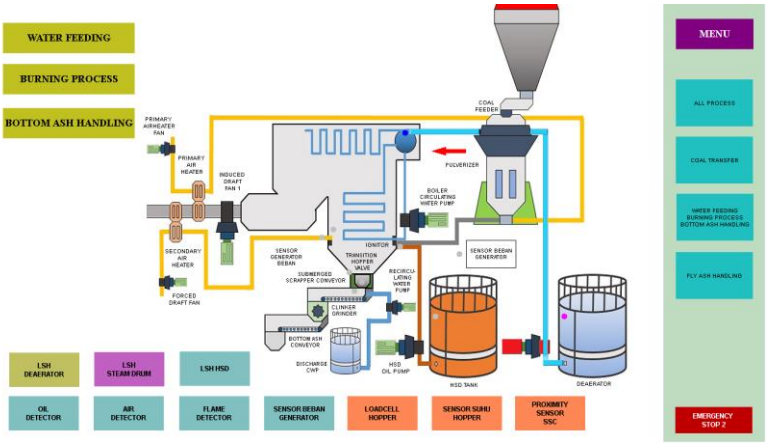
Tampilan menu water feeding, burning process, dan bottom ash handling



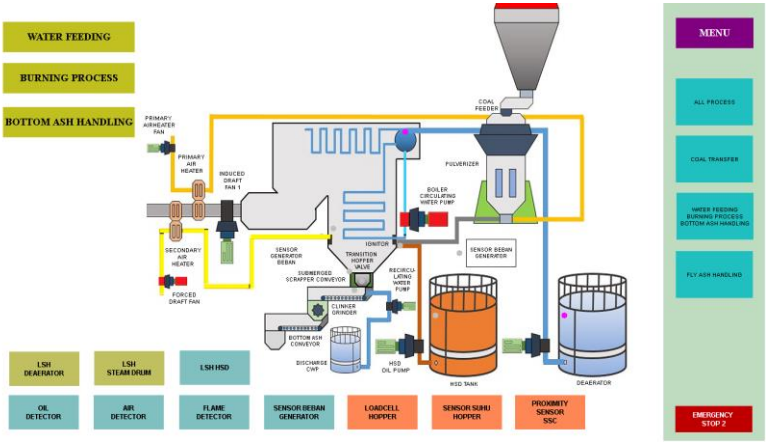
Tampilan menu Fly Ash Handling



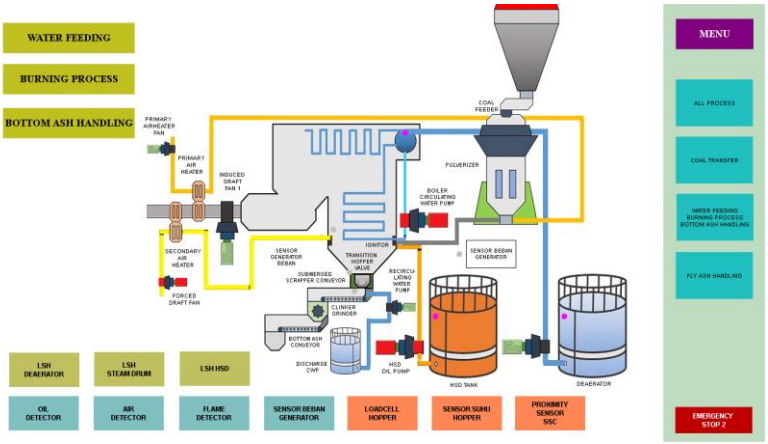
Sub proses 2 Langkah 1



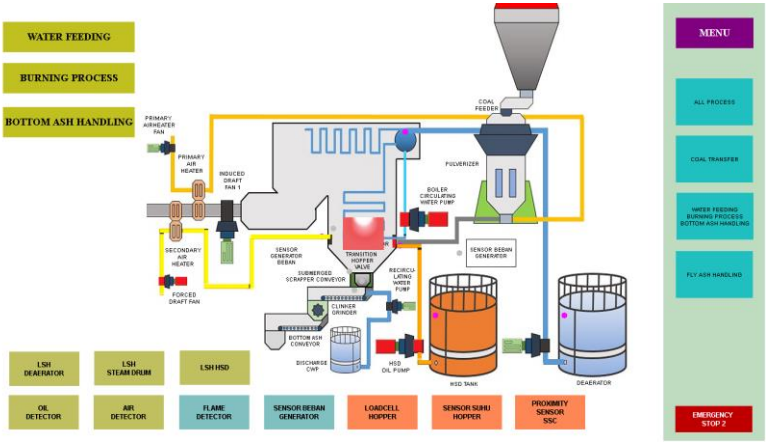
Sub proses 2 Langkah 2



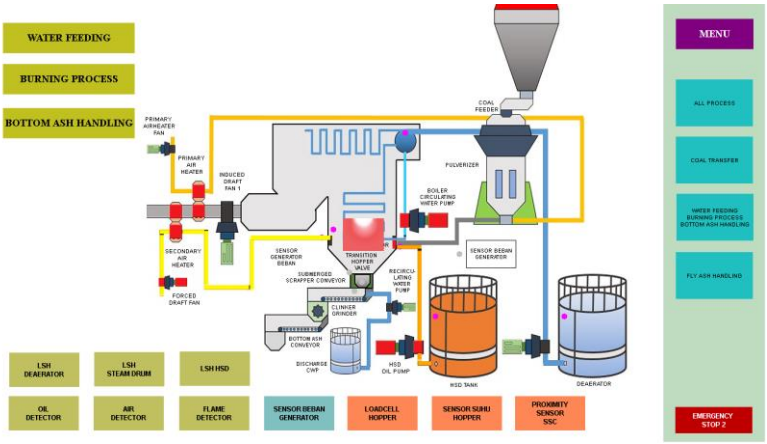
Sub proses 3 Langkah 1



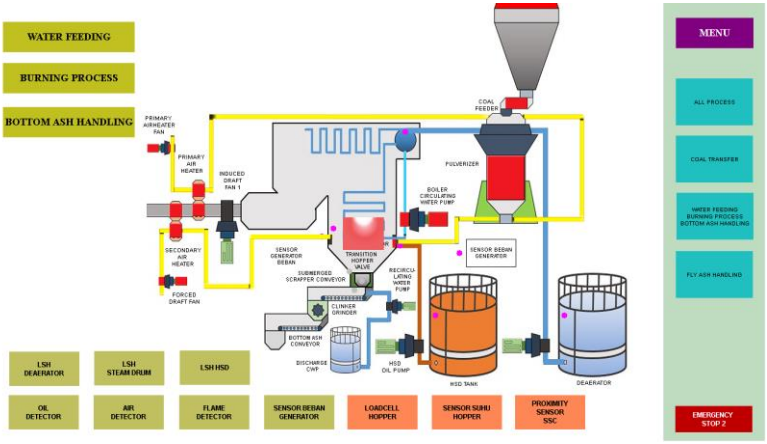
Sub proses 3 Langkah 2 dan 3



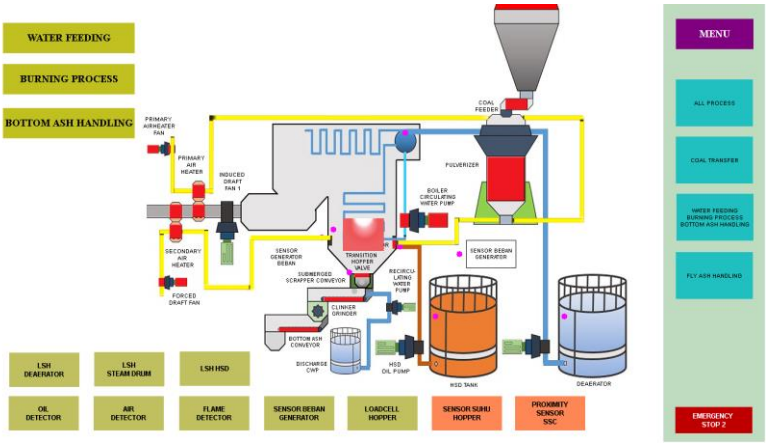
Sub proses 3 Langkah 4



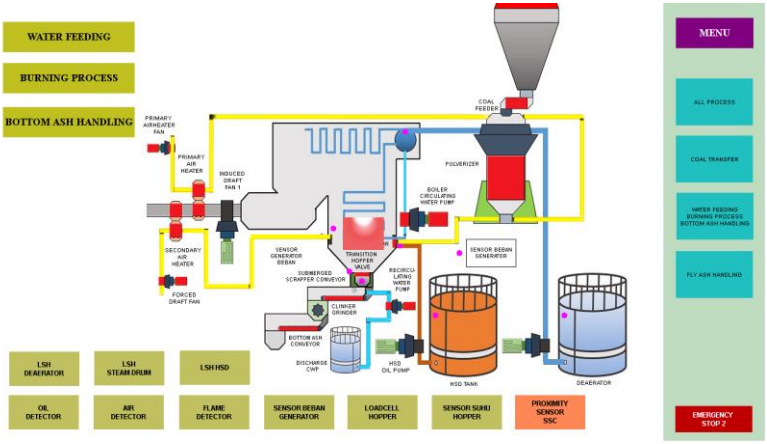
Sub proses 3 Langkah 5



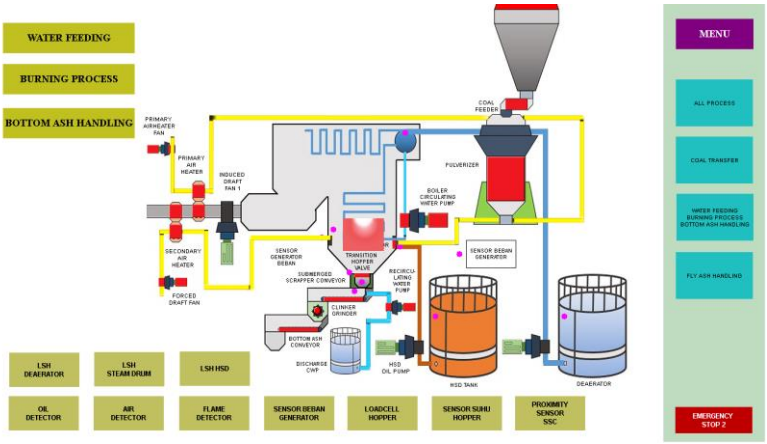
Sub proses 4 Langkah 1



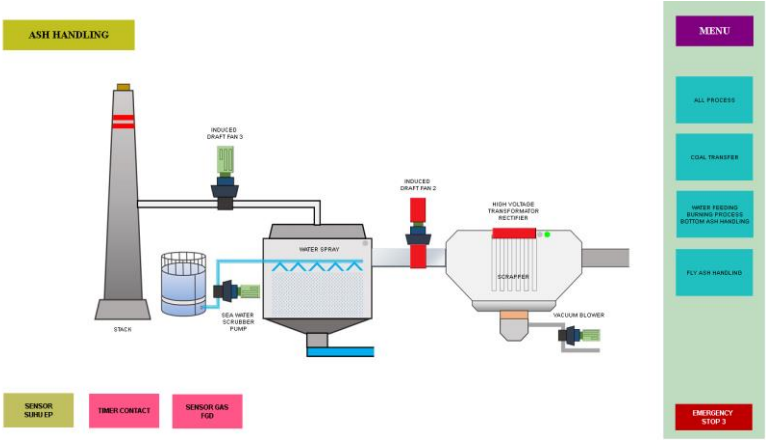
Sub proses 4 Langkah 2



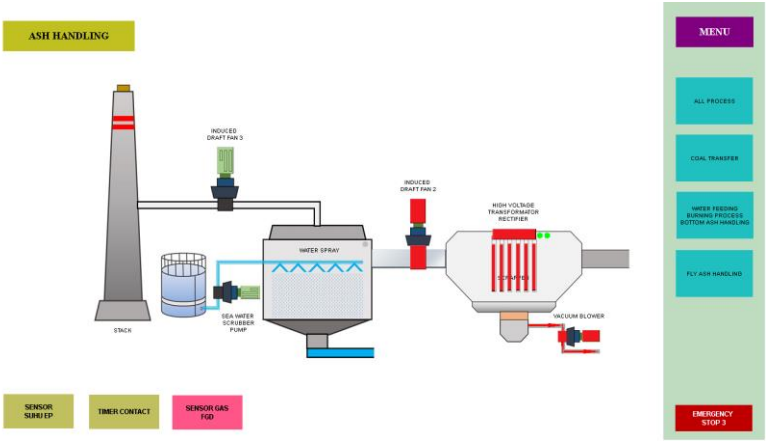
Sub proses 4 Langkah 3



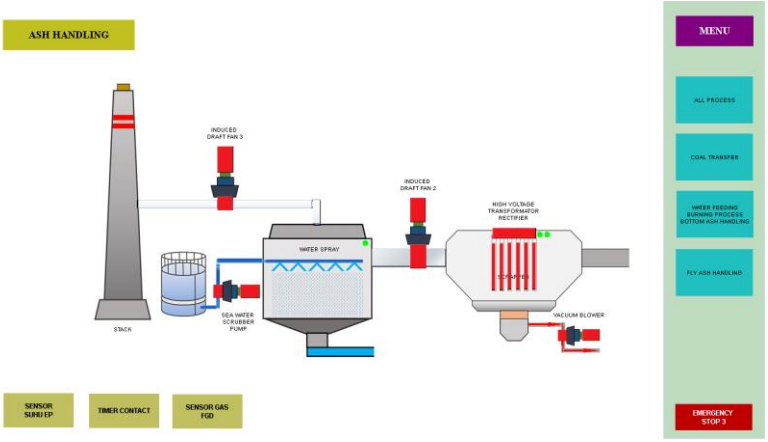
Sub proses 5 Langkah 1



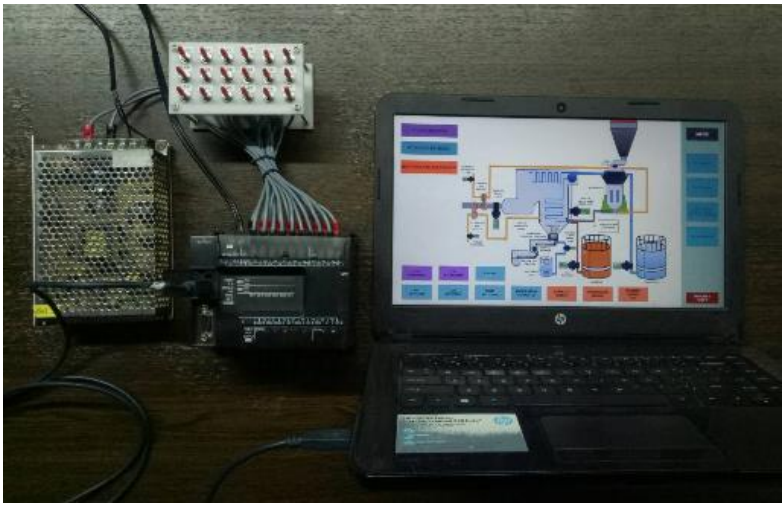
Sub proses 5 Langkah 2



Sub proses 5 Langkah 3



Integrasi Perancangan Sistem



[Halaman ini sengaja dikosongkan]

RIWAYAT HIDUP

Romadhona Ighifliyan Namaskara, lahir di Surabaya pada tanggal 27 Januari 1997. Penulis adalah anak kedua dari tiga bersaudara dari pasangan ayah Wahyudi dan Ibu Endah Kastiah. Setelah menempuh pendidikan di SDN Ketabang V Surabaya, SMPN 1 Surabaya, dan SMAN 1 Surabaya, pada tahun 2015 penulis melanjutkan studinya di S1 Teknik Elektro Institut Teknologi Sepuluh Nopember Surabaya. Selama berkuliah penulis juga aktif dalam kegiatan organisasi mahasiswa BEM FTI dan BEM FTE ITS.