

TUGAS AKHIR - TE 141599

**KONSTRUKSI *LADDER DIAGRAM* MENGGUNAKAN
METODE *STATE DIAGRAM* PADA *FACTORY
AUTOMATIC TRAINER***

Gustafian Fawally Al-Barr
NRP 07111440000058

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

DEPARTEMEN TEKNIK ELEKTRO
Fakultas Teknologi Elektro
Institut Teknologi Sepuluh Nopember
Surabaya 2018



TUGAS AKHIR - TE 141599

**KONSTRUKSI *LADDER DIAGRAM* MENGGUNAKAN
METODE *STATE DIAGRAM* PADA *FACTORY
AUTOMATIC TRAINER***

Gustafian Fawally Al-Barr
NRP 07111440000058

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

DEPARTEMEN TEKNIK ELEKTRO
Fakultas Teknologi Elektro
Institut Teknologi Sepuluh Nopember
Surabaya 2018



Final Project - TE 141599

***CONSTRUCTION OF LADDER DIAGRAM USING STATE
DIAGRAM METHOD ON FACTORY AUTOMATIC
TRAINER***

Gustafian Fawally Al-Barr
NRP 07111440000058

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

*Electrical Engineering Department
Faculty of Electrical Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2018*

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul "**Konstruksi *Ladder Diagram* Menggunakan Metode *State Diagram* Pada *Factory Automatic Trainer***" adalah merupakan hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri. Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, Juli 2018



Gustafian Fawally Al-Barr
NRP 07111440000058

[Halaman ini sengaja dikosongkan]

**KONSTRUKSI LADDER DIAGRAM MENGGUNAKAN
METODE STATE DIAGRAM PADA FACTORY
AUTOMATIC TRAINER**

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

Dr. Ir. Mochammad Rameli
NIP. 19541227 1981031002

Dosen Pembimbing II

Eka Iskandar, ST., MT.
NIP. 19800528 2008121001



[Halaman ini sengaja dikosongkan]

KONSTRUKSI *LADDER DIAGRAM* MENGGUNAKAN METODE *STATE DIAGRAM* PADA *FACTORY AUTOMATIC TRAINER*

Gustafian Fawally Al-Barr –07111440000058

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

ABSTRAK

Problematika sistem di industri yang semakin kompleks mendorong dikembangkan Programable Logic Controller (PLC) dengan spesifikasi yang lebih mumpuni. PLC kini telah didukung dengan beberapa bahasa pemrograman yang relatif lebih mudah dipahami dibandingkan *ladder diagram*. Hal ini berdampak baik bagi seorang designer saat melakukan perancangan program pada sebuah project baru. Selain itu akan memudahkan designer dalam menjelaskan project tersebut kepada pihak-pihak berkepentingan yang masih awam dengan perangkat PLC. Namun kemudahan ini tidak dapat terwujud bagi pelaku industri yang masih belum mampu mengupgrade perangkat PLC di perusahaanya. Oleh karena itu dibutuhkan metode yang mampu untuk mengatasi keterbatasan PLC tersebut Pada penelitian ini penulis akan memodelkan sistem yang ada pada *Factory Automatic Trainer* sebagai representasi sistem yang ada di industri menggunakan *State Diagram*. Setelah itu *State Diagram* yang sudah diperoleh akan dikonversi menjadi *ladder diagram*. Konstruksi *Ladder diagram* terdiri atas 49 rung, menggunakan sebanyak 23 *relay* dan 6 *timer*. I/O yang dipergunakan sebanyak 22 *input* dan 20 *output*. Sementara rata-rata waktu proses pada benda kerja 1 yaitu 25.998 s sedangkan pada benda kerja 2 yaitu 23.668 s.

Kata kunci : *Factory Automatic Trainer, State Diagram, Ladder diagram, PLC.*

[Halaman ini sengaja dikosongkan]

CONSTRUCTION OF LADDER DIAGRAM USING STATE DIAGRAM METHOD ON FACTORY AUTOMATIC TRAINER

Gustafian Fawally Al-Barr –07111440000058

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

ABSTRACT

The increasingly complex problem of system in industry encourages the development of Programmable Logic Controller (PLC) with more specification. PLC has now been supported with several programming languages that are relatively easier to understand than ladder diagrams. This is good for a designer when designing a program on a new project. In addition it will facilitate the designer in explaining the project to the people who still lay with PLC devices. But this ease can't be realized for industry players who still haven't been able to upgrade their PLC. Therefore the required method that is able to overcome the limitations of the PLC In this study the authors will model the existing system in Factory Automatic Trainer as a representation of existing systems in the industry using State Diagram. After that the obtained State Diagram will be converted into ladder diagram. The Construction Ladder diagram consists of 49 rung, using 23 relay and 6 timers. I/O used 22 inputs and 20 outputs. While the average time on the workpiece 1 is 25.998 s while the workpiece 2 is 23.668 s

Keyword : Factory Automatic Trainer, State Diagram, Ladder diagram, PLC.

[Halaman ini sengaja dikosongkan]

KATA PENGANTAR

Puji syukur penulis panjatkan kehadiran Tuhan Yang Maha Esa karena berkat rahmat dan karunia-Nya penulis dapat menyelesaikan tugas akhir yang berjudul “**Konstruksi Ladder Diagram Menggunakan Metode State Diagram Pada Factory Automatic Trainer**” 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 pengarahannya, 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, staf dan karyawan di Teknik Elektro Institut Teknologi Sepuluh Nopember.
5. 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, Juli 2018

Gustafian Fawally Al-Barr
NRP 07111440000058

[Halaman ini sengaja dikosongkan]

DAFTAR ISI

HALAMAN JUDUL.....	i
PERNYATAAN KEASLIAN TUGAS AKHIR	v
HALAMAN PENGESAHAN	vii
ABSTRAK	ix
<i>ABSTRACT</i>	xi
KATA PENGANTAR	xiii
DAFTAR ISI.....	xv
DAFTAR GAMBAR	xviii
DAFTAR TABEL	xxi
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	2
1.3 Tujuan	2
1.4 Batasan Masalah	2
1.4 Metodologi.....	2
1.5 Sistematika.....	3
1.6 Relevansi.....	3
BAB 2 DASAR TEORI	5
2.1 <i>Factory Automatic Trainer</i>	5
2.1.1 Separation Module	6
2.1.2 Pick and Place Module	7
2.1.3 Stopper Module	7
2.1.4 Line Movement Module	8
2.1.5 Control unit	9
2.1.6 Benda kerja.....	10
2.1.7 Photo Electric Sensor	11
2.1.8 <i>Fiber Optic Sensor</i>	12
2.1.9 <i>Proximity Sensor</i>	13
2.1.10 Rotary Cylinder	14
2.2 Programmable Logic Controller (PLC)	15
2.2.1 Sistem pada Programmable Logic Controller	15
2.2.1.1 Central Processing Unit.....	15
2.2.1.2 Power Supply	15
2.2.1.3 <i>Input/Output</i> Interface	16
2.2.1.4 Memory Unit	16

2.2.1.5 Communication Unit	16
2.2.1.6 Programming device.....	16
2.2.2 PLC GLOFA seri GM4	17
2.2.2.1 Spesifikasi PLC LG GLOFA seri GM4	17
2.2.2.2 Cara Pengoperasian PLC LG GLOFA seri GM4	18
2.2.2.3 Pengalamatan pada PLC LG GLOFA seri GM4	18
2.3 <i>Ladder diagram</i>	19
2.3.1 Simbol dasar pada <i>Ladder diagram</i>	20
2.4 <i>State Diagram</i>	22
2.4.1 Pendiskripsian urutan proses	22
2.4.2 Pendiskripsian I/O	23
2.4.3 Pembuatan <i>State Diagram</i> I/O	24
2.4.4 Penyusunan <i>Primitive Flow Table</i>	24
2.4.5 Penyusunan <i>Merged Flow Table</i>	27
2.4.6 Pembuatan <i>State Diagram</i> R/O	29
2.5 Konversi <i>State Diagram</i> ke <i>Ladder diagram</i>	30
BAB 3 PERANCANGAN SISTEM	33
3.1 Perumusan Sistem <i>Factory Automatic Trainer</i>	33
3.1.1 Langkah Kerja Sistem	33
3.1.2 <i>Input/Output</i> Sistem	41
3.2 Perancangan <i>State Diagram</i>	45
3.2.1 Pendeskripsian Urutan Proses	46
3.2.2 Pendiskripsian <i>Input</i> dan <i>Output</i>	48
3.2.3 Pembuatan <i>State Diagram</i> I/O (<i>Input/Output</i>)	49
3.2.4 Pembuatan <i>Primitive Flow Table</i>	50
3.2.5 Pembuatan <i>Merged Flow Table</i>	50
3.2.6 Pembuatan <i>State Diagram</i> R/O (<i>Relay/Output</i>)	51
3.3 Konversi <i>State Diagram</i> R/O ke <i>Ladder diagram</i>	52
BAB 4 PENGUJIAN DAN ANALISA.....	55
4.1 Pengkabelan	55
4.2 Pengujian Sistem.....	56
BAB 5 KESIMPULAN DAN SARAN.....	65
5.1 Kesimpulan	65
5.2 Saran	65
DAFTAR PUSTAKA	67
LAMPIRAN.....	69

RIWAYAT HIDUP	99
---------------------	----

[Halaman ini sengaja dikosongkan]

DAFTAR GAMBAR

Gambar 2. 1	<i>Factory Automatic Trainer</i> [1].....	5
Gambar 2. 2	Separation module [1].....	6
Gambar 2. 3	pick and place module [1].....	7
Gambar 2. 4	Stopper module [1]	8
Gambar 2. 5	line movement module [1]	9
Gambar 2. 6	<i>Control unit</i> pada <i>Factory Automatic Trainer</i> [1].....	9
Gambar 2. 7	benda kerja pada <i>Factory Automatic Trainer</i>	10
Gambar 2. 8	(a) Retroreflective photoelectric sensor (b) through beam photoelectric sensor [2].....	12
Gambar 2. 9	Optical fiber sensor tipe diffuse reflective [2]	12
Gambar 2. 10	bentuk secara umum proximity sensor [2]	13
Gambar 2. 11	Prinsip kerja capacitive proximity sensor [2].....	14
Gambar 2. 12	Rotary Cylinder [6]	14
Gambar 2. 13	Diagram sistem pada Programmable Logic Controller [3]	16
Gambar 2. 14	PLC LG GLOFA seri GM4	17
Gambar 2. 15	Aturan pengalamatan pada PLC LG	19
Gambar 2. 16	Proses scanning pada ladder diagram [3]	20
Gambar 2. 17	<i>State Diagram</i>	22
Gambar 2. 18	Langkah-langkah pada Metode <i>State Diagram</i>	22
Gambar 2. 19	<i>State Diagram</i> I/O.....	24
Gambar 2. 20	<i>State Diagram</i> (R/O).....	30
Gambar 2. 21	Contoh <i>State Diagram</i> yang akan dikonversi ke <i>ladder diagram</i>	32
Gambar 2. 22	Hasil konversi <i>State Diagram</i> ke <i>Ladder diagram</i>	32
Gambar 3. 1	Perumusan sistem <i>Factory Automatic Trainer</i> secara umum	33
Gambar 3. 2	komponen yang berperan pada subproses 1 (memasukkan benda) [1]	35
Gambar 3. 3	komponen yang berperan pada subproses 2, 3,dan 4 [1]..	36
Gambar 3. 4	komponen yang berperan dalam subproses 5 [1]	38
Gambar 3. 5	komponen yang berperan dalam subproses 6 [1]	39
Gambar 3. 6	komponen yang berperan dalam subproses 7,8,9,10 [1] ..	41
Gambar 3. 7	langkah perancangan state diagram	45
Gambar 3. 8	<i>State Diagram</i> I/O sub proses 1	50
Gambar 3. 9	<i>State Diagram</i> R/O untuk subproses 1.....	51
Gambar 3. 10	konversi <i>State Diagram</i> ke <i>ladder diagram</i> suproses 1 .	52

Gambar 3. 11 Konversi <i>State Diagram</i> ke <i>ladder diagram</i> subproses 1	53
Gambar 4. 1 wiring pada <i>Factory Automatic Trainer</i>	55
Gambar 4. 2 cylinder insert melakukan gerak eject.....	58
Gambar 4. 3 implementasi <i>ladder diagram</i> untuk insert bergerak eject.....	58
Gambar 4. 4 lengan robot bergerak UP	59
Gambar 4. 5 Implementasi <i>Ladder diagram</i> untuk lengan robot bergerak UP	59
Gambar 4. 6 conveyor 1 aktif	60
Gambar 4. 7 Implementasi <i>ladder diagram</i> untuk conveyor 1 aktif.....	60
Gambar 4. 8 Grafik Waktu proses benda kerja 1 dan benda kerja 2.....	63

DAFTAR TABEL

Tabel 2. 1 Klasifikasi benda kerja berdasarkan warna dan bahan	11
Tabel 2. 2 Klasifikasi benda kerja berdasarkan deteksi sensor	11
Tabel 2. 3 Spesifikasi PLC LG GLOFA seri GM4 [4]	17
Tabel 2. 4 Simbol dasar pada ladder diagram [3]	21
Tabel 2. 5 Pendiskripsian Urutan Proses	23
Tabel 2. 6 Tabel Deskripsi <i>Input</i>	23
Tabel 2. 7 Tabel Deskripsi <i>Output</i>	23
Tabel 2. 8 <i>Primitive Flow Table</i>	25
Tabel 2. 9 kolom bit <i>input</i> serta row pada <i>primitive flow Table</i>	25
Tabel 2. 10 state stabil pada <i>primitive flow Table</i>	26
Tabel 2. 11 state tidak stabil pada <i>primitive flow Table</i>	26
Tabel 2. 12 kondisi dont care pada <i>primitive flow Table</i>	27
Tabel 2. 13 penulisan kombinasi bit <i>output</i> pada <i>primitive flow Table</i> .	27
Tabel 2. 14 Penggabungan baris pada <i>Merged Flow Table</i>	28
Tabel 2. 15 <i>Merged Flow Table</i> setelah penggabungan baris.....	28
Tabel 2. 16 <i>Merged flow Table</i> yang sudah dilengkapi kolom <i>relay</i>	29
Tabel 3. 1 Tabel data <i>Input</i>	41
Tabel 3. 2 Tabel data <i>Output</i>	43
Tabel 3. 3 Tabel data <i>timer</i> yang dipergunakan pada sistem	44
Tabel 3. 4 Tabel Deskripsi Urutan Proses	46
Tabel 3. 5 Pendiskripsian <i>Input</i> dan <i>Output</i>	48
Tabel 3. 6 Tabel deskripsi urutan subproses 1 (memasukkan benda kerja)	49
Tabel 3. 7 <i>Primitive Flow Table</i> Subproses 1	50
Tabel 3. 8 <i>Merged Flow Table</i> Subproses 1	51
Tabel 4. 1 Pengalamatan <i>input</i> dan <i>output</i>	56
Tabel 4. 2 Urutan Subproses 1	57
Tabel 4. 3 Jumlah <i>relay</i> yang dibutuhkan tiap subproses	61
Tabel 4. 4 Waktu proses	62

[Halaman ini sengaja dikosongkan]

BAB 1

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

Efisiensi serta peningkatan produktivitas merupakan faktor utama kesuksesan industri manufaktur yang semakin kompetitif di era modern. Hal tersebut ditanggapi para pelaku industri dengan cara melakukan peningkatan kemampuan teknologi yang dimiliki. Terdapat dua cara untuk meningkatkan kemampuan teknologi yaitu mengganti teknologi lama dengan teknologi baru yang lebih canggih atau mengoptimalkan teknologi yang sudah ada.

Salah satu teknologi yang sudah lazim digunakan dalam industri manufaktur adalah *Programmable Logic Controller* (PLC). Saat ini PLC jauh lebih berkembang dari sebelumnya demi menjawab problematika sistem di industri yang semakin kompleks. PLC sudah tersedia dalam bahasa pemrograman seperti *sequential function chart* serta *Function Block Diagram* yang relatif lebih mudah dipahami dibandingkan *Ladder diagram*. Hal ini tentunya akan mempermudah designer pada saat perancangan *project* baru serta pada saat menjelaskan *project* tersebut kepada pihak-pihak yang berkepentingan. Namun hal ini tidak bisa diwujudkan apabila kenyataannya perusahaan belum mampu memfasilitasi PLC dengan spesifikasi tersebut. Oleh karena itu dibutuhkan solusi lain yang tetap mampu mengakomodasi kebutuhan designer dengan keterbatasan PLC yang hanya mendukung bahasa pemrograman *ladder diagram*.

Berdasarkan permasalahan tersebut maka penulis mengajukan solusi berupa perancangan *ladder diagram* dengan menggunakan metode *State Diagram*. Sistem yang ada di industri akan dimodelkan terlebih dahulu menggunakan *State Diagram* sebelum dikonversi ke dalam bentuk *ladder diagram*. Pada tugas akhir ini, penulis akan memodelkan sistem pada *plant Factory Automatic Trainer* yang merupakan representasi sistem yang ada di industri sesungguhnya. Serta hasil perancangan *ladder diagram*nya akan diimplementasikan ke *plant* yang tersebut.

1.2 Perumusan Masalah

Dari latar belakang, didapatkan rumusan masalah sebagai berikut:

1. Bagaimana menjalankan proses otomatisasi pada *Factory Automatic Trainer* sesuai dengan urutan proses yang diinginkan
2. Bagaimana merancang *ladder diagram* menggunakan metode *State Diagram* pada *Factory Automatic Trainer*

1.3 Tujuan

Dari perumusan masalah, didapatkan tujuan penulisan sebagai berikut:

1. Mendapatkan proses otomatisasi sesuai urutan proses yang diinginkan
2. Mendapatkan perancangan *ladder diagram* menggunakan metode *State Diagram* pada *Factory Automatic Trainer*

1.4 Batasan Masalah

Berikut merupakan batasan masalah dari tugas akhir ini :

1. Metode yang dipergunakan untuk memodelkan sistem adalah *State Diagram*
2. Plant yang digunakan merupakan *Factory Automatic Trainer* tipe CPE8030N
3. Benda kerja diproses satu persatu

1.4 Metodologi

Kegiatan penelitian pada tugas akhir ini mencakup studi literatur yang dilakukan dengan membaca beberapa referensi mengenai topik yang akan diangkat sampai diperoleh gambaran yang jelas mengenai sistem yang akan dirancang. Pada penelitian yang dilakukan diketahui bahwa permasalahan pada sistem dapat diselesaikan dengan menggunakan metode *State Diagram*.

Dari permasalahan yang telah diketahui selanjutnya dipelajari mengenai metode yang dapat mengatasi permasalahan tersebut dan penggunaannya pada sistem. Metode *State Diagram* dapat mengatasi banyaknya penggunaan *relay* pada sistem. Dimulai dengan membuat *State Diagram* (I/O) dari sistem *Factory Automatic Trainer* sesuai dengan kondisi sistem yang telah dideskripsikan. Informasi pada diagram tersebut disusun ke dalam *primitive flow Table* dan disederhanakan menjadi *merged flow Table* untuk mengetahui jumlah *relay* yang dibutuhkan sistem.

Merged flow Table yang sudah didapatkan diubah menjadi gambar *State Diagram* (R/O). *State* yang didapatkan menggambarkan

jumlah *relay* yang digunakan. Setelah itu konstruksi *ladder diagram* sistem dapat dilakukan sesuai dengan informasi *State Diagram* (R/O). Program yang telah dibuat kemudian disimulasikan terlebih dahulu dengan device programming bawaan dari PLC yaitu *GMWIN* sebelum diimplementasikan ke *plant Factory Automatic Trainer*.

1.5 Sistematika

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

Bab I Pendahuluan

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

Bab II Dasar Teori

Bab ini membahas tinjauan pustaka yang membantu penelitian, di antaranya adalah teori metode *State Diagram*, teori tentang *plant Factory Automatic Trainer*, teori instrumentasi sistem tentang sensor dan aktuator yang digunakan serta teori otomasi sistem tentang pemrograman *ladder diagram* pada PLC .

Bab III Perancangan Sistem

Bab ini membahas perancangan sistem yang meliputi perancangan langkah kerja pada *Factory Automatic Trainer*, perancangan sistem *plant Factory Automatic Trainer* dengan metode *State Diagram*, serta hasil konversi state diagram ke *ladder diagram*.

Bab IV Hasil dan Analisa

Bab ini memuat hasil penerapan pemodelan yang telah dibuat dan analisisnya.

Bab V Penutup

Bab ini berisi kesimpulan dan saran dari hasil pembahasan yang telah diperoleh.

1.6 Relevansi

Perancangan *ladder diagram* yang terstruktur dan sederhana sangat dibutuhkan apabila dipergunakan untuk permasalahan otomasi yang lebih kompleks. Oleh karena itu dibutuhkan metode *State Diagram* agar mampu mengatasi hal tersebut. Dengan menggunakan metode tersebut jumlah *relay* dapat diminimalkan sehingga memudahkan operator dalam mengidentifikasi adanya kesalahan pada sistem. Serta hasil penelitian yang dilakukan oleh penulis diharapkan

mampu menjadi referensi baik untuk pelaku industri juga mahasiswa dalam hal studi yang mempelajari tentang penyederhanaan jumlah *relay* pada suatu plant.

BAB 2

DASAR TEORI

Pada bab ini akan dijelaskan teori-teori yang berkaitan dengan plant *Factory Automatic Trainer*, Metode *State Diagram*, serta *Programmabke Logic Controller*.

2.1 *Factory Automatic Trainer* [1]

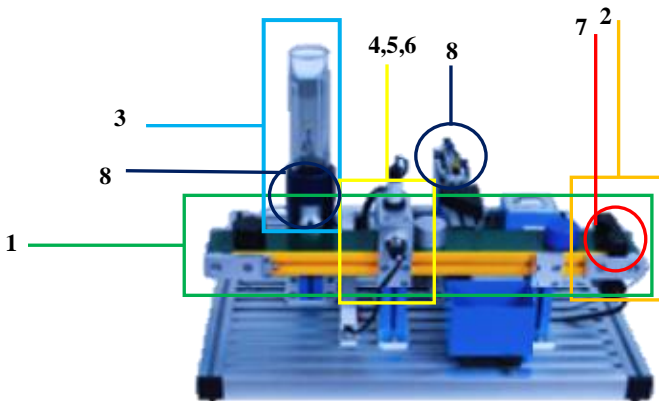
Factory Automatic Trainer atau FAT merupakan serangkaian perangkat yang berfungsi sebagai media pembelajaran tentang sistem otomasi pada industri. Perangkat ini mengadaptasi proses otomasi yang terdapat di industri seperti seleksi benda, pemindahan benda, serta proses drilling. Perangkat ini terdiri atas empat modul pembelajaran yaitu *separation module*, *pick and place module*, *stopper module*, dan *line movement module*. Keempat modul ini akan bekerja sesuai program yang telah dibuat dan disematkan pada PLC LG GLOFA seri GM4. *Factory Automatic Trainer* dapat dioperasikan dengan syarat sumber listrik bertegangan 85V-264V serta frekuensi pada 50Hz/60 Hz. Sementara sumber energi untuk menggerakkan aktuator pada tiap modul berasal dari *air compressor*. Bentuk dari *Factory Automatic Trainer* dapat dilihat pada Gambar 2. 1.



Gambar 2. 1 *Factory Automatic Trainer* [1]

2.1.1 Separation Module

Separation Module merupakan modul yang mempunyai 3 fungsi yaitu memasukkan benda kerja ke dalam sistem, melakukan transfer benda kerja, dan melakukan seleksi benda sesuai standardisasi yang diinginkan. Benda kerja yang akan diproses disimpan sementara pada work supply magazine. Ketika proses sudah dimulai benda kerja akan dideteksi oleh fiber sensor yang ada di work supply magazine. Terdeteksinya benda kerja menyebabkan double acting cylinder eject dan masuk ke conveyor. Benda kerja akan disalurkan menggunakan conveyor. Capacitive sensor, proximity sensor, photo sensor akan mengklasifikasi benda kerja. Benda kerja yang tidak diinginkan untuk diproses dapat diseleksi dengan double acting cylinder. Separation module dapat dilihat pada Gambar 2. 2.



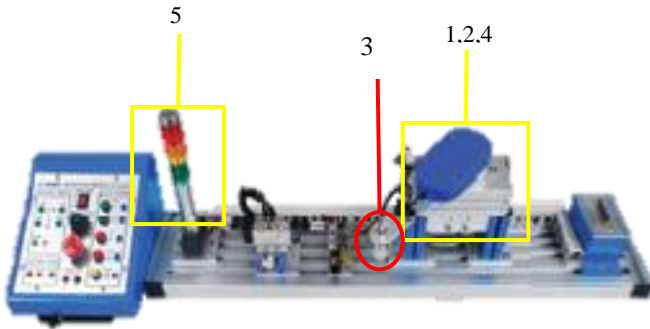
Gambar 2. 2 Separation module [1]

Separation Module terdiri atas komponen-komponen berikut:

1. Conveyor Belt
2. Motor Speed Control
3. Work Supply Magazine
4. Proximity Sensor
5. Capacitive Sensor
6. Photo Sensor
7. Photo Fiber Sensor
8. Double Acting Cylinder

2.1.2 Pick and Place Module

Pick and Place Module merupakan modul yang mempunyai fungsi utama memindahkan benda dari satu conveyor ke conveyor yang lain[5]. Benda kerja yang sudah dideteksi fiber sensor pada Separation Module akan dihisap menggunakan vacuum pad oleh lengan robot dan akan dipindahkan ke conveyor yang ada pada Stopper Module. Pick and place module dapat dilihat pada Gambar 2. 3.



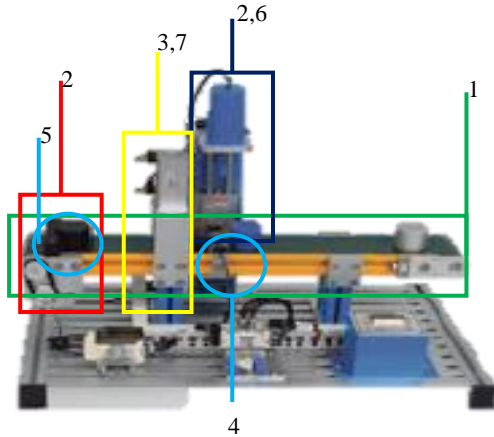
Gambar 2. 3 pick and place module [1]

Pick and Place Module terdiri atas komponen-komponen berikut:

1. Rotary Cylinder
2. Horizontal dan Vertical Cylinder
3. Vacuum Pad
4. Speed control
5. Indicator lamp

2.1.3 Stopper Module

Stopper Module merupakan modul yang mempunyai fungsi menyalurkan benda, proses drilling, menghentikan serta memposisikan benda untuk mempermudah proses drilling. Benda kerja akan disalurkan menggunakan conveyor hingga dideteksi optical sensor dan dihentikan oleh stopper. Benda yang sudah dideteksi oleh optical sensor akan masuk proses drilling. Setelah proses drilling selesai benda kerja kembali akan disalurkan oleh conveyor Gambar 2. 4 menunjukkan bentuk daristopper module.



Gambar 2. 4 Stopper module [1]

Stopper Module terdiri atas komponen-komponen berikut:

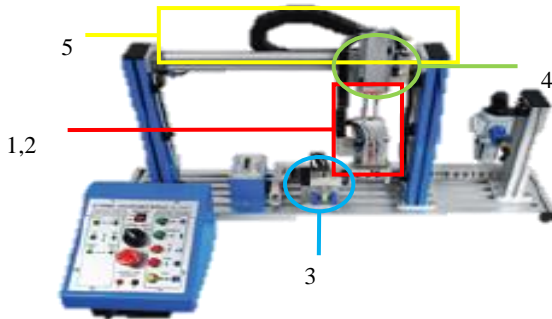
1. Conveyor belt
2. Drilling Machine
3. Stopper
4. Optical sensor
5. Photo Fiber sensor
6. Double Acting Cylinder
7. Single Acting Cylinder

2.1.4 Line Movement Module

Line movement module merupakan modul yang berfungsi memindahkan benda kerja dari satu tempat ke tempat lain[5]. Benda kerja yang sudah dideteksi oleh fiber sensor di Stopper module akan dipindahkan menggunakan finger grip. Line movement Module dapat dilihat pada Gambar 2. 5

Line Movement Module terdiri atas komponen-komponen berikut ini:

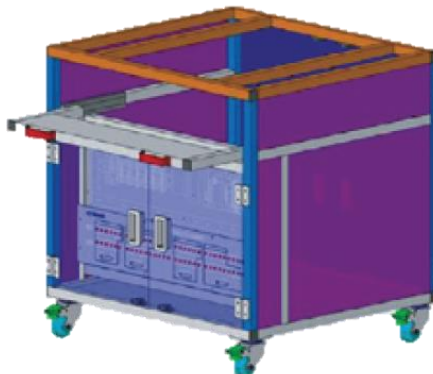
1. horizontal cylinder
2. vertical cylinder
3. Finger grip
4. speed control
5. Robo chain



Gambar 2. 5 line movement module [1]

2.1.5 Control unit

Untuk menjalankan proses pada tiap-tiap modul, *Factory Automatic Trainer* dilengkapi dengan control unit. Terdapat dua jenis control unit pada FAT yaitu control panel dan PLC. Control panel berfungsi sebagai controller untuk mode manual sedangkan PLC berfungsi sebagai controller untuk mode otomatis. Control unit pada *Factory Automatic Trainer* dapat dilihat pada Gambar 2. 6 .



Gambar 2. 6 Control unit pada *Factory Automatic Trainer* [1]

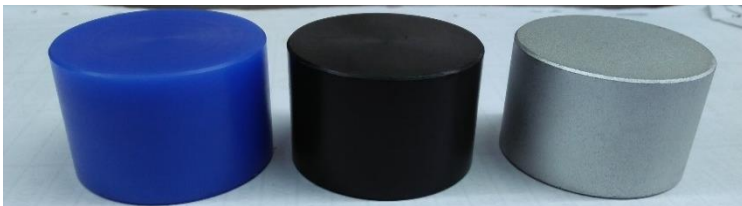
Control panel merupakan controller untuk mode manual. Control panel terdiri atas tombol-tombol yang berfungsi menggerakkan aktuator-aktuator yang terdapat pada tiap modul. Untuk melakukan

proses pada tiap modul, operator perlu menekan *power supply*, mengatur *switch mode* berada pada mode manual, dan menekan tombol aktuator sesuai kebutuhan. Kelebihan dari control unit mode manual adalah operator dapat mengamati proses secara real-time dan bebas mengatur aktuator mana yang ingin digerakkan sesuai kebutuhan. Sementara kekurangannya adalah operator harus terus mengoperasikan control panel hingga proses selesai sehingga memerlukan tenaga operator yang cukup banyak.

PLC merupakan controller untuk mode otomatis. PLC yang digunakan pada FAT adalah PLC LG GLOFA seri GM4. Kelebihan menggunakan *controller* berupa PLC adalah operator tidak perlu mengoperasikan FAT secara terus menerus hingga proses berakhir karena peran tersebut sudah dijalankan oleh program. Sedangkan kekurangannya adalah operator harus membuat program terlebih dahulu di *programming device*.

2.1.6 Benda kerja

Benda kerja merupakan obyek yang menerima perlakuan dari modul-modul yang tersedia pada *Factory Automatic Trainer*. Benda kerja terdiri atas 3 jenis yaitu benda kerja 1, benda kerja 2, dan benda kerja 3. Ketiga jenis benda kerja berbentuk tabung pejal namun mempunyai bahan serta warna yang berbeda. **Error! Reference source not found.** menunjukkan wujud dari benda kerja yang digunakan pada sistem *Factory Automatic Trainer*.



Gambar 2. 7 benda kerja pada *Factory Automatic Trainer*

Tiga benda kerja mempunyai karakteristik masing-masing. Benda kerja tersebut dapat diklasifikasikan berdasarkan warna dan bahan. Variasi benda kerja ini memungkinkan seorang designer program untuk melatih kemampuan dalam memprogram proses automasi yang melibatkan proses seleksi atau klasifikasi benda. Klasifikasi benda kerja berdasarkan warna dan bahan dapat dilihat pada **Error! Not a valid bookmark self-reference..**

Tabel 2. 1 Klasifikasi benda kerja berdasarkan warna dan bahan

NO	Nama Benda	Warna	Bahan
1	benda kerja 1	hitam	plastik
2	benda kerja 2	biru	plastik
3	benda kerja 3	silver	logam

Pada *Factory Automatic Trainer* 3 jenis benda kerja dapat diberikan perlakuan berbeda pada modul-modul yang tersedia sesuai kebutuhan. Untuk dapat melakukan hal tersebut, benda kerja harus diklasifikasi-kan. Proses klasifikasi dilakukan dengan cara mendeteksi benda kerja dengan sensor-sensor terkait. Data mengenai sensor yang aktif terhadap benda kerja dinyatakan pada Tabel 2. 2 .

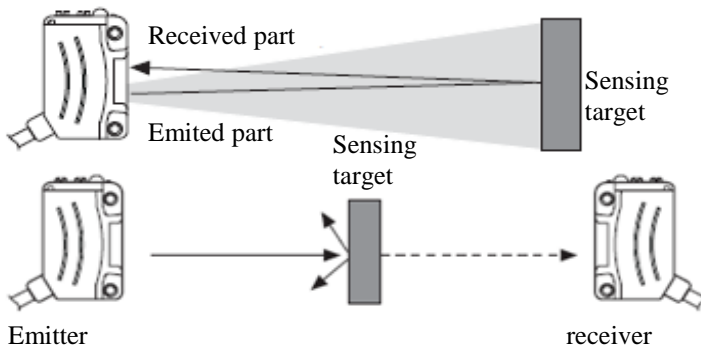
Tabel 2. 2 Klasifikasi benda kerja berdasarkan deteksi sensor

No	Nama Benda	Capacitive Sensor	Photo Sensor	Proximity Sensor
1	benda kerja 1	on	-	-
2	benda kerja 2	on	on	-
3	benda kerja 3	on	on	on

2.1.7 Photo Electric Sensor [2]

Photo electric sensor merupakan sensor yang mendeteksi kehadiran benda dengan prinsip memanfaatkan pancaran cahaya. Terdapat dua tipe photo electric sensor yaitu tipe through-beam photoelectric sensor dan retroreflective photoelectric sensor. [2] Kedua tipe tersebut ditunjukkan oleh

Gambar 2. 8. Pada gambar tersebut menunjukkan perbedaan hardware sensor sekaligus cara kerja nya.

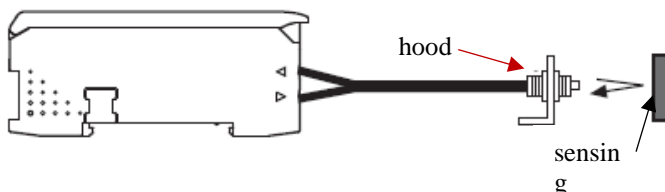


Gambar 2. 8 (a) Retroreflective photoelectric sensor (b) through beam photoelectric sensor [2]

Tipe Retroreflective photoelectric sensor memanfaatkan pantulan dari pancaran cahaya yang dikirimkan oleh emitter untuk dideteksi oleh receiver. Tipe through beam photoelectric sensor memanfaatkan intensitas cahaya yang diterima oleh receiver. Kemudian hal ini akan dirubah kebesaran listrik dan diterjemahkan.

2.1.8 Fiber Optic Sensor [2]

Fiber optic sensor digunakan untuk mendeteksi tanda ataupun benda menggunakan kabel Fiber optic sensor merupakan sensor yang menggunakan kabel fiber sebagai pengganti dari lensa yang biasa digunakan pada photo electric sensor. Hal ini memungkinkan sensor dapat ditempatkan dimanapun karena fleksibilitas dari optical fiber.



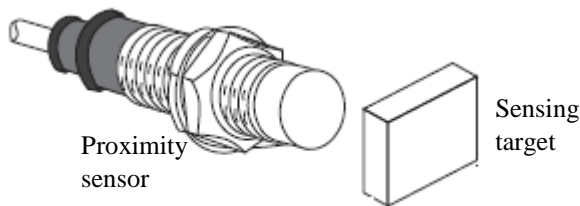
Gambar 2. 9 Optical fiber sensor tipe diffuse reflective [2]

Terdapat dua jenis fiber optic sensor yaitu tipe through beam dan tipe diffuse reflective seperti pada Gambar 2. 9. Pada *Factory Automatic Trainer* yang diteliti pada tugas akhir ini fiber optic yang

digunakan adalah tipe diffuse reflective. Cara kerja dari jenis sensor ini adalah hood akan menembakkan cahaya menuju benda target. Kemudian pantulan dari benda target akan kembali dideteksi receiver dan disalurkan melalui kabel fiber untuk diterjemahkan.

2.1.9 Proximity Sensor [2]

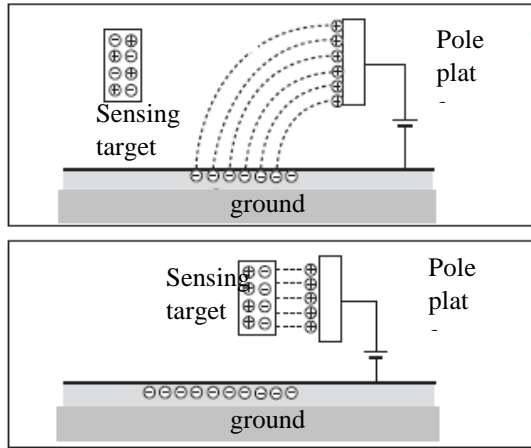
Proximity sensor merupakan sensor yang mendeteksi keberadaan benda yang mempunyai jarak yang dekat. Terdapat dua jenis proximity sensor yaitu *inductive proximity sensor* dan *capacitive proximity sensor*. Gambar 2. 10 merupakan bentuk secara umum dari proximity sensor



Gambar 2. 10 bentuk secara umum proximity sensor [2]

Prinsip kerja dari *inductive proximity sensor* adalah apabila benda logam berada pada daerah medan magnet bfrekuensi tinggi. Aliran arus yang terinduksi yang ada pada logam menyebabkan kondisi thermal loss dan menghasilkan reduction atau berhentinya kedaan oscillation. Perubahan kedaan ini akan dideteksi oleh *oscillation sensing* yang selanjutnya akan mengoperasikan *output circuit*. [2]

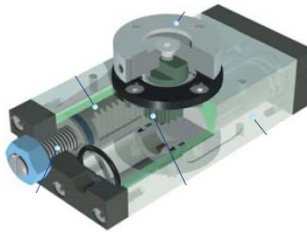
Sedangkan prinsip kerja dari *capacitive proximity sensor* adalah ketika tidak ada benda di jangkauan area sensor, maka permukaan pole plate akan dipenuhi beban positif sedangkan permukaan ground akan dipenuhi beban negatif sehingga ruang diantara pole plate dan ground akan terjadi medan listrik. Namun apabila diantara ruang tersebut terdapat sebuah benda maka sisi benda yang menghadap pole plate akan dipenuhi beban negative, sedangkan sisi satunya akan dipenuhi beban positif. Fenomena ini disebut dengan polarisasi. Pembacaan keberadaan benda bergantung pada besarnya polarisasi. Semakin dekat benda maka polarisasi semakin tinggi. [2] Prinsip kerja capacitive proximity sensor ditunjukkan pada Gambar 2. 11



Gambar 2. 11 Prinsip kerja capacitive proximity sensor [2]

2.1.10 Rotary Cylinder

Rotary cylinder merupakan peralatan yang merubah energi tekanan udara menjadi gerakan rotasi. Rotary cylinder mempunyai banyak tipe yaitu tipe vane, tipe Rack and pynion, tipe Screw, tipe Crank, dan tipe Wheel gear dan Chain. Pada plant *Factory Automatic Trainer* rotary cylinder terletak pada pick and place modul. Gambar 2. 12 menunjukkan bentuk dari rotary cylinder.



Gambar 2. 12 Rotary Cylinder [6]

Rotary cylinder ini dalam *Factory Automatic Trainer* adalah sebagai aktuator untuk menggerakkan lengan robot. Lengan robot tersebut dipergunakan untuk memindahkan benda kerja dari *separation module* ke *stopper module*. Di dalam rotary cylinder

terdapat double acting cylinder yang pergerakannya akan mendorong gerigi sehingga bergerak rotasi

2.2 Programmable Logic Controller (PLC) [3]

Programmable Logic Controller merupakan perangkat microprocessor berbasis controller yang menggunakan memori terprogram untuk menjalankan suatu instruksi serta fungsi khusus seperti aritmatika, logika, sequencing (pengurutan), timing, counting. Pada umumnya PLC dirancang untuk melakukan pengendalian mesin serta pengendalian proses yang ada pada industri. Keunggulan PLC dibandingkan controller yang lain adalah penggunaannya yang jauh lebih mudah, fleksibel, serta tahan terhadap lingkungan industri yang keras.

PLC pertama kali dikembangkan pada tahun 1969. Latar belakang dimulainya pengembangan PLC adalah ditemukannya kelemahan pada controller berbasis *relay* seiring dengan kebutuhan plant yang semakin beragam. Kelemahan tersebut berupa sistem yang sudah ada sulit untuk dimodifikasi ulang. Sekarang penggunaan PLC sudah secara luas digunakan serta berkembang pesat dari yang hanya mampu mewadahi 20 digital *input/output* hingga telah dikembangkan PLC bersistem modular yang mampu mewadahi *input/output* yang jauh lebih banyak, mampu menangani digital ataupun analog *input/output*, bahkan juga dapat melakukan control proportional-integral-derivative.

2.2.1 Sistem pada Programmable Logic Controller [3]

Pada dasarnya sistem pada Programmable Logic Controller terdiri atas enam komponen utama yaitu *processor unit*, *memory*, *power supply unit*, *input/output interface*, *communication interface*, dan *programming device*. Diagram sistem yang ada pada PLC seperti yang ditunjukkan oleh Gambar 2. 13.

2.2.1.1 Central Processing Unit

Processor unit atau *Central Processing Unit* (CPU) adalah komponen yang berfungsi menerjemahkan sinyal *input*, kemudian mengolahnya, dan pada akhirnya mengirimkan sinyal aksi berdasarkan instruksi pemrograman yang sudah ditentukan.

2.2.1.2 Power Supply

Power Supply adalah komponen yang berfungsi mengkonversi tegangan a.c. ke tegangan d.c. (5 V). Tegangan ini dipergunakan untuk mengaktifkan processor serta *Input/Output Interface*.

2.2.1.3 Input/Output Interface

Input/Output Interface adalah komponen yang berfungsi sebagai perantara antara komponen fisik *input/output* dengan processor. Melalui komponen ini processor dapat menerima sinyal *input* dari komponen fisik *input* seperti *capacitive sensor*, *temperature sensor* dan lain-lain. Serta mengirimkan sinyal aksi menuju komponen fisik *output* seperti *motor starter coils*, *solenoid valves* dan lain-lain [3].

2.2.1.4 Memory Unit

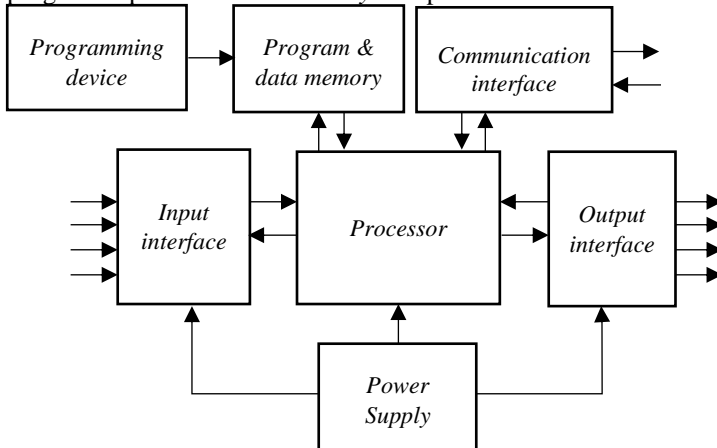
Memory unit adalah komponen yang berfungsi sebagai tempat penyimpanan program yang berisi instruksi-instruksi untuk control aksi. Selain itu komponen ini menyimpan data-data yang berkaitan dengan *input* serta *output* sistem.

2.2.1.5 Communication Unit

Communication unit adalah komponen yang berfungsi mengirim serta menerima data pada jaringan komunikasi antar perangkat PLC yang ber-beda.

2.2.1.6 Programming device

Programming device merupakan peralatan yang berfungsi untuk mendesain program yang dibutuhkan oleh plant. Melalui peralatan ini program dapat dikirim ke *memory unit* pada PLC.



Gambar 2. 13 Diagram sistem pada Programmable Logic Controller [3]

2.2.2 PLC GLOFA seri GM4 [4]

Pada *Factory Automatic Trainer* terdapat PLC yang berfungsi sebagai *controller* untuk tiap modul yang tersedia. PLC yang digunakan merupakan jenis PLC LG GLOFA seri GM4 seperti yang ditunjukkan pada Gambar 2. 14 . PLC ini merupakan jenis PLC modular. PLC jenis ini memungkinkan adanya modifikasi berupa penambahan ataupun pengurangan jumlah modul *input/output* serta modul-modul khusus sesuai kebutuhan. Seperti produk PLC LG yang lain, PLC GLOFA seri GM4 mempunyai programming device yang disebut GMWIN.



Gambar 2. 14 PLC LG GLOFA seri GM4

2.2.2.1 Spesifikasi PLC LG GLOFA seri GM4

Berikut merupakan spesifikasi dari PLC LG GLOFA seri GM4 ditunjukkan oleh Tabel 2. 3 .

Tabel 2. 3 Spesifikasi PLC LG GLOFA seri GM4 [4]

NO	Spesifikasi	Keterangan
1	<i>Input power</i>	AC85~ 264 V 50/60 Hz
2	<i>Jumlah Input</i>	32 point
3	<i>Jumlah Output</i>	32 point

4	Bahasa pemrograman	Instruction List, <i>Ladder diagram</i> , Sequential Function Chart
5	Kapasitas memori programming	128K
6	Mode pengoperasian	RUN, STOP, PAUSE, DEBUG
7	Kecepatan pemrosesan	2 μ S/instruction

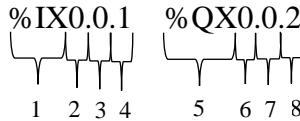
2.2.2.2 Cara Pengoperasian PLC LG GLOFA seri GM4

PLC LG GLOFA seri GM4 dioperasikan ketika operator memilih mode control otomatis. Mode ini memungkinkan operator untuk menerapkan program PLC yang sudah didesain sebelumnya ke plant *Factory Automatic Trainer*. Berikut merupakan cara pengoperasian PLC LG GLOFA seri GM4 :

1. Siapkan program yang telah didesain sebelumnya di GMWIN
2. Pastikan kabel *power* sudah terpasang
3. Atur *circuit breaker* dalam keadaan on
4. Atur *control select switch* pada keadaan auto
5. Hubungkan PC dengan PLC menggunakan *connector* RS-232
6. Pastikan *running mode* dari CPU pada keadaan *PAUSE*
7. *Compile* serta *connect* program PLC melalui GMWIN
8. Apabila lampu indikator hijau sudah menyala, atur running mode dari CPU menjadi keadaan *RUN*
9. Eksekusi program melalui GMWIN
10. PLC LG GLOFA seri GM4 akan beroperasi sesuai program yang telah didesain

2.2.2.3 Pengalamanatan pada PLC LG GLOFA seri GM4

Agar program yang sudah didesain di GMWIN dapat dijalankan, komponen *input* serta *output* pada program harus diberi pengalamanatan sesuai dengan *input* serta *output* pada plant. Tiap vendor PLC mempunyai aturan tersendiri dalam hal pengalamanatan atau *addressing*. Berikut merupakan aturan pengalamanatan pada PLC LG seperti yang ditunjukkan oleh Gambar 2. 15



- | | |
|-----------------|------------------|
| 1. <i>Input</i> | 5. <i>Output</i> |
| 2. <i>Base</i> | 6. <i>Base</i> |
| 3. <i>Slot</i> | 7. <i>Slot</i> |
| 4. <i>Bit</i> | 8. <i>Bit</i> |

Gambar 2. 15 Aturan pengalamatan pada PLC LG

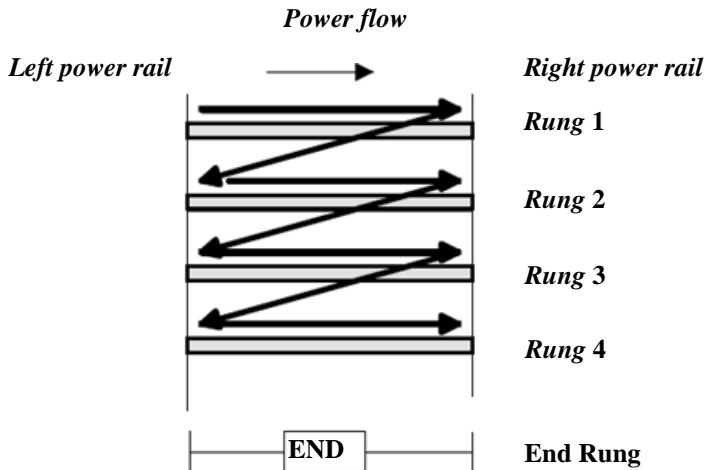
Pada Gambar 2. 15 menunjukkan dua contoh pengalamatan untuk satu perangkat *input* dan satu perangkat *output*. Pengalamatan pada perangkat *input* menunjukkan perangkat tersebut berada pada base 0 slot 0 dan bit ke 1. Sedangkan pengalamatan pada perangkat *output* menunjukkan perangkat berada pada base 0 slot 0 dan bit ke 2.

2.3 *Ladder diagram* [3]

Ladder diagram merupakan metode yang umum digunakan pada pemrograman PLC saat ini. *Ladder diagram* terdiri atas dua garis vertical yang merepresentasikan power rail. Kedua sirkuit ini terhubung dengan garis horizontal yang merepresentasikan rung dari *ladder diagram* [3].

Dalam menggambarkan *ladder diagram* perlu diperhatikan hal-hal berikut:

1. Garis vertical merepresentasikan power rails yang dihubungkan oleh garis horizontal (rung dari *ladder diagram*)
2. Setiap rung mendefinisikan satu operasi pada suatu *control process*
3. Pembacaan *ladder diagram* adalah dari kiri ke kanan dan dari atas ke bawah. Gambar 2. 16 menunjukkan bagaimana proses scanning pada *ladder diagram*. Rung paling atas dibaca dari kiri ke kanan. Selanjutnya pada rung kedua pembacaan dari kiri ke kanan kembali begitu seterusnya. Ketika PLC sedang dalam keadaan mode run. Maka program akan terus berjalan hingga rung pada *ladder diagram* berakhir. Prosedur ini biasa disebut dengan cycle. Berakhirnya rung pada *ladder diagram* biasanya disimbolkan dengan kata END atau RET.



Gambar 2. 16 Proses *scanning* pada *ladder diagram* [3]

4. Setiap rung harus mempunyai minimal 1 *input* dan 1 *output*. Istilah *input* digunakan untuk tindakan kontrol seperti penutupan kontak switch, dan digunakan sebagai *input* ke PLC. Istilah *output* digunakan untuk perangkat yang terhubung dengan *output* PLC seperti motor ataupun pneumatik cylinder

5. Dalam *ladder diagram* akan ditunjukkan perangkat listrik dalam kondisi normalnya. Misalkan normally open merupakan kondisi perangkat listrik yang akan terbuka hingga ada perangkat lain yang menutupnya. Kondisi lain adalah normally closed yang akan ditunjukkan dengan symbol tertutup.

6. Dalam serangkaian *ladder diagram* bisa saja ditemukan perangkat khusus yang terpasang dilebih dari satu rung. Begitu juga dengan huruf serta penomoran bisa saja digunakan untuk melabeli lebih dari satu perangkat.





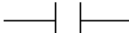

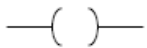
7. *Input* dan *output* semuanya diidentifikasi oleh addresses nya masing-masing. Notasi yang digunakan tergantung dengan pabrikan PLC

2.3.1 Simbol dasar pada *Ladder diagram*

Berdasarkan standar IEC 1131-3 ditetapkan simbol-simbol dasar untuk perangkat *input* dan *output* pada *ladder diagram*. Simbol-simbol tersebut ditunjukkan pada Tabel 2. 4. Perangkat *input* direpresentasi-

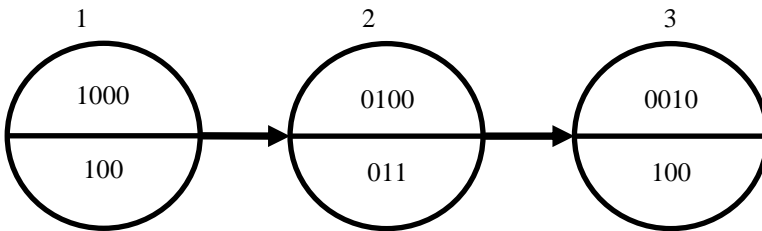
kan dengan dua simbol yang berbeda yaitu normally open contact serta normally close contact. Normally open contact berfungsi untuk mengaktifkan suatu switch sebaliknya normally close switch berfungsi mematikan suatu switch. Sedangkan untuk perangkat *output* hanya direpresentasikan oleh satu simbol yaitu *output coil*.

Tabel 2. 4 Simbol dasar pada *ladder diagram* [3]

No	Simbol	Keterangan
1		<i>Horizontal link</i> yaitu tempat dimana <i>power</i> dapat mengalir
2		<i>Interconnection link</i> merupakan link penghubung antar dua atau lebih horizontal link
3		<i>Left hand power rail</i>
4		<i>Right hand power rail</i>
5		<i>Normally open contact</i> merupakan jenis contact yang apabila <i>energized</i> maka <i>power</i> pada rung terkait akan tetap mengalir
6		<i>Normally close contact</i> merupakan jenis contact yang apabila <i>energized</i> maka <i>power</i> yang mengalir pada rung terkait akan diputus
7		<i>Output coil</i> merupakan simbol yang merepresentasikan aktif atau tidaknya suatu <i>output</i> . <i>Output coil</i> akan aktif ketika <i>energized</i> .

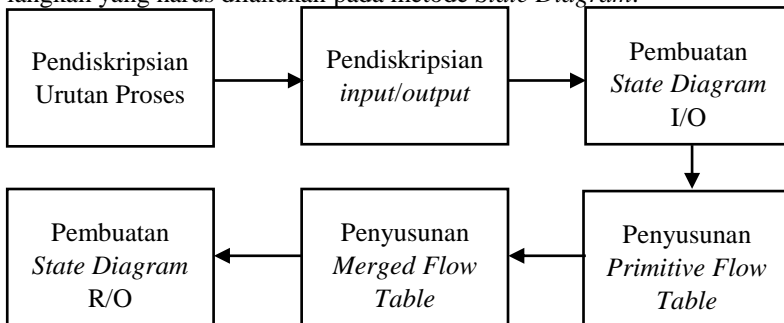
2.4 State Diagram [5]

State Diagram merupakan sebuah grafik yang merepresentasikan kejadian ataupun keadaan suatu sistem dalam bentuk lingkaran. Lingkaran tersebut biasa disebut dengan *state*. *State* tersebut berisikan *input* serta *output* suatu sistem yang diwakili dengan bilangan biner seperti pada Gambar 2. 17 .



Gambar 2. 17 *State Diagram*

State Diagram dapat diterapkan sebagai metode dalam perancangan *ladder diagram*. Gambar 2. 18 menunjukkan langkah-langkah yang harus dilakukan pada metode *State Diagram*.



Gambar 2. 18 Langkah-langkah pada Metode *State Diagram*

2.4.1 Pendiskripsian urutan proses

Langkah pertama pada metode *State Diagram* adalah pendeskripsian urutan proses. Pendeskripsian urutan proses adalah upaya penyusunan urutan state beserta penjelasan terkait bagaimana kondisi *input* serta *output* pada *state* tersebut. Hal ini akan mempermudah dalam pembuatan *State Diagram* berdasarkan I/O. Tabel 2. 5 menunjukkan format tabel deskripsi state.

Tabel 2. 5 Pendiskripsian Urutan Proses

State	Kondisi <i>Input</i>	Kondisi <i>Output</i>
S	START on	CONV1 on
S+1	MAG on	SEP1 on
⋮	⋮	⋮
Sn	(.....)	(.....)

2.4.2 Pendiskripsian I/O

Langkah kedua dalam metode *State Diagram* adalah pendiskripsian *input/output*. Hal ini dilakukan agar mempermudah operator dalam memahami komponen apa saja yang berperan dalam sebuah sistem yang direpresentasikan dalam bentuk *State Diagram* nantinya. *Input* akan diberikan penamaan ataupun simbol dalam *State Diagram* berupa huruf X_1 hingga X_n dan Z_1 hingga Z_n untuk *output*. Pemberian simbol ini bersifat bebas atau boleh diganti dengan huruf lain. Hal yang terpenting simbol itu harus singkat sehingga memudahkan ketika menggambar *State Diagram*. Tabel 2. 6 menunjukkan format tabel deskripsi *input* serta Tabel 2. 7 menunjukkan format tabel deskripsi *output*.

Tabel 2. 6 Tabel Deskripsi *Input*

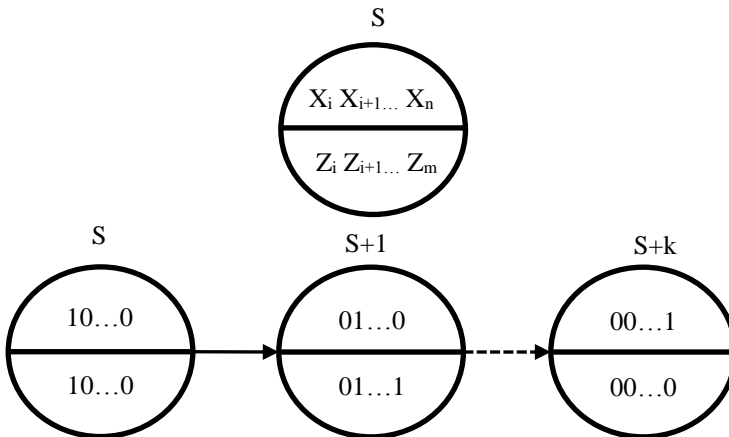
No	Nama <i>Input</i>	Simbol pada <i>State Diagram</i>
i	START	X_i
i+1	(.....)	X_{i+1}
⋮	⋮	⋮
n	(.....)	X_n

Tabel 2. 7 Tabel Deskripsi *Output*

No	Nama <i>Output</i>	Simbol pada <i>State Diagram</i>
i	CONV1	Z_i
i+1	(.....)	Z_{i+1}
⋮	⋮	⋮
N	(.....)	Z_n

2.4.3 Pembuatan *State Diagram* I/O [5]

Langkah ketiga dalam metode *State Diagram* adalah pembuatan *State Diagram* berdasarkan I/O. Pembuatan *State Diagram* dimulai dengan menggambarakan lingkaran sebagai representasi state yang sudah disusun pada tabel deskripsi *state* sebelumnya. Setelah itu, pada bagian atas lingkaran-lingkaran yang sudah digambar diberi label sesuai urutan state yang ada. Lingkaran diisi dengan format penulisan kondisi *input* per kondisi *output* pada state tersebut. Penulisan *input/output* yang aktif dinyatakan dalam digit biner “1” sebaliknya penulisan untuk *input/output* yang tidak aktif dinyatakan dalam digit biner “0”[4]. Gambar 2. 19 merupakan format pembuatan *State Diagram* berdasarkan I/O beserta contohnya. Simbol X menyatakan *input* sedangkan Z menyatakan *output*, dan S menunjukkan urutan state.



Gambar 2. 19 *State Diagram* I/O

2.4.4 Penyusunan *Primitive Flow Table* [5]

Primitive Flow Table adalah tabel yang terdiri atas kondisi *input* serta *output* setiap state yang dinyatakan dalam digit bit sesuai *State Diagram* I/O yang dibuat sebelumnya. *Primitive Flow Table* berisi kombinasi bit *input* serta *output* yang sesuai dengan *State Diagram* I/O yang telah dibuat sebelumnya [5]. Tabel 2. 8 menunjukkan format penulisan *primitive flow Table*.

Tabel 2. 8 *Primitive Flow Table*

<i>Primitive flow Table</i>									
Row	<i>Input (Xi Xi+1...Xn)</i>					<i>Output</i>			
	10...0	01...0	01...1	...	00...1	Zi	Zi+1	...	Zm
S	S	S+1	-	...	-	1	0	...	0
S+1	-	S+1	S+2	...	-	0	1	...	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
S+k	-	-	-	...	S+k	1	0		0

Hal pertama yang dilakukan adalah mengisi row sesuai urutan penomoran state yang ada pada *State Diagram* I/O. Setelah itu mengisi kolom *input* sesuai kombinasi bit *input*. Misal terdapat 3 state dengan kombinasi *input* 100, 010, 001. Maka jumlah kolom *input* ditulis sebanyak 3 kolom sesuai dengan jumlah state dan diisi dengan 3 kombinasi bit tersebut sesuai urutannya. Bila terdapat 2 atau lebih state yang memiliki kombinasi bit *input* yang sama maka cukup dituliskan ke dalam satu kolom. Pengisian kombinasi bit *input* dan row dapat dilihat di Tabel 2. 9

Tabel 2. 9 kolom bit *input* serta row pada *primitive flow Table*

<i>Primitive flow Table</i>									
Row	<i>Input (Xi Xi+1...Xn)</i>					<i>Output</i>			
	10...0	01...0	01...1	...	00...1	Zi	Zi+1	...	Zm
S	S	S+1	-	...	-	1	0	...	0
S+1	-	S+1	S+2	...	-	0	1	...	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
S+k	-	-	-	...	S+k	1	0		0

Apabila kolom *input* sudah lengkap, langkah selanjutnya adalah mengisi kolom-kolom *input* dengan “state stabil”, “state tidak stabil”, dan “*don’t care*” secara berurutan. State stabil adalah state yang mempunyai kondisi *input* dan *output* yang sama sesuai dengan *State Diagram* I/O. Pada *State Diagram* I/O state stabil diwakilkan oleh pasangan *input output* yang terdapat pada masing-masing lingkaran. Pada menunjukkan “**S**”, “**S+1**” dan “**S+k**” yang bercetak tebal

menunjukkan state stabil. Tabel 2. 10 menunjukkan pengisian state stabil pada *primitive flow Table*.

Tabel 2. 10 state stabil pada *primitive flow Table*

<i>Primitive flow Table</i>									
Row	<i>Input (Xi Xi+1...Xn)</i>					<i>Output</i>			
	10...0	01...0	01...1	...	00...1	Zi	Zi+1	...	Zm
S	S	S+1	-	...	-	1	0	...	0
S+1	-	S+1	S+2	...	-	0	1	...	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
S+k	-	-	-	...	S+k	1	0		0

Sedangkan state tidak stabil merupakan kondisi transisi perpindahan satu state ke state yang lain. Di *State Diagram I/O* state tidak stabil direpresentasikan dengan tanda panah. Sedangkan pada *primitive flow Table* state tidak stabil ditulis dengan angka bercetak normal. Angka yang dituliskan untuk state tidak stabil merupakan angka yang sama dengan state stabil yang dituju oleh panah. Contoh penulisan state tidak stabil dapat dilihat pada Tabel 2. 11

Tabel 2. 11 state tidak stabil pada *primitive flow Table*

<i>Primitive flow Table</i>									
Row	<i>Input (Xi Xi+1...Xn)</i>					<i>Output</i>			
	10...0	01...0	01...1	...	00...1	Zi	Zi+1	...	Zm
S	S	S+1	-	...	-	1	0	...	0
S+1	-	S+1	S+2	...	-	0	1	...	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
S+k	-	-	-	...	S+k	1	0		0

Terdapat kondisi don't care yang mana merupakan kondisi yang diabaikan. Sel-sel yang tidak terisi oleh state stabil maupun state tidak stabil akan diisi dengan keadaan don't care. Kondisi don't care direpresentasikan dengan tanda "-". Contoh penulisan kondisi don't care pada *primitive flow Table* dapat dilihat pada Tabel 2. 12.

Tabel 2. 12 kondisi dont care pada *primitive flow Table*

<i>Primitive flow Table</i>									
Row	<i>Input (Xi Xi+1...Xn)</i>					<i>Output</i>			
	10...0	01...0	01...1	...	00...1	Zi	Zi+1	...	Zm
S	S	S+1	-	...	-	1	0	...	0
S+1	-	S+1	S+2	...	-	0	1	...	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
S+k	-	-	-	...	S+k	1	0		0

Langkah selanjutnya adalah pengisian kolom *output*. Jumlah kolom *output* bergantung pada jumlah variable *output* yang terlibat. Misal terdapat 2 variable *output* yang terlibat yaitu Z_1 dan Z_2 maka jumlah kolom *output* dibuat 2 kolom. Kolom-kolom tersebut diisi sesuai kombinasi bit *output* yang terdapat pada *State Diagram* I/O. Penulisan kombinasi bit *output* pada *primitive flow Table* dapat dilihat pada Tabel 2. 13

Tabel 2. 13 penulisan kombinasi bit *output* pada *primitive flow Table*

<i>Primitive flow Table</i>									
Row	<i>Input (Xi Xi+1...Xn)</i>					<i>Output</i>			
	10...0	01...0	01...1	...	00...1	Zi	Zi+1	...	Zm
S	S	S+1	-	...	-	1	0	...	0
S+1	-	S+1	S+2	...	-	0	1	...	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
S+k	-	-	-	...	S+k	1	0		0

2.4.5 Penyusunan *Merged Flow Table* [5]

Merged Flow Table merupakan tabel penyerdehanaan *output* dan penentuan jumlah *relay*. Informasi yang terdapat pada *Merged Flow Table* akan dipergunakan dalam pembuatan *State Diagram* R/O (*Relay/Output*). Dalam format tabel ini baris-baris yang memiliki *output* yang sama pada *Primitive Flow Table* dapat digabungkan. Tujuan dari penggabungan ini adalah untuk meminimalkan jumlah *relay* yang akan dipergunakan nantinya. Hal ini didasarkan pada jumlah *relay* yang dibutuhkan berbanding lurus dengan jumlah baris

yang ada. Berikut merupakan aturan yang harus dipenuhi saat menggabungkan baris pada *Merged Flow Table*:

1. Apabila terdapat kondisi state stabil dan kondisi state tidak stabil dalam satu kolom yang sama, maka baris hasil penggabungan harus diisi dengan state stabil.
2. Apabila terdapat kondisi state stabil dan kondisi don't care dalam satu kolom yang sama, maka baris hasil penggabungan harus diisi dengan state stabil
3. Apabila terdapat kondisi state tidak stabil dan kondisi don't care dalam satu kolom yang sama, maka baris hasil penggabungan harus diisi dengan state tidak stabil
4. Apabila terdapat 2 kondisi don't care dalam satu kolom yang sama maka baris hasil penggabungan harus diisi dengan kondisi don't care

Tabel 2. 14 Penggabungan baris pada *Merged Flow Table*

<i>Merged flow Table</i>									
Row	<i>Input (Xi Xi+1...Xn)</i>					<i>Output</i>			
	10...0	01...0	01...1	...	00...1	Zi	Zi+1	...	Zm
S	S	S+1	-	...	-	1	0	...	0
S+1	-	S+1	S+2	...	-	0	1	...	1
S+2	-	-	S+2	S+3	-	1	0	...	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
S+k	-	-	-	...	S+k	1	0		0

Contoh pada Tabel 2. 14 penggabungan baris pada *Merged Flow Table* ditunjukkan pada baris ke S dan ke S+2 yang mempunyai *output* yang sama, kedua baris ini dapat digabungkan. Sesuai aturan penggabungan baris, maka *merged flow Table* menjadi seperti yang ditunjukkan pada Tabel 2. 15.

Tabel 2. 15 *Merged Flow Table* setelah penggabungan baris

<i>Merged Flow Table</i>									
Row	<i>Input (Xi Xi+1...Xn)</i>					<i>Output</i>			
	10...0	01...0	01...1	...	00...1	Zi	Zi+1	...	Zm
S,S+2	S	S+1	S+2	S+3	-	1	0	...	0
S+1	-	S+1	S+2	...	-	0	1	...	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
S+k	-	-	-	...	S+k	1	0		0

Langkah selanjutnya pada *merged flow Table* adalah penentuan *relay*. *Relay* yang dibuat harus mampu mengakomodasi seluruh baris pada *merged flow Table*. *Relay* pada *merged flow Table* menggunakan teori RS flip flop. Satu *relay* dapat menghasilkan 2 kombinasi bit yang dapat mengakomodasi 2 baris. Jika n adalah jumlah *relay*, m adalah jumlah baris, dan 2^n adalah kombinasi bit, maka $2^n \geq m$. Sebagai contoh apabila terdapat 3 baris pada *merged flow Table* maka jumlah *relay* yang dibutuhkan adalah 2 buah sehingga menghasilkan 4 kombinasi bit. Empat kombinasi bit tersebut mengakomodasi 3 baris meskipun terdapat 1 sisa kombinasi bit yang tidak terpakai.

Kombinasi bit pada kolom *relay* harus bermengikuti sistem gray code. Gray code merupakan salah satu sistem penomoran biner yang hanya berubah satu bit tiap keadaan. Penggunaan sistem gray code ini dapat menghindari kesalahan pembacaan program pada saat perubahan dari satu state ke state selanjutnya. Misal terdapat state dengan kombinasi bit 1100 sedangkan state selanjutnya mempunyai kombinasi bit 1010. Pada saat perpindahan state bisa saja terjadi logika 1110 sehingga program akan melakukan instruksi lain di sela-sela pergantian state tersebut. Berikut merupakan contoh *merged flow Table* yang sudah dilengkapi *relay* dinyatakan pada Tabel 2. 16.

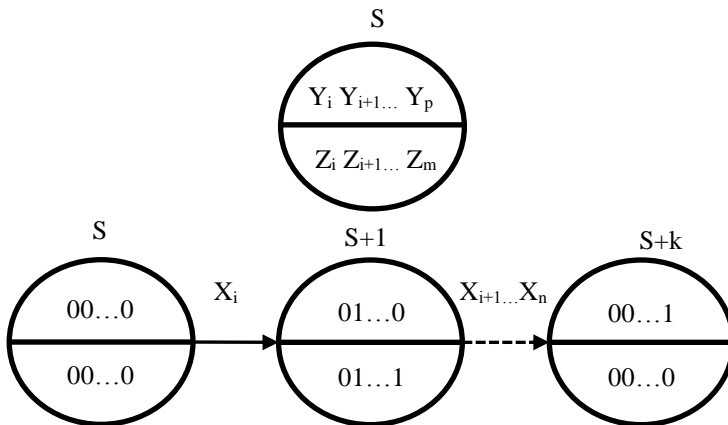
Tabel 2. 16 *Merged flow Table* yang sudah dilengkapi kolom *relay*

<i>Merged Flow Table</i>								
Row	Input (X ₁ X ₂ X ₃ X ₄)				Output (Z ₁ Z ₂)		Relay (Y ₁ Y ₂)	
	1000	0100	0010	0001	Z ₁	Z ₂	Y ₁	Y ₂
1,3	1	2	3	4	1	0	1	0
2	-	2	3	-	0	1	1	1
4	-	-	-	4	0	0	0	1

2.4.6 Pembuatan *State Diagram* R/O

Pembuatan *State Diagram* berdasarkan (R/O) merupakan langkah terakhir pada metode *State Diagram*. Format penulisan *State Diagram* R/O hampir sama dengan *State Diagram* (I/O). Hal yang membedakan hanya pada isi pada setiap state diganti dari format *input/output* menjadi *relay/output*. Selain itu di setiap panah pergantian state diberikan informasi berupa *input* apa saja yang berperan dalam

pergantian state tersebut. Simbol Y menunjukkan *relay*, simbol X menunjukkan *input*, dan S menunjukkan urutan state. Pada lingkaran bagian atas diisi dengan kombinasi bit *Relay* sedangkan bagian bawah diisi dengan kombinasi bit *output*. Sedangkan *Input* diletakkan pada simbol-simbol panah. Hal ini bermakna *input-input* tersebut yang berperan mengaktifkan *output* pada state yang dituju oleh panah tersebut. Berikut merupakan *State Diagram* (R/O) dengan data yang termuat pada Tabel 2. 16 yang ditunjukkan oleh Gambar 2. 20.



Gambar 2. 20 *State Diagram* (R/O)

2.5 Konversi *State Diagram* ke *Ladder diagram* [4]

Untuk membuat program instruksi pada PLC, *State Diagram* (R/O) harus diterjemahkan dahulu ke dalam bentuk *ladder diagram*. Proses penerjemahan dimulai dengan mengkonstruksi *ladder diagram* untuk setiap *relay* serta *output*. Untuk mempermudah pemahaman pada Gambar 2. 21 terdapat contoh *State Diagram* (R/O) dengan 2 *relay* yaitu “Y₁” dan “Y₂” serta 2 *output* “Z₁” dan “Z₂”.

Langkah pertama adalah membuat konstruksi *ladder diagram* untuk *relay* “Y₁” dan “Y₂”. Hal yang harus dilakukan adalah memahami perubahan bit *relay* pada setiap state. Bit *relay* dinyatakan aktif apabila bit bernilai “1” dan tidak aktif apabila bit bernilai “0”. Perubahan bit *relay* dari “0” (state asal) ke “1” (state tujuan) disebut *set*, Sedangkan perubahan bit *relay* dari “1” ke “0” disebut dengan *reset*.

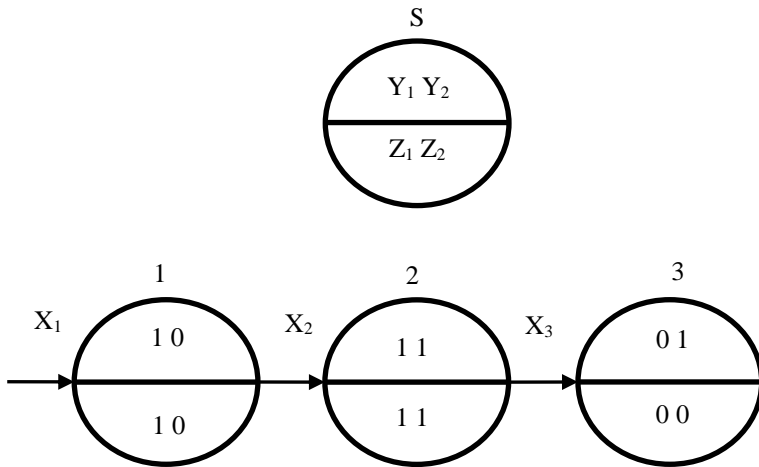
Berikut merupakan langkah -langkah penerjemahan *relay* pada *State Diagram* ke *ladder diagram* :

1. mencari kondisi *set* untuk *relay* tersebut pada *State Diagram*
2. mencatat *input* yang berperan pada kondisi *set* tersebut
3. mencari kondisi reset pada *State Diagram*
4. mencatat *input* yang berperan pada kondisi reset tersebut
5. membuat Normally Open contact(NO) untuk *input* yang berperan pada kondisi *set* serta membuat Normally Open contact(NO) untuk *relay* tersebut sebagai *self holding*. Self holding berfungsi untuk menyatakan bahwa *relay* akan terus aktif sampai ada kondisi *reset*.
6. menyusun secara parallel NO contact *input* yang berperan pada kondisi set dengan NO contact *relay* tersebut.
7. membuat normally close contact (NC) untuk *input* yang berperan dalam kondisi reset.
8. membuat coil dengan penamaan sesuai *relay* tersebut
9. contact yang berperan pada kondisi set serta kondisi reset dan coil *relay* disusun secara seri

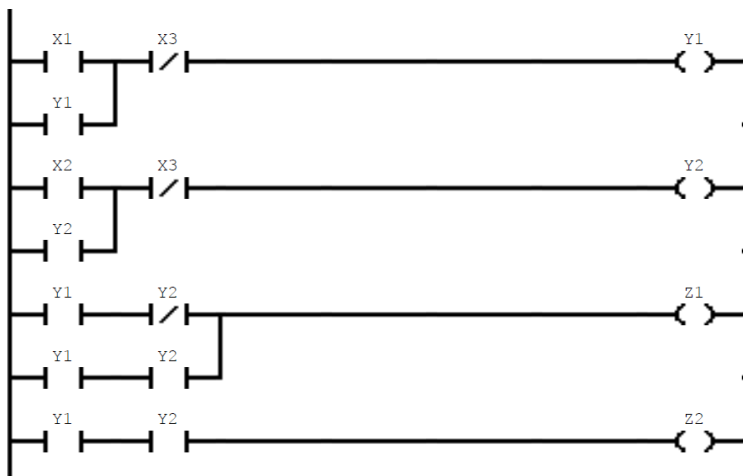
Langkah kedua adalah membuat konstruksi *ladder diagram* untuk *output* “Z₁” dan “Z₂”. Berikut merupakan langkah-langkah yang perlu dilakukan untuk penerjemahan *output* menjadi *ladder diagram*,:

1. mencari state yang berisi kondisi aktif dari *output* terkait
2. mencatat kombinasi bit *relay* yang berada pada state-state tersebut
3. membuat contact untuk kombinasi *input* pada state-state tersebut. NO contact untuk bit yang bernilai “1” dan NC contact untuk bit yang bernilai “0”
4. membuat simbol coil untuk *output*
5. menyusun seri contact untuk kombinasi bit *relay* dengan coil *output*

Contoh hasil konversi *State Diagram* pada Gambar 2. 21 ke *ladder diagram* dapat dilihat pada Gambar 2. 22



Gambar 2. 21 Contoh *State Diagram* yang akan dikonversi ke *ladder diagram*



Gambar 2. 22 Hasil konversi *State Diagram* ke *Ladder diagram*

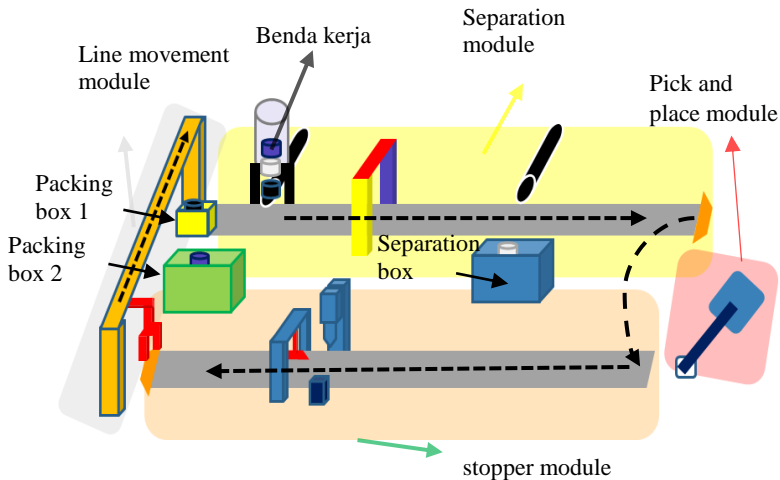
BAB 3

PERANCANGAN SISTEM

Pada bab ini membahas tentang perancangan sistem yang terdiri atas 3 tahap yaitu perumusan sistem *Factory Automatic Trainer*, perancangan *State Diagram*, dan konstruksi *ladder diagram*.

3.1 Perumusan Sistem *Factory Automatic Trainer*

Pada penelitian ini, dirumuskan sebuah sistem produksi benda yang terdiri atas proses seleksi benda, pemindahan benda, proses drilling dan proses *packing*. Oleh karena itu pada perumusan sistem ini digunakan 4 module yang terdapat pada *Factory Automatic Trainer* yaitu separation Module, Pick and Place Module, Stopper Module, dan Line Movement Module. Gambar 3. 1 menunjukkan perumusan sistem *Factory Automatic Trainer* secara umum



Gambar 3. 1 Perumusan sistem *Factory Automatic Trainer* secara umum

Pada sistem ini dipergunakan 3 jenis benda kerja yang akan mendapat perlakuan dari modul-modul yang terdapat pada *Factory Automatic Trainer*. Benda kerja 1 dan benda kerja 2 akan diproses menggunakan keempat modul yang terdapat pada *Factory Automatic Trainer* yaitu proses seleksi benda di separation module, pemindahan benda pada pick and place module, proses drilling pada stopper module, dan proses *packing* pada line movement module. Sedangkan benda kerja 3 hanya akan melewati proses seleksi benda pada

separation module. Benda kerja 3 akan didorong keluar menuju separation box.

Secara umum, proses yang pertama adalah benda kerja akan masuk ke separation module untuk proses seleksi benda. Pada modul ini benda kerja yang tidak dibutuhkan akan dikeluarkan dari modul. Setelah itu benda kerja akan memasuki pick and place module. Pada pick and place module terdapat lengan robot yang akan memindahkan benda kerja dari separation module menuju stopper module. Pada stopper module benda kerja akan diproses menggunakan drilling. Proses yang terakhir adalah benda kerja akan dipindahkan oleh lengan gripper dari stopper modul menuju tempat packing. Benda kerja 1 akan ditempatkan pada *packing box* 1 sedangkan benda kerja 2 ditempatkan pada *packing box* 2.

3.1.1 Langkah Kerja Sistem

Berdasarkan perumusan sistem *Factory Automatic Trainer* terdapat 4 proses utama yaitu proses seleksi benda kerja di separation module, proses pemindahan benda kerja di pick and place module, proses drilling di stopper module, dan proses packing pada line movement module.

Keseluruhan proses pada *Factory Automatic Trainer* akan dibagi menjadi 10 subproses. Hal ini dilakukan untuk mempermudah designer program saat proses perancangan *State Diagram*. Berikut merupakan langkah kerja secara rinci pada perancangan sistem *Factory Automatic Trainer*:

3.1.1.1 Langkah kerja pada Separation Module

Proses seleksi benda yang terjadi di Separation Module dibagi menjadi 4 subproses sebagai berikut

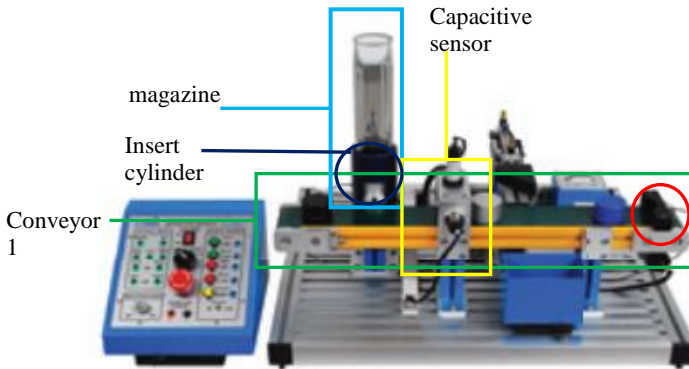
1. Subproses 1 (memasukkan benda kerja)

Pada subproses ini bertujuan untuk memasukkan benda kerja ke dalam plant *Factory Automatic Trainer*. Berikut merupakan langkah kerja pada subproses 1 :

- Proses dimulai dengan memasukkan benda kerja ke dalam magazine. Hingga benda kerja terdeteksi oleh magazine sensor.
- Setelah itu tombol start harus ditekan untuk memulai proses.
- Terdeteksinya benda kerja oleh magazine sensor dan ditekannya tombol start menyebabkan insert cylinder eject dan mendorong benda menuju conveyor 1.

- Ketika insert cylinder melakukan gerak eject sepenuhnya, conveyor 1 aktif. Conveyor 1 akan membawa benda kerja menuju ke arah pick and place module.

Komponen yang berperan pada subproses 1 dapat dilihat di Gambar 3. 2.



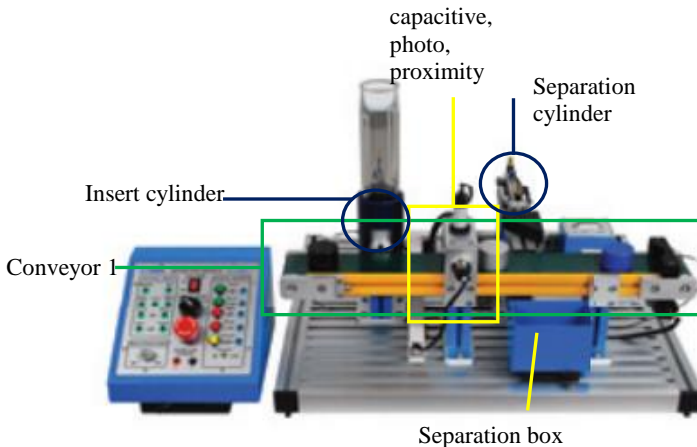
Gambar 3. 2 komponen yang berperan pada subproses 1 (memasukkan benda) [1]

2. Subproses 2 (Deteksi Benda Kerja 3 (silver))

Subproses ini bertujuan untuk memberikan perlakuan pada benda kerja 3. Benda kerja 3 akan terseleksi, dan akan dikeluarkan dari proses. Berikut merupakan langkah kerja pada subproses 2 :

- Ketika benda kerja 3 terdeteksi oleh capacitive sensor, photo sensor, dan inductive proximity sensor, maka separation cylinder bergerak eject selama 3 detik, insert cylinder bergerak return, dan conveyor 1 aktif
- Benda kerja 3 akan dibawa sampai separation cylinder, sehingga benda kerja terdorong keluar dari conveyor 1 menuju separation box
- Ketika separation cylinder sudah selesai bergerak eject selama 3 detik, maka separation cylinder akan bergerak return , dan insert cylinder bergerak eject untuk memasukkan benda kerja yang baru.
- Setelah insert cylinder bergerak eject sepenuhnya, maka conveyor 1 berhenti

Komponen yang berperan pada subproses 2 dapat dilihat di Gambar 3. 3.



Gambar 3. 3 komponen yang berperan pada subproses 2, 3,dan 4 [1]

3. Subproses 3 (Deteksi Benda Kerja 2 (biru))

Subproses ini bertujuan untuk memberikan perlakuan pada benda kerja 2. Benda kerja 2 akan dibawa sampai dideteksi fiber sensor 1 di akhir separation module. Berikut merupakan langkah kerja pada subproses 3 :

- Ketika benda kerja 2 dideteksi oleh capacitive sensor dan photo sensor, maka benda kerja 2 dibawa oleh conveyor 1 hingga sampai pada ujung separation module
- Ketika benda kerja 2 smpai pada ujung separation module dan dideteksi fiber sensor 1, maka conveyor 1 akan nonaktif

Komponen yang berperan pada subproses 3 ditunjukkan pada Gambar 3. 3

4. Subproses 4 (Deteksi Benda Kerja 1 (hitam))

Subproses ini bertujuan untuk memberikan perlakuan pada benda kerja 2. Benda kerja 2 akan dibawa sampai dideteksi fiber sensor 1 di akhir separation module. Berikut merupakan langkah kerja pada subproses 3 :

- Ketika benda kerja 2 dideteksi oleh capacitive sensor dan photo sensor, maka benda kerja 2 dibawa oleh conveyor 1 hingga sampai pada ujung separation module
- Ketika benda kerja 2 sampai pada ujung separation module dan dideteksi fiber sensor 1, maka conveyor 1 akan nonaktif

Komponen yang berperan pada subproses 4 ditunjukkan pada Gambar 3. 3

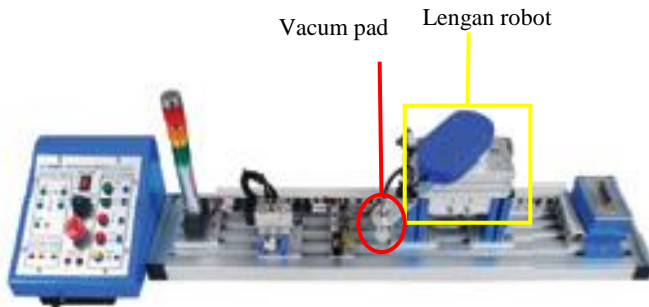
3.1.1.2 Langkah Kerja Pada Pick And Place Module (Subproses 5)

Berikut merupakan langkah kerja pada pick and place module yang disebut dengan Subproses 5 (Pick and Place) :

Subproses ini berfungsi untuk memindahkan baik benda kerja 1 maupun benda kerja 2 yang sudah terdeteksi oleh fiber sensor 1 ke stopper module. Berikut merupakan langkah kerja pada subproses 5:

- Ketika benda kerja terdeteksi oleh fiber sensor 1, maka lengan robot akan bergerak searah jarum jam .
- Lengan kerja yang sudah bergerak searah jarum jam sepenuhnya, selanjutnya akan turun
- Setelah lengan robot turun sepenuhnya, maka vacuum pad aktif dan akan menyedot benda kerja
- Benda kerja yang sudah disedot oleh vacuum pad, maka lengan robot akan naik
- Ketika lengan robot sudah naik sepenuhnya, maka selanjutnya lengan robot akan bergerak melawan jarum jam.
- Ketika lengan robot sudah bergerak melawan arah jarum jam sepenuhnya, maka lengan robot akan bergerak turun
- Lengan robot yang sudah bergerak turun sepenuhnya, maka vacuum pad nonaktif sehingga benda kerja berada di conveyor 2 yang berada di stopper module

Komponen yang berperan dalam subproses 5 ditunjukkan Gambar 3. 4.



Gambar 3. 4 komponen yang berperan dalam subproses 5 [1]

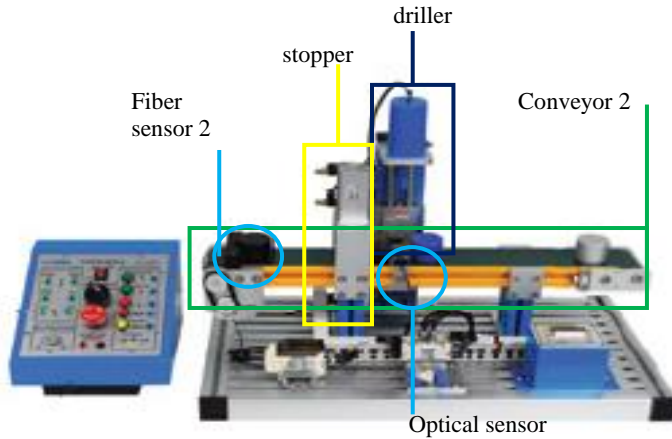
**3.1.1.3 Langkah Kerja Pada Stopper Module (Subproses 6) **

Berikut merupakan langkah kerja yang terdapat pada stopper module yang disebut subproses 6 (Stop dan Drilling):

Subproses ini berfungsi untuk menyalurkan benda kerja sampai workpoint dan dilakukan proses drilling. Berikut merupakan langkah kerja pada subproses 6:

- Ketika benda kerja sudah sampai di conveyor 2, maka benda kerja akan disalurkan sampai ke workpoint dan dihentikan oleh stopper
- Ketika benda kerja sudah dideteksi oleh optical sensor, maka driller akan turun dan conveyor 1 berhenti
- Driller yang sudah turun sepenuhnya, maka driller akan melakukan proses drilling selama 3 detik
- Ketika driller sudah selesai melakukan drilling, maka driller dan stopper akan naik dan conveyor 2 akan aktif kembali
- Benda kerja akan disalurkan oleh conveyor 2 hingga dideteksi fiber sensor 2.
- Benda kerja yang sudah dideteksi fiber sensor 2 menyebabkan stopper turun dan conveyor 2 akan berhenti.

Komponen yang berperan dalam subproses 6 ditunjukkan Gambar 3. 5



Gambar 3. 5 komponen yang berperan dalam subproses 6 [1]

3.1.1.4 Langkah Kerja Pada Line Movement Module

Pada line movement module akan dibagi menjadi 4 subproses sebagai berikut:

1. Subproses 7 (Gripping)

Subproses ini berfungsi menjepit produk baik produk 1 (benda kerja 1 yang sudah melewati proses drilling) maupun produk 2 yang (benda kerja 2 yang sudah melewati proses drilling) telah dideteksi fiber sensor untuk bersiap dipacking. Berikut merupakan langkah kerja pada subproses 7 :

- produk yang sudah dideteksi oleh fiber sensor 2 akan menyebabkan lengan gripper bergerak backward menuju arah stopper module
- Ketika lengan gripper sudah bergerak backward sepenuhnya, maka lengan gripper akan turun
- Lengan gripper yang sudah turun sepenuhnya, selanjutnya akan menyebabkan gripper menjepit produk

Komponen yang berperan pada subproses 7 ditunjukkan Gambar 3. 6

2.. Subproses 8 (Packing produk 1)

Subproses ini berfungsi melakukan perlakuan berupa packing pada produk 1 (benda kerja 1 yang sudah melewati proses drilling). Berikut merupakan langkah kerja pada pada subproses 8:

- Produk 1 yang sudah dijepit gripper, maka akan dibawa lengan gripper bergerak naik
- Lengan gripper yang sudah bergerak naik sepenuhnya, selanjutnya akan bergerak forward
- Lengan gripper yang sudah bergerak forward sepenuhnya, maka selanjutnya akan bergerak turun
- Lengan gripper yang sudah bergerak turun sepenuhnya maka menyebabkan gripper membuka sepenuhnya dan meletakkan produk 1 pada *packing box 1*

Komponen yang berperan pada subproses 8 ditunjukkan Gambar 3. 6

3. Subproses 9 (Packing produk 2)

Subproses ini berfungsi melakukan perlakuan berupa packing pada produk 2 (benda kerja 2 yang sudah melewati proses drilling). Berikut merupakan langkah kerja pada subproses 9:

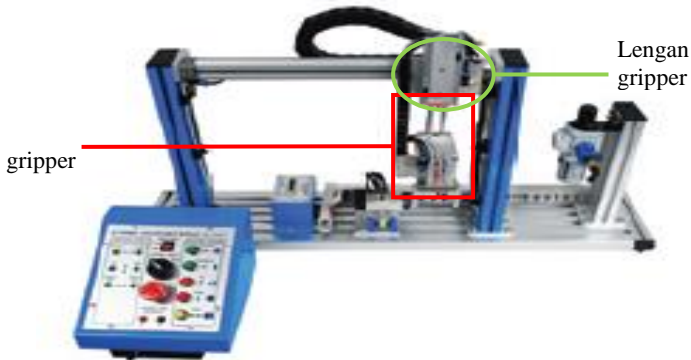
- Produk 2 yang sudah dijepit gripper, maka akan dibawa lengan gripper bergerak naik
- Lengan gripper yang sudah bergerak naik sepenuhnya, selanjutnya akan bergerak forward
- Ketika lengan gripper tepat mencapai setengah jalan maka lengan gripper akan turun.
- Lengan gripper yang sudah bergerak turun sepenuhnya maka menyebabkan gripper membuka sepenuhnya dan meletakkan produk 2 pada *packing box 2*
- Setelah produk 2 sudah ditempatkan di *packing box 2*, maka lengan gripper akan bergerak hingga forward sepenuhnya.

4.. Subproses 10 (Line Movement Stand by)

Subproses ini berfungsi memposisikan line movement module dalam keadaan *stand by* untuk menerima produk selanjutnya. Berikut merupakan langkah kerja pada subproses 10:

- Ketika lengan gripper dalam keadaan turun dan gripper terbuka sepenuhnya, maka lengan gripper akan naik sepenuhnya

Komponen yang berperan pada subproses 8 ditunjukkan Gambar 3. 6



Gambar 3. 6 komponen yang berperan dalam subproses 7,8,9,10 [1]

3.1.2 Input/Output Sistem

Pada tugas akhir ini, perancangan sistem kerja pada plant *Factory Automatic Trainer* menggunakan sebanyak 23 *input* serta 20 *output*. Tabel 3. 1 menunjukkan tabel yang berisi informasi terkait *input* yang digunakan pada sistem. Pendataan *input* dan *output* bertujuan untuk mempermudah designer dalam mengingat penamaan tiap *input* dan *output* dan juga fungsi dari masing-masingnya.

Tabel 3. 1 Tabel data *Input*

NO	NAMA	KETERANGAN	FUNGSI
1	START	tombol mulai	untuk memulai seluruh proses
2	STOP	tombol berhenti	untuk menghentikan seluruh proses
3	MAG	sensor magazine	pendeteksi keberadaan benda saat berada di magazine
4	SINSERT_E	<i>Limit switch</i> insert ketika eject	menunjukkan bahwa cylinder insert telah eject sepenuhnya
5	CAP	capacitive sensor	mendeteksi keberadaan benda kerja
6	PHOTO	proximity sensor	mendeteksi benda kerja berdasarkan warna
7	PROX	photo sensor	mendeteksi benda kerja berdasarkan bahan

8	END1	Photo fiber sensor 1	mendeteksi benda kerja yang sudah mencapai ujung subproses separation module
9	SUP	<i>Limit switch up</i>	menunjukkan bahwa lengan robot telah mencapai gerak naik sepenuhnya
10	SCW	<i>Limit switch clockwise</i>	menunjukkan bahwa lengan robot telah mencapai gerak searah jarum jam sepenuhnya
11	SDOWN	<i>Limit switch down</i>	menunjukkan bahwa lengan robot telah mencapai gerak turun sepenuhnya
12	SCCW	<i>Limit switch counter clockwise</i>	menunjukkan bahwa lengan robot telah mencapai gerak melawan arah jarum jam sepenuhnya
13	WORK	Optical sensor	Pendeteksi kehadiran bendakerja yang siap drilling
14	SDRILL_D	<i>Limit switch drill down</i>	Menunjukkan bahwa driller telah mencapai gerakan turun sepenuhnya
15	SSTOPPER_U	<i>Limit switch stopper up</i>	Menunjukkan bahwa stopper telah mencapai gerak naik sepenuhnya
16	END2	Photo Fiber sensor 2	Pendeteksi benda yang sudah mencapai ujung proses stopper
17	SUP2	<i>Limit switch UP2</i>	Menunjukkan bahwa lengan gripper telah mencapai gerak naik sepenuhnya
18	SBACKWARD	<i>Limit switch backward</i>	Menunjukkan bahwa lengan gripper telah mencapai gerak backward sepenuhnya
19	SDOWN2	<i>Limit switch DOWN2</i>	Menunjukkan bahwa lengan gripper telah mencapai gerak turun sepenuhnya
20	SGRIP	<i>Limit switch Grip</i>	Menunjukkan bahwa gripper dalam keadaan menutup
21	SFORWARD	<i>Limit switch forward</i>	Menunjukkan bahwa lengan gripper telah mencapai gerak forward sepenuhnya

22	SOPEN	Limit switch grip saat terbuka	Menunjukkan bahwa gripper dalam keadaan terbuka sepenuhnya
----	-------	--------------------------------	--

Sedangkan Tabel 3. 2 menunjukkan tabel yang berisi data *output* yang dipergunakan pada sistem. Tabel berisi nama *input*, keterangan *input* serta fungsi dari *output* tersebut.

Tabel 3. 2 Tabel data *Output*

NO	NAMA	KETERANGAN	FUNGSI
1	INSERT_E	Insert cylinder eject	Mendorong benda kerja dari magazine ke conveyor 1
2	CONV1	Conveyor 1	Menyalurkan benda kerja hingga ujung separation module
3	SEP_E	Separation Eject	Melakukan proses seleksi benda kerja dengan cara mendorongnya keluar dari conveyor 1
4	INSERT_R	Insert cylinder return	Menarik cylinder insert kembali
5	SEP_R	Separation return	Menarik cylinder separation kembali
6	UP	Gerak naik vertical cylinder 1	Memerintahkan lengan robot melakukan gerak naik
7	CW	Gerak clockwise	Memerintahkan lengan robot melakukan gerak searah jarum jam
8	DOWN	Gerak turun vertical cylinder 2	Memerintahkan lengan robot melakukan gerak turun
9	VACUUM	Vacuum pad	Mengaktifkan vacuum pad
10	CCW	Gerak Counter clockwise	Memerintahkan lengan robot untuk bergerak melawan arah jarum jam

11	CONV2	Conveyor 2	Mengaktifkan conveyor 2 untuk membawa benda kerja selama proses berjalan
12	DRILL_D	Drill down	Mengaktifkan driller untuk bergerak turun
13	DRILL	Proses drilling	Melakukan proses drilling
14	DRILL_U	Drill up	Mengaktifkan driller untuk bergerak naik
15	STOPPER_U	Stopper up	Mengaktifkan stopper untuk bergerak naik
16	UP2	Gerak naik	Mengaktifkan lengan gripper bergerak naik
17	BACKWARD	Gerak backward	Mengaktifkan lengan gripper bergerak backward (menuju stopper module)
18	DOWN2	Gerak turun	Mengaktifkan gerak turun pada lengan gripper
19	GRIP	Gripper	Mengaktifkan gripper untuk terbuka sepenuhnya
20	FORWARD	Gerak forward	Mengaktifkan gerakan forward pada lengan gripper

Berikut pada Tabel 3. 3 merupakan tabel *timer* yang dipergunakan pada sistem *Factory Automatic Trainer*. *Timer* dipergunakan untuk berbagai macam fungsi. Hal tersebut dapat dilihat pada tabel data *timer*.

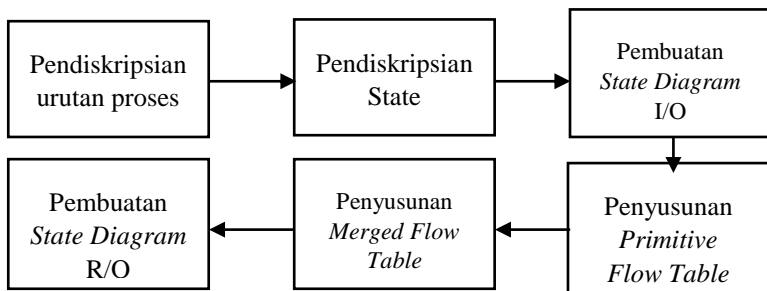
Tabel 3. 3 Tabel data *timer* yang dipergunakan pada sistem

NO	NAMA	KETERANGAN	FUNGSI
1	T1	<i>Timer 1</i>	Menghitung selama 3 detik apabila <i>input</i> CAP aktif, PHOTO aktif, PROX aktif

2	T2	<i>Timer 2</i>	Menghitung selama 0.1 detik apabila <i>input</i> END1 aktif, dan DOWN aktif
3	T3	<i>Timer 3</i>	Menghitung selama 0.1 detik apabila <i>input</i> SDOWN aktif dan SCCW aktif
4	T4	<i>Timer 4</i>	Menghitung selama 3 detik apabila <i>input</i> SDRILL_D aktif
5	T5	<i>Timer 5</i>	Menghitung selama 0.5 detik apabila <i>input</i> SUP2 aktif, SBACKWARD aktif, dan M2 aktif
6	T6	<i>Timer 6</i>	Menghitung selama 0.55 detik apabila <i>input</i> SUP2 SOPEN aktif

3.2 Perancangan *State Diagram*

Sesuai dengan yang sudah dijelaskan pada bab sebelumnya bahwa terdapat 6 langkah untuk merancang *State Diagram*. Hal ini bisa dilihat pada Gambar 3. 7.



Gambar 3. 7 langkah perancangan state diagram

3.2.1 Pendeskripsian Urutan Proses

Langkah pertama pada metode *State Diagram* adalah membuat tabel deskripsi state. Pada tabel dijelaskan bagaimana kondisi *input* dan *output* pada tiap state. Dalam penelitian ini dirancang sebanyak 40 state yang terbagi dalam 10 subproses. Dalam penyusunan urutan proses ini, ditambahkan. Pelabelan state tiap subproses dimulai dari state ke 1 hingga state ke -n. Tabel 3. 4 menunjukkan Tabel Deskripsi urutan proses pada Sistem *Factory Automatic Trainer*.

Tabel 3. 4 Tabel Deskripsi Urutan Proses

SUB PROSES	STATE	KONDISI <i>INPUT</i>	KONDISI <i>OUTPUT</i>
Memasukkan benda kerja	1	START on MAG on	INSERT_E on, UP on
	2	SINSERT_E on	CONV1 on,
	3	CAP on	CONV1 off, INSERT_E off
Deteksi benda kerja 3 (silver)	1	CAP on, PHOTO on PROX on	SEP_E on, T1 on, INSERT_R on, CONV1 on
	2	T1 contact on	SEP_R on, INSERT_E on
	3	SINSERT_E	CONV1 off, SEP_R off, INSERT_E off
Deteksi benda kerja 2 (biru)	1	CAP on, PHOTO on, PROX off	INSERT_R on, CONV1 on
	2	END1 on	INSERT_R off, CONV1 off,
Deteksi benda hitam 1 (hitam)	1	CAP on PHOTO off, PROX off	INSERT_R on, CONV1 on
	2	END1 on	INSERT_R off, CONV1 off
Pick and place	1	END1 on SCCW on	INSERT_E on, CW on
	2	END1 on, SCW on	DOWN on
	3	END1 on, SDOWN on	VACUUM on, T2 on

	4	END1 on,SDOWN on, T2 contact on	UP on, VACUUM on
	5	END 1 off, SUP on, SCW on	CCW on, VACUUM on
	6	SCCW on	DOWN on, VACUUM on,
	7	SDOWN on	VACUUM off
Stop dan drilling	1	SDOWN on SCCW on	T3 on, UP on
	2	T3 contact on	CONV2 on
	3	WORK on	CONV2 off, DRILL_D on
	4	SDRILL_D on	DRILL on, T4 on
	5	T4 contact on	DRILL on, DRILL_U on, STOPPER_U on, CONV2 on
	6	SSTOPPER on	DRILL off, CONV2 on, STOPPER_U on
Stop and drilling	7	END2 on	CONV2 off, STOPPER_U off
Gripping	1	END2 on	BACKWARD on
	2	SBACKWARD on	DOWN2 on
	3	SBACKWARD on SDOWN2 on	GRIP on
	4	SGRIP on	GRIP off
Packing produk 1	1	SGRIP on BACKWARD on M1 on	UP2 on, GRIP on,
	2	SUP2 on BACKWARD on M1 on	FORWARD on GRIP on
	3	SFORWARD on M1 on	DOWN2 on GRIP on
	4	SDOWN2 on	GRIP off
Packing produk 2	1	SGRIP on SBACKWARD on M2 on	UP2 on, GRIP on

	2	SUP2 on SBACKWARD on M2 on	FORWARD on, T5 on, GRIP on
	3	Contact T5 on	DOWN2 on, GRIP on
	4	SDOWN2 on	GRIP off,
Line Movement Stand by	1	SDOWN2 on	Grip off
	2	SOPEN on SDOWN2 on	UP2 on,
	3	SUP2 on, SOPEN on	T6 on UP2 on
	4	Contact t6 on	UP2 off

3.2.2 Pendiskripsian *Input dan Output*

Berikut merupakan tabel pendiskripsian *input* dan *output* sistem yang ditunjukkan oleh Tabel 3. 5.

Tabel 3. 5 Pendiskripsian *Input dan Output*

NO	INPUT		OUTPUT	
	NAMA	SIMBOL	NAMA	SIMBOL
1.	START	X ₁	INSERT_E	Z ₁
2.	MAG	X ₂	CONV1	Z ₂
3.	SINSERT_E	X ₃	SEP_E	Z ₃
4.	CAP	X ₄	INSERT_R	Z ₄
5.	PHOTO	X ₅	SEP_R	Z ₅
6.	PROX	X ₆	UP	Z ₆
7.	END1	X ₇	CW	Z ₇
8.	SUP	X ₈	DOWN	Z ₈
9.	SCW	X ₉	VACUUM	Z ₉
10.	SDOWN	X ₁₀	CCW	Z ₁₀
11.	SCCW	X ₁₁	CONV2	Z ₁₁
12.	WORK	X ₁₂	DRILL_D	Z ₁₂
13.	SDRILL_D	X ₁₃	DRILL	Z ₁₃
14.	SSTOPPER_U	X ₁₄	DRILL_U	Z ₁₄
15.	END2	X ₁₅	STOPPER_U	Z ₁₅
16.	SUP2	X ₁₆	UP2	Z ₁₆

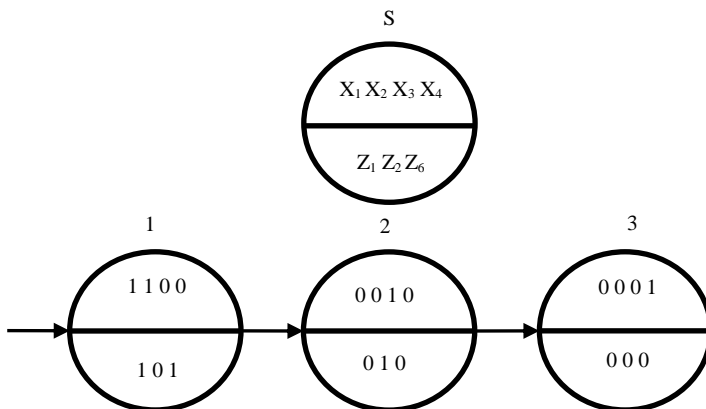
17.	SBACKWARD	X ₁₇	BACKWARD	Z ₁₇
18.	SDOWN2	X ₁₈	DOWN2	Z ₁₈
19.	SGRIP	X ₁₉	GRIP	Z ₁₉
20.	SFORWARD	X ₂₀	FORWARD	Z ₂₀
21.	SOPEN	X ₂₁	T1 contact	Z ₂₁
22.	T1	X ₂₂	T2 contact	Z ₂₂
23.	T2	X ₂₃	T3 contact	Z ₂₃
24.	T3	X ₂₄	T4 contact	Z ₂₄
25.	T4	X ₂₅	T5 contact	Z ₂₅
26.	T5	X ₂₆	T6 contact	Z ₂₆
27.	T6	X ₂₇	-	-
28.	STOP	X ₂₈	-	-
29.	M1	X ₂₉	-	-
30.	M2	X ₃₀	-	-

3.2.3 Pembuatan *State Diagram I/O (Input/Output)*

Urutan proses *Factory Automatic Trainer* yang sudah dibagi menjadi 10 subproses selanjutnya akan dibuat *State Diagram* berdasarkan I/O. *State Diagram* untuk subproses 1 ditunjukkan oleh Gambar 3. 8 yang dibuat berdasarkan tabel deskripsi urutan subproses 1 yang ditunjukkan Tabel 3. 6. Setiap 1 state pada *Table* deskripsi digambarkan menjadi 1 lingkaran state dan dihubungkan oleh anak panah. Di atas tiap lingkaran state terdapat nomor urut state. Pada bagian atas lingkaran diisi kondisi *input* dan bagian bawah kondisi *output*. Kondisi *input* atau *output* yang aktif diwakilkan bit “1” sedangkan yang tidak aktif diwakilkan bit “0”. *State diagram I/O* untuk seluruh subproses dapat dilihat pada Lampiran.

Tabel 3. 6 Tabel deskripsi urutan subproses 1 (memasukkan benda kerja)

SUB PROSES	STATE	KONDISI <i>INPUT</i>	KONDISI <i>OUTPUT</i>
Memasukkan benda kerja	1	- START on (X ₁) - MAG on (X ₂)	- INSERT_E on, (Z ₁) - UP on (Z ₆)
	2	- SINSERT_E on (X ₃)	- CONV1 on (Z ₂)
	3	CAP on (X ₄)	- CONV1 off, (Z ₂) - INSERT_E off (Z ₁)



Gambar 3. 8 State Diagram I/O sub proses 1

3.2.4 Pembuatan *Primitive Flow Table*

Informasi yang terdapat pada *State Diagram I/O* setiap subproses, akan dipergunakan untuk menyusun *primitive Flow Table*. Tabel 3. 7 merupakan *primitive flow Table* dari subproses 1. Primitive flow table untuk seluruh subproses terdapat pada Lampiran.

Tabel 3. 7 *Primitive Flow Table* Subproses 1

<i>Primitive Flow Table</i> Subproses 1						
Row	Input (X ₁ X ₂ X ₃ X ₄)			Output (Z ₁ Z ₂ Z ₆)		
	1100	0010	0001	Z ₁	Z ₂	Z ₆
1	1	2	-	1	0	1
2	-	2	3	0	1	0
3	-	-	3	0	0	0

3.2.5 Pembuatan *Merged Flow Table*

Tahapan selanjutnya adalah pembuatan *merged flow Table* untuk setiap subproses. Tabel 3. 8 menunjukkan *merged flow Table* dari subproses 1. Karena tidak ada baris yang memiliki kombinasi bit *output* yang sama maka tidak ada penggabungan baris pada *merged flow Table* dari subproses 1. Penentuan kolom *relay* didasarkan pada rumus $2^n \geq m$. Dimana n merupakan jumlah *relay*, 2^n merupakan jumlah kombinasi bit *relay* yang harus dibuat, dan m merupakan jumlah baris. Diketahui bahwa terdapat 3 baris, sehingga diperlukan jumlah kombinasi bit *relay* yang lebih dari atau sama dengan 3.

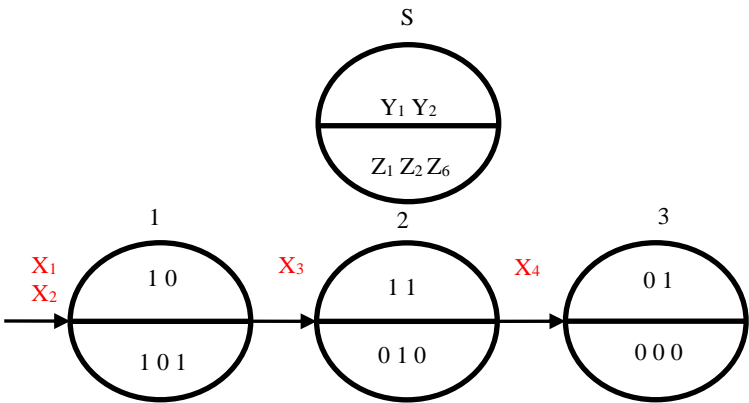
Sehingga didapatkan $n = 2$ dan $2^n = 4$. Jadi jumlah *relay* yang harus dibuat adalah 2. *Relay* tersebut diberikan nama Y_1 dan Y_2 . Sedangkan penyusunan untuk kombinasi bit *relay* dapat disusun hingga 4 kemungkinan, namun karena memang jumlah baris hanya 3 maka hanya 3 kombinasi bit saja yang dipakai. *Merged flow table* untuk seluruh subproses terdapat pada Lampiran.

Tabel 3. 8 *Merged Flow Table* Subproses 1

<i>Merged Flow Table</i> Subproses 1								
Row	Input (X ₁ X ₂ X ₃ X ₄)			Output (Z ₁ Z ₂ Z ₆)			Relay (Y ₁ Y ₂)	
	1100	0010	0001	Z ₁	Z ₂	Z ₆	Y ₁	Y ₂
1	1	2	-	1	0	1	1	0
2	-	2	3	0	1	0	1	1
3	-	-	3	0	0	0	0	1

3.2.6 Pembuatan *State Diagram* R/O (*Relay/Output*)

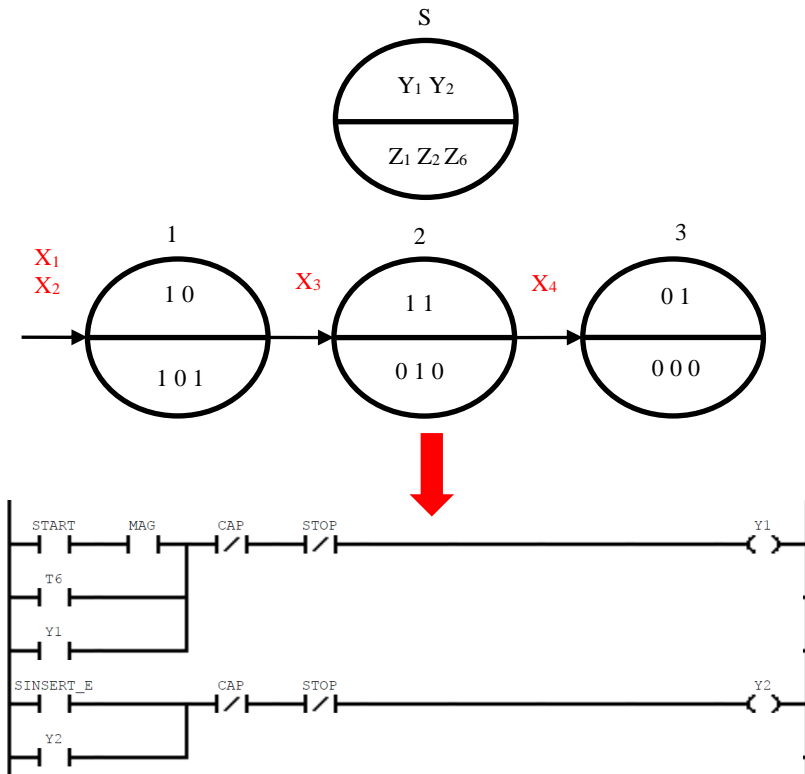
Tahapan terakhir adalah pembuatan *State Diagram* R/O yang disusun untuk setiap subproses. *State Diagram* R/O disusun berdasarkan informasi yang terdapat pada *merged flow Table*.Bagian atas lingkaran diisi dengan informasi berupa kombinasi bit *relay*. Gambar 3. 9 menunjukkan *State Diagram* R/O pada subproses 1.State Digram R/O untuk seluruh subproses dapat dilihat pada Lampiran.



Gambar 3. 9 *State Diagram* R/O untuk subproses 1

3.3 Konstruksi *Ladder diagram*

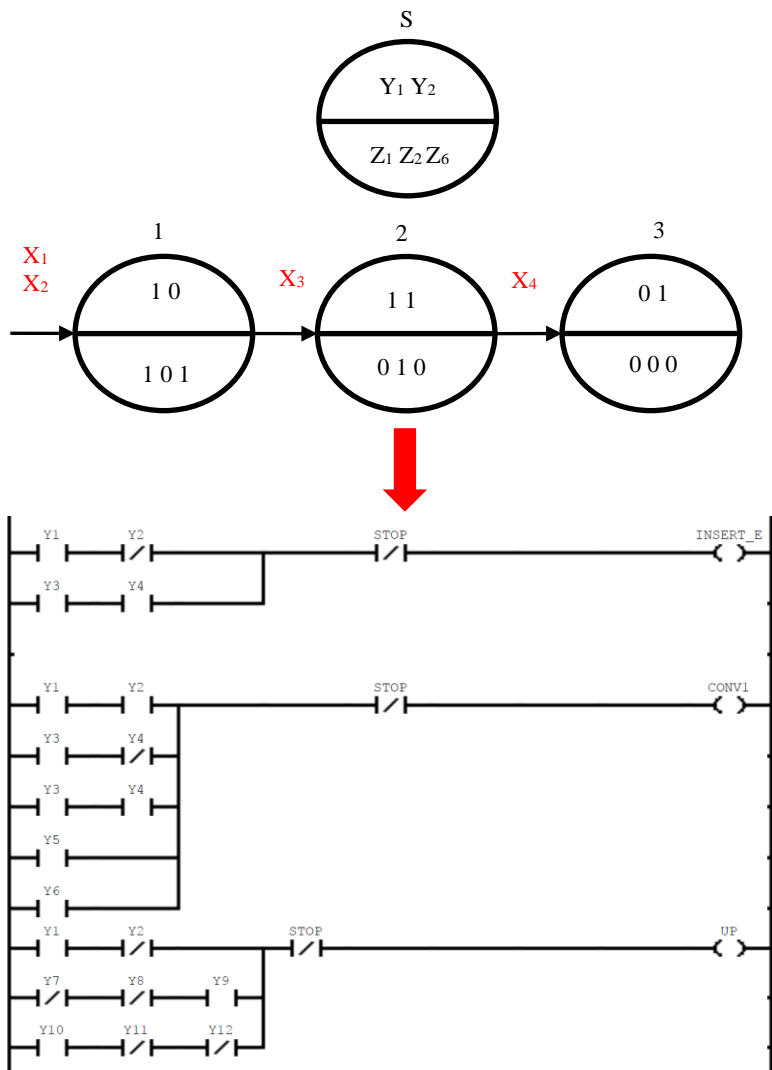
State Diagram R/O yang telah dibuat untuk tiap subproses harus dikonversikan ke dalam *ladder diagram*. Gambar 3. 10 dan Gambar 3. 11 menunjukkan konversi *State Diagram* R/O ke *Ladder diagram* untuk subproses 1. Hasil konstruksi *ladder diagram* untuk seluruh subproses dapat dilihat pada Lampiran.



Gambar 3. 10 konversi *State Diagram* ke *ladder diagram* supproses 1

Kondisi set terjadi saat state 0 menuju state 1, dan kondisi reset terjadi saat state 2 menuju state 3. Sehingga *Input* yang berperan dalam kondisi set diwakilkan dengan contact normally open, sedangkan *input* yang berperan dalam kondisi reset akan diwakilkan contact normally close. Gambar 3. 10 menunjukkan *relay* Y₁ dan Y₂ hasil konversi *State*

Diagram R/O subproses 1 menjadi *ladder diagram*. Sedangkan Gambar 3. 11 menunjukkan rung *output* hasil konversi *State Diagram* ke *ladder diagram*. State Diagram R/O untuk seluruh subproses dapat dilihat di Lampiran.



Gambar 3. 11 Konversi *State Diagram* ke *ladder diagram* subproses 1

[Halaman ini sengaja dikosongkan]

BAB 4

PENGUJIAN DAN ANALISA

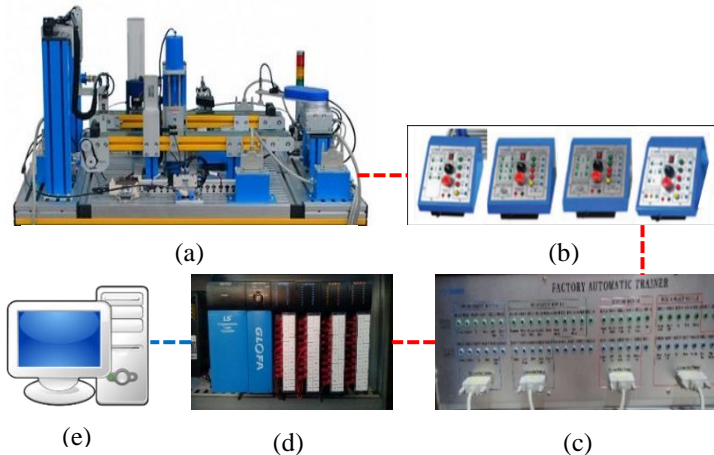
Pada bab ini akan dijelaskan langkah-langkah pengujian sistem, hasil uji sistem, serta analisa berdasarkan hasil uji sistem yang sudah dilakukan.

4.1 Pengujian Sistem

Pengujian sistem merupakan serangkaian kegiatan yang bertujuan untuk membuktikan apakah hasil perancangan sudah sesuai urutan proses yang diharapkan ketika diimplementasikan ke plant *Factory Automatic Trainer*. Dalam pengujian sistem dilakukan langkah-langkah sebagai berikut:

4.1.1 Pengkabelan

Untuk dapat melakukan pengujian sistem peneliti harus melakukan pengkabelan atau wiring pada komponen yang terdapat pada *Factory Automatic Trainer* terlebih dahulu. Skema pengkabelan dapat dilihat pada Gambar 4. 1



Keterangan:

- (a) Plant module
- (b) Control Panel
- (c) Main Panel
- (d) PLC
- (e) Computer

Gambar 4. 1 wiring pada *Factory Automatic Trainer*

Sesuai Gambar 4. 1 komponen *Factory Automatic Trainer* yaitu plant module, control panel, main panel, dan PLC LG GLOFA seri GM4 dihubungkan oleh D-SUB 25p connector. Sedangkan antara PLC dengan PC dihubungkan dengan RS-232connector. Dengan skema pengkabelan tersebut operator tidak perlu lagi melakukan pengkabelan dari *panel control* ke *main panel* secara manual per point menggunakan kabel. Sehingga operator hanya perlu membuat program dengan pengalaman *input/output* yang sesuai pada main panel.

4.1.2 Pengalaman *Input/Output*

Tabel 4. 1 menunjukkan pengalaman pada *input* serta *output* pada *Factory Automatic Trainer*. Didapatkan hasil berupa 22 *input* dan 20 *output* yang dihubungkan pada plant.

Tabel 4. 1 Pengalaman *input* dan *output*

DIGITAL INPUT		DIGITAL OUTPUT	
%IX0.0.0	SFORWARD	%QX0.2.0	FORWARD
%IX0.1.1	SBACKWARD	%QX0.2.1	BACKWARD
%IX0.1.2	SUP2	%QX0.2.2	UP2
%IX0.1.3	SDOWN2	%QX0.2.3	DOWN2
%IX0.1.4	SOPEN	%QX0.2.4	GRIP
%IX0.1.5	SGRIP	%QX0.2.5	-
%IX0.1.6	MAG	%QX0.2.6	INSERT_E
%IX0.1.7	PHOTO	%QX0.2.7	INSERT_R
%IX0.1.8	PROX	%QX0.2.8	SEP_E
%IX0.1.9	CAP	%QX0.2.9	SEP_R
%IX0.1.10	SINSERT_E	%QX0.2.10	CONV1
%IX0.0.14	END1	%QX0.3.0	STOPPER UP
%IX0.1.0	SSTOPPER_U	%QX0.3.1	DRILL_U
%IX0.1.1	START	%QX0.3.2	DRILL_D
%IX0.1.2	WORK	%QX0.3.3	DRILL
%IX0.1.3	STOP	%QX0.3.4	CONV2
%IX0.1.4	SDRILL_D	%QX0.3.5	-
%IX0.1.5	END2	%QX0.3.6	CCW
%IX0.1.6	SCCW	%QX0.3.7	CW
%IX0.1.7	SCW	%QX0.3.8	UP
%IX0.1.8	SUP	%QX0.3.9	DOWN
%IX0.1.9	SDOWN	%QX0.3.10	VACUUM

4.1.3 Implementasi Konstruksi *Ladder diagram* Pada FAT

Setelah dilakukan wiring dan pengalamatan *input/output* maka tahap selanjutnya adalah melakukan implementasi program *ladder diagram*. Dalam pengujian sistem yang pertama ini, terdapat beberapa langkah – langkah yang harus dilakukan. Berikut merupakan tahapan pada implementasi program *ladder diagram* :

1. Menyiapkan program yang telah didesain sebelumnya di GMWIN
2. Memastikan kabel *power* sudah terpasang
3. Mengatur *circuit breaker* dalam keadaan on
4. Mengatur *control select switch* pada keadaan auto
5. Menghubungkan PC dengan PLC menggunakan *connector* RS-232
6. Memastikan *running mode* dari CPU pada keadaan *PAUSE*
7. Melakukan *Compile* serta *connect* program PLC melalui GMWIN
8. Apabila lampu indikator hijau sudah menyala, atur running mode dari CPU menjadi keadaan *RUN*
9. Mengeksekusi program melalui GMWIN
10. Melakukan pengecekan secara langsung apakah plant sudah berjalan sesuai program yang dirancang

Berikut merupakan contoh implementasi *ladder diagram* pada subproses 1 :

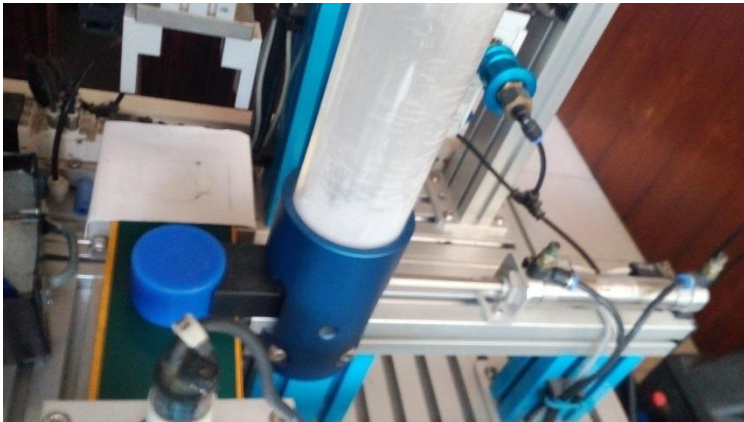
Berdasarkan urutan subproses 1 pada **Error! Reference source not found.**

Tabel 4. 2 maka akan diimplementasi *ladder diagram* yang ditunjukkan oleh

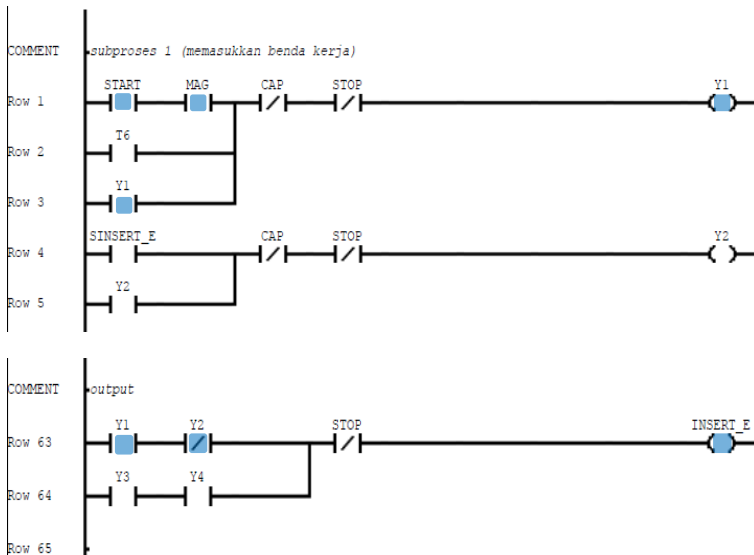
Tabel 4. 2 Urutan Subproses 1

SUB PROSES	STATE	KONDISI <i>INPUT</i>	KONDISI <i>OUTPUT</i>
Memasukkan benda kerja	1	- START on (X ₁) - MAG on (X ₂)	- INSERT_E on, (Z ₁) - UP on (Z ₆)
	2	- SINSERT_E on (X ₃)	- CONV1 on (Z ₂)
	3	CAP on (X ₄)	- CONV1 off, (Z ₂) - INSERT_E off (Z ₁)

Pada Gambar 4. 2 dapat dilihat bahwa insert cylinder melakukan gerakan eject yang akan mendorong benda kerja masuk ke conveyor 1

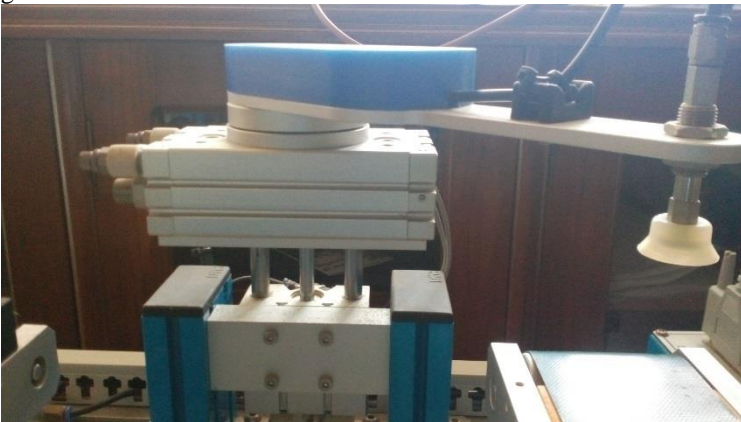


Gambar 4. 2 cylinder insert melakukan gerak eject

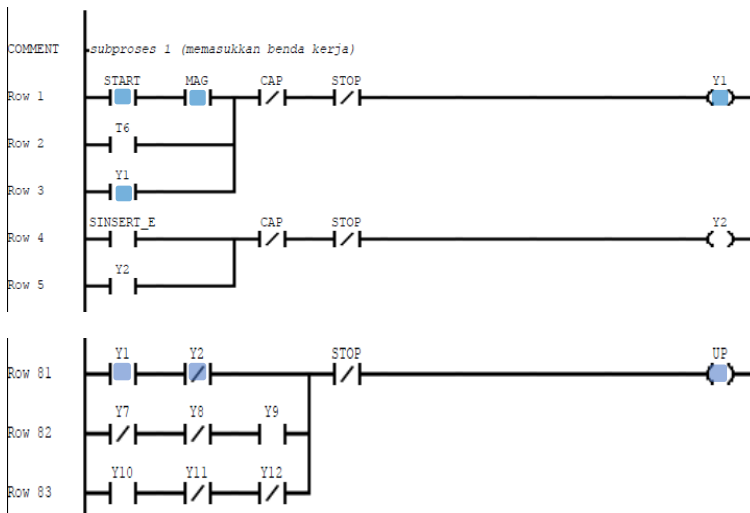


Gambar 4. 3 implementasi *ladder diagram* untuk insert bergerak eject

Pada Gambar 4. 4 menunjukkan lengan robot sedang melakukan gerakan UP

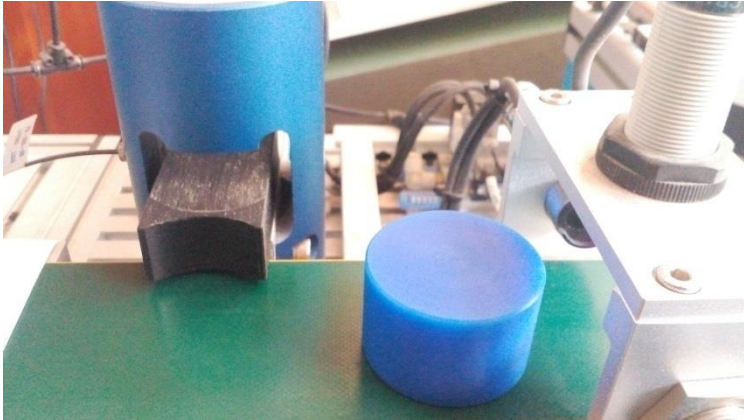


Gambar 4. 4 lengan robot bergerak UP

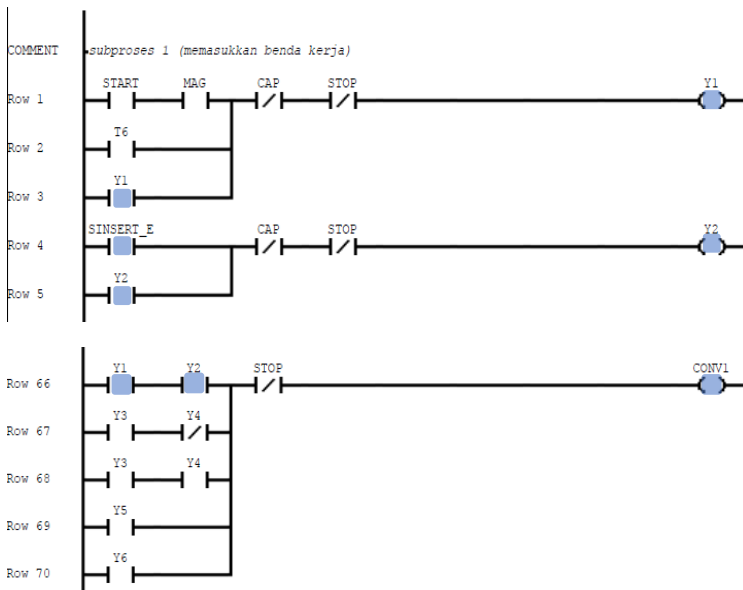


Gambar 4. 5 Implementasi *Ladder diagram* untuk lengan robot bergerak UP

Gambar 4. 6 menunjukkan conveyor 1 yang aktif pada subproses 1. Sehingga benda kerja bergerak menuju arah pick and place module.



Gambar 4. 6 conveyor 1 aktif



Gambar 4. 7 Implementasi *ladder diagram* untuk conveyor 1 aktif

Pengecekan dilakukan dengan cara memastikan hasil simulasi dan implementasi sesuai seperti pada subproses 1. Cara yang sama dilakukan untuk subproses 2 hingga 10.

4.1.4. Pencatatan waktu proses

Dalam industri, efisiensi waktu produksi mempunyai peranan penting. Oleh karena itu dibutuhkan informasi awal terkait berapa waktu proses produksi. Pada penelitian ini produk 1 dan produk 2 akan dicatat waktu prosesnya dengan cara pengambilan data masing-masing sebanyak 10 siklus.

4.2 Hasil Uji Sistem

Hasil uji sistem terdiri atas hasil implementasi konstruksi *ladder diagram* dan hasil uji waktu proses. Berikut merupakan penguraian dari hasil uji sistem:

4.2.1 Implementasi dari konstruksi *ladder diagram*

Implementasi konstruksi *ladder diagram* yang telah dirancang pada *Factory Automatic Trainer* sesuai dengan urutan proses. Konstruksi *ladder diagram* menghasilkan total 49 rung. Jumlah *timer* yang dipergunakan adalah 6 *timer*. Sedangkan jumlah *relay* yang diperlukan adalah sebanyak 23 *relay*. Tabel 4. 3 menunjukkan jumlah *relay* yang dibutuhkan tiap subproses.

Tabel 4. 3 Jumlah *relay* yang dibutuhkan tiap subproses

No	SUB PROSES	Jumlah Relay
1	Insert benda	2 Relay
2	Deteksi benda kerja 1	2 Relay
3	Deteksi benda kerja 2	1 Relay
4	Deteksi benda kerja 3	1 Relay
5	Pick and Place	3 Relay
6	Stopper & drilling	3 Relay
7	Line Movement	2 relay
8.	Packing produk 1	3 relay
9.	Packing produk 2	3 relay
10	Line movement Standby	2 relay

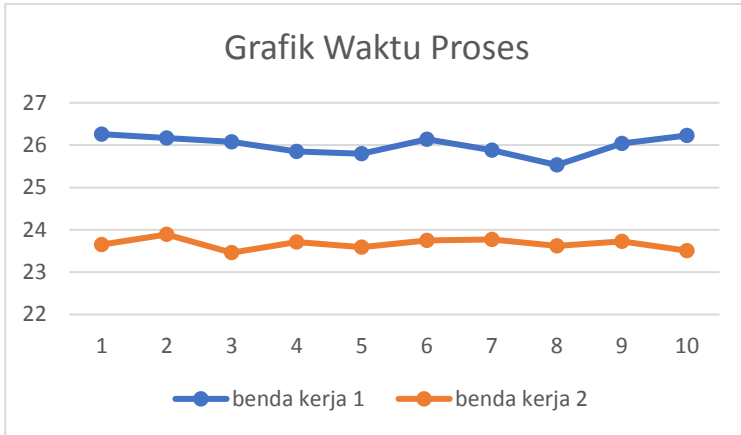
4.2.2 Waktu proses

Tabel 4. 4 menunjukkan waktu proses dari benda kerja 1 dan benda kerja 2. Pengambilan data dilakukan sebanyak 10 siklus.

Tabel 4. 4 Waktu proses

Pengambilan data ke-	Waktu Proses (sekon)	
	Benda kerja 1	Benda kerja 2
1	26.26	23.65
2	26.17	23.89
3	26.08	23.46
4	25.85	23.71
5	25.80	23.59
6	26.14	23.75
7	25.88	23.77
8	25.53	23.62
9	26.04	23.73
10	26.23	23.51

Dari pengambilan data untuk waktu proses benda kerja 1, tidak terjadi perbedaan yang signifikan pada waktu proses dari pengambilan data ke 1 hingga pengambilan data ke 10. Hal yang sama juga terjadi pada pengambilan data waktu proses untuk benda kerja 2. Hal ini menunjukkan bahwa kinerja sensor dan aktuator pada plant masih berfungsi sebagaimana mestinya meskipun diambil pendataan hingga 10 siklus pemrosesan. Hal yang turut andil dalam stabilnya waktu proses hingga pengambilan data ke 10 adalah terjaganya tekanan udara pada compressor, sehingga tidak menghambat pergerakan dari aktuator-aktuator yang terdapat pada *Factory Automatic Trainer*.



Gambar 4. 8 Grafik Waktu proses benda kerja 1 dan benda kerja 2

Gambar 4. 8 merupakan grafik waktu proses benda kerja 1 dan benda kerja 2. Dari data diatas didapatkan rata-rata waktu pemrosesan benda kerja 1 sebesar 25.998 detik sedangkan rata-rata untuk pemrosesan benda kerja 2 sebesar 23.668 detik. Dari data rata-rata waktu proses didapatkan waktu proses benda kerja 2 relatif lebih cepat dibandingkan waktu proses benda kerja 1. Hal ini dapat terjadi dikarenakan terjadi perbedaan perlakuan pada saat benda kerja masuk ke line movement module. Benda kerja 1 akan ditempatkan pada wadah *packing box* 1 yang terletak di ujung lintasan line movement. Sedangkan benda kerja 2 ditempatkan pada wadah *packing box* 2 yang terletak tepat di tengah lintasan line movement. Dengan kata lain jarak lintasan yang dilalui oleh benda kerja 1 dua kali lebih panjang daripada lintasan yang dilalui benda kerja 2. Sehingga waktu tempuh yang dibutuhkan oleh benda kerja 2 jauh lebih cepat daripada benda kerja 1.

[Halaman ini sengaja dikosongkan]

BAB 5

KESIMPULAN DAN SARAN

Pada bab ini berisi kesimpulan serta saran yang didapatkan setelah melakukan pembahasan pada penelitian ini.

5.1 Kesimpulan

Berdasarkan proses penelitian dengan judul Konstruksi *Ladder diagram* menggunakan Metode *State Diagram* pada *Factory Automatic Trainer* didapatkan kesimpulan sebagai berikut:

1. Konstruksi *Ladder diagram* menggunakan Metode *State Diagram* pada *Factory Automatic Trainer* menghasilkan total 49 Rung
2. Konstruksi *Ladder diagram* menggunakan Metode *State Diagram* pada *Factory Automatic Trainer* membutuhkan *relay* sebanyak 23 *relay* dan 6 *timer*
3. Konstruksi *Ladder diagram* menggunakan Metode *State Diagram* pada *Factory Automatic Trainer* menggunakan 22 *input* dan 20 *output*
4. Rata-rata waktu proses yang dibutuhkan untuk benda kerja 1 adalah 25.998 detik sedangkan untuk benda kerja 2 adalah 23.668 detik

5.2 Saran

Berikut merupakan saran yang diberikan untuk penelitian selanjutnya antara lain:

1. Sebaiknya mengembangkan sistem kerja untuk pemrosesan benda kerja yang kontinyu bukan lagi satu persatu
2. Sebaiknya dilakukan optimalisasi pada waktu pemrosesan benda kerja

[Halaman ini sengaja dikosongkan]

DAFTAR PUSTAKA

- [1] CPE-AT8030N, "Manual of *Factory Automatic Trainer*,"
CHUNGPA EMT CO. LTD.
- [2] 9th TOTAL CATALOGUE, "Manual of Sensor," Autonics.
- [3] W. Bolton, "Programmable Logic Controller 5th Edition",
London: Elsevier Newnes, 2009.
- [4] LG Industrial Systems, "LG GLOFA GM series, Handbook,"
LG Industrial Systems, Seoul-Korea, 2003.
- [5] D. Pessen, "Industrial Automation : Circuit Design and
Component", Israel: John Wiley & Sons, 1989.
- [6] Global Reader JRT, "Established FA (Factory Automation)
Pneumatic Research Institute," Korea, 1995.

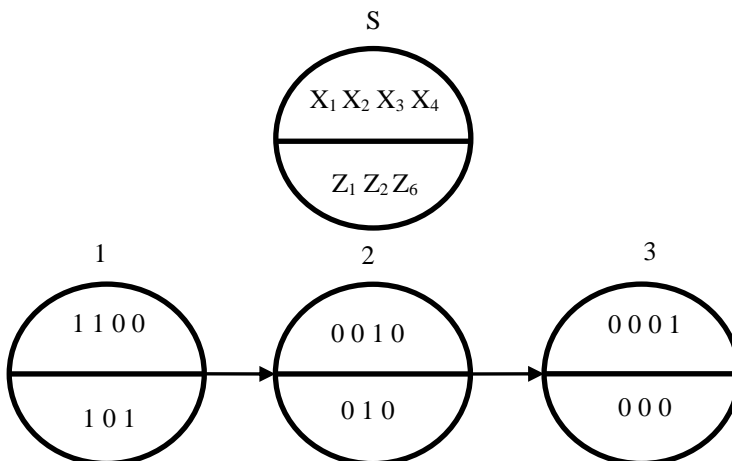
[Halaman ini sengaja dikosongkan]

LAMPIRAN

A. STATE DIAGRAM I/O UNTUK SELURUH SUBPROSES

Tabel A. 1 Deskripsi Urutan Subproses 1

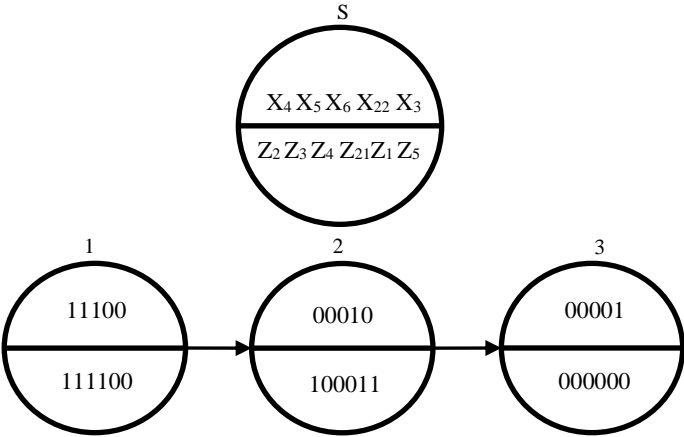
SUB PROSES	STATE	KONDISI INPUT	KONDISI OUTPUT
Memasukkan benda kerja	1	- START on (X ₁) - MAG on (X ₂)	- INSERT_E on, (Z ₁) - UP on (Z ₆)
	2	- SINSERT_E on (X ₃)	- CONV1 on (Z ₂)
	3	CAP on (X ₄)	- CONV1 off, (Z ₂) - INSERT_E off (Z ₁)



Gambar A. 1 State Diagram I/O Subproses 1

Tabel A. 2 Deskripsi Urutan Subproses 2

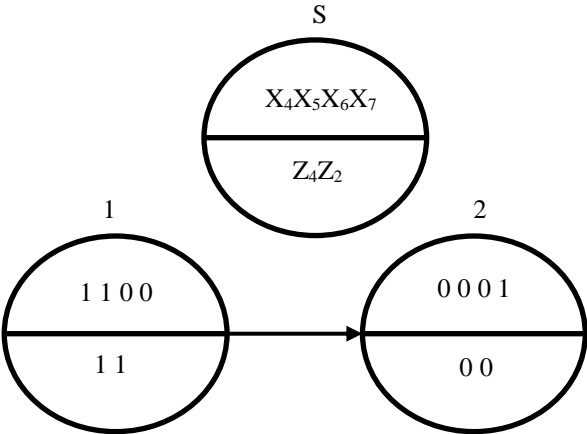
SUBPROSES	STATE	KONDISI INPUT	KONDISI OUTPUT
Deteksi benda kerja 3 (silver)	1	<ul style="list-style-type: none">• CAP (X₄) on,• PHOTO (X₅) on,• PROX (X₆) on	<ul style="list-style-type: none">• SEP_E (Z₃) on,• T1 (Z₂₁) on,• INSERT_R (Z₄) on,• CONV1 (Z₂) on
	2	<ul style="list-style-type: none">• T1 contact (X₂₂) on	<ul style="list-style-type: none">• SEP_R (Z₅) on,• CONV1 (Z₂) on• INSERT_E (Z₁) on
	3	<ul style="list-style-type: none">• SINSERT_E (X₃) on	<ul style="list-style-type: none">• CONV1 (Z₂) off,• SEP_R (Z₅) off,• INSERT_E (Z₁) off,• INSERT_R (Z₄) off,• SEP_E (Z₃) off,• T1 (Z₂₁) off



Gambar A. 2 State Diagram I/O Subproses 2

Tabel A. 3 Deskripsi Urutan Subproses 3

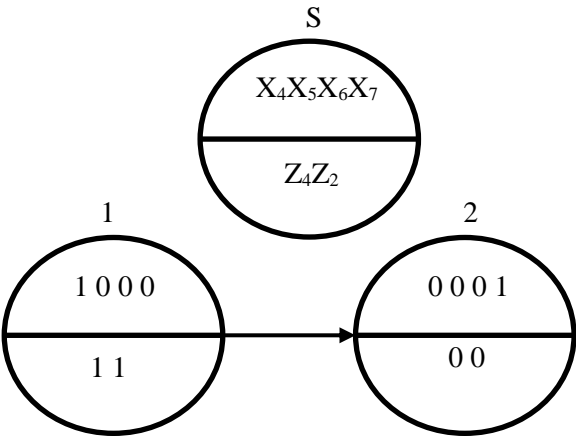
SUBPROSES	STATE	KONDISI INPUT	KONDISI OUTPUT
Deteksi benda kerja 2 (biru)	1	<ul style="list-style-type: none">• CAP (X4)on,• PHOTO (X5)on,• PROX (X6)off	<ul style="list-style-type: none">• INSERT_R (Z4) on,• CONV1 (Z2) on
	2	<ul style="list-style-type: none">• END1 (X7)on	<ul style="list-style-type: none">• INSERT_R (Z4) off,• CONV1 (Z2) off,



Gambar A. 3 State Diagram I/O Subproses 3

Tabel A. 4 Deskripsi Urutan Subproses 4

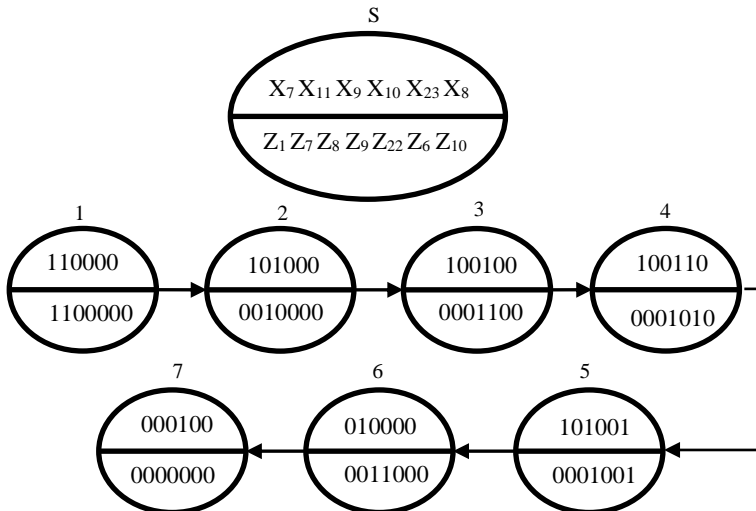
SUBPROSES	STATE	KONDISI INPUT	KONDISI OUTPUT
Deteksi benda kerja 1 (hitam)	1	<ul style="list-style-type: none">• CAP (X₄)on,• PHOTO (X₅)off,• PROX (X₆)off	<ul style="list-style-type: none">• INSERT_R (Z₄) on,• CONV1 (Z₂) on
	2	<ul style="list-style-type: none">• END1 (X₇)on	<ul style="list-style-type: none">• INSERT_R (Z₄) off,• CONV1 (Z₂) off,



Gambar A. 4 State Diagram I/O Subproses 4

Tabel A. 5 Deskripsi Urutan Subproses 5

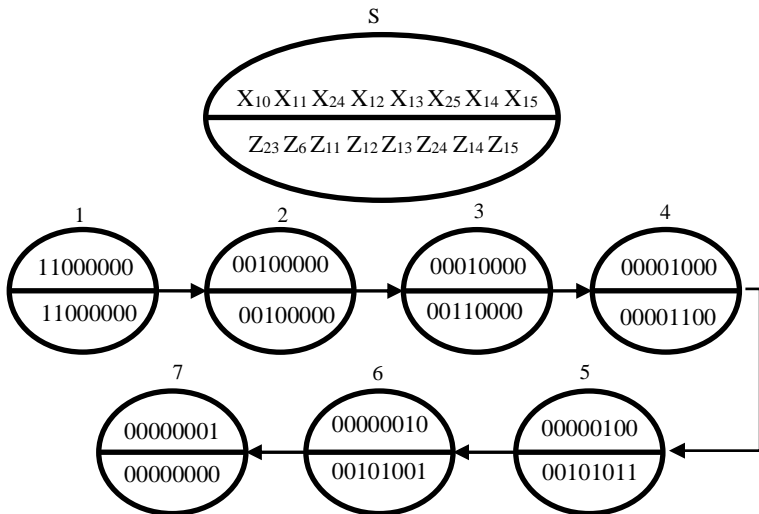
SUBPROSES	STATE	KONDISI INPUT	KONDISI OUTPUT
Pick and place	1	<ul style="list-style-type: none"> • END1 (X₇) on, • SCCW(X₁₁) on 	<ul style="list-style-type: none"> • INSERT _E (Z₁) on, • CW (Z₇) on
	2	<ul style="list-style-type: none"> • END1 (X₇) on, • SCW (X₉) on 	<ul style="list-style-type: none"> • DOWN (Z₈) on
	3	<ul style="list-style-type: none"> • END1 (X₇) on, • SDOWN (X₁₀) on 	<ul style="list-style-type: none"> • VACUUM (Z₉) on, • T2 (Z₂₂) on
	4	<ul style="list-style-type: none"> • END1 (X₇) on, • SDOWN (X₁₀) on, • T2 contact (X₂₃) on 	<ul style="list-style-type: none"> • UP (Z₆) on, • VACUUM (Z₉) on
	5	<ul style="list-style-type: none"> • END 1 (X₇) off, • SUP (X₈) on, • SCW (X₉) on 	<ul style="list-style-type: none"> • CCW (Z₁₀) on, • VACUUM (Z₉) on
	6	<ul style="list-style-type: none"> • SCCW (X₁₁) on 	<ul style="list-style-type: none"> • DOWN (Z₈) on, • VACUUM (Z₉) on,
	7	<ul style="list-style-type: none"> • SDOWN (X₁₀) on 	<ul style="list-style-type: none"> • VACUUM (Z₉) off



Gambar A. 5 State Diagram I/O Subproses 5

Tabel A. 6 Deskripsi Urutan Subproses 6

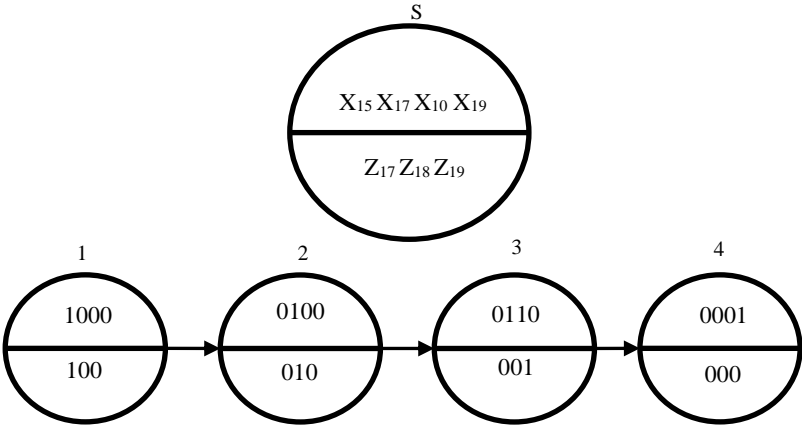
SUBPROSES	STATE	• KONDISI INPUT	• KONDISI OUTPUT
Stop dan drilling	1	• SDOWN (X ₁₀) on • SCCW (X ₁₁) on	• T3 (Z ₂₃) on, • UP (Z ₆) on
	2	• T3 contact (X ₂₄) on	• CONV2 (Z ₁₁) on
	3	• WORK (X ₁₂) on	• CONV2 (Z ₁₁) off, • DRILL_D (Z ₁₂) on
	4	• SDRILL_D (X ₁₃) on	• DRILL (Z ₁₃) on, • T4 (Z ₂₄) on
	5	• T4 contact (X ₂₅) on	• DRILL (Z ₁₃) on, • DRILL_U (Z ₁₄) on, • STOPPER_U (Z ₁₅) on, • CONV2 (Z ₁₁) on
	6	• SSTOPPER_U (X ₁₄) on	• DRILL (Z ₁₃) off, • CONV2 (Z ₁₁) on, • STOPPER_U (Z ₁₅) on
	7	• END2 (X ₁₅) on	• CONV2 (Z ₁₁) off, • STOPPER_U (Z ₁₅) off



Gambar A. 6 State Diagram I/O Subproses 6

Tabel A. 7 Deskripsi Urutan Subproses 7

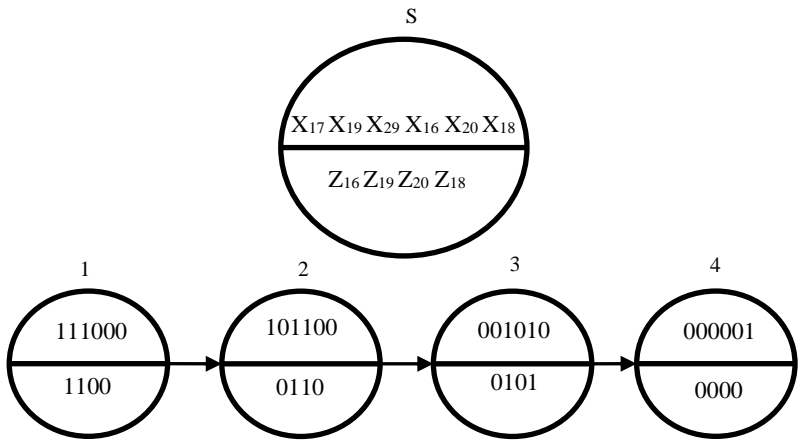
SUBPROSES	STATE	KONDISI INPUT	KONDISI OUTPUT
Gripping	1	• END2 (X ₁₅) on	• BACKWARD (Z ₁₇) on
	2	• SBACKWARD (X ₁₇) on	• DOWN2 (Z ₁₈) on
	3	• SBACKWARD (X ₁₇) on • SDOWN2 (X ₁₀) on	• GRIP (Z ₁₉) on
	4	• SGRIP (X ₁₉) on	• GRIP (Z ₁₉) off



Gambar A. 7 State Diagram I/O Subproses 7

Tabel A. 8 Deskripsi Urutan Subproses 8

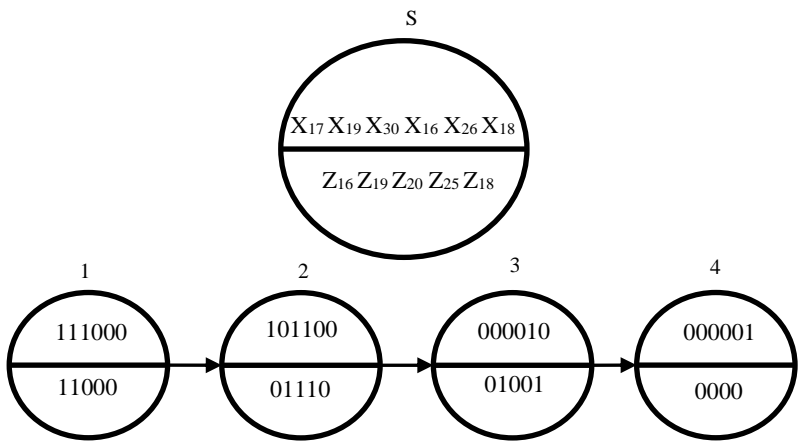
SUBPROSES	STATE	KONDISI INPUT	KONDISI OUTPUT
Packing produk 1	1	<ul style="list-style-type: none">• SGRIP (X₁₉) on,• SBACKWARD (X₁₇) on,• M1 (X₂₉) on	<ul style="list-style-type: none">• UP2 (Z₁₆) on,• GRIP (Z₁₉) on,
	2	<ul style="list-style-type: none">• SUP2 (X₁₆) on,• SBACKWARD (X₁₇) on,• M1 (X₂₉) on	<ul style="list-style-type: none">• FORWARD (Z₂₀) on,• GRIP (Z₁₉) on
	3	<ul style="list-style-type: none">• SFORWARD (X₂₀) on,• M1 (X₂₉) on	<ul style="list-style-type: none">• DOWN2 (Z₁₈) on,• GRIP (Z₁₉) on
	4	<ul style="list-style-type: none">• SDOWN2 (X₁₈) on	<ul style="list-style-type: none">• GRIP (Z₁₉) off



Gambar A. 8 State Diagram I/O Subproses 8

Tabel A. 9 Deskripsi Urutan Subproses 9

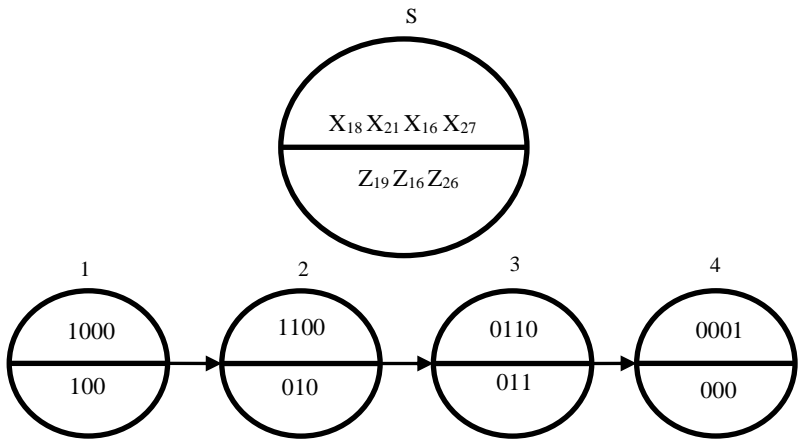
SUBPROSES	STATE	KONDISI <i>INPUT</i>	KONDISI <i>OUTPUT</i>
Packing produk 2	1	<ul style="list-style-type: none">• SGRIP (X₁₉) on• SBACKWARD (X₁₇) on• M2 (X₃₀) on	<ul style="list-style-type: none">• UP2 (Z₁₆) on,• GRIP (Z₁₉) on
	2	<ul style="list-style-type: none">• SUP2 (X₁₆) on• SBACKWARD (X₁₇) on• M2 (X₃₀) on	<ul style="list-style-type: none">• FORWARD (Z₂₀) on,• T5 (Z₂₅) on,• GRIP (Z₁₉) on
	3	<ul style="list-style-type: none">• T5 Contact (X₂₆) on	<ul style="list-style-type: none">• DOWN2 (Z₁₈) on,• GRIP (Z₁₉) on
	4	<ul style="list-style-type: none">• SDOWN2 (X₁₈) on	<ul style="list-style-type: none">• GRIP (Z₁₉) off



Gambar A. 9 State Diagram I/O Subproses 9

Tabel A. 10 Deskripsi Urutan Subproses 10

SUBPROSES	STATE	KONDISI INPUT	KONDISI OUTPUT
Line Movement Stand by	1	• SDOWN2 (X ₁₈) on	• Grip (Z ₁₉) off
	2	• SOPEN (X ₂₁) on • SDOWN2 (X ₁₈) on	• UP2 (Z ₁₆) on,
	3	• SUP2 (X ₁₆) on, • SOPEN (X ₂₁) on	• T6 (Z ₂₆) on • UP2 (Z ₁₆) on
	4	• T6 Contact (X ₂₇) on	• UP2 (Z ₁₆) off



Gambar A. 10 State Diagram I/O Subproses 10

B. PRIMITIVE FLOW TABLE UNTUK SELURUH SUBPROSES

Tabel B. 1 *Primitive Flow Table* Subproses 1

Primitive Flow Table Subproses 1						
Row	Input (X ₁ X ₂ X ₃ X ₄)			Output (Z ₁ Z ₂ Z ₆)		
	1100	0010	0001	Z ₁	Z ₂	Z ₆
1	1	2	-	1	0	1
2	-	2	3	0	1	0
3	-	-	3	0	0	0

Tabel B. 2 *Primitive Flow Table* Subproses 2

Primitive Flow Table Subproses 2									
Row	Input (X ₄ X ₅ X ₆ X ₂₂ X ₃)			Output (Z ₂ Z ₃ Z ₄ Z ₂₁ Z ₁ Z ₅)					
	11100	00010	00001	Z ₂	Z ₃	Z ₄	Z ₂₁	Z ₁	Z ₅
1	1	2	-	1	1	1	1	0	0
2	-	2	3	1	0	0	0	1	1
3	-	-	3	0	0	0	0	0	0

Tabel B. 3 *Primitive Flow Table* Subproses 3

Primitive Flow Table Subproses 3				
Row	Input (X ₄ X ₅ X ₆ X ₇)		Output (Z ₄ Z ₂)	
	1100	0001	Z ₄	Z ₂
1	1	2	1	1
2	-	2	0	0

Tabel B. 4 *Primitive Flow Table* Subproses 4

Primitive Flow Table Subproses 4				
Row	Input (X ₄ X ₅ X ₆ X ₇)		Output (Z ₄ Z ₂)	
	1000	0001	Z ₄	Z ₂
1	1	2	1	1
2	-	2	0	0

Tabel B. 5 *Primitive Flow Table* Subproses 5

Primitive Flow Table Subproses 5														
Row	Input (X ₇ X ₁₁ X ₉ X ₁₀ X ₂₃ X ₈)							Output (Z ₁ Z ₇ Z ₈ Z ₉ Z ₂₂ Z ₆ Z ₁₀)						
	110000	101000	100100	100110	101001	010000	000100	Z ₁	Z ₇	Z ₈	Z ₉	Z ₂₂	Z ₆	Z ₁₀
1	1	2	-	-	-	-	-	1	1	0	0	0	0	0
2	-	2	3	-	-	-	-	0	0	1	0	0	0	0
3	-	-	3	4	-	-	-	0	0	0	1	1	0	0
4	-	-	-	4	5	-	-	0	0	0	1	0	1	0
5	-	-	-	-	5	6	-	0	0	0	1	0	0	1
6	-	-	-	-	-	6	7	0	0	1	1	0	0	0
7	-	-	-	-	-	-	7	0	0	0	0	0	0	0

Tabel B. 6 *Primitive Flow Table* Subproses 6

Primitive Flow Table Subproses 6															
Row	Input							Output							
	(X ₁₀ X ₁₁ X ₂₄ X ₁₂ X ₁₃ X ₂₅ X ₁₄ X ₁₅)							(Z ₂₃ Z ₆ Z ₁₁ Z ₁₂ Z ₁₃ Z ₂₄ Z ₁₄ Z ₁₅)							
	11000000	00100000	00010000	00001000	00000100	00000010	00000001	Z ₂₃	Z ₆	Z ₁₁	Z ₁₂	Z ₁₃	Z ₂₄	Z ₁₄	Z ₁₅
1	1	2	-	-	-	-	-	1	1	0	0	0	0	0	0
2	-	2	3	-	-	-	-	0	0	1	0	0	0	0	0
3	-	-	3	4	-	-	-	0	0	0	1	1	0	0	0
4	-	-	-	4	5	-	-	0	0	0	0	1	1	0	0
5	-	-	-	-	5	6	-	0	0	1	0	1	0	1	1
6	-	-	-	-	-	6	7	0	0	1	0	1	0	0	1
7	-	-	-	-	-	-	7	0	0	0	0	0	0	0	0

Tabel B. 7 *Primitive Flow Table* Subproses 7

<i>Primitive Flow Table</i> Subproses 7							
Row	<i>Input</i> (X ₁₅ X ₁₇ X ₁₀ X ₁₉)				<i>Output</i> (Z ₁₇ Z ₁₈ Z ₁₉)		
	1000	0100	0110	0001	Z ₁₇	Z ₁₈	Z ₁₉
1	1	2	-	-	1	0	0
2	-	2	3	-	0	1	0
3	-	-	3	4	0	0	1
4	-	-	-	4	0	0	0

Tabel B. 8 *Primitive Flow Table* Subproses 8

Primitive Flow Table Subproses 8								
Row	Input (X ₁₇ X ₁₉ X ₂₉ X ₁₆ X ₂₀ X ₁₈)				Output (Z ₁₆ Z ₁₉ Z ₂₀ Z ₁₈)			
	111000	101100	001010	000001	Z₁₆	Z₁₉	Z₂₀	Z₁₈
1	1	2	-	-	1	1	0	0
2	-	2	3	-	0	1	1	0
3	-	-	3	4	0	1	0	1
4	-	-	-	4	0	0	0	0

Tabel B. 9 *Primitive Flow Table* Subproses 9

Primitive Flow Table Subproses 9									
Row	Input (X ₁₇ X ₁₉ X ₃₀ X ₁₆ X ₂₆ X ₁₈)				Output (Z ₁₆ Z ₁₉ Z ₂₀ Z ₂₅ Z ₁₈)				
	111000	101100	000010	000001	Z₁₆	Z₁₉	Z₂₀	Z₂₅	Z₁₈
1	1	2	-	-	1	1	0	0	0
2	-	2	3	-	0	1	1	1	0
3	-	-	3	4	0	1	0	0	1
4	-	-	-	4	0	0	0	0	0

Tabel B. 10 *Primitive Flow Table* Subproses 10

Primitive Flow Table Subproses 10							
Row	Input (X ₁₈ X ₂₁ X ₁₆ X ₂₇)				Output (Z ₁₉ Z ₁₆ Z ₂₆)		
	1000	1100	0110	0001	Z₁₉	Z₁₆	Z₂₆
1	1	2	-	-	1	0	0
2	-	2	3	-	0	1	0
3	-	-	3	4	0	1	1
4	-	-	-	4	0	0	0

[Halaman ini sengaja dikosongkan]

C. MERGED *FLOW TABLE* UNTUK SELURUH SUBPROSES

Table C. 1 Merged *Flow Table* Subproses 1

Merged <i>Flow Table</i> Subproses 1								
Row	Input (X ₁ X ₂ X ₃ X ₄)			Output (Z ₁ Z ₂ Z ₆)			Relay (Y ₁ Y ₂)	
	1100	0010	0001	Z ₁	Z ₂	Z ₆	Y ₁	Y ₂
1	1	2	-	1	0	1	1	0
2	-	2	3	0	1	0	1	1
3	-	-	3	0	0	0	0	1

Table C. 2 Merged *Flow Table* Subproses 2

Merged <i>Flow Table</i> Subproses 2											
Row	Input (X ₄ X ₅ X ₆ X ₂₂ X ₃)			Output (Z ₂ Z ₃ Z ₄ Z ₂₁ Z ₁ Z ₅)						Relay (Y ₃ Y ₄)	
	11100	00010	00001	Z ₂	Z ₃	Z ₄	Z ₂₁	Z ₁	Z ₅	Y ₃	Y ₄
1	1	2	-	1	1	1	1	0	0	1	0
2	-	2	3	1	0	0	0	1	1	1	1
3	-	-	3	0	0	0	0	0	0	0	1

Table C. 3 Merged *Flow Table* Subproses 3

Merged <i>Flow Table</i> Subproses 3					
Row	Input (X ₄ X ₅ X ₆ X ₇)		Output (Z ₄ Z ₂)		Relay (Y ₅)
	1100	0001	Z ₄	Z ₂	Y ₅
1	1	2	1	1	1
2	-	2	0	0	0

Table C. 4 Merged *Flow Table* Subproses 4

Merged <i>Flow Table</i> Subproses 4					
Row	Input (X ₄ X ₅ X ₆ X ₇)		Output (Z ₄ Z ₂)		Relay (Y ₆)
	1000	0001	Z ₄	Z ₂	Y ₆
1	1	2	1	1	1
2	-	2	0	0	0

Table C. 5 Merged Flow Table Subproses 5

Merged Flow Table Subproses 5																	
Row	Input (X ₇ X ₁₁ X ₉ X ₁₀ X ₂₃ X ₈)							Output (Z ₁ Z ₇ Z ₈ Z ₉ Z ₂₂ Z ₆ Z ₁₀)							Relay (Y ₇ Y ₈ Y ₉)		
	110000	101000	100100	100110	101001	010000	000100	Z ₁	Z ₇	Z ₈	Z ₉	Z ₂₂	Z ₆	Z ₁₀	Y ₇	Y ₈	Y ₉
1	1	2	-	-	-	-	-	1	1	0	0	0	0	0	1	1	0
2	-	2	3	-	-	-	-	0	0	1	0	0	0	0	1	1	1
3	-	-	3	4	-	-	-	0	0	0	1	1	0	0	1	0	1
4	-	-	-	4	5	-	-	0	0	0	1	0	1	0	0	0	1
5	-	-	-	-	5	6	-	0	0	0	1	0	0	1	0	1	1
6	-	-	-	-	-	6	7	0	0	1	1	0	0	0	0	1	0
7	-	-	-	-	-	-	7	0	0	0	0	0	0	0	0	0	0

Table C. 6 Merged Flow Table Subproses 6

Merged Flow Table Subproses 6																		
Row	Input (X ₁₀ X ₁₁ X ₂₄ X ₁₂ X ₁₃ X ₂₅ X ₁₄ X ₁₅)							Output (Z ₂₃ Z ₆ Z ₁₁ Z ₁₂ Z ₁₃ Z ₂₄ Z ₁₄ Z ₁₅)								Relay (Y ₁₀ Y ₁₁ Y ₁₂)		
	11000000	00100000	00010000	00001000	00000100	00000010	00000001	Z ₂₃	Z ₆	Z ₁₁	Z ₁₂	Z ₁₃	Z ₂₄	Z ₁₄	Z ₁₅	Y ₁₀	Y ₁₁	Y ₁₂
1	1	2	-	-	-	-	-	1	1	0	0	0	0	0	0	1	0	0
2	-	2	3	-	-	-	-	0	0	1	0	0	0	0	0	1	1	0
3	-	-	3	4	-	-	-	0	0	0	1	1	0	0	0	1	1	1
4	-	-	-	4	5	-	-	0	0	0	0	1	1	0	0	1	0	1
5	-	-	-	-	5	6	-	0	0	1	0	1	0	1	1	0	0	1
6	-	-	-	-	-	6	7	0	0	1	0	1	0	0	1	0	1	1
7	-	-	-	-	-	-	7	0	0	0	0	0	0	0	0	0	1	0

Table C. 7 Merged Flow Table Subproses 7

Merged Flow Table Subproses 7									
Row	Input (X ₁₅ X ₁₇ X ₁₀ X ₁₉)				Output (Z ₁₇ Z ₁₈ Z ₁₉)			Relay (Y ₁₃ Y ₁₄)	
	1000	0100	0110	0001	Z ₁₇	Z ₁₈	Z ₁₉	Y ₁₃	Y ₁₄
1	1	2	-	-	1	0	0	1	0
2	-	2	3	-	0	1	0	1	1
3	-	-	3	4	0	0	1	0	1
4	-	-	-	4	0	0	0	0	0

Table C. 8 Merged Flow Table Subprose 8

Merged Flow Table Subproses 8										
Row	Input (X ₁₇ X ₁₉ X ₂₉ X ₁₆ X ₂₀ X ₁₈)				Output (Z ₁₆ Z ₁₉ Z ₂₀ Z ₁₈)				Relay (Y ₁₅ Y ₁₆)	
	111000	101100	001010	000001	Z ₁₆	Z ₁₉	Z ₂₀	Z ₁₈	Y ₁₅	Y ₁₆
1	1	2	-	-	1	1	0	0	1	0
2	-	2	3	-	0	1	1	0	1	1
3	-	-	3	4	0	1	0	1	0	1
4	-	-	-	4	0	0	0	0	0	0

Table C. 9 Merged Flow Table Subproses 9

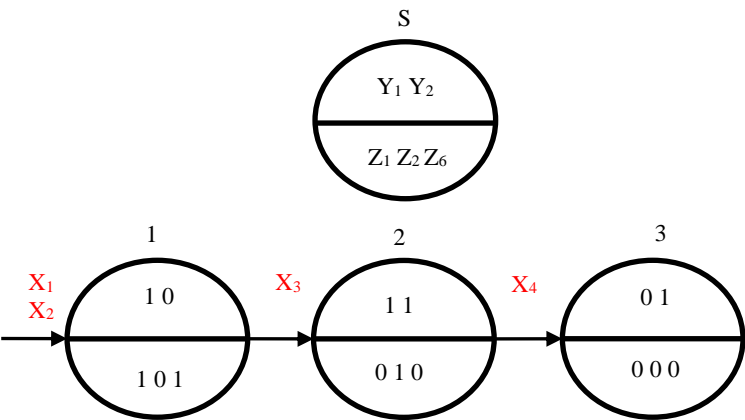
Merged Flow Table Subproses 9											
Row	Input (X ₁₇ X ₁₉ X ₃₀ X ₁₆ X ₂₆ X ₁₈)				Output (Z ₁₆ Z ₁₉ Z ₂₀ Z ₂₅ Z ₁₈)					Relay (Y ₁₇ Y ₁₈)	
	111000	101100	000010	000001	Z ₁₆	Z ₁₉	Z ₂₀	Z ₂₅	Z ₁₈	Y ₁₇	Y ₁₈
1	1	2	-	-	1	1	0	0	0	1	0
2	-	2	3	-	0	1	1	1	0	1	1
3	-	-	3	4	0	1	0	0	1	0	1
4	-	-	-	4	0	0	0	0	0	0	0

Table C. 10 Merged Flow Table Subproses 10

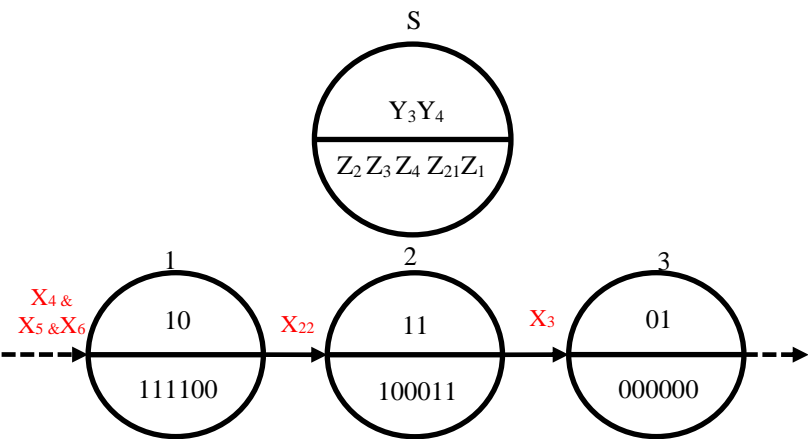
Merged Flow Table Subproses 10									
Row	Input (X ₁₈ X ₂₁ X ₁₆ X ₂₇)				Output (Z ₁₉ Z ₁₆ Z ₂₆)			Relay (Y ₁₉ Y ₂₀)	
	1000	1100	0110	0001	Z ₁₉	Z ₁₆	Z ₂₆	Y ₁₉	Y ₂₀
1	1	2	-	-	1	0	0	1	0
2	-	2	3	-	0	1	0	1	1
3	-	-	3	4	0	1	1	0	1
4	-	-	-	4	0	0	0	0	0

[Halaman ini sengaja dikosongkan]

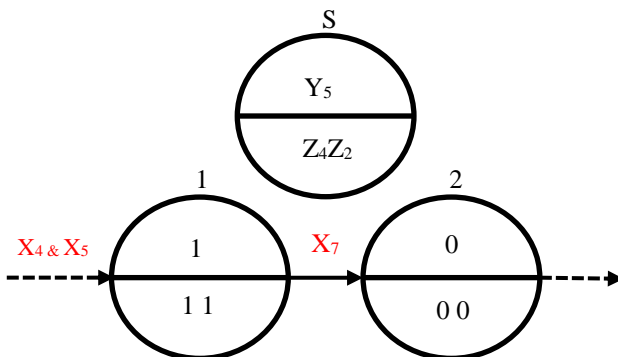
D. DIAGRAM R/O UNTUK SELURUH SUBPROSES



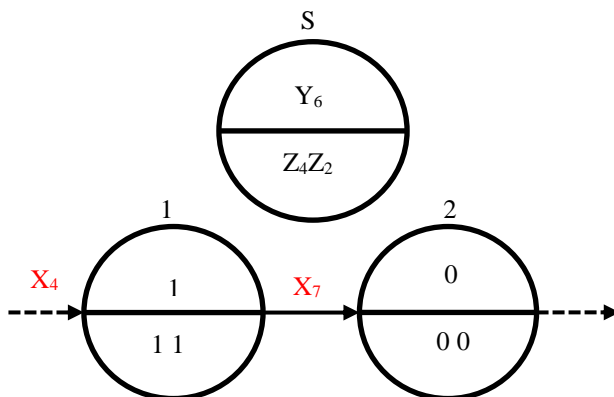
Gambar D. 1 Diagram R/O Subproses 1



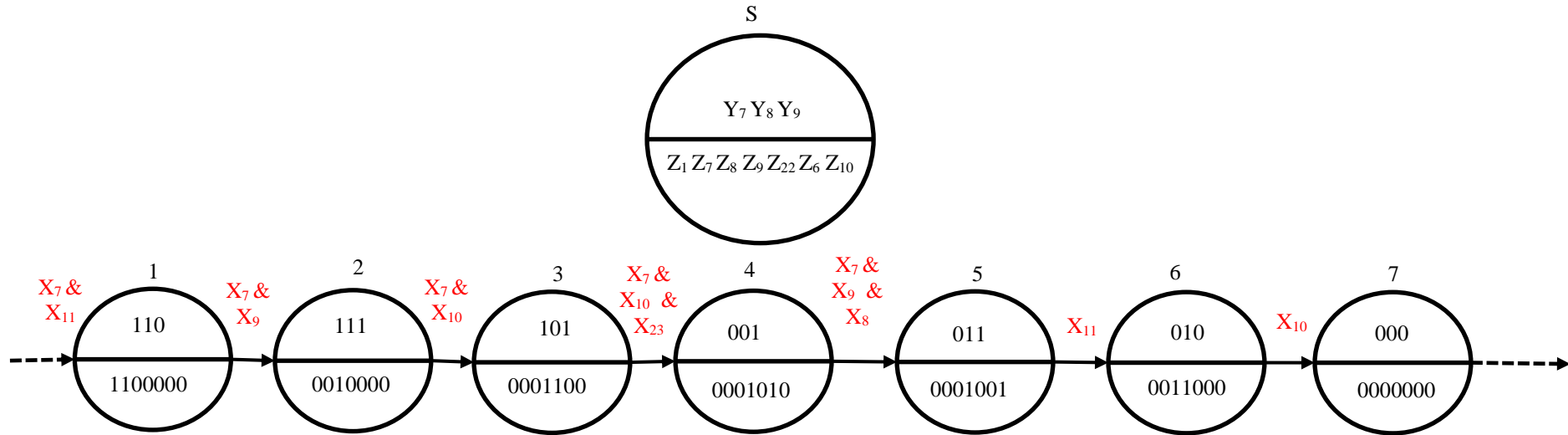
Gambar D. 2 Diagram R/O subproses 2



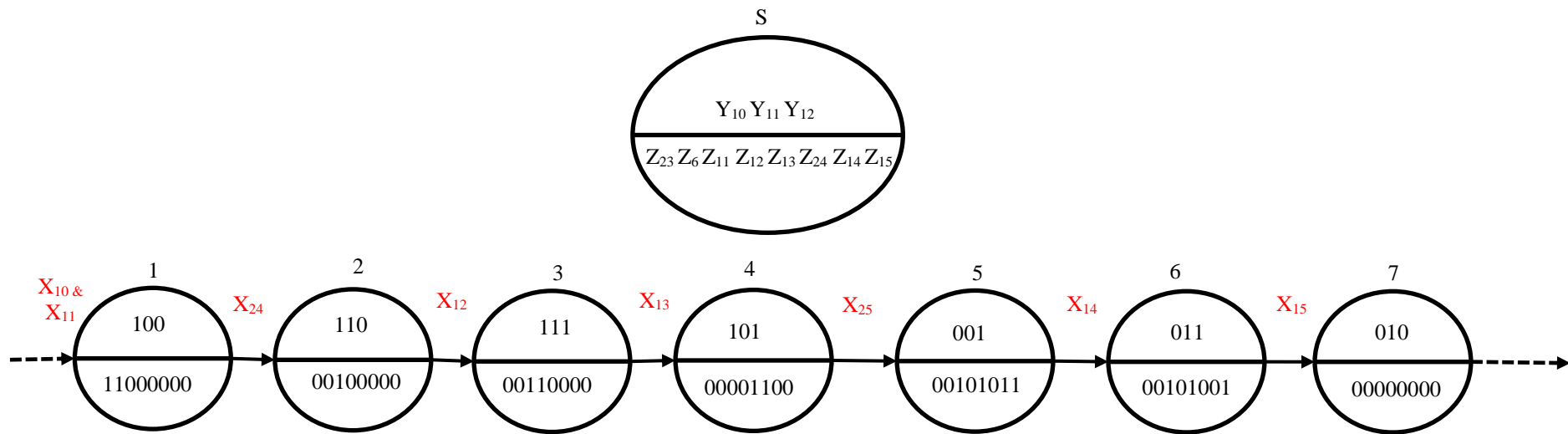
Gambar D. 3 Diagram R/O subproses 3



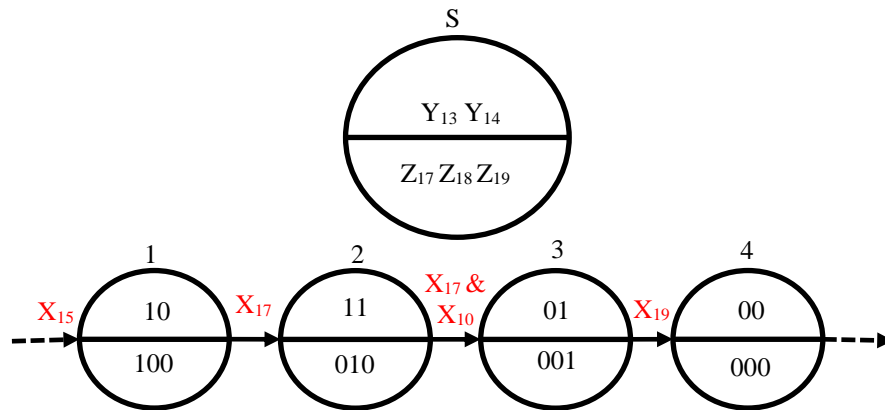
Gambar D. 4 Diagram R/O Subproses 4



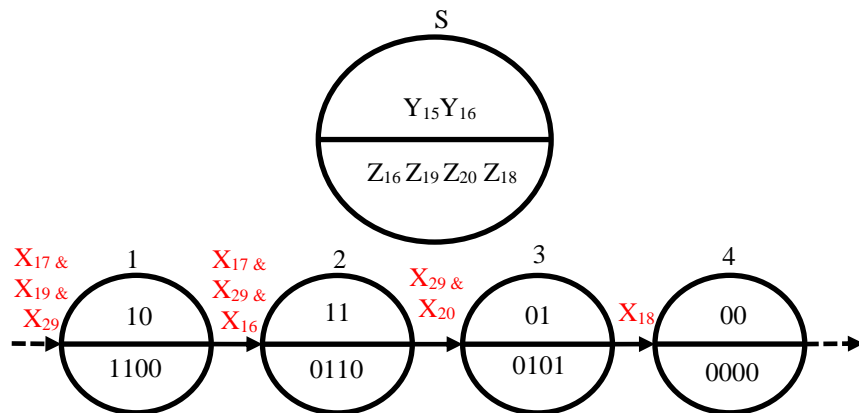
Gambar D. 5 State Diagram R/O Subproses 5



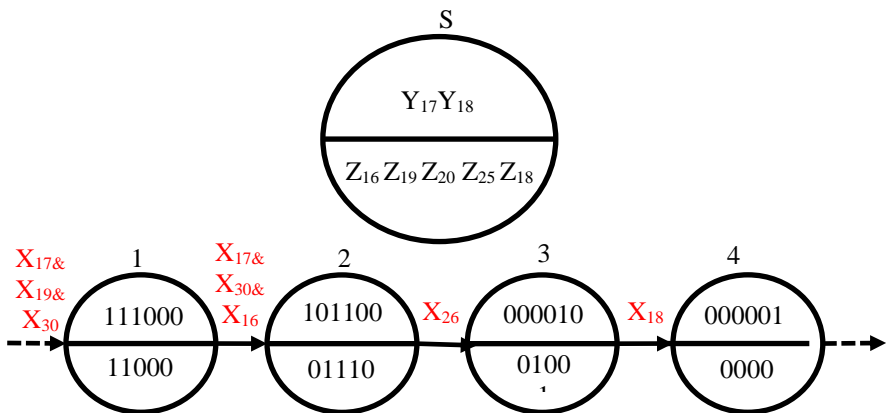
Gambar D. 6 State Diagram R/O Subproses 6



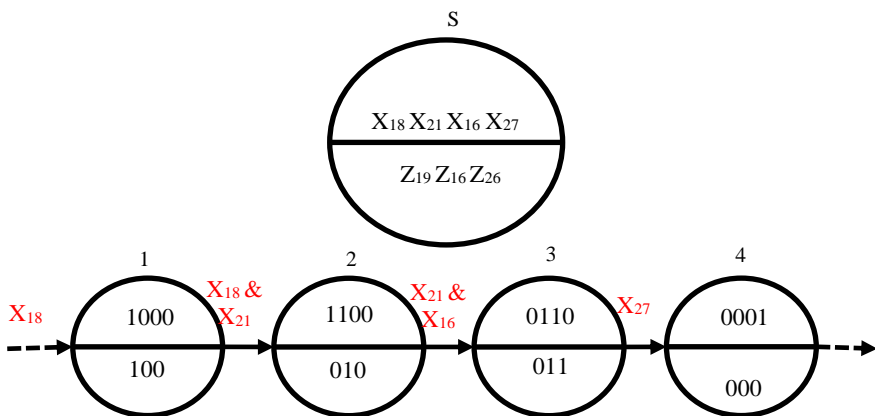
Gambar D. 7 State Diagram R/O Subproses 7



Gambar D. 8 State Diagram R/O Subproses 8



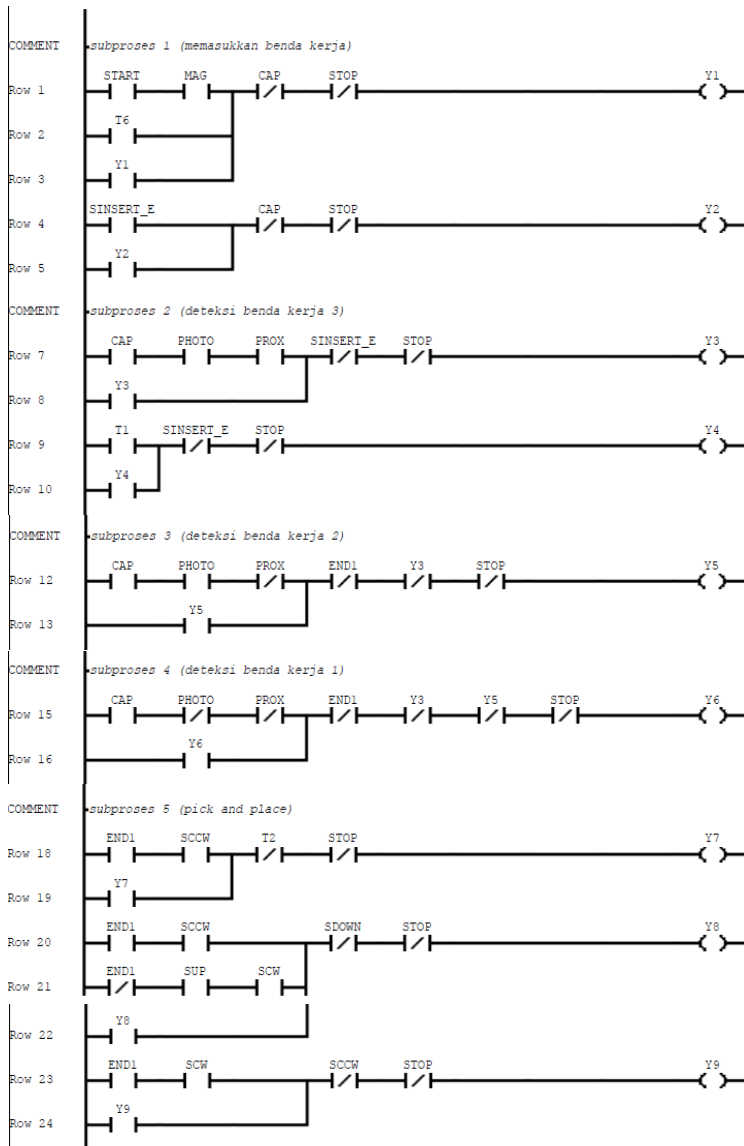
Gambar D. 9 State Diagram R/O Subproses 9

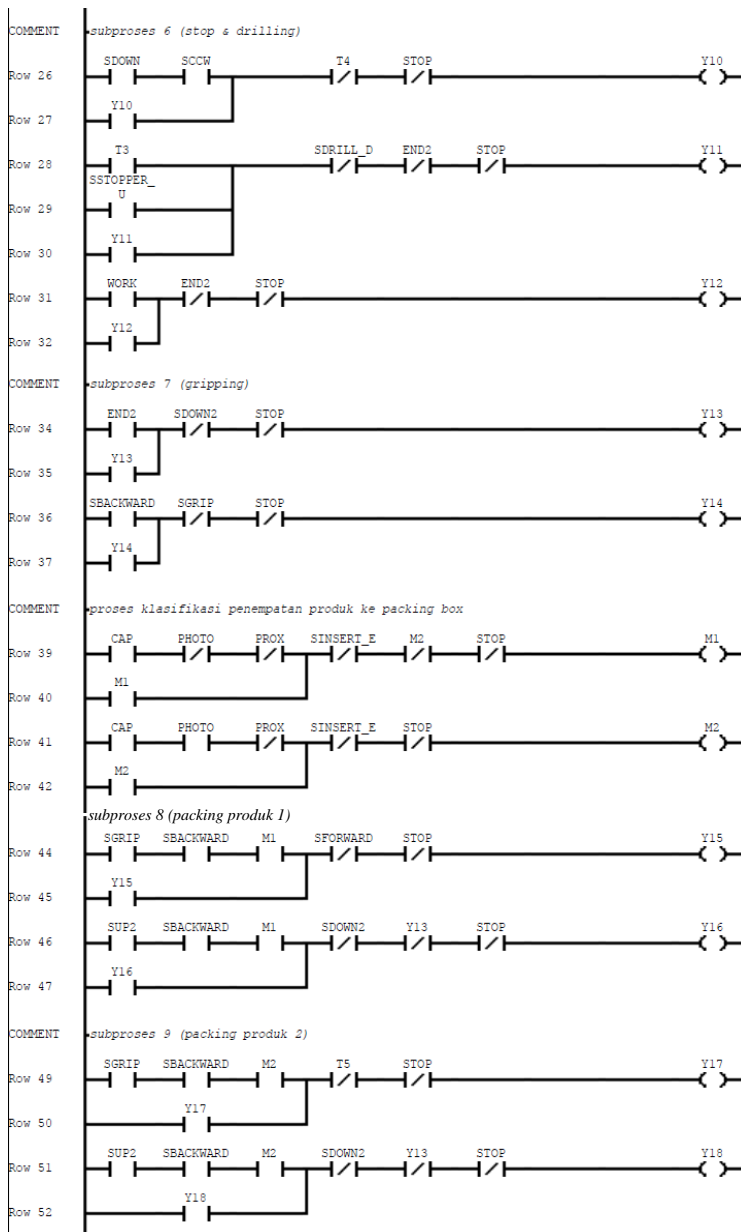


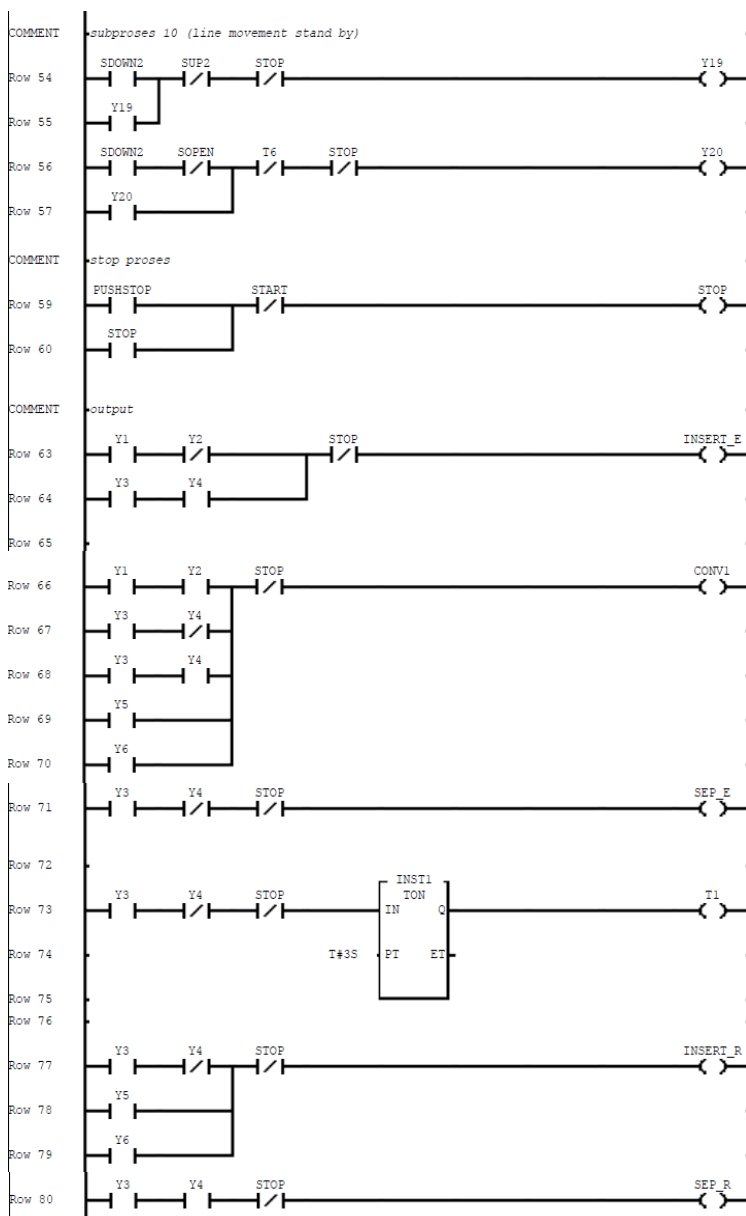
Gambar D. 10 State Diagram R/O Subproses 10

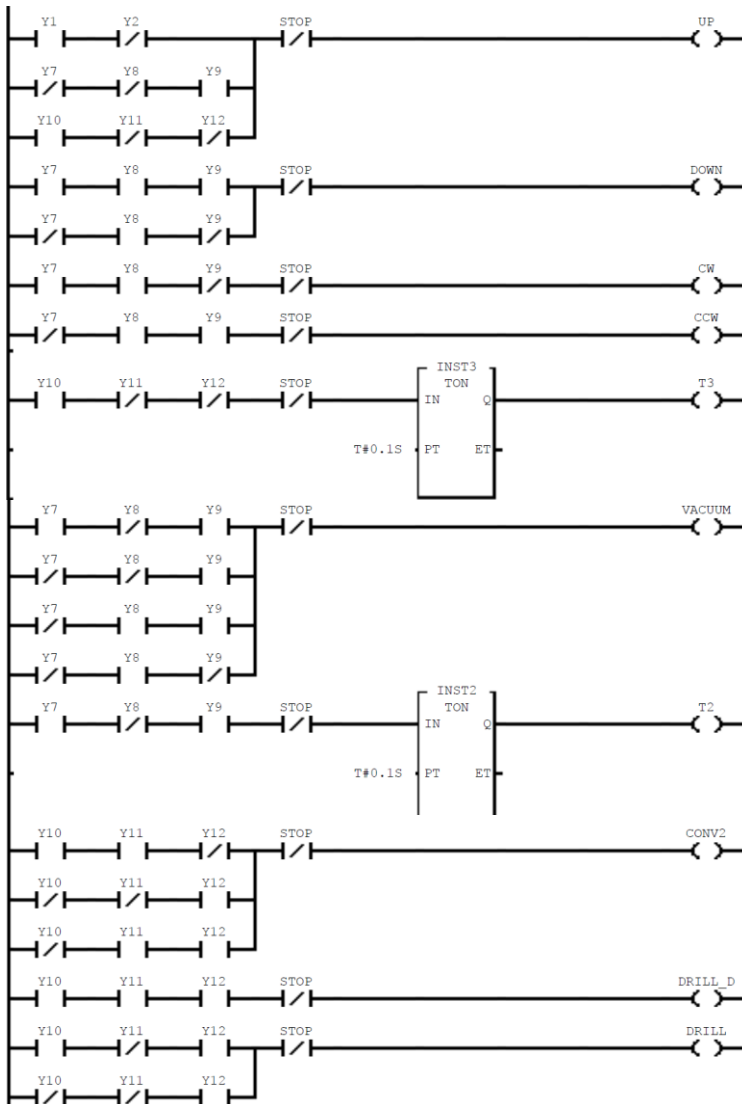
[Halaman ini sengaja dikosongkan]

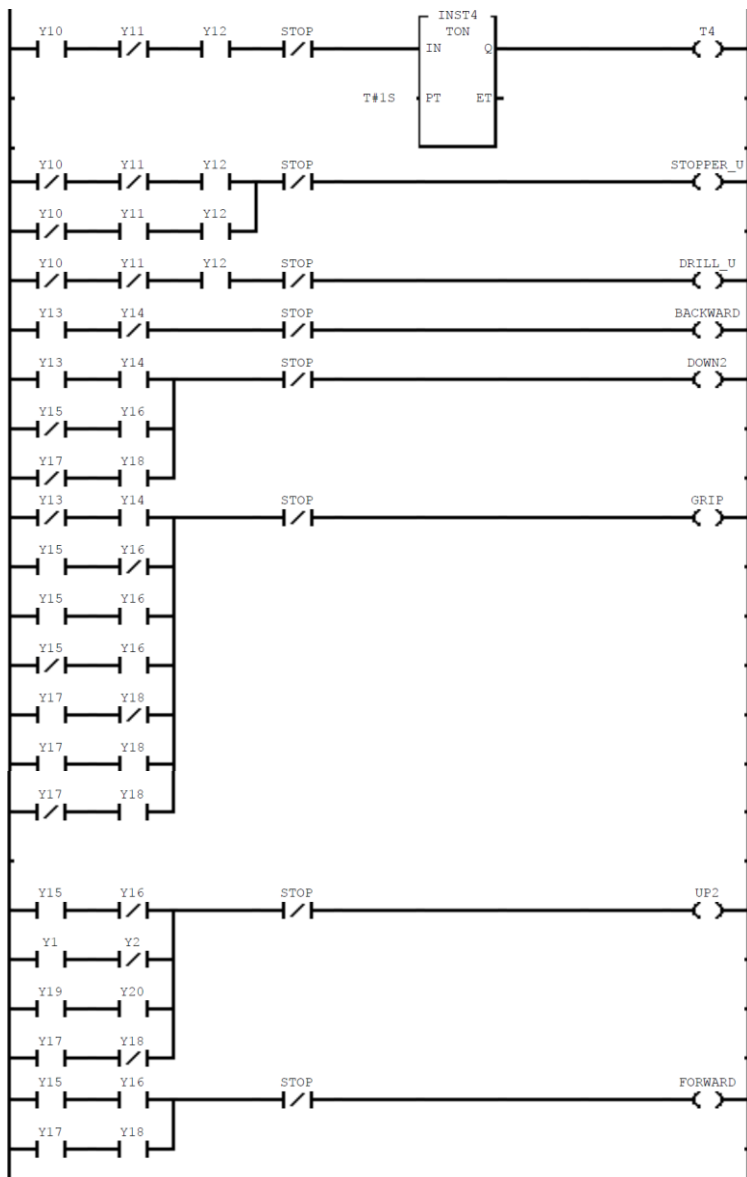
E. HASIL KONSTRUKSI *LADDER DIAGRAM* UNTUK SELURUH SUBPROSES

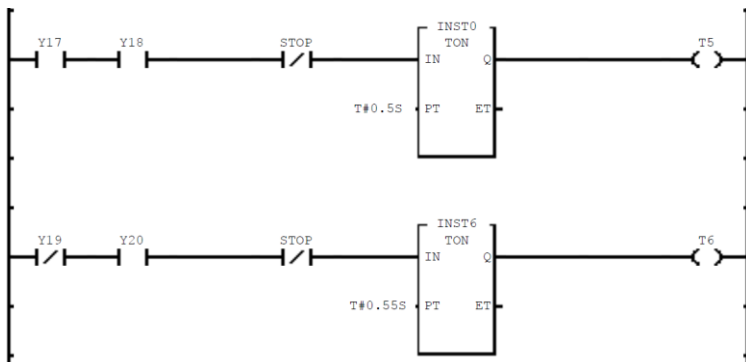












RIWAYAT HIDUP



Nama saya Gustafian Fawally Al-Barr. Lahir di Blitar 23 Januari 1996. Saya merupakan Mahasiswa Teknik Elektro angkatan 2014 Saya adalah anak pertama dari dua bersaudara. Keluarga saya kental akan dunia Pendidikan. putra-putra di keluarga kami tidak terlepas dari andil dari kedua orang tua kami yang bekerja di dunia pendidikan. Besar harapan saya ilmu-ilmu yang saya pergunakan di jurusan Teknik elektro ITS dapat bermanfaat seminimalnya bagi saya pribadi lebih-lebih bagi keluarga, masyarakat,

negara serta agama Islam. Untuk lebih jauh mengenal silahkan hubungi gustaf.f96@gmail.com