



TUGAS AKHIR - EE 184801

**DESAIN DIAGRAM *LADDER* UNTUK BANYAK OBJEK
PADA FACTORY AUTOMATIC TRAINER
MENGGUNAKAN METODE GRAFCET**

Innes Deviana
NRP 07111745000036

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

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



FINAL PROJECT - EE 184801

***LADDER DIAGRAM DESIGN FOR MULTIPLE OBJECTS
ON FACTORY AUTOMATIC TRAINER USING GRAFCET
METHOD***

Innes Deviana
NRP 07111745000036

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

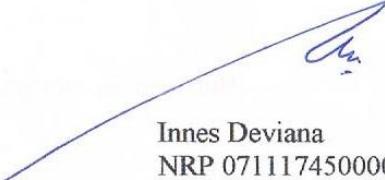
Electrical Engineering Department
Electrical Engineering Faculty
Institut Teknologi Sepuluh Nopember
Surabaya 2019

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul “**Desain Diagram Ladder untuk Banyak Objek pada Factory Automatic Trainer menggunakan Metode Grafset**” merupakan hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan – bahan yang tidak diijinkan dan buka 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 2019


Innes Deviana
NRP 07111745000036

-----Halaman ini sengaja dikosongkan-----



**DESAIN DIAGRAM LADDER UNTUK BANYAK OBJEK PADA
FACTORY AUTOMATIC TRAINER MENGGUNAKAN METODE
GRAFCET**



TUGAS AKHIR

Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik
Pada

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

Menyetujui :

Dosen Pembimbing I

Dosen Pembimbing II

Dr.Ir.Mochammad Rameli
NIP.19541227 1981031002

Eka Iskandar, ST., MT.
NIP. 198005282008121001



-----Halaman ini sengaja dikosongkan-----

DESAIN DIAGRAM LADDER UNTUK BANYAK OBJEK PADA FACTORY AUTOMATIC TRAINER MENGGUNAKAN METODE GRAFCET

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

ABSTRAK

Factory Automatic Trainer (FAT) merupakan sebuah miniatur yang merepresentasikan sistem otomasi pada industri penyortiran benda. *Factory Automatic Trainer* (FAT) dioperasikan oleh PLC dengan bahasa pemrograman yang digunakan adalah diagram *ladder*. Permasalahan yang dihadapi dengan adanya penambahan objek serta perulangan proses pada *Factory Automatic Trainer* (FAT) dapat meningkatkan resiko terjadinya *troubleshoot* ataupun *human error*. Dengan adanya sebuah metode dalam merancang diagram *ladder* diharapkan mampu mengurangi permasalahan tersebut. *Grafcet* merupakan salah satu metode dalam pembuatan diagram *ladder*. Keunggulan dari metode ini adalah dapat menggambarkan urutan dari proses tersebut sehingga mudah dipahami. Luaran yang diharapkan adalah hasil desain diagram *ladder* menjadi terstruktur dan dapat diimplementasi sesuai dengan yang operasi sistem yang diinginkan. Hasil desain diagram *ladder* dengan metode *grafcet* terdapat 85 memori (*step*), 23 *output* dan 28 *input* dengan 6 *timer* dan 12 *counter*. Data program sebesar 30 KB dengan jumlah *rung* sebanyak 72 *rung*. Waktu proses untuk benda tunggal berwarna hitam 22.12 detik, benda biru 21.28 detik dan benda *silver* 5.85 detik. Pengujian waktu benda berurutan sekitar 01.08.85 menit, benda acak warna *silver* di posisi terakhir memiliki waktu proses tercepat yaitu 01.03.48 menit, warna biru sekitar 01.06.39 menit dan benda hitam 01.08.60 menit. Dampak dari hasil diagram *ladder* yang dirancang dengan menggunakan metode *grafcet* apabila diterapkan ke dalam industri sortir benda membuat siklus produksi menjadi lebih cepat apabila dikerjakan secara multi proses.

Kata Kunci : *Factory Automatic Trainer* (FAT), *Programmable Logic Controller* (PLC), Diagram *Ladder*, *Grafcet*

-----Halaman ini sengaja dikosongkan-----

LADDER DIAGRAM DESIGN FOR MULTIPLE OBJECTS ON FACTORY AUTOMATIC TRAINER USING GRAFCET METHOD

Name : Innes Deviana
Advisor : 1. Dr.Ir.Mochammad Rameli
 2. Eka Iskandar, ST., MT.

ABSTRACT

Factory Automatic Trainer (FAT) is a miniature that represents an automation system in the product sorting industry. Factory Automatic Trainer (FAT) operated by PLC with the programming language used is a ladder diagram. The problems faced by the addition of objects and repetition of the process at the Factory Automatic Trainer (FAT) can increase the risk of troubleshoot or human error. With the existence of a method in designing a ladder diagram, it is expected to reduce these problems. Grafcet is one method in designing ladder diagrams. The advantage of this method is that it can describe the sequence of the process so that it is easy to understand. Expected output is the result of a ladder diagram design that is structured and can be implemented in accordance with the desired operating system. The design of the ladder diagram with the grafcet method contains several coils consisting of 85 for memory (steps) and 23 for outputs. While the use of inputs are 28. There are 6 timers and 12 counters. The program data produced is 30 KB with a total of 72 rungs. The processing time for single black objects is 22.12 seconds, blue objects are 21.28 seconds and silver objects are 5.85 seconds. Testing for sequential objects is around 01.08.85 minutes, random objects with silver in the last position have the fastest processing time, which is around 01.03.48 minutes, followed by blue around 01.06.39 minutes and black objects in the last position have the longest processing time, namely 01.08.60 minutes. The impact of the ladder diagram results designed using the grafcet method when applied to the object sorting industry makes the production cycle faster if it is done in a multi-process.

Keywords : Factory Automatic Trainer (FAT), Programmable Logic Controller (PLC), Ladder Diagram, Grafcet

-----Halaman ini sengaja dikosongkan-----

KATA PENGANTAR

Puji syukur penulis panjatkan kehadiran Allah SWT yang selalu memberikan rahmat dan hidayah-Nya sehingga Tugas Akhir ini dapat terselesaikan dengan baik. Shalawat serta salam semoga selalu dilimpahkan kepada Rasulullah Muhammad SAW, keluarga, sahabat, dan umat muslim yang senantiasa meneladani beliau.

Tugas Akhir ini disusun untuk memenuhi sebagian persyaratan guna menyelesaikan pendidikan S1 pada Bidang Studi Teknik Sistem Pengaturan, Departemen Teknik Elektro, Fakultas Teknologi Elektro, Institut Teknologi Sepuluh Nopember Surabaya dengan judul:

DESAIN DIAGRAM LADDER UNTUK BANYAK OBJEK PADA FACTORY AUTOMATIC TRAINER MENGGUNAKAN METODE GRAFCET

Penulis mengucapkan terima kasih kepada Ibu dan Bapak penulis yang memberikan berbagai bentuk doa serta dukungan tulus tiada henti, Bapak Dr.Ir.Mochammad Rameli dan Bapak Eka Iskandar, ST., MT. atas segala bimbingan ilmu, moral, dan spiritual dari awal hingga terselesaiannya Tugas Akhir ini. Penulis juga mengucapkan banyak terima kasih kepada semua pihak yang telah membantu baik secara langsung maupun tidak langsung dalam proses penyelesaian Tugas Akhir ini.

Penulis menyadari dan memohon maaf atas segala kekurangan pada Tugas Akhir ini. Akhir kata, semoga Tugas Akhir ini dapat bermanfaat dalam pengembangan keilmuan di kemudian hari.

Surabaya, Juli 2019

Penulis

-----Halaman ini sengaja dikosongkan-----

DAFTAR ISI

	HALAMAN
HALAMAN JUDUL	i
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	xix
DAFTAR TABEL	xxv
 BAB 1 PENDAHULUAN	 1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	3
1.4 Tujuan.....	3
1.5 Metodologi Penelitian	3
1.6 Sistematika Laporan	5
1.7 Relevansi.....	6
 BAB 2 TEORI PENUNJANG.....	 7
2.1 FAT (<i>Factory Automatic Trainer</i>)	7
2.1.1 <i>Separation Module</i>	8
2.1.2 <i>Pick & Place Module</i>	9
2.1.3 <i>Stopper Module</i>	9
2.1.4 <i>Line Movement Module</i>	10
2.1.5 <i>Control Unit</i>	11
2.1.6 Benda Kerja <i>Factory Automatic Trainer</i>	12
2.2 Perangkat Komponen Elektrik dan Pneumatik pada <i>Factory Automatic Trainer</i>	12
2.2.1 <i>Proximity Inductive Sensor</i>	13
2.2.2 <i>Proximity Capacitive Sensor</i>	14
2.2.3 Sensor Fotoelektrik	15
2.2.4 <i>Fiber Optic Sensor</i>	16
2.2.5 <i>Reed Switch</i>	17
2.2.6 <i>Belt Conveyor</i>	18

2.2.7 <i>Pneumatic Cylinder</i>	19
2.2.8 <i>Rodless Cylinder</i>	20
2.2.9 <i>Rotary Cylinder</i>	21
2.2.10 <i>Finger Cylinder</i>	22
2.2.11 <i>Solenoid Valve 5/2 Way</i>	22
2.2.12 <i>Vacuum Pad</i>	24
2.2.13 <i>Drilling Machine</i>	25
2.2.14 <i>Warning Light</i>	25
2.3 <i>PLC (Programmable Logic Controller)</i>	26
2.4 <i>Diagram Ladder</i>	28
2.5 <i>Graf cet</i>	32
2.5.1 Bagian – bagian <i>Graf cet</i>	32
2.5.2 Konversi <i>Graf cet</i> menjadi <i>Diagram Ladder</i>	35
2.6 <i>Fluidsim</i>	36
BAB 3 PERANCANGAN SISTEM	39
3.1 Identifikasi <i>Input</i> dan <i>Output Factory Automatic Trainer</i>	39
3.2 Perancangan <i>Graf cet</i>	48
3.2.1 Perancangan <i>Graf cet</i> pada <i>Separation Module</i>	48
3.2.2 Perancangan <i>Graf cet</i> pada <i>Pick and Place Module</i>	53
3.2.3 Perancangan <i>Graf cet</i> pada <i>Stopper Module</i>	56
3.2.4 Perancangan <i>Graf cet</i> pada <i>Line Movement Module</i>	59
3.3 Perancangan <i>Diagram Ladder</i>	67
3.3.1 Perancangan <i>Diagram Ladder Separation Module</i>	67
3.3.2 Perancangan <i>Diagram Ladder Pick and Place Module</i>	78
3.3.3 Perancangan <i>Diagram Ladder Stopper Module</i>	84
3.3.4 Perancangan <i>Diagram Ladder Line Movement Module</i> ...	89
BAB 4 IMPLEMENTASI DAN ANALISA	107
4.1 Konfigurasi <i>Factory Automatic Trainer</i>	107
4.2 Validasi	109
4.2.1 Validasi pada <i>Separation Module</i>	109
4.2.2 Validasi pada <i>Pick and Place Module</i>	112
4.2.3 Validasi pada <i>Stopper Module</i>	114
4.2.4 Validasi pada <i>Line Movement Module</i>	116
4.3 Implementasi	120
4.3.1 Implementasi pada <i>Separation Module</i>	120
4.3.2 Implementasi pada <i>Pick and Place Module</i>	130
4.3.3 Implementasi pada <i>Stopper Module</i>	136
4.3.4 Implementasi pada <i>Line Movement Module</i>	140

4.4 Hasil Pengujian	152
BAB 5 PENUTUP	155
5.1 Kesimpulan	155
5.2 Saran.....	156
DAFTAR PUSTAKA	157
LAMPIRAN	159
A. Keseluruhan <i>Graf cet</i>	159
B. Keseluruhan Diagram <i>Ladder</i>	163
C. Hasil Rangkaian Elektrik & Pneumatik pada Fluidsim.....	171
C.1 <i>Separation Module</i>	171
C.2 <i>Pick and Place Module</i>	172
C.3 <i>Stopper Module</i>	173
C.4 <i>Line Movement Module</i>	174
DAFTAR RIWAYAT HIDUP	177

-----Halaman ini sengaja dikosongkan-----

DAFTAR GAMBAR

HALAMAN

Gambar 2.1 <i>Factory Automatic Trainer</i> [3]	7
Gambar 2.2 <i>Separation Module</i>	8
Gambar 2.3 <i>Pick & Place Module</i>	9
Gambar 2.4 <i>Stopper Module</i>	10
Gambar 2.5 <i>Line Movement Module</i>	11
Gambar 2.6 <i>Control Unit</i> [3]	11
Gambar 2.7 Benda Kerja <i>Factory Automatic Trainer</i>	12
Gambar 2.8 <i>Proximity Inductive Sensor</i> [4].....	13
Gambar 2.9 Prinsip Kerja <i>Proximity Inductive Sensor</i>	13
Gambar 2.10 <i>Proximity Capacitive Sensor</i> [4].....	14
Gambar 2.11 Prinsip Kerja <i>Capacitive Proximity Sensor</i>	15
Gambar 2.12 Sensor Fotoelektrik [7].....	15
Gambar 2.13 <i>Fiber Optic Sensor</i>	17
Gambar 2.14 <i>Reed Switch</i>	18
Gambar 2.15 Konveyor pada <i>Factory Automatic Trainer</i>	18
Gambar 2.16 Silinder Pneumatik pada <i>Factory Automatic Trainer</i>	19
Gambar 2.17 <i>Rodless Cylinder</i> [10].....	20
Gambar 2.18 <i>Rodless Cylinder</i> pada <i>Factory Automatic Trainer</i>	21
Gambar 2.19 <i>Rotary Cylinder</i> pada <i>Factory Automatic Trainer</i>	21
Gambar 2.20 <i>Finger Cylinder</i> pada <i>Factory Automatic Trainer</i>	22
Gambar 2.21 <i>Single Acting</i> (NO) dan <i>Single Acting</i> (NC) [10]	22
Gambar 2.22 <i>Solenoid Valve 5/2 Way</i>	23
Gambar 2.23 <i>Solenoid valve Single acting</i> [10]	23
Gambar 2.24 <i>Solenoid valve Double acting</i> [10].....	24
Gambar 2.25 <i>Vacuum Pad</i>	24
Gambar 2.26 <i>Drilling Machine</i>	25
Gambar 2.27 <i>Warning Light</i>	26
Gambar 2.28 PLC LG Glofa GM4	26
Gambar 2.29 Bagian – bagian PLC [6].....	27
Gambar 2.30 <i>Ladder Diagram</i> [6].....	29
Gambar 2.31 Isyarat <i>Input & Output Timer on Delay</i> [11].....	30
Gambar 2.32 Isyarat <i>Input & Output Timer off Delay</i> [11]	31
Gambar 2.33 Isyarat <i>Input & Output Timer Pulse</i> [11]	31
Gambar 2.34 Struktur <i>Grafset</i> Sederhana [14].....	35

Gambar 2.35 Hasil Diagram <i>Ladder</i> pada <i>Grafcet</i> Sederhana [14]	36
Gambar 2.36 <i>Software Festo Fluidsim</i>	37
Gambar 3.1 Prosedur Pembuatan Diagram <i>Ladder</i> dengan Metode <i>Grafcet</i>	39
Gambar 3.2 Letak I/O pada <i>Separation Module</i>	40
Gambar 3.3 Letak Aktuator pada <i>Pick and Place Module</i>	42
Gambar 3.4 Letak I/O pada <i>Stopper Module</i>	45
Gambar 3.5 Letak Aktuator pada <i>Line Movement Module</i>	47
Gambar 3.6 Diagram Waktu <i>Separation Module</i> Benda Silver.....	50
Gambar 3.7 Diagram Waktu <i>Separation Module</i> Benda Hitam	50
Gambar 3.8 Perancangan <i>Grafcet</i> pada <i>Separation Module</i>	53
Gambar 3.9 Diagram Waktu <i>Pick and Place Module</i>	54
Gambar 3.10 Perancangan <i>Grafcet</i> pada <i>Pick and Place Module</i>	56
Gambar 3.11 Diagram Waktu <i>Stopper Module</i>	57
Gambar 3.12 Perancangan <i>Grafcet</i> pada <i>Stopper Module</i>	58
Gambar 3.13 Perancangan <i>Grafcet</i> Pemilihan Warna Benda.....	62
Gambar 3.14 Diagram Waktu <i>Line Movement Module</i> Benda Biru	63
Gambar 3.15 Diagram Waktu <i>Line Movement Module</i> Benda Hitam ..	65
Gambar 3.16 Perancangan <i>Grafcet</i> pada <i>Line Movement Module</i>	67
Gambar 3.17 <i>Grafcet Step1</i>	68
Gambar 3.18 Diagram <i>Ladder Step1</i>	68
Gambar 3.19 <i>Grafcet Step2</i>	69
Gambar 3.20 Diagram <i>Ladder Step2</i>	69
Gambar 3.21 <i>Grafcet Step3A</i> dan <i>Step3B</i>	70
Gambar 3.22 Diagram <i>Ladder Step3A</i> dan <i>Step3B</i>	71
Gambar 3.23 <i>Grafcet Step4</i>	71
Gambar 3.24 Diagram <i>Ladder Step4</i>	72
Gambar 3.25 <i>Grafcet Step5A</i> dan <i>Step5B</i>	72
Gambar 3.26 Diagram <i>Ladder Step5A</i> dan <i>Step5B</i>	73
Gambar 3.27 <i>Grafcet Step6</i>	73
Gambar 3.28 Diagram <i>Ladder Step6</i>	74
Gambar 3.29 <i>Grafcet Step7</i>	74
Gambar 3.30 Diagram <i>Ladder Step7</i>	74
Gambar 3.31 <i>Grafcet Step5C</i>	75
Gambar 3.32 Diagram <i>Ladder Step5C</i>	75
Gambar 3.33 <i>Grafcet Step5D</i> dan <i>Step5E</i>	76
Gambar 3.34 Diagram <i>Ladder Step5D</i>	76
Gambar 3.35 Diagram <i>Ladder Step5E(TP)</i>	77
Gambar 3.36 <i>Grafcet Step8</i>	77

Gambar 3.37 Diagram <i>Ladder Step8</i>	78
Gambar 3.38 <i>Graf cet Step9</i>	78
Gambar 3.39 Diagram <i>Ladder Step9A</i>	79
Gambar 3.40 <i>Graf cet Step10A</i> dan <i>Step10B</i>	79
Gambar 3.41 Diagram <i>Ladder Step10A</i> dan <i>Step10B</i>	80
Gambar 3.42 <i>Graf cet Step11</i>	81
Gambar 3.43 Diagram <i>Ladder Step11</i>	81
Gambar 3.44 <i>Graf cet Step12</i>	82
Gambar 3.45 Diagram <i>Ladder Step12</i>	82
Gambar 3.46 <i>Graf cet Step13A</i>	83
Gambar 3.47 Diagram <i>Ladder Step13A</i>	83
Gambar 3.48 <i>Graf cet Step14A</i>	84
Gambar 3.49 Diagram <i>Ladder Step14A</i>	84
Gambar 3.50 <i>Graf cet Step13B</i>	85
Gambar 3.51 Diagram <i>Ladder Step13B</i>	85
Gambar 3.52 <i>Graf cet Step14B</i> dan <i>Step14C</i>	86
Gambar 3.53 Diagram <i>Ladder Step14B</i> dan <i>Step14C</i>	86
Gambar 3.54 <i>Graf cet Step15</i>	87
Gambar 3.55 Diagram <i>Ladder Step15</i>	87
Gambar 3.56 <i>Graf cet Step16</i>	88
Gambar 3.57 Diagram <i>Ladder Step16</i>	88
Gambar 3.58 <i>Graf cet Step17</i>	89
Gambar 3.59 Diagram <i>Ladder Step17</i>	89
Gambar 3.60 <i>Graf cet Step10C</i>	90
Gambar 3.61 <i>Graf cet Step10D</i>	90
Gambar 3.62 Diagram <i>Ladder Step10C</i> atau <i>Step10D</i>	91
Gambar 3.63 <i>Graf cet Step18A</i>	91
Gambar 3.64 Diagram <i>Ladder Step18A</i>	92
Gambar 3.65 <i>Graf cet Step19A</i> dan <i>Step19B</i>	93
Gambar 3.66 Diagram <i>Ladder Step19A</i> dan <i>Step19B</i>	94
Gambar 3.67 <i>Graf cet Step20A</i> dan <i>Step20B</i>	95
Gambar 3.68 Diagram <i>Ladder Step20A</i> dan <i>Step20B</i>	95
Gambar 3.69 <i>Graf cet Step21A</i>	96
Gambar 3.70 Diagram <i>Ladder Step21A</i>	96
Gambar 3.71 <i>Graf cet Step22A</i>	97
Gambar 3.72 Diagram <i>Ladder Step22A</i>	97
Gambar 3.73 <i>Graf cet Step18B</i>	98
Gambar 3.74 Diagram <i>Ladder Step18B</i>	99
Gambar 3.75 <i>Graf cet Step19C</i> dan <i>Step19D</i>	100

Gambar 3.76 Diagram <i>Ladder Step19C</i> dan <i>Step19D</i>	101
Gambar 3.77 <i>Graf cet Step20C</i>	101
Gambar 3.78 Diagram <i>Ladder Step20C</i>	102
Gambar 3.79 <i>Graf cet Step21B</i>	102
Gambar 3.80 Diagram <i>Ladder Step21B</i>	103
Gambar 3.81 <i>Graf cet Step22B</i>	104
Gambar 3.82 Diagram <i>Ladder Step22B</i>	104
Gambar 3.83 <i>Graf cet Step23</i>	105
Gambar 3.84 Diagram <i>Ladder Step23</i>	105
Gambar 4.1 Konfigurasi <i>Factory Automatic Trainer</i>	107
Gambar 4.2 <i>Wiring Input PLC LG</i>	108
Gambar 4.3 <i>Wiring Output PLC LG</i>	108
Gambar 4.4 Hasil Pengkabelan	109
Gambar 4.5 Validasi <i>Separation Module</i> Benda Silver.....	110
Gambar 4.6 Validasi <i>Separation Module</i> Benda Hitam.....	111
Gambar 4.7 Validasi <i>Pick and Place Module</i>	113
Gambar 4.8 Validasi <i>Stopper Module</i>	115
Gambar 4.9 Validasi <i>Line Movement Module</i> Benda Biru.....	117
Gambar 4.10 Validasi <i>Line Movement Module</i> Benda Hitam	119
Gambar 4.11 Implementasi Diagram <i>Ladder Step1</i>	121
Gambar 4.12 Hasil Implementasi Aksi <i>Step1</i>	121
Gambar 4.13 Implementasi Diagram <i>Ladder Step2</i>	122
Gambar 4.14 Hasil Implementasi Aksi <i>Step2</i>	122
Gambar 4.15 Implementasi Diagram <i>Ladder Step3A & Step3B</i>	123
Gambar 4.16 Hasil Implementasi Aksi <i>Step3A & Step3B</i>	123
Gambar 4.17 Implementasi Diagram <i>Ladder Step4</i>	124
Gambar 4.18 Hasil Implementasi Aksi <i>Step4</i>	124
Gambar 4.19 Implementasi Diagram <i>Ladder Step5A & Step5B</i>	125
Gambar 4.20 Hasil Implementasi <i>Step5A & Step5B</i>	125
Gambar 4.21 Hasil Implementasi Aksi <i>Step5B</i>	126
Gambar 4.22 Implementasi Diagram <i>Ladder Step5C</i>	126
Gambar 4.23 Hasil Implementasi <i>Step5C</i>	127
Gambar 4.24 Implementasi Diagram <i>Ladder Step5D</i>	127
Gambar 4.25 Hasil Implementasi <i>Step5D</i>	128
Gambar 4.26 Implementasi Diagram <i>Ladder Step6 & Step7</i>	128
Gambar 4.27 Hasil Implementasi Aksi <i>Step7</i>	129
Gambar 4.28 Implementasi Diagram <i>Ladder Step8</i>	129
Gambar 4.29 Hasil Implementasi <i>Step8</i>	130
Gambar 4.30 Implementasi Diagram <i>Ladder Step9A</i>	130

Gambar 4.31 Hasil Implementasi Aksi <i>Step9A</i>	131
Gambar 4.32 Implementasi Diagram <i>Ladder Step10A & Step10B</i>	131
Gambar 4.33 Hasil Implementasi Aksi <i>Step10A & Step10B</i>	132
Gambar 4.34 Implementasi Diagram <i>Ladder Step11</i>	132
Gambar 4.35 Hasil Implementasi Aksi <i>Step11</i>	133
Gambar 4.36 Implementasi Diagram <i>Ladder Step12</i>	133
Gambar 4.37 Hasil Implementasi <i>Step12</i>	134
Gambar 4.38 Implementasi Diagram <i>Ladder Step13A</i>	134
Gambar 4.39 Hasil Implementasi Aksi <i>Step13A</i>	135
Gambar 4.40 Implementasi Diagram <i>Ladder Step14A</i>	135
Gambar 4.41 Hasil Implementasi Aksi <i>Step14A</i>	136
Gambar 4.42 Implementasi Diagram <i>Ladder Step13B</i>	136
Gambar 4.43 Implementasi Diagram <i>Ladder Step14B & Step14C</i>	137
Gambar 4.44 Hasil Implementasi Aksi <i>Step14B & Step14C</i>	137
Gambar 4.45 Implementasi Diagram <i>Ladder Step15</i>	138
Gambar 4.46 Implementasi Diagram <i>Ladder Step16</i>	138
Gambar 4.47 Hasil Implementasi Aksi <i>Step16</i>	139
Gambar 4.48 Implementasi Diagram <i>Ladder Step17</i>	139
Gambar 4.49 Hasil Implementasi <i>Step17</i>	140
Gambar 4.50 Implementasi Diagram <i>Ladder Step18A</i>	141
Gambar 4.51 Hasil Implementasi Aksi <i>Step18A</i>	141
Gambar 4.52 Implementasi Diagram <i>Ladder Step19A & Step19B</i>	142
Gambar 4.53 Hasil Implementasi Aksi <i>Step19A & Step19B</i>	142
Gambar 4.54 Implementasi Diagram <i>Ladder Step20B & Step20B</i>	143
Gambar 4.55 Hasil Implementasi Aksi <i>Step20A & Step20B</i>	143
Gambar 4.56 Implementasi Diagram <i>Ladder Step21A</i>	144
Gambar 4.57 Implementasi Diagram <i>Ladder Step22A</i>	144
Gambar 4.58 Hasil Implementasi Aksi <i>Step22A</i>	145
Gambar 4.59 Implementasi Diagram <i>Ladder Step18B</i>	146
Gambar 4.60 Hasil Implementasi Aksi <i>Step18B</i>	146
Gambar 4.61 Implementasi Diagram <i>Ladder Step19C & Step19D</i>	147
Gambar 4.62 Hasil Implementasi Aksi <i>Step19C & Step19D</i>	148
Gambar 4.63 Implementasi Diagram <i>Ladder Step20C</i>	148
Gambar 4.64 Hasil Implementasi Aksi <i>Step20C</i>	149
Gambar 4.65 Implementasi Diagram <i>Ladder Step21B</i>	149
Gambar 4.66 Hasil Implementasi Aksi <i>Step21B</i>	150
Gambar 4.67 Implementasi Diagram <i>Ladder Step22B</i>	150
Gambar 4.68 Hasil Implementasi Aksi <i>Step22B</i>	151

Gambar 4.69 Implementasi Diagram <i>Ladder Step23</i>	151
Gambar 4.70 Hasil Implementasi Aksi <i>Step23</i>	152

DAFTAR TABEL

	HALAMAN
Tabel 2.1 Spesifikasi Benda Kerja.....	12
Tabel 2.2 Nama, Bagian-bagian dan Fungsi <i>Grafcet</i>	33
Tabel 3.1 Alamat <i>Input</i> pada <i>Separation Module</i>	40
Tabel 3.2 Alamat <i>Output</i> pada <i>Separation Module</i>	41
Tabel 3.3 Alamat <i>Input</i> pada <i>Pick and Place Module</i>	43
Tabel 3.4 Alamat <i>Output</i> pada <i>Pick and Place Module</i>	43
Tabel 3.5 Alamat <i>Input</i> pada <i>Stopper Module</i>	45
Tabel 3.6 Alamat <i>Output</i> pada <i>Stopper Module</i>	46
Tabel 3.7 Alamat <i>Input</i> pada <i>Line Movement Module</i>	47
Tabel 3.8 Alamat <i>Output</i> pada <i>Line Movement Module</i>	48
Tabel 3.9 Tahapan Proses <i>Separation Module</i>	51
Tabel 3.10 Tahapan Proses <i>Pick and Place Module</i>	54
Tabel 3.11 Tahapan Proses <i>Stopper Module</i>	57
Tabel 3.12 Tahapan Proses Pemilihan Warna Benda	60
Tabel 3.13 Tahapan Proses <i>Line Movement Module</i> Benda Biru.....	63
Tabel 3.14 Tahapan Proses <i>Line Movement Module</i> Benda Hitam.....	65
Tabel 4.1 Hasil Validasi <i>Separation Module</i>	112
Tabel 4.2 Hasil Validasi <i>Pick and Place Module</i>	114
Tabel 4.3 Hasil Validasi <i>Stopper Module</i>	116
Tabel 4.4 Hasil Validasi <i>Line Movement Module</i> Benda Biru	118
Tabel 4.5 Hasil Validasi <i>Line Movement Module</i> Benda Hitam	119
Tabel 4.6 Hasil Pengujian Waktu Proses Objek Tunggal	152
Tabel 4.7 Hasil Pengujian Multi Proses Benda Berurutan.....	153
Tabel 4.8 Hasil Pengujian Multi Proses Benda Acak	154

-----Halaman ini sengaja dikosongkan-----

BAB 1

PENDAHULUAN

Pada bab ini akan membahas mengenai hal apa yang melatar belakangi dalam pembuatan tugas akhir ini, permasalahan yang timbul dari tugas akhir ini, batasan masalah, solusi untuk menyelesaikan permasalahan tersebut yang disajikan dalam tujuan tugas akhir, metodologi penelitian yang menjadi pedoman dalam pengerjaan tugas akhir ini, sistematika laporan yang secara garis besar menjadi isi dalam laporan tugas akhir ini serta relevansi.

1.1 Latar Belakang

Seiring dengan perkembangan teknologi sebagian pekerjaan manusia telah digantikan dengan mesin. Hal ini tentu berkaitan dengan peningkatan kebutuhan masyarakat di berbagai bidang, sehingga mendorong kebutuhan suatu sistem agar dapat mempermudah dan meningkatkan produktivitas dan efektivitas dalam berbagai pekerjaan. Mahasiswa sebaiknya telah dipersiapkan untuk menuju ke arah industri dikarenakan mahasiswa memiliki peran penting dalam perkembangan teknologi. Dengan adanya wawasan kegiatan usaha di industri mahasiswa dapat mengaplikasikan ilmu tentang otomasi sistem. *Factory Automatic Trainer* (FAT) merupakan salah satu *trainer* yang dapat mencakup kegiatan usaha di industri dan dapat beroperasi secara otomatis. Proses kerja dari *Factory Automatic Trainer* (FAT) adalah mengoperasikan objek dengan menyortir objek yang tidak sesuai dengan spesifikasi berdasarkan bahan material serta warna dari objek, memindahkan dari konveyor pertama menuju konveyor kedua untuk dilakukan proses drilling dan menjepit objek untuk diletakkan pada suatu wadah. Pada sistem *Factory Automatic Trainer* (FAT) terdapat pusat kendali yang mengatur kinerja dari keseluruhan sistem dengan perangkat bernama *Programmable Logic Controller* (PLC).

Programmable Logic Controller (PLC) merupakan seperangkat alat elektronik yang dapat mengerjakan berbagai fungsi kontrol pada tingkatan yang kompleks serta dapat diterapkan sebagai pengendalian sistem yang membutuhkan *input* ataupun *output* dalam jumlah banyak. Diagram *ladder* adalah bahasa pemrograman *Programmable Logic Controller* (PLC) yang dikembangkan dari kontak - kontak *relay* terstruktur dan menggambarkan aliran arus listrik. Diagram *ladder* terdiri

dari kontak *relay*, koil *relay*, *timer* serta *counter* yang digunakan untuk mengoperasikan suatu proses pada *plant* tertentu seperti pada suatu industri yang memiliki proses cukup kompleks. Dalam membuat diagram *ladder* dibutuhkan sebuah metode untuk mempermudah menyusun langkah-langkah setiap proses yang akan dijalankan.

Witanto, Aji P. (2018) pada jurnal yang berjudul : “Konstruksi Diagram *Ladder* dengan Metode *Huffman* untuk *Factory Automatic Trainer*”. Hasil dari tugas akhir tersebut adalah diagram *ladder* yang dirancang mampu mengoperasikan benda tunggal dari *factory automatic trainer* [1]. Pada jurnal ini menjadi acuan untuk mempelajari *plant* yang sama.

Nugraha, Raka A. (2018) pada jurnal yang berjudul : “Konstruksi Diagram *Ladder* dengan Bantuan *Grafset* untuk *Factory Automatic Trainer*”. Penggunaan metode *grafset* memiliki konsep bahasa grafis sehingga lebih mudah dipahami dan dilakukan modifikasi program apabila terjadi perubahan operasi sistem. Metode *grafset* tepat digunakan untuk pemrograman sistem otomatisasi. Hasil dari tugas akhir tersebut adalah diagram *ladder* yang dirancang dengan menggunakan metode *grafset* mampu mengoperasikan benda tunggal pada *factory automatic trainer* [2]. Pada pengembangan tugas akhir ini dengan metode dan *plant* yang sama mampu mengoperasikan *factory automatic trainer* secara multi proses.

1.2 Rumusan Masalah

Factory Automatic Trainer (FAT) merupakan sebuah *trainer* yang memiliki proses penggerjaan sesuai dengan proses industri manufaktur. Proses kerja dari *Factory Automatic Trainer* (FAT) berjalan dari beberapa objek yang diseleksi berdasarkan dengan bahan material dan warna, objek yang sesuai dengan spesifikasi yang diinginkan akan diambil dan diletakkan (*pick and place*) pada konveyor selanjutnya. Pada konveyor yang kedua objek akan dilakukan proses *drilling* dan kemudian dialokasikan ke dalam suatu wadah. Objek akan beroperasi secara terus-menerus hingga tombol *stop* ditekan. Banyaknya perulangan proses yang dikerjakan oleh *Factory Automatic Trainer* (FAT) menimbulkan banyak langkah yang harus diproses dikarenakan adanya penambahan objek. Oleh karena itu apabila dalam perancangan diagram *ladder* ini diselesaikan dengan tanpa adanya metode akan menjadi rumit serta memiliki resiko adanya *troubleshoot* ataupun *human error* yang membuat operasi sistem tidak dapat beroperasi dengan semestinya. Dengan adanya

sebuah metode dalam pembuatan desain diagram *ladder* diharapkan dapat mengurangi permasalahan tersebut.

1.3 Batasan Masalah

Pada penelitian Tugas Akhir ini penulis memiliki beberapa batasan masalah dalam penggerjaannya. Beberapa batasan masalah tersebut diantaranya:

1. Diagram *ladder* yang dibuat menggunakan *software* GMWIN
2. Metode yang digunakan dalam merancang diagram *ladder* adalah dengan metode *grafcet*.
3. Alat yang digunakan sebagai penerapan sistem otomasi adalah *Factory Automatic Trainer*
4. Kontroler yang digunakan untuk mengoperasikan *Factory Automatic Trainer* adalah PLC LG Glofa GM4 yang terdiri dari *power supply*, CPU, 2 modul *digital input* dan 2 modul *digital output*.
5. Benda kerja dari *Factory Automatic Trainer* berjumlah 9 buah dengan spesifikasi benda berwarna *silver* 3 buah, benda berwarna hitam 3 buah dan benda berwarna hitam 3 buah.

1.4 Tujuan

Metode *grafcet* merupakan salah satu metode yang dapat digunakan untuk membuat desain diagram *ladder*. Metode *grafcet* sendiri adalah metode yang dikembangkan untuk menspesifikasikan runtun kendali industri secara diagram. Keunggulan dari metode *grafcet* sendiri adalah mudah dipahami dan mudah dikoreksi apabila terjadi *error*. Dari metode *grafcet* akan menghasilkan diagram *ladder*. Dengan adanya sebuah metode dalam menyelesaikan operasi sistem pada *Factory Automatic Trainer* (FAT) yang memiliki banyak perulangan tersebut diharapkan akan membentuk rancangan diagram *ladder* secara terstruktur sesuai urutan operasi sistem serta dapat mengoperasikan keseluruhan proses pada *Factory Automatic Trainer* (FAT) dengan banyak objek.

1.5 Metodologi Penelitian

1. Studi Literatur

Pada proses penggerjaan tugas akhir ini terlebih dahulu mencari literatur yang mempelajari mengenai proses kerja dari plant yang akan dibahas yaitu *Factory Automatic Trainer* (FAT). Literatur mengenai kontroler PLC juga diperlukan untuk memahami kaidah dari tipe PLC

yang digunakan yaitu PLC LG Glofa. Literatur yang digunakan dari *manual book* PLC LG Glofa. Bahasa pemrograman dari PLC LG Glofa yaitu diagram *ladder* juga perlu dipelajari. Salah satu metode dalam membuat diagram *ladder* yang digunakan adalah metode *grafcet*.

2. Pembuatan daftar input dan output dari sistem

Pembuatan daftar input dan output dari *Factory Automatic Trainer* (FAT) sendiri berfungsi untuk menentukan perangkat input dan output yang terhubung ke PLC serta mempermudah dalam penggerjaan pekerjaan selanjutnya dikarenakan I/O sudah terdata.

3. Menentukan deskripsi dari sistem

Dalam menentukan urutan operasi dari sistem perlu terlebih dahulu mengidentifikasi sistem. Dalam *Factory Automatic Trainer* sendiri terdapat 4 modul bagian yang beroperasi dan dapat diidentifikasi alur proses kerjanya. Tahap awal adalah menentukan urutan operasi sistem dari *separation module*, selanjutnya adalah menentukan urutan operasi sistem dari *pick and place module*, kemudian menentukan urutan operasi sistem dari *stopper module* dan yang terakhir menentukan urutan operasi sistem dari *line movement module*. Keseluruhan modul yang telah ditentukan urutan operasi sistemnya akan digabungkan untuk menjadi keseluruhan urutan operasi sistem dari *Factory Automatic Trainer* (FAT) sendiri.

4. Perancangan diagram *ladder* dengan Metode *Grafset*

Berdasarkan urutan operasi sistem yang telah dibuat dapat dikonversi menjadi sebuah diagram grafis yang menggambarkan alur dari operasi sistem yang berjalan dengan menggunakan sebuah metode yaitu metode *grafcet*. Dari diagram grafis tersebut dapat dikonversikan menjadi diagram *ladder*. Keunggulan dari metode *grafcet* sendiri adalah mudah dipahami dan mudah dikoreksi apabila terjadi *error*.

5. Validasi

Kegiatan validasi ditujukan untuk membuktikan apakah hasil rancangan diagram *ladder* yang dibuat dengan menggunakan metode *grafcet* telah sesuai dengan deskripsi operasi sistem yang diinginkan.

6. Simulasi dan Implementasi

Dari hasil *ladder diagram* yang dibentuk kemudian dilakukan simulasi dari *software* PLC LG Glofa. Simulasi dilakukan untuk menjalankan hasil dari pemrograman ladder diagram yang telah dibuat. Dari diagram *ladder* yang telah disimulasikan menggunakan *software* dari GMWIN perlu diterapkan ke dalam *Factory Automatic Trainer* (FAT). Tujuan dari implementasi adalah untuk mengetahui kesesuaian

hasil *ladder diagram* yang telah dirancang dengan aksi aktuator dari *plant Factory Automatic Trainer* (FAT).

7. Penyusunan Buku Tugas Akhir

Setelah semua prosedur diselesaikan dilakukan penyusunan buku tugas akhir. Penyusunan buku tugas akhir berpedoman pada studi literatur di awal.

1.6 Sistematika Laporan

Pembahasan Tugas Akhir ini akan dibagi menjadi lima Bab dengan sistematika sebagai berikut:

Bab 1 Pendahuluan

Bab ini meliputi latar belakang, rumusan masalah, tujuan, metodologi penelitian, sistematika laporan, dan relevansi.

Bab 2 Teori Penunjang

Bab ini menjelaskan tentang teori penunjang yang terdiri dari pembahasan *factory automatic trainer* yang didalamnya dibahas tentang *separation module*, *pick & place module*, *stopper module*, *line movement module*, *control unit* dan benda kerja pada *factory automatic trainer*. Selanjutnya dijelaskan mengenai perangkat komponen elektrik dan pneumatik pada *factory automatic trainer*, pembahasan dilanjutkan dengan membahas PLC, Diagram *Ladder*, *grafcet* dan *fluidsim*.

Bab 3 Perancangan Sistem

Bab ini membahas mengenai prosedur pembuatan diagram *ladder* dengan menggunakan metode *grafcet*. Dimulai dari identifikasi *input* dan *output*, perancangan *grafcet* dan perancangan diagram *ladder*.

Bab 4 Implementasi dan Analisa

Bab ini membahas mengenai prosedur implementasi dari hasil rancangan diagram *ladder* ke alat. Pembahasan dimulai dari konfigurasi *factory automatic trainer*, validasi, implementasi dan hasil pengujian

Bab 5 Penutup

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

1.7 Relevansi

Hasil yang diperoleh dari Tugas Akhir ini diharapkan dapat menjadi sumber referensi bagi pembaca ataupun mahasiswa pelaksana tugas akhir mendatang yang akan mengambil topik tugas akhir yang sama.

BAB 2

TEORI PENUNJANG

Bab ini menjelaskan pembahasan mengenai *factory automatic trainer* yang terdiri dari *separation module*, *pick and place module*, *stopper module*, *line movement module*, *control unit* serta benda kerja *factory automatic trainer*. Pembahasan kedua mengenai perangkat komponen penyusun dari *factory automatic trainer* yang terdiri dari elektrik maupun pneumatik. Pembahasan ketiga mengenai definisi prinsip kerja dari PLC. Pembahasan keempat mengenai diagram *ladder* yang menjadi salah satu bahasa pemrograman pada PLC. Pembahasan selanjutnya mengenai *grafcet* yaitu salah satu metode yang digunakan untuk membuat diagram *ladder*. Pembahasan terakhir mengenai *software fluidsim*.

2.1 FAT (*Factory Automatic Trainer*)

Factory Automatic Trainer merupakan sebuah *trainer* yang dapat merepresentasikan otomasi sistem di dunia industri dengan menggunakan PLC. Proses kerja dari *Factory Automatic Trainer* adalah mengoperasikan objek dengan menyortir objek yang tidak sesuai dengan spesifikasi berdasarkan bahan material serta warna dari objek, memindahkan dari konveyor pertama menuju konveyor kedua untuk dilakukan proses *drill* dan menjepit (*grip*) objek untuk diletakkan pada suatu wadah. Pada *Factory Automatic Trainer* sendiri terdapat 4 modul utama diantaranya yaitu modul *separation*, modul *pick and place*, modul *stopper* serta modul *line and movement*. Bentuk fisik dari *Factory Automatic Trainer* dapat dilihat pada Gambar 2.1 di bawah ini.

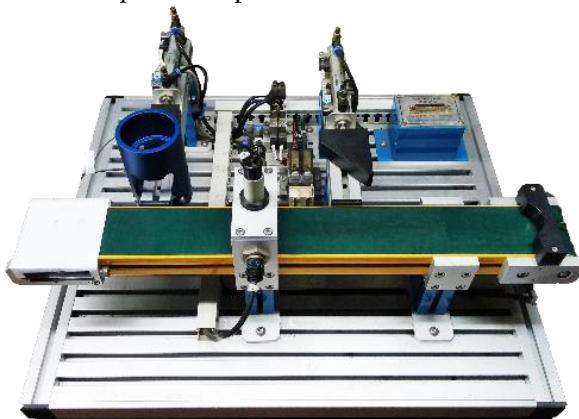


Gambar 2.1 *Factory Automatic Trainer* [3]

2.1.1 Separation Module

Separation Module memiliki peran dalam mengelompokkan benda berdasarkan unsur logam dan unsur non-logam. Komponen pada *separation module* diantaranya yaitu *Conveyor Belt*, *Motor DC*, *Inductive Proximity Sensor*, *Capasitive Proximity Sensor*, *Photo Sensor* , *Fiber Optic Sensor*, *Reed Switch Sensor*, *Pneumatic Cylinder* serta *Solenoid Valve 5/2 way* [3].

Modul ini mempelajari tentang pengelompokan benda berdasarkan ciri - cirinya. Pada modul ini terdapat konveyor, *feeding tube and feeder system*, dan beberapa sensor yang mendukung kinerja dari modul ini. Konveyor yang digerakkan dengan motor DC, digunakan sebagai jalur pemindahan untuk benda yang akan dikelompokan. *Feeding tube and feeder system* yang dilengkapi dengan *pneumatic cylinder* digunakan untuk mengeluarkan benda dari tabung penampungan menuju konveyor untuk disortir. Setelah disortir oleh sensor, maka selanjutnya benda yang tidak sesuai dengan ciri yang diinginkan akan didorong menuju kotak pembuangan, sedangkan benda yang diinginkan akan jalan terus di konveyor menuju ke modul selanjutnya. Alat pendorong yang digunakan berupa *pneumatic cylinder*. Beberapa jenis sensor yang digunakan pada modul ini antara lain sensor *proximity* (*inductive* dan *capacitive*), sensor *fiber optic*, dan *photo sensor*. *Control Panel* juga disediakan untuk memudahkan interaksi proses kontrol yang dilakukan operator terhadap modul ini, baik secara otomatis maupun secara manual. Bentuk fisik dari *separation module* dapat dilihat pada Gambar 2.2 di bawah ini.



Gambar 2.2 Separation Module

2.1.2 *Pick & Place Module*

Pick & place module berperan dalam memindahkan benda dari konveyor di *separation module* menuju *stopper module* dengan menggunakan *vacuum* yang digerakkan oleh silinder yang dapat bergerak secara rotasi maupun vertikal. Pada *pick & place module* juga terdapat beberapa komponen yang menjadi aktuator diantaranya yaitu *vacuum*, *rotational cylinder* serta *vertical cylinder*. *Vacuum* berfungsi menghisap benda untuk memindahkan benda. *Rotational cylinder* berfungsi untuk menggerakkan *vacuum pad* bergerak searah atau berlawanan dengan jarum jam. *Vertical Cylinder* berfungsi untuk menggerakkan *vacuum pad* bergerak ke atas atau ke bawah. Pada *pick & place module* dilengkapi dengan *warning light* berwarna merah, kuning dan hijau yang berfungsi untuk mengindikasikan operasi dari sistem *factory automatic trainer*. Sistem yang sedang beroperasi diindikasikan dengan *warning light* berwarna merah, untuk sistem yang sedang dalam proses *drilling* pada *stopper module* diindikasikan dengan *warning light* berwarna kuning dan untuk sistem yang diberhentikan akan diindikasikan dengan *warning light* berwarna hijau. Komponen lain pada *pick and place module* yaitu *solenoid valve 5/2 way* [3]. Bentuk fisik dari *pick & place module* dapat dilihat pada Gambar 2.3 di bawah ini. Pada *pick & place module* terdapat sebuah *control panel* untuk memudahkan interaksi proses kontrol yang dilakukan operator terhadap modul ini, baik secara otomatis maupun secara manual.



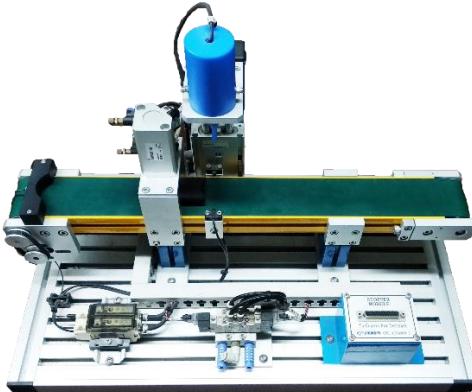
Gambar 2.3 *Pick & Place Module*

2.1.3 *Stopper Module*

Stopper module merupakan kelanjutan proses dari benda yang telah berhasil diseleksi pada *separation module* serta telah dipindahkan menuju konveyor selanjutnya dengan bantuan *pick & place module*. Komponen pada *stopper module* diantaranya terdapat *Conveyor Belt*, Motor DC,

Stopper, Mesin Drilling, Fiber Optic Sensor serta Selenoid Valve 5/2 way [3].

Pada *stopper module* terdapat konveyor yang berfungsi menjalankan benda dan akan diberhentikan pada saat *optical sensor (work point)* mendeteksi keberadaan benda di ujung konveyor. Benda yang diberhentikan oleh *stopper* akan dilakukan proses *drilling* untuk dapat melanjutkan ke proses berikutnya. Sensor *photo fiber* yang akan mendeteksi benda kerja berada pada ujung konveyor. Pada *stopper module* terdapat sebuah *control panel* untuk memudahkan interaksi proses kontrol yang dilakukan operator terhadap modul ini, baik secara otomatis maupun secara manual. Gambar fisik dari modul *stopper* dapat dilihat pada Gambar 2.4 di bawah ini.



Gambar 2.4 Stopper Module

2.1.4 Line Movement Module

Pada *line movement module* merupakan kelanjutan proses dari benda yang berhasil menyelesaikan keseluruhan proses pada ketiga modul sebelumnya. Dalam pengoperasian *line movement module* akan terdapat 2 perlakuan berbeda untuk benda berwarna biru dan juga benda berwarna hitam. Benda berwarna hitam akan dialokasikan menuju pangkal dari modul *separation* sedangkan untuk benda berwarna biru akan diletakkan pada kotak di tengah pergerakan dari *finger cylinder*.

Komponen yang terdapat pada *line movement module* diantaranya *Horizontal Cylinder*, *Vertical Cylinder*, *Finger Cylinder* dan *Selenoid Valve 5/2 way* [3]. Pada *line movement module* terdapat sebuah *control panel* untuk memudahkan interaksi proses kontrol yang dilakukan

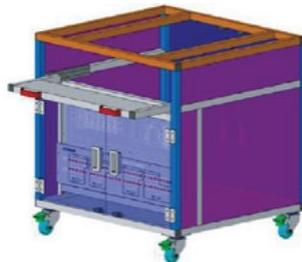
operator terhadap modul ini, baik secara otomatis maupun secara manual. Bentuk fisik dari modul *line movement* dapat dilihat pada Gambar 2.5 di bawah ini.



Gambar 2.5 Line Movement Module

2.1.5 Control Unit

Modul ini berfungsi sebagai pusat kendali dari keseluruhan modul yang ada. Dilengkapi dengan *circuit breaker* dan *control panel* untuk memudahkan interaksi proses kontrol yang dilakukan operator. Terdapat *main panel* serta *control panel* di dalam *control unit*. *Main panel* terdiri dari alamat *input* dan *output* dari masing – masing modul yang dihubungkan dengan PLC. Sedangkan *control panel* terdiri *button* yang terdiri dari *input* dan *output* dari masing – masing modul, ketika tombol ditekan akan terintegrasi dengan aktuator yang terhubung. Pusat kendali yang digunakan berupa PLC (*Programmable Logic Controller*) LG GLOFA tipe GM4 dengan jumlah slot untuk *input* dan *output* masing-masing sebanyak 32 slot. Jenis pemrograman yang digunakan pada PLC ini adalah diagram *ladder* [3]. Bentuk fisik untuk *control unit* pada *factory automatic trainer* dapat dilihat pada Gambar 2.6 berikut ini.



Gambar 2.6 Control Unit [3]

2.1.6 Benda Kerja *Factory Automatic Trainer*

Benda kerja di *Factory Automatic Trainer* terdapat 3 jenis yaitu berwarna *silver*, hitam dan biru seperti pada Gambar 2.7. Benda dengan warna *silver* merupakan logam sedangkan untuk benda warna hitam dan biru merupakan benda non logam.



Gambar 2.7 Benda Kerja *Factory Automatic Trainer*

Masing – masing dari benda memiliki bahan material yang berbeda serta warna yang berbeda yang mempengaruhi masing – masing sensor. Spesifikasi mengenai benda kerja pada *factory automatic trainer* dapat ditinjau pada Tabel 2.1 di bawah ini.

Tabel 2.1 Spesifikasi Benda Kerja

Warna Benda	Bahan Dasar	Ukuran		Deteksi Sensor		
		Diameter	Tinggi	Proximity Inductive	Photo sensor	Proximity Capacitive
Silver	Logam	4 cm	2,5 cm	✓	✓	✓
Biru				-	✓	✓
Hitam				-	-	✓

2.2 Perangkat Komponen Elektrik dan Pneumatik pada *Factory Automatic Trainer*

Dalam *factory automatic trainer* terdapat beberapa perangkat komponen elektrik dan pneumatik yang menunjang pengoperasian dari sistem. Terdapat beberapa komponen yang berfungsi menjadi aktuator serta sensor diantaranya yaitu *proximity inductive sensor*, *proximity capacitive sensor*, *photoelectric sensor*, *fiber optic sensor*, *reed switch*, *belt conveyor*, *pneumatic cylinder*, *rodless cylinder*, *rotary cylinder*, *finger cylinder*, *solenoid valve 5/2 way*, *vacuum pad*.

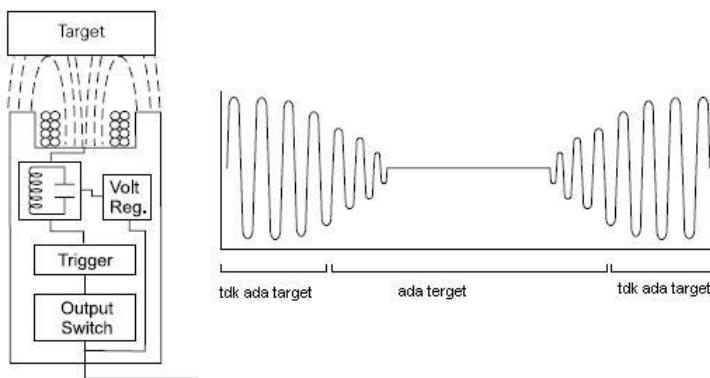
2.2.1 Proximity Inductive Sensor

Proximity inductive sensor dapat mendeteksi target berbahan metal yang mendekat ke sensor tanpa adanya sentuhan fisik. Bentuk fisik dari *proximity inductive sensor* dapat dilihat pada Gambar 2.8 di bawah ini.



Gambar 2.8 Proximity Inductive Sensor [4]

Ada beberapa jenis sensor *proximity* berdasarkan cara kerjanya, yaitu menggunakan induksi elektromagnetik, magnet, dan perubahan kapasitansi. Pada sensor dengan menggunakan induksi elektromagnetik, medan magnet frekuensi tinggi dihasilkan oleh lilitan pada rangkaian osilasi. Prinsip kerja pada sensor *proximity* induktif dapat dilihat pada Gambar 2.9 di bawah ini.



Gambar 2.9 Prinsip Kerja Proximity Inductive Sensor

Ketika target mendekat medan magnet, arus induksi atau *eddy current* mengalir pada target karena induksi elektromagnetik. Semakin dekat target dengan sensor maka arus induksi semakin besar dan

mengakibatkan beban pada rangkaian osilasi meningkat. Sensor mendeteksi perubahan amplitudo osilasi pada rangkaian dan menghasilkan sinyal deteksi. Ketika tidak ada benda kerja amplitudo sinyal lebih besar dibanding ketika ada benda kerja. Perbedaan inilah yang digunakan sebagai penentu ada tidaknya benda. Akan tetapi, untuk benda bersifat *non-ferrous* seperti alumunium, cara kerjanya sedikit berbeda. Pada metal *non-ferrous* akan terjadi kehilangan energi karena arus induksi yang mengalir pada target sehingga menyebabkan perubahan frekuensi osilasi. Ketika benda *non-ferrous* mendekat, frekuensi osilasi meningkat. Sementara itu ketika benda *ferrous* seperti besi mendekat, frekuensi osilasi berkurang. Ketika osilasi frekuensi lebih besar dibanding frekuensi referensi maka sensor akan mengeluarkan sinyal deteksi [5].

2.2.2 Proximity Capacitive Sensor

Proximity capacitive sensor berbeda dengan *proximity inductive sensor*. Sensor ini mampu mendeteksi semua jenis benda, tidak hanya mendeteksi benda logam. Contohnya bisa air, gelas, plastik, kayu, logam, dan lain – lain [6]. Bentuk fisik sensor *proximity capacitive sensor* ditunjukkan pada Gambar 2.10 di bawah ini.

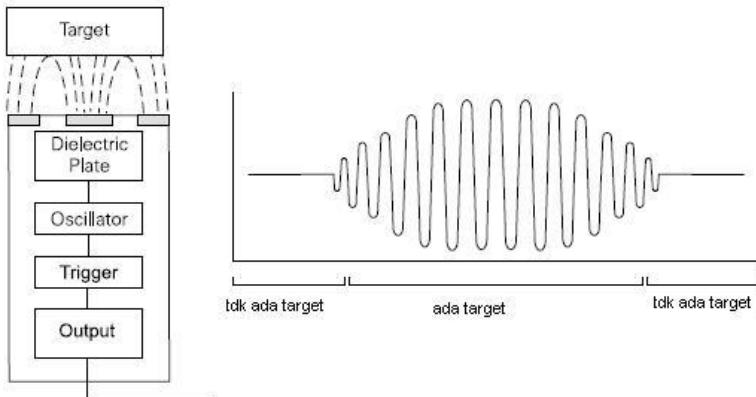


Gambar 2.10 *Proximity Capacitive Sensor* [4]

Prinsip kerja dari *proximity capacitive* adalah dengan cara mengukur perubahan kapasitansi medan listrik sebuah kapasitor yang disebabkan oleh sebuah objek yang mendekatinya. Prinsip kerja fisikanya adalah sama dengan kapasitor pelat: semakin pendek jarak antara pelat, semakin besar luas pelat atau semakin tinggi permitivitas dielektrik menghasilkan kapasitansi yang lebih tinggi.

Medan listrik berosilasi digunakan untuk mendeteksi perubahan kapasitansi ini dengan menghasilkan arus. Ketika bidang berubah melebihi batas tertentu, *output* sensor (*output relay* atau *transistor*) diaktifkan. Sensor kapasitif ada 2 jenis. Jenis yang paling umum dapat mendeteksi objek apa pun tanpa memandang materi. Dalam jenis ini,

kedua pelat dibangun ke dalam sensor. Karena pelat memiliki area tertentu dan jarak di antara mereka ditetapkan, kapasitansi hanya dapat diubah dengan perubahan dalam permitivitas. Tanpa objek oleh sensor, dielektrik hanya terdiri dari udara dan permitivitas relatif sama dengan 1. Dengan objek di tempatnya di sebelah sensor, permitivitas relatif meningkat dan kapasitansi meningkat [6]. Prinsip kerja dari *capacitive proximity sensor* dapat dilihat pada Gambar 2.11 di bawah ini.



Gambar 2.11 Prinsip Kerja *Capacitive Proximity Sensor*

Jenis utama lainnya dapat mendeteksi benda konduktif seperti logam atau air dan juga berfungsi di jarak yang lebih jauh. Ini dapat terjadi bahkan ketika objek berada di belakang bahan isolasi. Contoh aplikasi semacam itu adalah deteksi tingkat cairan dalam tangki [6].

2.2.3 Sensor Fotoelektrik

Sensor ini menggunakan elemen peka cahaya untuk mendeteksi benda-benda dan terdiri dari *transmitter/emitor* (sumber cahaya) dan penerima (*receiver*). Bentuk fisik dari sensor fotoelektrik ditunjukkan pada Gambar 2.12 di bawah ini.



Gambar 2.12 Sensor Fotoelektrik [7]

Terdapat empat jenis sensor fotoelektrik yang tersedia :

a. Pemantulan Langsung (*Direct Reflection*).

Transmitter dan *receiver* ditempatkan bersama-sama dan menggunakan cahaya yang dipantulkan langsung dari objek untuk melakukan deteksi. Pemilihan *photosensor* jenis ini harus mempertimbangkan warna dan tipe permukaan objek (kasar, licin, buram, terang). Dengan permukaan buram, jarak *sensing* akan dipengaruhi oleh warna objek. Warna-warna terang berpengaruh terhadap jarak *sensing* maksimum dan warna gelap berpengaruh terhadap jarak *sensing* minimum. Jika permukaan objek mengkilap, efek permukaan yang lebih penting daripada warna.

b. Pemantulan dengan reflektor (*Reflection with Reflector*)

Transmitter dan *receiver* ditempatkan bersama-sama dan membutuhkan reflektor. Obyek terdeteksi karena memotong cahaya antara sensor dan reflektor sehingga *receiver* tidak menerima cahaya. Sensor ini memungkinkan jarak *sensing* lebih jauh. Dengan adanya reflektor sinar yang dipancarkan akan dipantulkan sepenuhnya ke *receiver*.

c. Pemantulan terpolarisasi dengan reflektor (*Polarized Reflection with Reflector*).

Hampir sama dengan Pemantulan dengan reflektor, sensor fotoelektrik ini menggunakan perangkat anti-refleks. Jadi reflektor tidak mengkilap. Sensor ini mendasarkan fungsi pada sebuah pita cahaya terpolarisasi, memberikan keuntungan dan deteksi akurat bahkan ketika permukaan obyek sangat mengkilap. Data teknik tidak ada karena sangat dipengaruhi oleh pemantulan acak (benda apa saja).

d. *Through Beam*.

Transmitter dan *Receiver* ditempatkan secara terpisah dan deteksi obyek terjadi ketika memotong sinar antara *transmitter* dan *receiver* sehingga *receiver* kehilangan cahaya sesaat. Sensor fotoelektrik ini memiliki jarak *sensing* terpanjang [8].

2.2.4 Fiber Optic Sensor

Sensor optik adalah piranti masukan suatu sistem kendali otomatis yang dibuat dengan komponen optikal yang berfungsi untuk menangkap/mengumpulkan informasi mengenai kondisi lingkungan yang ada di sekitar sensor dengan bantuan cahaya. Bentuk fisik dari *fiber optic sensor* dapat dilihat pada Gambar 2.13 di bawah ini.



Gambar 2.13 Fiber Optic Sensor

Komponen yang sering digunakan dalam pembuatan sensor optik adalah *Light Dependent Resistor* (LDR), *photo-diode*, dan *photo-transistor*. Oleh sebab sensor optik adalah sensor yang bekerja dengan bantuan cahaya, maka proses pensaklaran tidak dapat dilakukan oleh komponen saklar mekanik. Pada sensor optik, proses pensaklaran dilakukan oleh komponen yang bekerja dengan bantuan cahaya, yaitu komponen optik (LDR/*photo-diode*/*photo-transistor*). Dari sini maka dapat disimpulkan bahwa sistem kerja sensor optik menyerupai sebuah saklar, yaitu menghubungkan dan memutuskan aliran arus listrik. Perbedaannya, proses pensaklarannya komponen saklar membutuhkan bantuan manusia sedangkan komponen optik proses pensaklarannya dibantu dengan cahaya, yaitu cahaya yang mengenai bagian *photo-conductive* komponen optik [9].

2.2.5 Reed Switch

Reed Switch adalah sebuah saklar listrik yang dioperasikan oleh medan magnet. Benda ini ditemukan di *Bell Telephone Laboratories* pada tahun 1936 oleh WB Ellwood. Dalam keadaan biasa kontak terbuka, kontak akan menutup ketika medan magnet terdeteksi. Setelah medan magnet ditarik dari saklar, saklar *reed* akan kembali ke posisi semula.

Reed switch atau sensor magnet berfungsi untuk mendeteksi gerakan dari penggerak silinder naik, turun atau maju, mundur. Cara kerja dari sensor ini adalah ketika ada medan magnet mengenai bagian depan sensor, maka sensor akan bekerja sehingga menghubungkan kontaknya, medan magnet ini terdapat dari bagian dalam silinder sebelah atas dan bawah kemudian posisi sensor dengan badan silinder melekat pada saat silinder bergerak naik atau turun maka akan ada medan magnet yang

mengenai *reed switch*. Bentuk fisik dari *reed switch* ditunjukkan pada Gambar 2.14 di bawah ini.



Gambar 2.14 *Reed Switch*

2.2.6 *Belt Conveyor*

Konveyor adalah suatu sistem mekanik yang mempunyai fungsi memindahkan barang dari satu tempat ke tempat yang lain. Konveyor banyak dipakai di industri untuk transportasi barang yang jumlahnya sangat banyak dan berkelanjutan. Konveyor terutama berguna dalam aplikasi yang melibatkan transportasi bahan berat atau besar. Sistem konveyor memungkinkan transportasi cepat dan efisien untuk berbagai bahan. Banyak jenis sistem konveyor yang tersedia dan digunakan sesuai dengan kebutuhan berbagai industri yang berbeda. Dalam kondisi tertentu, konveyor banyak dipakai karena mempunyai nilai ekonomis dibanding transportasi berat seperti truk dan mobil pengangkut. Bentuk fisik konveyor pada *factory automatic trainer* dapat dilihat pada Gambar 2.15 di bawah ini.



Gambar 2.15 Konveyor pada *Factory Automatic Trainer*

2.2.7 Pneumatic Cylinder

Silinder pneumatik adalah aktuator atau perangkat mekanis yang menggunakan kekuatan udara bertekanan (udara yang terkompresi) untuk menghasilkan kekuatan dalam gerakan bolak – balik piston secara linier (gerakan keluar - masuk). Pada *Factory Automatic Trainer* silinder *insert* yang digunakan untuk memasukkan benda kerja ke konveyor serta silinder *eject* yang digunakan untuk membuang benda kerja berbahan logam menggunakan silinder jenis ini seperti pada Gambar 2.16 di bawah ini.



Gambar 2.16 Silinder Pneumatik pada *Factory Automatic Trainer*

Berikut merupakan dua jenis dari silinder pneumatik yaitu:

- a) Silinder kerja tunggal (*single acting cylinder*)

Silinder kerja tunggal (*single acting cylinder*) merupakan jenis silinder yang hanya memiliki satu *port* untuk masuknya udara bertekanan. Silinder ini menggunakan kekuatan udara bertekanan untuk mendorong ataupun menekan piston dalam satu arah saja (umumnya keluar). Dan menggunakan pegas pada sisi yang lain untuk mendorong piston kembali pada posisi semula. Silinder kerja tunggal mempunyai *seal* piston tunggal yang dipasang pada sisi suplai udara bertekanan. Pembuangan udara pada sisi batang piston silinder dikeluarkan ke atmosfir melalui saluran pembuangan. Jika lubang pembuangan tidak diproteksi dengan sebuah penyaring akan memungkinkan masuknya partikel halus dari debu ke dalam silinder yang bisa merusak *seal*. Apabila lubang pembuangan ini tertutup akan membatasi atau menghentikan udara yang akan dibuang pada saat silinder gerakan keluar dan gerakan akan menjadi terhenti. *Seal* terbuat dari bahan yang fleksibel yang ditanamkan di dalam piston dari logam atau plastik. Selama bergerak permukaan *seal* bergeser dengan permukaan silinder.

b) Silinder kerja ganda (*double acting cylinder*)

Silinder kerja ganda (*double acting cylinder*) merupakan silinder yang memiliki dua *port* untuk *instroke* dan *outstroke*. Silinder jenis ini menggunakan kekuatan udara bertekanan untuk mendorong piston keluar dan mendorong piston untuk kembali ke posisi awal (menarik kedalam). Sehingga silinder ini membutuhkan lebih banyak udara dan katup pengontrol arah yang lebih kompleks bila dibandingkan dengan silinder kerja tunggal. Konstruksi silinder kerja ganda adalah sama dengan silinder kerja tunggal, tetapi tidak mempunyai pegas pengembali. Silinder kerja ganda mempunyai dua saluran (saluran masukan dan saluran pembuangan) [10].

2.2.8 Rodless Cylinder

Rodless Cylinder adalah silinder kerja ganda yang terdiri dari tabung bulat dan piston tanpa batang. Sebuah penggeser menempel pada tabung. Piston dalam tabung dapat bergerak bebas berdasarkan tekanan udara yang diberikan padanya. Pada piston dipasang magnet permanen. Kopling magnet dihasilkan antara penggeser dan piston. Bila piston bergerak maka penggeser mengikutinya secara serempak menggerakkan penggesernya [10]. Gambar fisik untuk *rodless cylinder* ditunjukkan pada Gambar 2.17 berikut ini.



Gambar 2.17 Rodless Cylinder [10]

Prinsip kerja silinder ini adalah ketika ada *supply* udara dari kiri maka udara akan memenuhi ruang sebelah kiri sehingga silinder akan teraktuasi / bergerak kearah kanan dan secara otomatis udara yang memenuhi ruang kanan akan keluar dan sebaliknya apabila ada *supply* dari kanan, maka udara akan memenuhi ruang sebelah kanan sehingga silinder akan teraktuasi bergerak ke arah kiri dan secara otomatis udara yang memenuhi ruang kiri akan keluar.

Pada Gambar 2.18 ditunjukkan *rodless cylinder* pada *factory automatic trainer* yang terletak pada *line movement module*. *Rodless cylinder* berperan dalam pergerakan silinder secara horizontal.



Gambar 2.18 Rodless Cylinder pada Factory Automatic Trainer

2.2.9 Rotary Cylinder

Silinder putar adalah aktuator yang menghasilkan gerakan berputar dari 0° sampai 360°. Batang piston mempunyai profil bergigi. Profil bergigi ini menggerakkan roda gigi. Gerakan linear piston diubah menjadi gerakan putar dari roda gigi dengan sudut putaran dari 0° sampai 360°. Untuk pemakaian *gripper* atau *suction cup*, sumber udara dapat dialirkkan melalui batang piston yang berlubang [10]. Gambar 2.19 menunjukkan penggunaan *rotary cylinder* pada *factory automatic trainer*.



Gambar 2.19 Rotary Cylinder pada Factory Automatic Trainer

2.2.10 Finger Cylinder

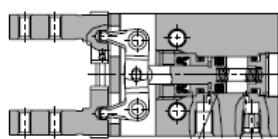
Finger cylinder terdapat pada *line movement module plant factory automatic trainer* (FAT) seperti pada Gambar 2.20 digunakan untuk memindahkan benda dengan cara menjepit benda menggunakan *Finger grip*. Pneumatik jenis *finger* ini mampu bergerak dengan *rotary* yang berdasarkan roda gigi.



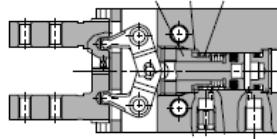
Gambar 2.20 Finger Cylinder pada Factory Automatic Trainer

Cara kerja *finger grip* adalah saat udara bertekanan masuk menjadi gerakan mekanik dari roda gigi agar dapat menjepit seperti pada Gambar 2.21 di bawah ini.

Single acting/Normally open



Single acting/Normally closed



Gambar 2.21 Single Acting (NO) dan Single Acting (NC) [10]

2.2.11 Solenoid Valve 5/2 Way

Solenoid valve adalah katup yang berfungsi sebagai saluran untuk mengeluarkan udara bertekanan ke silinder. Sebagai elemen kontrol, katup ini memiliki sebuah piston kontrol yang dengan gerakan horizontalnya menghubungkan atau memisahkan saluran yang sesuai. Pada *plant factory automatic trainer* menggunakan *solenoid valve 5/2 way* seperti pada Gambar 2.22 berikut ini.

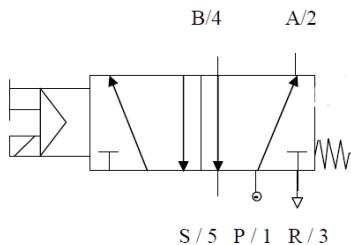


Gambar 2.22 Solenoid Valve 5/2 Way

Solenoid valve 5/2 terdapat 2 jenis yaitu:

a. *Single Acting*

Solenoid jenis ini mempunyai satu *solenoid* dan *spring* yang mana fungsi dari *spring* itu sendiri adalah sebagai penarik batang pelat yang ada pada *valve* agar pada saat tidak ada arus ada tegangan *supply* yang masuk maka *valve* akan nonaktif. Saat solenoid valve ini mendapat tegangan dan arus, induksi yang terjadi dalam solenoid tersebut menarik batang pelat yang mempunyai gaya tarik lebih besar dari gaya *spring* dan akibatnya sumber *supply* masukan menyalurkan sumbernya pada keluaran yang lainnya, akibatnya keluaran yang satunya dapat aktif. Symbol untuk *solenoid valve single acting* pada pneumatik digambarkan pada Gambar 2.23 berikut ini.

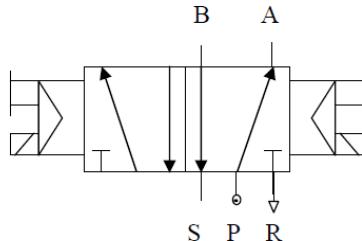


Gambar 2.23 Solenoid valve Single acting [10]

b. *Double Acting*

Solenoid valve ini mempunyai cara kerja dan konstruksi bagian dalamnya sama dengan *valve pneumatic 5/2-Way single acting* dan yang menjadi perbedaan dengan *input* dari *trigger*, yang mana pada jenis *solenoid valve* ini terdapat dua keluaran yang tiap keluarannya tersebut mempunyai masing - masing *solenoid*, sehingga cara kerjanya

berdasarkan *input trigger solenoid* dan pada saat *input 1* tegangan dan arus aktif maka silinder akan maju sedangkan jika *input 2* aktif dan *input 1* tidak aktif maka silinder akan mundur. Symbol untuk *solenoid valve double acting* pada pneumatik digambarkan pada Gambar 2.24 berikut ini.



Gambar 2.24 Solenoid valve Double acting [10]

2.2.12 Vacuum Pad

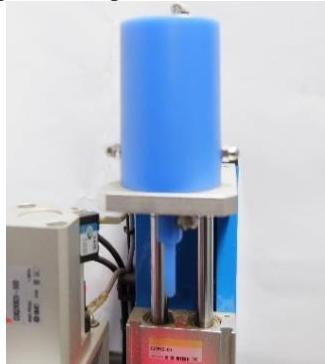
Vacuum suction cup/Vacuum pad adalah aktuator yang menghasilkan udara vakum yang dipakai untuk mengangkat suatu benda. *suction cup* terbuat dari bahan elastis, fleksibel dan memiliki permukaan melengkung. Ketika pusat cangkir penyedot ditekan terhadap permukaan yang datar dan tidak berpori maka benda akan terhisap dan terangkat menuju tempat yang diinginkan. *Vacuum suction cup* terdiri dari *vacuum suction nozzle* dan *sucker* (penghisap). Bentuk fisik dari *vacuum pad* pada *factory automatic trainer* dapat dilihat pada Gambar 2.25 berikut ini.



Gambar 2.25 Vacuum Pad

2.2.13 Drilling Machine

Proses *drilling* adalah proses permesinan untuk membuat lubang bulat pada benda kerja. *Drilling* biasanya dilakukan memakai pahat silindris yang memiliki dua ujung potong yang disebut *drill*. Pahat diputar pada porosnya dan diumpulkan pada benda kerja yang diam sehingga menghasilkan lubang berdiameter sama dengan diameter pahat. Mesin yang digunakan disebut *drill press*, tetapi mesin lain dapat juga digunakan untuk proses ini. Lubang yang dihasilkan dapat berupa lubang tembus (*through holes*) dan tak tembus (*blind holes*). Terdapat beberapa jenis yang serupa dengan *drilling machine* dengan fungsi sama yakni untuk proses pembuatan lubang dengan mata bor, namun dengan adanya modifikasi sesuai dengan kebutuhan maka berikut merupakan proses *drilling* beserta fungsinya. Bentuk *drilling machine* pada *factory automatic trainer* dapat dilihat pada Gambar 2.26 berikut ini.



Gambar 2.26 Drilling Machine

2.2.14 Warning Light

Warning Light umumnya digunakan pada peralatan di industri manufaktur dan lingkungan pengendalian proses untuk memberikan indikator visual dan suara sehingga operator dapat melihat dan mendengar keadaan mesin atau proses secara keseluruhan. *Warning light* digunakan dalam aplikasi yang sama seperti lampu suar, namun informasi yang ditampilkan mencakup lebih banyak kondisi dan proses mesin [1]. *Warning light* secara visual ditunjukkan pada Gambar 2.27 berikut ini.



Gambar 2.27 Warning Light

2.3 PLC (*Programmable Logic Controller*)

Programmable Logic Controller secara umum digunakan di teknologi pengendalian proses di industri. PLC adalah *grade industrial* komputer yang dapat diprogram untuk fungsi kontrol. PLC menghilangkan banyak pengkabelan yang ada di relai konvensional.

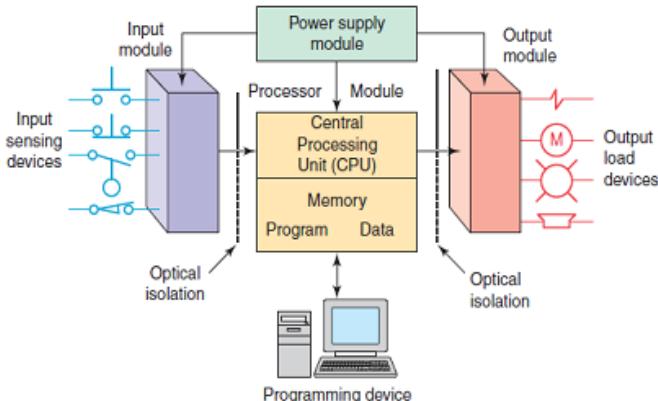
Pada tahun 1960, relai elektromekanik, *timer* dan *counter* dipakai sebagai standard. Banyak panel kontrol berisi ratusan peralatan relai juga bermil mil kabelnya. Reliabilitinya rendah dan biaya perawatannya tinggi. Biaya tinggi untuk memodifikasi atau *upgrade panel control*. Pada tahun 1968 divisi *hidramatic* di general motor menggunakan peralatan yang sekarang kita kenal sebagai PLC [6]. Bentuk fisik dari PLC ditunjukkan pada Gambar 2.28 berikut ini.



Gambar 2.28 PLC LG Glofa GM4

PLC sendiri terdiri dari dua jenis, yaitu *compact* dan *modular*. PLC *compact* adalah jenis PLC yang seluruh komponennya menjadi satu dan

tidak dapat ditambah lagi jumlah masukan dan keluarannya. Sebaliknya, PLC *modular* adalah jenis PLC yang komponennya terpisah menjadi beberapa modul-modul, sehingga dapat ditambahkan lagi modul *input* dan *outputnya*. Pada dasarnya PLC mempunyai beberapa bagian utama yaitu ditunjukkan pada Gambar 2.29 di bawah ini.



Gambar 2.29 Bagian – bagian PLC [6]

a. *Input dan Output*

Input yang akan masuk ke dalam CPU berupa sinyal dari sensor atau transduser. Sinyal dari sensor ini terdapat dua jenis, yaitu: sinyal diskrit dan sinyal analog. Sinyal diskrit berupa saklar dimana hanya sebuah *on/off* (1 atau 0), biasanya dalam bentuk *push button*, *limit switch* dan *level sensor*. Sedangkan sinyal analog menggunakan rentang nilai antara nol hingga skala penuh, biasanya dalam bentuk potensiometer. Hasil pemrosesan data yang diolah pada CPU akan berupa sinyal keluaran digital yang dikirim ke modul *output* untuk menjalankan aktuator. Aktuator ini dapat berupa motor listrik, *solenoid*, *heater*, *led display*, *injector*, pompa dan lain-lain. Aktuator ini akan berfungsi sesuai instruksi dari CPU.

b. CPU (*Central Processing Unit*)

Semua aktivitas atau pemprosesan data yang diambil dari sensor (data *input*) terjadi pada *Central Processing Unit* (CPU). CPU ini memiliki tiga bagian utama, yaitu: *Processor*, *Memory System* dan *Power Supply*. *Processor* akan memproses sinyal input secara aritmatik dan *logic*, yaitu: melakukan operasi logika, *sequential*, *timer*, *counter* dan mengolah fungsi-fungsi yang

diinginkan berdasarkan program yang telah ditentukan. Selain itu, *processor* juga mengolah program yang ada di dalam memori, serta mengatur komunikasi antara *input-output*, memori dengan *processor* itu sendiri.

c. Unit catu daya

Unit ini diperlukan untuk mengonversikan tegangan sumber menjadi tegangan rendah DC. (5V) yang dibutuhkan oleh prosesor dan rangkaian - rangkaian di dalam modul - modul antarmuka *input* dan *output*.

d. Perangkat pemrograman

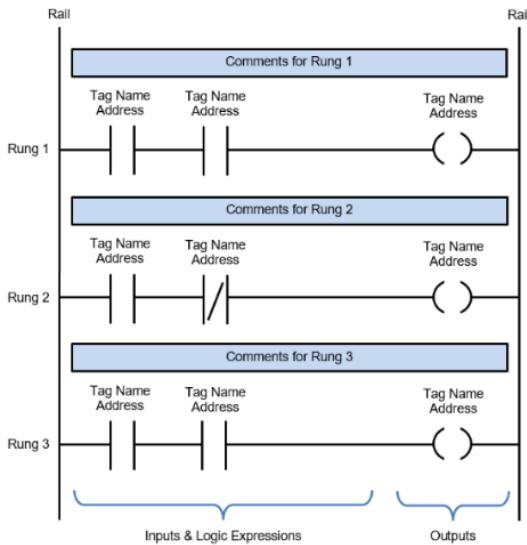
Dipergunakan untuk memasukkan program yang dibutuhkan kedalam memori. Program tersebut dibuat dengan menggunakan perangkat ini dan kemudian dipindahkan kedalam unit memori PLC. Pada PLC terdapat beberapa jenis bahasa pemrograman, antara lain *Ladder Diagram* (LAD), *Function Block Diagram* (FBD), dan *Statement List* (STL).

e. Unit memori

Unit ini merupakan tempat dimana program yang digunakan untuk melaksanakan tindakan - tindakan kendali oleh mikroprosesor yang disimpan [6].

2.4 Diagram Ladder

Bahasa pemrograman ini dinamai *Ladder Diagram* karena faktanya menyerupai seperti struktur tangga, dengan dua garis vertikal dinamai rel, yang dihubungkan oleh anak tangga horizontal dinamakan sirkuit. Setiap anak tangga terdiri dari satu atau lebih banyak kondisi *input* yang mengeksekusi satu instruksi *output* atau satu kondisi *input* yang mengeksekusi satu atau lebih *output* instruksi. Contoh dari konstruksi diagram *ladder* dapat dilihat pada Gambar 2.30 berikut ini.



Gambar 2.30 Ladder Diagram [6]

Bagian-bagian dari lader diagram dalam pemrograman PLC.

- **Bus bar**
Merupakan garis tebal di sisi kiri dan kanan dari ladder diagram, yang mana merupakan simbol dari kutub (+) dan kutub (-) yang akan mengalirkan listrik ke komponen-komponen yang akan dipasang dalam ladder diagram tersebut.
- **Input**
Merupakan masukan dari luar PLC, baik dari *Switch*, *Sensor*, *Relay*, *Timer*, *Potentiometer* ataupun peralatan listrik yang lain, yang secara fisik ada di rangkaian listrik dari mesin, yang dihubungkan ke unit *Input* PLC, bisa berupa *digital input* maupun *analog input*. Biasanya dilambangkan dengan kontak NO dan/atau NC yang berfungsi sebagai syarat untuk berlakunya suatu operasi yang kita inginkan.
- **Output**
Merupakan hasil keluaran dari PLC, yang mana bisa berupa *digital output* maupun *analog output*, yang bisa langsung dihubungkan kerangkaian listrik yang lain di mesin tersebut melalui *unit Output* PLC.

- *Internal relay*

Merupakan *relay* memori dari PLC itu sendiri, dimana bisa berupa *relay*, *timer*, *counter*, atau operasi-operasi logika yang lain. Seperti *Input* dan *Output*, simbol-simbol dari *internal relay* ini cukup beragam dan berbeda antara pabrikan yang satu dengan yang lain.

- *Normally Open*

Normally Open adalah kondisi dimana saat kontak tersebut tidak ditekan maka kontak tersebut dalam kondisi tidak terhubung atau putus. Sebaliknya, saat kontak tersebut ditekan atau bekerja maka kontak tersebut dalam kondisi terhubung.

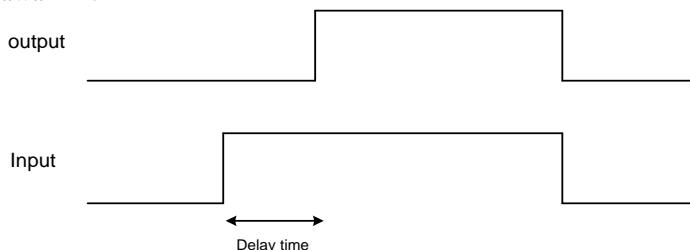
- *Normally Close*

Normally Close adalah kondisi dimana saat kontak tersebut tidak ditekan maka kontak tersebut dalam kondisi terhubung. Sebaliknya, saat kontak tersebut ditekan/bekerja maka kontak tersebut dalam kondisi tidak terhubung atau putus.

Selain tersebut terdapat juga fungsi *timer* dan *counter* yang terdapat pada PLC. *Timer* digunakan sebagai pengatur waktu proses. Terdapat 3 jenis *timer* yang umum digunakan diantaranya yaitu:

- a) *Timer on delay*

Output akan berlogika *high* apabila *input* diberi sinyal dengan logika *high* yang lamanya melebihi setting tundaan waktunya. *Output* akan kembali berlogika *low* saat sinyal *input* berlogika *low*. Isyarat untuk *input* dan *output timer on delay* ditunjukkan pada Gambar 2.31 di bawah ini.



Gambar 2.31 Isyarat *Input & Output Timer on Delay* [11]

- b) *Timer off delay*

Output akan berlogika *low* apabila *input* diberi sinyal dengan logika *low* yang lamanya melebihi *setting* tundaan waktu. Diasumsikan kondisi awal *timer* mendapatkan sinyal *input high* kemudian sinyal *input* tersebut diubah menjadi *low*. Saat diberi *input* diberi sinyal

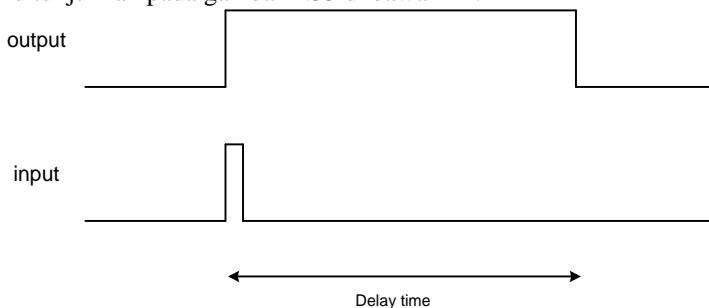
high maka *output high*, kemudian saat input diberi sinyal *low*, maka *output* akan *low* jika lamanya sinyal *input* dengan logika *low* tersebut melebihi *setting* tundaan waktunya. Isyarat *input* dan *output* mode ini ditunjukkan pada gambar 2.32 di bawah ini.



Gambar 2.32 Isyarat *Input & Output Timer off Delay* [11]

c) *Timer pulse*

Output akan berlogika *high* selama *setting* tundaan waktu apabila *input* diberi *trigger* berlogika *high*. Isyarat input dan output mode ini ditunjukkan pada gambar 2.33 di bawah ini.



Gambar 2.33 Isyarat *Input & Output Timer Pulse* [11]

Counter dalam PLC bekerja seperti halnya *counter* mekanik atau elektronik yaitu membandingkan nilai yang terkumpul dengan nilai yang diinginkan, hasil perbandingan digunakan sebagai acuan keluaran. Ada dua jenis counter yang sering digunakan dalam PLC yaitu :

- Counter Up* : menghitung dari 0 sampai nilai yang ditentukan
- Counter Down*: menghitung dari besaran yang ditentukan sampai 0 [12].

2.5 Grafcet

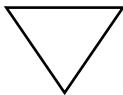
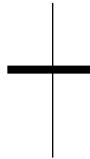
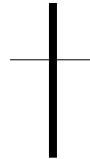
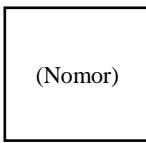
Grafcet ditemukan di Eropa pada tahun 1979. Tim desain yang didirikan oleh *French Association for Economical and Applied Cybernetics* (AFCET) termasuk anggota dari universitas dan industri. GRAFCET merupakan akronim dari GRAphe Fonctionnel de Commande Etape/Transition atau di dalam bahasa inggris *Step Transition Function Charts*. Selama paruh pertama 1980-an, popularitas *grafcet* meningkat pesat di Eropa orang-orang menemukan bahwa itu membantu mereka menyelesaikan masalah otomatisasi secara efisien. Hari ini banyak sekali perusahaan besar dan kecil di seluruh dunia mengandalkan *Grafcet* sebagai kunci untuk otomatisasi yang memiliki produktifitas lebih besar. *Grafcet* diajarkan di universitas - universitas Eropa sebagai bagian integral dari regular program teknik. Metode *Grafcet* didukung oleh standar nasional dan internasional. Dokumen penentu asli adalah Komisi Standar Prancis NFC-03-190. Dengan nama alternatif, *Sequential Function Charts* (SFC), *Grafcet* juga telah distandardisasi oleh Komisi Elektroteknik Internasional di bawah IEC 848. IEC juga telah merilis dokumen pra-standardisasi DIS 1131-3 untuk memberikan panduan dalam implementasi bahasa pemrograman PLC [13].

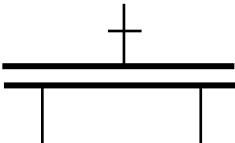
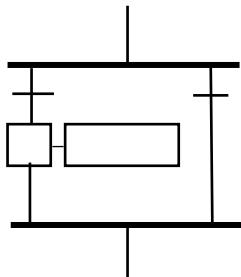
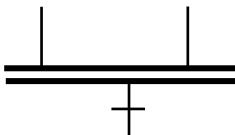
Grafcet merupakan sebuah metode dalam pembuatan *ladder diagram* yang dapat digunakan untuk memodelkan suatu alur proses menjadi sebuah grafis standar yang dapat menggambarkan urutan dari proses tersebut [2]. Dari hasil permodelan tersebut kemudian pengembangannya dapat digunakan dan di implementasikan ke dalam perangkat kontroler PLC.

2.5.1 Bagian – bagian *Grafcet*

Dalam menggambarkan urutan dari suatu proses perlu terlebih dahulu untuk memahami nama, bagian – bagian serta fungsi *grafcet* seperti pada Tabel 2.2 berikut ini. Pengenalan terhadap setiap bagian dari *grafcet* membantu dalam menggambarkan rancangan *grafcet* berdasarkan deskripsi sistem yang ada.

Tabel 2.2 Nama, Bagian-bagian dan Fungsi *Grafset*

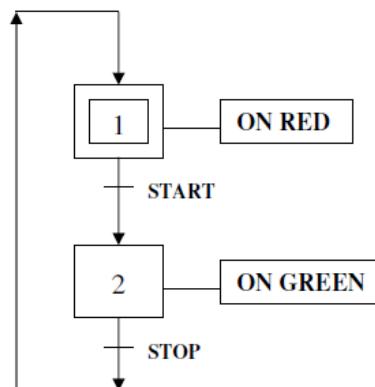
Bagian-bagian <i>grafset</i>	Nama	Fungsi
	<i>Labelled Connector</i>	menunjukkan kontrol dari konektor asal
	<i>Initial Step</i>	mendefinisikan kondisi/ <i>state</i> awal dari sistem otomatis
	<i>Transition/ transisi</i>	antara kondisi/ <i>state/ step</i> tempat kondisi terkait
	<i>Directed link</i>	menghubungkan <i>step</i> ke transisi juga transisi ke <i>step</i>
	<i>Step</i>	Menunjukkan <i>memory</i> yang akan digunakan untuk mengaktifkan <i>output</i>

Bagian-bagian grafceet	Nama	Fungsi
	<i>Output</i>	Keluaran yang ingin dihasilkan dari sebuah sistem
	<i>Simultaneous activation/AND Divergence</i>	Menunjukkan bahwa terdapat satu transisi yang mengaktifkan 2 step atau lebih secara bersamaan
	<i>Start of sequence selection/ OR Divergence</i>	Menunjukkan bahwa terdapat 2 atau lebih kondisi yang mengaktifkan step atau output yang berbeda dan hanya aktif apabila syaratnya terpenuhi.
	<i>End of sequence selection/ OR Convergence</i>	Mengakhiri dari start of sequence selection
	<i>Simultaneous deactivation/AND Convergence</i>	Mengakhiri simultaneous activation

Bagian-bagian grafset	Nama	Fungsi
	<i>Labelled connector</i>	menunjukkan di mana kontrol akan ke konektor tujuan

2.5.2 Konversi *Grafset* menjadi Diagram *Ladder*

Dalam mengkonversi *grafset* menjadi diagram *ladder* prosedur awal adalah mengenali *step* dan transisi yang digunakan. Dalam sebuah Kasus terdapat struktur *grafset* sederhana dimana terdiri dari 2 *step* dan 2 transisi. *Step1* untuk mengaktifkan ON RED dan *Step2* digunakan untuk mengaktifkan ON GREEN. Struktur *grafset* sederhana dapat dilihat pada Gambar 2.34 di bawah ini.



Gambar 2.34 Struktur *Grafset* Sederhana [14]

1st cycle bit muncul ketika tombol *power* dari PLC aktif. *Step1* aktif dikarenakan *step* sebelumnya aktif yaitu *step2* dan ketika tombol *stop* ditekan. Untuk membuat *step1* aktif tanpa harus menahan tombol *stop* digunakan *latching* atau *self holding* dari *step1* dan dinonaktifkan (*reset*) dengan *step* setelahnya yaitu *step2* yang diberi tanda *bar* menandakan komplemen. *Switching function* dari *step1* adalah sebagai berikut.

$$Step1 = 1_{st_cycle_bit} + Step1.\overline{Step2} + Step2.Stop$$

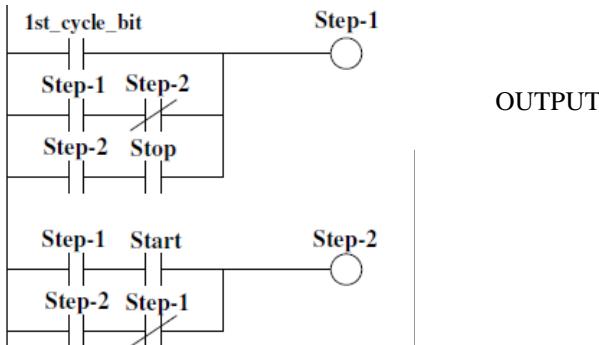
Step2 aktif ketika *step1* aktif dan tombol *start* ditekan. Untuk membuat *step2* aktif tanpa harus menahan tombol *start* digunakan *latching* atau *self holding* dari *step2* dan pemutus dari *step2* adalah *step* setelahnya yaitu *step1* maka dari itu *switching function* untuk *step2* adalah sebagai berikut.

$$\overline{Step2} = Step1.Start + Step2.Step1$$

Keluaran yang diinginkan adalah *step1* untuk mengaktifkan ON RED dan *step2* untuk mengaktifkan ON GREEN dituliskan ke dalam *switching function* untuk bagian *output* adalah sebagai berikut. Dari hasil *switching function* menjadi diagram *ladder* untuk struktur *grafcet* sederhana di atas ditunjukkan melalui Gambar 2.35 di bawah ini.

$$On\ Red = Step1$$

$$On\ Green = Step2$$



Gambar 2.35 Hasil Diagram Ladder pada Grafset Sederhana [14]

2.6 Fluidsim

Festo Fluidsim adalah sebuah *software* yang digunakan untuk simulasi sistem kontrol pneumatik dan hidraulik. Fluid simulation pneumatik ini dikembangkan oleh Festo Didactic, Jerman. *Software* ini merupakan suatu program pendukung untuk demo simulasi aliran fluida (angin) yang khususnya pada sistem rangkaian pneumatik dan hidrolik [15].

Dalam pengoperasian pneumatik pada *software* fluidsim dapat mensimulasikan rangkaian elektrik, rangkaian pneumatik, rangkaian digital serta rangkaian *grafcet*. Rangkaian pneumatik terdiri dari elemen

supply (kompresor), aktuator (silinder), katup (*valve*) serta instrument pengukuran dan sensor. Pada kontrol elektrik terdapat aktuator berupa *solenoid* dan motor, *power supply*, instrumen pengukuran dan sensor, *relay*, *switches*, kontroler serta simbol pada *ladder diagram*. Pada bagian rangkaian digital terdapat beberapa fungsi logika sedangkan pada *grafcet* terdapat bagian bagian *grafcet* seperti transisi, *step*, aksi, *grafcet* I/O serta *simultaneous activation*. Tampilan *software festo fluidsim* dapat dilihat pada Gambar 2.36 berikut ini.



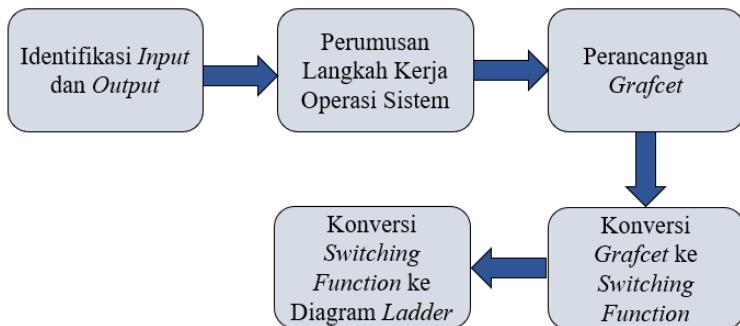
Gambar 2.36 *Software Festo Fluidsim*

-----Halaman ini sengaja dikosongkan---

BAB 3

PERANCANGAN SISTEM

Dalam pembuatan diagram *ladder* dengan menggunakan metode *grafcet* terdapat beberapa prosedur yang perlu diperhatikan. Langkah pertama adalah melakukan identifikasi *input* dan *output* dari *plant*. Langkah kedua adalah membuat perumusan langkah kerja dari operasi sistem yang ingin dijalankan. Langkah ketiga adalah membuat perancangan *grafcet* dari uraian langkah kerja yang telah dibuat. Langkah keempat adalah mengkonversi *grafcet* ke dalam *switching function*. Langkah yang terakhir adalah mengkonversi *switching function* menjadi diagram *ladder*. Prosedur dari perancangan pembuatan diagram *ladder* dengan menggunakan metode *grafcet* ditunjukkan pada Gambar 3.1

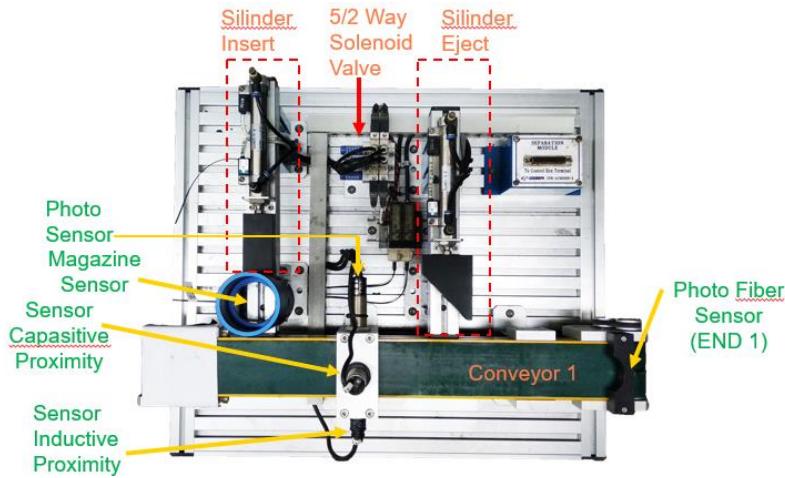


Gambar 3.1 Prosedur Pembuatan Diagram *Ladder* dengan Metode *Grafset*

3.1 Identifikasi *Input* dan *Output* *Factory Automatic Trainer*

Tujuan dari identifikasi *input* dan *output* adalah untuk mengklasifikasikan komponen – komponen dari *plant* yang menjadi *input* dan *output*. Identifikasi *input* dan *output* mempermudah dalam merancang diagram *ladder* dikarenakan alamat *input* dan *output* sudah diarsipkan terlebih dahulu sehingga masing – masing alamat mewakili setiap *input* maupun *output*.

Gambar 3.2 menunjukkan peletakkan *input* dan *output* pada *separation module*. Tulisan berwarna hijau menunjukkan *input* sedangkan untuk tulisan berwarna jingga menunjukkan *output* dari *separation module*.



Gambar 3.2 Letak I/O pada *Separation Module*

Alamat *input* pada *separation module* beserta nama, label dan fungsinya ditunjukkan pada Tabel 3.1 di bawah ini. Pemberian nama label digunakan untuk mempersingkat penamaan pada saat akan membuat diagram *ladder*.

Tabel 3.1 Alamat *Input* pada *Separation Module*

No	Alamat	Nama	Label	Fungsi
1.	%IX0.1.11	Tombol Start	START	Untuk memulai program
2.	%IX0.1.12	Tombol Stop	STOP	Untuk mengentikan program
3.	%IX0.0.6	Magazine Sensor	MAGAZINE	Mendeteksi keberadaan benda kerja pada <i>feeding system</i>
4.	%IX0.0.7	Photo Sensor	PHOTO_SENSOR	Mendeteksi warna benda
5.	%IX0.0.8	Proximity Sensor	PROXIMITY	Mendeteksi keberadaan benda kerja yang

No	Alamat	Nama	Label	Fungsi
				mengandung logam
6.	%IX0.0.9	Capasitive Sensor	CAPASITIVE	Mendeteksi keberadaan benda kerja berbahan logam maupun non logam pada <i>separation</i> proses
7.	%IX0.0.10	Auto Switch (Insert)	AS_INSERT	Mendeteksi silinder <i>insert</i> pada posisi <i>extend</i>
8.	%IX0.0.11	Auto Switch (Insert Return)	AS_INSERT_RETURN	Mendeteksi silinder <i>insert</i> pada posisi <i>retract</i>
9.	%IX0.0.12	Auto Switch (Eject)	AS_EJECT	Mendeteksi silinder <i>eject</i> pada posisi <i>extend</i>
10.	%IX0.0.13	Auto Switch (Eject Return)	AS_EJECT_RETURN	Mendeteksi silinder <i>eject</i> pada posisi <i>retract</i>
11.	%IX0.0.14	Photo Fiber Sensor 1	END1	Mendeteksi benda kerja pada ujung konveyor 1

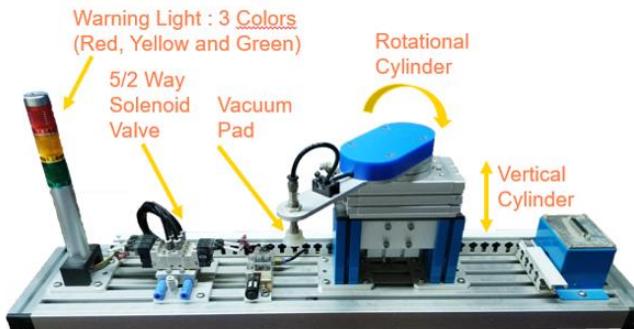
Alamat *output* pada *separation module* beserta nama, label dan fungsinya ditunjukkan pada Tabel 3.2 di bawah ini. Pemberian nama label digunakan untuk mempersingkat penamaan pada saat akan membuat diagram *ladder*.

Tabel 3.2 Alamat *Output* pada *Separation Module*

No	Alamat	Nama	Label	Fungsi
1.	%QX0.2.6	Silinder Insert	INSERT	Mendorong benda kerja memasuki konveyor 1

No	Alamat	Nama	Label	Fungsi
				pada <i>separation module</i> oleh silinder <i>insert</i>
2.	%QX0.2.7	Silinder Insert Return	INSERT_RETURN	Mengembalikan silinder <i>insert</i> ke posisi semula
3.	%QX0.2.8	Silinder Eject	EJECT	Mengeluarkan benda kerja yang mengandung unsur logam (warna <i>silver</i>) oleh silinder <i>eject</i>
4.	%QX0.2.9	Silinder Eject Return	EJECT_RETURN	Mengembalikan silinder <i>eject</i> ke posisi semula
5.	%QX0.2.10	Conveyor1	CONVEYOR1	Membawa benda kerja bergerak pada separation proses dari <i>feeding system</i> menuju sensor <i>photo fiber 1</i> .

Pada *pick and place module* letak aktuator ditunjukkan pada Gambar 3.3 di bawah ini dengan tanda anak panah dan tulisan berwarna jingga.



Gambar 3.3 Letak Aktuator pada *Pick and Place Module*

Alamat *input* pada *pick and place module* beserta nama, label dan fungsinya ditunjukkan pada Tabel 3.3 di bawah ini.

Tabel 3.3 Alamat *Input* pada *Pick and Place Module*

No	Alamat	Nama	Label	Fungsi
1.	%IX0.1.6	Auto Switch (Rotary Cylinder CCW)	AS_CCW	Mendeteksi <i>rotary cylinder</i> bergerak searah dengan arah jarum jam
2.	%IX0.1.7	Auto Switch (Rotary Cylinder CW)	AS_CW	Mendeteksi <i>rotary cylinder</i> bergerak berlawanan dengan arah jarum jam
3.	%IX0.1.8	Auto Switch (Vertical Cylinder Up)	AS_UP_V	Mendeteksi <i>vertical cylinder</i> pada <i>vacuum pad</i> berada di posisi atas
4.	%IX0.1.9	Auto Switch (Vertical Cylinder Down)	AS_DOWN_V	Mendeteksi <i>vertical cylinder</i> pada <i>vacuum pad</i> berada di posisi bawah
5.	%IX0.1.10	Auto Switch (Vacuum On)	AS_VACUUM	Mendeteksi <i>vacuum</i> sedang aktif

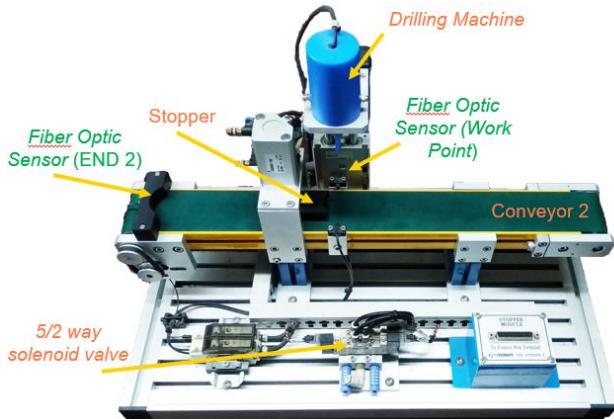
Alamat *output* pada *pick and place module* beserta nama, label dan fungsinya ditunjukkan pada Tabel 3.1 di bawah ini.

Tabel 3.4 Alamat *Output* pada *Pick and Place Module*

No	Alamat	Nama	Label	Fungsi
1.	%QX0.3.11	Sign Red	SIGN_RED	Mengindikasikan sistem sedang berjalan.
2.	%QX0.3.12	Sign Yellow	SIGN_YELLOW	Mengindikasikan <i>machine driller</i>

No	Alamat	Nama	Label	Fungsi
				sedang beroperasi.
3.	%QX0.3.13	Sign Green	SIGN_GREEN	Mengindikasikan sistem sedang diberhentikan.
4.	%QX0.3.6	Rotary Cylinder Counter Clock Wise	CCW	Mengerakkan <i>rotary cylinder</i> bergerak berlawanan dengan arah jarum jam
5.	%QX0.3.7	Rotary Cylinder Clock Wise	CW	Mengerakkan <i>rotary cylinder</i> bergerak searah dengan arah jarum jam
6.	%QX0.3.8	Vacuum Pad Up	UP_VACUUM	Mengerakkan <i>vacuum pad</i> di posisi atas
7.	%QX0.3.9	Vacuum Pad Down	DOWN_VACUUM	Mengerakkan <i>vacuum pad</i> di posisi bawah
8.	%QX0.3.10	Vacuum	VACUUM	Proses <i>vacuum pad</i> menghisap benda kerja untuk dipindah letakkan.

Pada Gambar 3.4 berikut ini menunjukkan peletakan *input* dan *output* pada *stopper module*. Tulisan berwarna hijau menggambarkan *input* sedangkan untuk tulisan berwarna jingga menggambarkan *output* dari sistem.



Gambar 3.4 Letak I/O pada Stopper Module

Alamat *input* pada *stopper module* beserta nama, label dan fungsinya ditunjukkan pada Tabel 3.5 di bawah ini.

Tabel 3.5 Alamat *Input* pada Stopper Module

No	Alamat	Nama	Label	Fungsi
1.	%IX0.1.0	Auto Switch (Stopper Up)	AS_STOPPER_UP	Mendeteksi keadaan stopper posisi di atas
2.	%IX0.1.1	Auto Switch (Stopper Down)	AS_STOPPER_DOWN	Mendeteksi keadaan stopper posisi di bawah
3.	%IX0.1.2	Sensor fiber optic (work point)	WORK_POINT	Mendeteksi benda pada <i>stopper module</i> dan menginformasikan bahwa benda pada posisi tepat untuk memasuki proses <i>drilling</i>
4.	%IX0.1.3	Auto Switch (Drill Up)	AS_DRILL_UP	Mendeteksi mesin <i>drill</i> posisi di atas pada <i>stopper module</i> di proses <i>drilling</i>

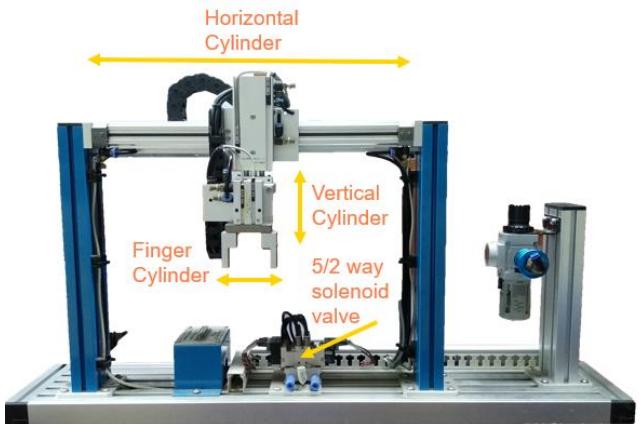
No	Alamat	Nama	Label	Fungsi
5.	%IX0.1.4	Auto Switch (Drill Down)	AS_DRILL_DOWN	Mendeteksi mesin <i>drill</i> posisi di bawah pada <i>stopper module</i> pada proses <i>drilling</i>
6.	%IX0.1.5	Photo Fiber Sensor 2	END2	Mendeteksi benda kerja pada ujung conveyor 2

Alamat *output* pada *stopper module* beserta nama, label dan fungsinya ditunjukkan pada Tabel 3.6 di bawah ini.

Tabel 3.6 Alamat *Output* pada *Stopper Module*

No	Alamat	Nama	Label	Fungsi
1.	%QX0.3.0	Stopper Up	STOPPER_UP	Mengerakkan <i>stopper</i> berada di posisi atas
2.	%QX0.3.1	Drill Up	DRILL_UP	Mengangkat mesin <i>drill</i> ke atas menempatkan ke posisi semula
3.	%QX0.3.2	Drill Down	DRILL_DOWN	Menurunkan mesin <i>drill</i> ke posisi bawah tepat ke benda kerja
4.	%QX0.3.3	Drill On	DRILL_ON	Mengaktifkan mesin <i>drill</i> untuk bekerja
5.	%QX0.3.4	Conveyor2	CONVEYOR2	Membawa benda kerja bergerak pada <i>stopper</i> proses dari <i>line movement</i> ke <i>pick&place</i>

Pada *line movement module* letak aktuator ditunjukkan pada Gambar 3.5 di bawah ini.



Gambar 3.5 Letak Aktuator pada Line Movement Module

Alamat *input* pada *line movement module* beserta nama, label dan fungsinya ditunjukkan pada Tabel 3.7 di bawah ini

Tabel 3.7 Alamat *Input* pada Line Movement Module

No	Alamat	Nama	Label	Fungsi
1.	%IX0.0.0	Auto Switch (Forward)	AS_FORWARD	Mendeteksi silinder <i>forward</i> tepat di atas <i>separation module</i>
2.	%IX0.0.1	Auto Switch (Backward)	AS_BACKWARD	Mendeteksi silinder <i>backward</i> tepat di atas <i>stopper module</i>
3.	%IX0.0.2	Auto Switch (Up)	AS_UP	Mendeteksi keadaan <i>finger</i> silinder berada di posisi atas
4.	%IX0.0.3	Auto Switch (Down)	AS_DOWN	Mendeteksi keadaan <i>finger</i> silinder berada di posisi bawah
5.	%IX0.0.4	Auto Switch (Finger Open)	AS_OPEN	Mendeteksi <i>finger</i> silinder pada keadaan <i>Grip Off</i>
6.	%IX0.0.5	Auto Switch (Finger Grip)	AS_GRIP	Mendeteksi <i>finger</i> silinder pada keadaan <i>Grip On</i>

Alamat *output* pada *line movement module* beserta nama, label dan fungsinya ditunjukkan pada Tabel 3.8 di bawah ini

Tabel 3.8 Alamat *Output* pada *Line Movement Module*

No	Alamat	Nama	Label	Fungsi
1.	%QX0.2.0	Forward	FORWARD	Menggerakkan <i>finger silinder</i> ke arah <i>stopper module</i>
2.	%QX0.2.1	Backward	BACKWARD	Menggerakkan <i>finger silinder</i> ke arah <i>separation module</i>
3.	%QX0.2.2	Up	UP	Menggerakkan <i>finger silinder</i> berada di posisi atas
4.	%QX0.2.3	Down	DOWN	Menggerakkan <i>finger silinder</i> berada di posisi bawah
5.	%QX0.2.4	Finger Grip	FINGER_GRIP	Menggenggam benda kerja untuk dapat dipindahkan dari posisi satu ke posisi yang lain.

3.2 Perancangan *Grafset*

Dalam merancang struktur *grafset* diperlukan rincian spesifikasi dari langkah kerja proses *plant* yang akan dioperasikan. Langkah kerja dalam bentuk uraian kalimat berisi susunan proses yang diinginkan dari sebuah sistem adalah kunci di dalam menyusun *grafset*. Pada perancangan *grafset* untuk *factory automatic trainer* dibagi menjadi perancangan *grafset* pada *separation module*, perancangan *grafset* pada *pick & place module*, perancangan *grafset* pada *stopper module* serta perancangan *grafset* pada *line movement module*.

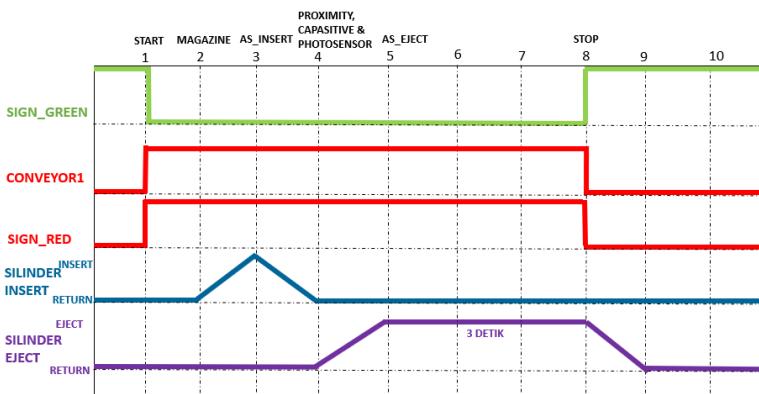
3.2.1 Perancangan *Grafset* pada *Separation Module*

Langkah kerja pengoperasian sistem pada *separation module* adalah sebagai berikut:

1. Kondisi awal sistem dalam keadaan non-aktif, indikator lampu hijau (*Sign Green*) aktif

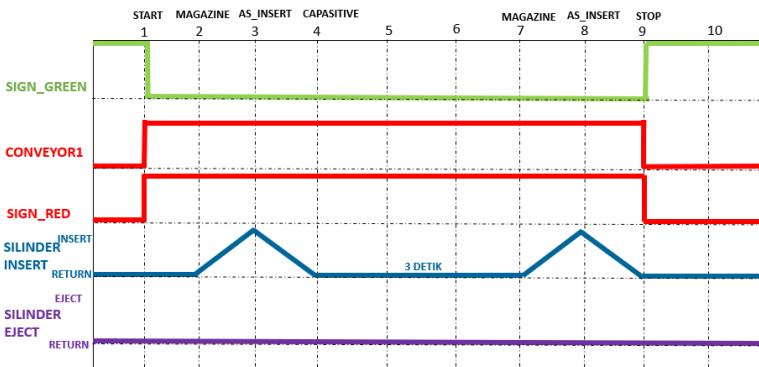
2. Ketika Tombol Start ditekan maka *conveyor* 1 aktif dan lampu indikator berganti warna merah (*Sign Red*).
3. Magazine sensor mendeteksi keberadaan benda untuk mengaktifkan silinder *insert* yang berperan mendorong benda memasuki *conveyor1*. Disini *conveyor1* dan indikator lampu merah masih beroperasi terus hingga tombol stop ditekan.
4. Silinder *insert energize* menyentuh *auto switch* (AS_Insert) dan membuat silinder *insert* kembali ke posisi awal (*insert return*)
5. Pada kondisi benda berwarna silver yang sedang beroperasi sensor *proximity*, *capasitive* dan *photo sensor* mendeteksi keberadaan benda berwarna terang dan mengandung unsur logam. Hal ini mengakibatkan silinder *eject* aktif bersama dengan *timer 3s* yang membuat silinder *insert* mendorong benda selanjutnya.
6. Untuk benda berwarna *silver* silinder *eject* akan menyentuh *auto switch* (AS_Eject) yang mengakibatkan silinder *eject* kembali ke posisi awal (*eject return*)
7. Pada kondisi benda berwarna biru yang sedang beroperasi sensor *capasitive* dan *photo sensor* mendeteksi keberadaan benda berwarna terang dan tidak mengandung unsur logam. Hal ini membuat *timer 3s* aktif untuk membuat silinder *insert* mendorong benda selanjutnya. Hal ini juga mengakibatkan *timer pulse 5s* aktif.
8. Pada kondisi benda berwarna hitam yang berjalan sensor *capasitive* mendeteksi keberadaan benda tidak mengandung unsur logam. Hal ini membuat *timer 3s* aktif untuk membuat silinder *insert* mendorong benda selanjutnya.

Deskripsi sistem di atas dapat ditunjukkan melalui diagram waktu. Diagram waktu dibuat untuk menunjukkan proses dari pergerakan aktuator yang dipengaruhi oleh *input* atau transisi. Terdapat perbedaan aksi untuk benda berbahan metal dan non-metal. Diagram waktu untuk benda berbahan metal (*silver*) ditunjukkan pada Gambar 3.6 berikut ini. Dimana aksi silinder *eject* aktif ditunjukkan dengan garis berwarna ungu.



Gambar 3.6 Diagram Waktu *Separation Module* Benda Silver

Diagram waktu untuk benda biru dan benda hitam memiliki kesamaan aksi aktuator. Pada Gambar 3.7 ditunjukkan diagram waktu dari benda hitam yang melintas. Perbedaan yang terlihat jika dibandingkan dengan benda silver terletak pada aksi silinder *eject*.



Gambar 3.7 Diagram Waktu *Separation Module* Benda Hitam

Dari uraian deskripsi sistem tersebut kemudian diklasifikasikan ke dalam setiap langkah (*step*), aksi yang dihasilkan dari setiap *step*, transisi yang menjadi syarat cukup, *step* sebelumnya yang mempengaruhi *step* tersebut aktif menjadi syarat perlu serta *reset* yang membuat *step* tersebut non-aktif. Tahapan proses tersebut pada *separation module* ditinjau pada Tabel 3.9 berikut ini.

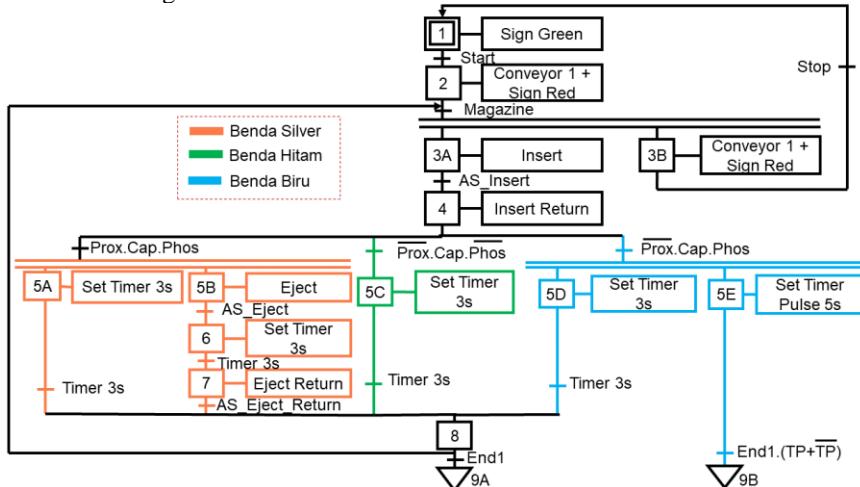
Tabel 3.9 Tahapan Proses *Separation Module*

Step	Aksi	Syarat Cukup	Syarat Perlu	Reset
<i>Step1</i>	Lampu Indikator dengan warna hijau menyala (<i>Sign_Green</i> aktif)	Tombol <i>Stop</i>	Sistem non-aktif	<i>Step 2</i> aktif
<i>Step2</i>	Lampu indikator dengan warna merah menyala (<i>Sign_Red</i> aktif) + <i>Conveyor1</i> aktif.	Tombol <i>Start</i>	Sistem non-aktif	Tombol <i>Stop</i>
<i>Step3A</i>	<i>Silinder Insert</i> mendorong benda kerja menuju <i>conveyor1</i>	Sensor <i>Magazine</i> mendeteksi benda kerja atau Kontak <i>Timer 3s</i> (<i>Step8</i> aktif)	<i>Step2</i> aktif	Tombol <i>Stop</i> dan <i>Step4</i> aktif
<i>Step3B</i>	Lampu indikator dengan warna merah menyala (<i>Sign_Red</i> aktif) dan <i>Conveyor1</i> aktif.	Sensor <i>Magazine</i> mendeteksi benda kerja	<i>Step2</i> aktif	Tombol <i>Stop</i> dan <i>Step1</i> aktif
<i>Step4</i>	<i>Silinder Insert Return</i> mengembalikan posisi <i>silinder</i>	<i>Auto Switch</i> (<i>AS_Insert</i>) aktif	<i>Step3A</i> aktif	Tombol <i>Stop</i> dan <i>Step8</i> aktif
<i>Step5A</i>	Mengaktifkan Timer 3s untuk memasukkan benda kerja kembali setelah benda kerja berwarna <i>silver</i> melewati sensor.	Sensor <i>proximity, capacitive</i> dan <i>photo_sensor</i> aktif.	<i>Step4</i> aktif	Tombol <i>Stop</i> dan <i>Step8</i> aktif

<i>Step</i>	Aksi	Syarat Cukup	Syarat Perlu	<i>Reset</i>
<i>Step5B</i>	<i>Silinder Eject</i> menyisihkan benda berwarna silver	Sensor <i>proximity</i> , <i>capasitive</i> dan <i>photo_sensor</i> aktif.	<i>Step4</i> aktif	Tombol <i>Stop</i> dan <i>Step6</i> aktif
<i>Step5C</i>	Mengaktifkan <i>Timer 3s</i> untuk memasukkan benda kerja kembali setelah benda kerja berwarna hitam melewati sensor.	Sensor <i>capasitive</i> aktif dan sensor <i>proximity</i> beserta <i>photo_sensor</i> non-aktif	<i>Step4</i> aktif	Tombol <i>Stop</i> dan <i>Step8</i> aktif
<i>Step5D</i>	Mengaktifkan <i>Timer 3s</i> untuk memasukkan benda kerja kembali setelah benda kerja berwarna biru melewati sensor.	Sensor <i>capasitive</i> dan <i>photo_sensor</i> aktif serta sensor <i>proximity</i> non-aktif	<i>Step4</i> aktif	Tombol <i>Stop</i> dan <i>Step8</i> aktif.
<i>Step5E (TP)</i>	Mengaktifkan <i>timer pulse</i> selama 5s	Sensor <i>capasitive</i> dan <i>photosensor</i> aktif serta sensor <i>proximity</i> non aktif.	<i>Step4</i> aktif	-
<i>Step6</i>	Mengaktifkan <i>Timer 3s</i> untuk menahan silinder <i>eject</i> pada posisi <i>energize</i> selama 3 detik.	<i>Auto Switch (AS_Eject)</i> aktif	<i>Step5B</i> aktif	Tombol <i>Stop</i> dan <i>Step7</i> aktif.
<i>Step7</i>	Silinder <i>eject</i> kembali ke posisi semula (<i>eject return</i>)	<i>Timer</i> selesai menghitung selama 3 detik.	<i>Step6</i> aktif	Tombol <i>Stop</i> dan <i>Auto Switch (AS_Eject_Return)</i> aktif

Step	Aksi	Syarat Cukup	Syarat Perlu	Reset
Step8	Mengindikasi timer 3s selesai menghitung.	Timer selesai menghitung selama 3 detik.	Step5A aktif + step5C aktif + step5D aktif	Tombol Stop

Dari uraian langkah kerja dan spesifikasi Tabel 3.9 di atas maka dapat dibuat rancangan *grafset* untuk *separation module* seperti pada Gambar 3.8 di bawah ini. Pada Gambar 3.8 garis berwarna orange adalah jalur untuk benda *silver*, garis berwarna hijau diperuntukkan jalur benda hitam dan di garis berwarna biru untuk benda berwarna biru.



Gambar 3.8 Perancangan *Grafset* pada *Separation Module*

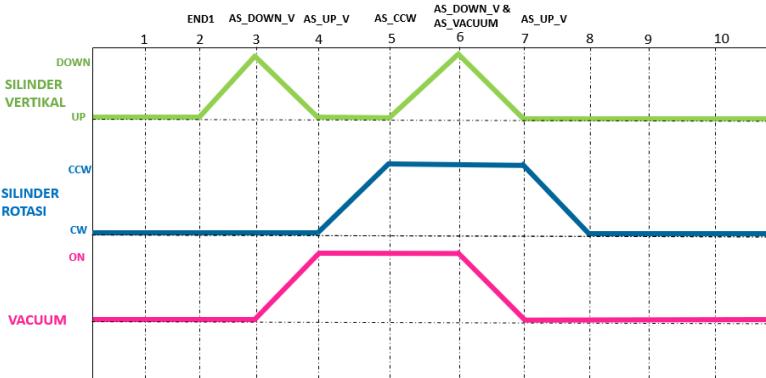
3.2.2 Perancangan *Grafset* pada *Pick and Place Module*

Langkah kerja pengoperasian sistem pada *pick and place module* adalah sebagai berikut :

1. Pada saat benda menyentuh sensor *photo fiber* 1 (END1) di ujung *conveyor 1* maka *vertical cylinder* pada *pick and place module* akan turun (DOWN) mengarah ke benda di *conveyor 1*.
2. *Auto switch* (AS_Down) terdeteksi maka membuat *vertical cylinder* naik (UP) serta *Vacuum* aktif menghisap benda kerja.
3. *Auto Switch* (AS_Up) terdeteksi maka membuat *rotational cylinder* bergerak berlawanan dengan arah jarum jam *counter clock wise* (CCW).

4. Auto Switch (AS_CCW) terdeteksi maka membuat *vertical cylinder* turun (DOWN) membawa benda mengarah ke *conveyor 2*.
5. Auto Switch (AS_Down) dan Auto Switch (AS_Vacuum) terdeteksi maka membuat *vertical cylinder* naik (UP) mengarah ke konveyor 2 bersamaan dengan konveyor 2 yang berjalan. Hal ini juga membuat *vacuum* non-aktif.
6. Auto Switch (AS_Up) terdeteksi maka membuat *rotational cylinder* bergerak searah dengan arah jarum jam atau *clockwise* (CW).

Deskripsi sistem di atas kemudian digambarkan ke dalam diagram waktu yang menunjukkan aksi dari aktuator yang aktif karena pengaruh dari transisi. Diagram waktu *pick and place module* digambarkan pada Gambar 3.9 di bawah ini.



Gambar 3.9 Diagram Waktu *Pick and Place Module*

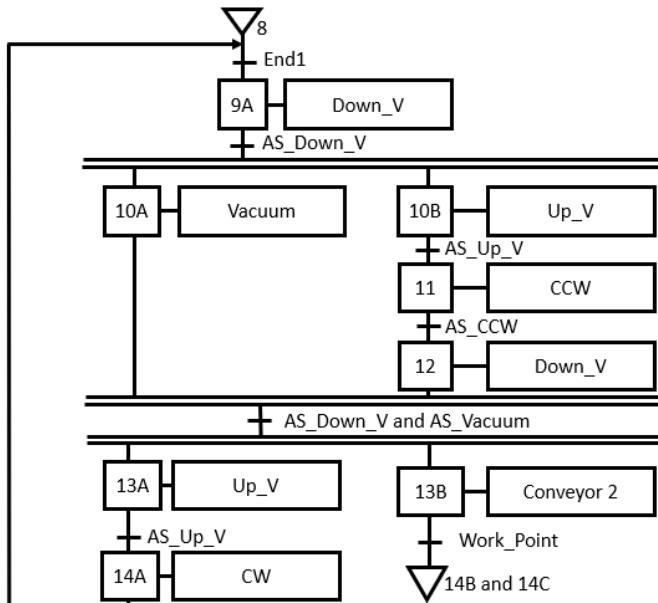
Dari uraian deskripsi sistem tersebut kemudian diklasifikasikan ke dalam setiap langkah (*step*), aksi yang dihasilkan dari setiap *step*, transisi yang menjadi syarat cukup, *step* sebelumnya yang mempengaruhi *step* tersebut aktif menjadi syarat perlu serta *reset* yang membuat *step* tersebut non-aktif. Tahapan proses tersebut pada *pick and place module* ditinjau pada Tabel 3.10 berikut ini.

Tabel 3.10 Tahapan Proses *Pick and Place Module*

Step	Aksi	Syarat Cukup	Syarat Perlu	Reset
Step 9A	Silinder Vertikal pada <i>pick and place module</i> turun (Down_Vacuum)	<i>Sensor photo fiber 1</i> (END1) aktif	-	Tombol Stop, Step10A dan

<i>Step</i>	Aksi	Syarat Cukup	Syarat Perlu	<i>Reset</i>
				<i>Step10B</i> aktif
<i>Step 10A</i>	<i>Vacuum Pad</i> menghisap benda kerja	<i>Auto Switch</i> (AS_Down_V) aktif	<i>Step9A</i> aktif	Tombol <i>Stop</i> dan <i>Step13A</i> aktif
<i>Step 10B</i>	Silinder Vertikal pada <i>pick and place module</i> naik (UP_Vacuum)	<i>Auto Switch</i> (AS_Down_V) aktif	<i>Step9A</i> aktif	Tombol <i>Stop</i> dan <i>Step11</i> aktif
<i>Step 11</i>	Silinder Rotasi pada <i>pick and place module</i> bergerak berlawanan dengan arah jarum jam (CCW)	<i>Auto Switch</i> (AS_Up_V) aktif	<i>Step10B</i> aktif	Tombol <i>Stop</i> , <i>Step12</i> dan <i>Step14A</i> aktif.
<i>Step 12</i>	Silinder Vertikal pada <i>pick and place module</i> turun (Down_Vacuum)	<i>Auto Switch</i> (AS_CCW) aktif	<i>Step11</i> aktif	Tombol <i>Stop</i> , <i>Step13A</i> dan <i>Step14A</i> aktif
<i>Step 13A</i>	Silinder Vertikal pada <i>pick and place module</i> naik (UP_V)	<i>Auto Switch</i> (AS_Down_V) dan <i>Auto Switch</i> (AS_Vacuum) aktif	<i>Step12</i> dan <i>Step10A</i> aktif	Tombol <i>Stop</i> dan <i>Step14A</i> aktif
<i>Step 13B</i>	<i>Conveyor 2</i> aktif	<i>Auto Switch</i> (AS_Down_V) dan <i>Auto Switch</i> (AS_Vacuum) aktif	<i>Step12</i> dan <i>Step10A</i> aktif atau <i>Step16</i> aktif	Tombol <i>Stop</i> , <i>Step14B</i> dan <i>Step14C</i> aktif
<i>Step 14A</i>	Silinder Rotasi pada <i>pick and place module</i> bergerak searah dengan arah jarum jam (CW)	<i>Auto Switch</i> (AS_Up_V) aktif	<i>Step13A</i> aktif	Tombol <i>Stop</i> dan <i>Step9A</i> aktif

Dari uraian langkah kerja dan spesifikasi Tabel 3.10 di atas maka dapat dibuat rancangan *grafcet* untuk *pick and place module* seperti pada Gambar 3.10 di bawah ini.



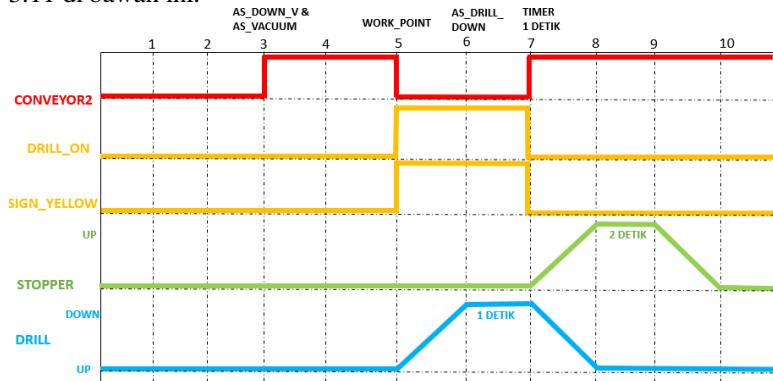
Gambar 3.10 Perancangan *Grafcet* pada *Pick and Place Module*

3.2.3 Perancangan *Grafcet* pada *Stopper Module*

Langkah kerja pengoperasian sistem pada *stopper module* adalah sebagai berikut:

1. *Conveyor 2* yang membawa benda akan terhenti ketika benda dideteksi oleh sensor *work point* hal ini mengakibatkan *drilling machine* aktif (*drill on*) bersamaan dengan *drilling machine* turun (*drill down*).
2. Pada saat *Auto Switch* (AS_Drill_Down) terdeteksi maka mengakibatkan *timer* 1s aktif
3. Setelah terhitung selama 1s *drilling machine* naik (*drill up*) bersamaan dengan *stopper* naik (*stopper up*) dan juga *conveyor 2* kembali beroperasi.
4. Ketika *Auto Switch* (AS_Stopper_Up) terdeteksi hal ini mengakibatkan *timer* 2s aktif.

5. Setelah terhitung 2s *stopper* akan kembali ke posisi awal yakni di bawah sedangkan untuk *drilling machine* tetap berada di posisi atas
- Deskripsi sistem di atas kemudian digambarkan ke dalam diagram waktu yang menunjukkan aksi dari aktuator yang aktif karena pengaruh dari transisi. Diagram waktu *stopper module* digambarkan pada Gambar 3.11 di bawah ini.



Gambar 3.11 Diagram Waktu *Stopper Module*

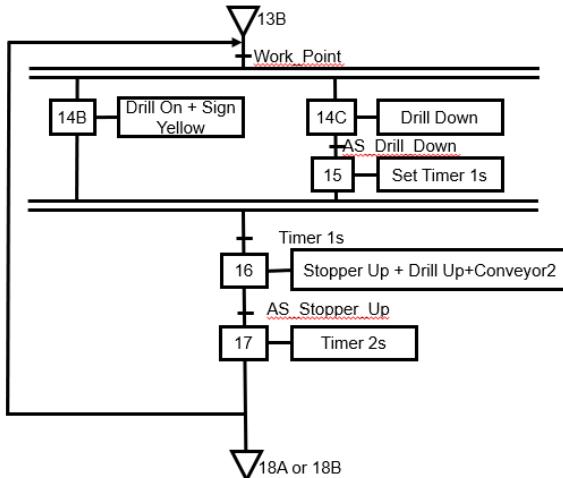
Dari uraian deskripsi sistem tersebut kemudian diklasifikasikan ke dalam setiap langkah (*step*), aksi yang dihasilkan dari setiap *step*, transisi yang menjadi syarat cukup, *step* sebelumnya yang mempengaruhi *step* tersebut aktif menjadi syarat perlu serta *reset* yang membuat *step* tersebut non-aktif. Tahapan proses tersebut pada *stopper module* ditinjau pada Tabel 3.11 berikut ini.

Tabel 3.11 Tahapan Proses *Stopper Module*

Step	Aksi	Syarat Cukup	Syarat Perlu	Reset
Step14B	Mesin <i>driller</i> pada kondisi menyala (ON) + Lampu indikator berwarna kuning (<i>Sign_Yellow</i>) menyala	<i>Work_Point</i> aktif	-	Tombol <i>Stop</i> dan Step16 aktif

Step	Aksi	Syarat Cukup	Syarat Perlu	Reset
Step14C	Mesin driller turun (<i>Drill_Down</i>)	<i>Work_Point</i> aktif	-	Tombol <i>Stop</i> dan <i>Step 15</i> aktif
Step15	Mengaktifkan <i>timer</i> selama 1 detik	<i>Auto Switch</i> (<i>AS_Drill_Down</i>) aktif	<i>Step 14C</i> aktif	Tombol <i>Stop</i> dan <i>Step15_Tim</i> aktif
Step15_Tim	Mengindikasi <i>timer</i> 1s selesai menghitung	Timer selesai menghitung selama 1 detik.	<i>Step15</i> aktif	Tombol <i>Stop</i>
Step16	<i>Stopper</i> naik (<i>stopper up</i>) + mesin driller naik (<i>Drill_up</i>) + <i>Conveyor 2</i> aktif	-	<i>Step15_Tim</i> aktif	Tombol <i>Stop</i> dan <i>Step17</i>
Step17	Mengaktifkan <i>timer</i> selama 2 detik	<i>Auto Switch</i> (<i>AS_Stopper_Up</i>) aktif	<i>Step16</i> aktif	Tombol <i>Stop</i>

Dari uraian langkah kerja dan spesifikasi Tabel 3.11 di atas maka dapat dibuat rancangan *grafset* untuk *stopper module* seperti pada Gambar 3.12 berikut ini.



Gambar 3.12 Perancangan *Grafset* pada *Stopper Module*

3.2.4 Perancangan Graft pada Line Movement Module

Benda yang berhasil melewati proses seleksi di *separation module* hanya terdapat 2 kemungkinan yaitu berwarna biru atau hitam. Sistem tidak dapat mendeteksi benda tersebut biru atau hitam karena di *line movement module* tidak terdapat sensor *proximity*, *capasitive* maupun *photo sensor*. Letak sensor hanya terdapat di *separation module*. Dalam mengatasi hal tersebut dibentuklah *timer pulse* di *Step5E* yang menjadi penanda bahwa benda tersebut berwarna biru ketika kontak *timer pulse* non-aktif dapat disimpulkan bahwa benda berwarna hitam.

Pada *line movement module* akan ada 2 tindakan berbeda untuk benda berwarna biru ataupun hitam. Benda dengan warna biru akan diletakkan di kotak yang berada di tengah sedangkan benda berwarna hitam akan diletakkan di pangkal *conveyor1*.

Langkah kerja pengoperasian sistem pada pemilihan benda biru atau hitam yang akan memasuki *line movement module* adalah sebagai berikut:

1. Pada saat benda pertama melintas menyentuh sensor END1 maka *preset value* untuk *counter* ($C=1$) akan terpenuhi. Apabila kontak *timer pulse* 5s aktif memori Step10C (Biru1) aktif namun sebaliknya apabila kontak *timer pulse* 5s tidak aktif maka Step10D (Hitam1) yang akan aktif.
2. Pada saat benda kedua melintas menyentuh sensor END1 maka *preset value* untuk *counter* ($C=2$) akan terpenuhi. Apabila kontak *timer pulse* 5s aktif memori Step10E (Biru2) aktif namun sebaliknya apabila kontak *timer pulse* 5s tidak aktif maka Step10F (Hitam2) yang akan aktif.
3. Pada saat benda ketiga melintas menyentuh sensor END1 maka *preset value* untuk *counter* ($C=3$) akan terpenuhi. Apabila kontak *timer pulse* 5s aktif memori Step10G (Biru3) aktif namun sebaliknya apabila kontak *timer pulse* 5s tidak aktif maka Step10H (Hitam3) yang akan aktif.
4. Pada saat benda keempat melintas menyentuh sensor END1 maka *preset value* untuk *counter* ($C=4$) akan terpenuhi. Apabila kontak *timer pulse* 5s aktif memori Step10I (Biru4) aktif namun sebaliknya apabila kontak *timer pulse* 5s tidak aktif maka Step10J (Hitam4) yang akan aktif.
5. Pada saat benda kelima melintas menyentuh sensor END1 maka *preset value* untuk *counter* ($C=5$) akan terpenuhi. Apabila kontak *timer pulse* 5s aktif memori Step10K (Biru5) aktif namun sebaliknya

apabila kontak *timer pulse* 5s tidak aktif maka Step10L (Hitam5) yang akan aktif.

6. Pada saat benda keenam melintas menyentuh sensor END1 maka *preset value* untuk *counter* ($C=6$) akan terpenuhi. Apabila kontak *timer pulse* 5s aktif memori Step10M (Biru6) aktif namun sebaliknya apabila kontak *timer pulse* 5s tidak aktif maka Step10N (Hitam6) yang akan aktif.

Dari uraian deskripsi sistem tersebut kemudian diklasifikasikan ke dalam setiap langkah (*step*), aksi yang dihasilkan dari setiap *step*, transisi yang menjadi syarat cukup, *step* sebelumnya yang mempengaruhi *step* tersebut aktif menjadi syarat perlu serta *reset* yang membuat *step* tersebut non-aktif. Tahapan proses tersebut pada pemilihan aksiditnjau pada Tabel 3.12 berikut ini.

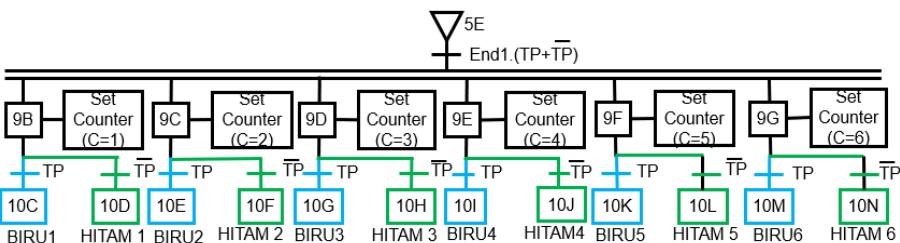
Tabel 3.12 Tahapan Proses Pemilihan Warna Benda

<i>Step</i>	Aksi	Syarat Cukup	Syarat Perlu	<i>Reset</i>
<i>Step10C</i> (Biru1)	Mengaktifkan <i>counter</i> ($C=1$) dan <i>memory</i> untuk benda pertama berwarna biru	Sensor END1 aktif dan kontak <i>timer pulse</i> aktif	<i>Preset value</i> dari <i>counter</i> harus terpenuhi untuk ($C=1$)	-
<i>Step10D</i> (Hitam1)	Mengaktifkan <i>counter</i> ($C=1$) dan <i>memory</i> untuk benda kedua berwarna hitam	Sensor END1 aktif dan kontak <i>timer pulse</i> non aktif	<i>Preset value</i> dari <i>counter</i> harus terpenuhi untuk ($C=1$)	-
<i>Step10E</i> (Biru2)	Mengaktifkan <i>counter</i> ($C=2$) dan <i>memory</i> untuk benda kedua berwarna biru	Sensor END1 aktif dan kontak <i>timer pulse</i> aktif	<i>Preset value</i> dari <i>counter</i> harus terpenuhi untuk ($C=2$)	-
<i>Step10F</i> (Hitam2)	Mengaktifkan <i>counter</i> ($C=2$) dan <i>memory</i> untuk benda	Sensor END1 aktif dan kontak <i>timer pulse</i> non aktif	<i>Preset value</i> dari <i>counter</i> harus	-

<i>Step</i>	Aksi	Syarat Cukup	Syarat Perlu	<i>Reset</i>
	kedua berwarna hitam		terpenuhi untuk (C=2)	
<i>Step10G</i> (Biru3)	Mengaktifkan <i>counter</i> (C=3) dan <i>memory</i> untuk benda ketiga berwarna biru	Sensor END1 aktif dan kontak <i>timer pulse</i> aktif	<i>Preset value</i> dari <i>counter</i> harus terpenuhi untuk (C=3)	-
<i>Step10H</i> (Hitam3)	Mengaktifkan <i>counter</i> (C=3) dan <i>memory</i> untuk benda ketiga berwarna hitam	Sensor END1 aktif dan kontak <i>timer pulse</i> non aktif	<i>Preset value</i> dari <i>counter</i> harus terpenuhi untuk (C=3)	-
<i>Step10I</i> (Biru4)	Mengaktifkan <i>counter</i> (C=4) dan <i>memory</i> untuk benda keempat berwarna biru	Sensor END1 aktif dan kontak <i>timer pulse</i> aktif	<i>Preset value</i> dari <i>counter</i> harus terpenuhi untuk (C=4)	-
<i>Step10J</i> (Hitam4)	Mengaktifkan <i>counter</i> (C=4) dan <i>memory</i> untuk benda keempat berwarna hitam	Sensor END1 aktif dan kontak <i>timer pulse</i> non aktif	<i>Preset value</i> dari <i>counter</i> harus terpenuhi untuk (C=4)	-
<i>Step10K</i> (Biru5)	Mengaktifkan <i>counter</i> (C=5) dan <i>memory</i> untuk benda kelima berwarna biru	Sensor END1 aktif dan kontak <i>timer pulse</i> aktif	<i>Preset value</i> dari <i>counter</i> harus terpenuhi untuk (C=5)	-

Step	Aksi	Syarat Cukup	Syarat Perlu	Reset
Step10L (Hitam5)	Mengaktifkan <i>counter</i> ($C=5$) dan <i>memory</i> untuk benda kelima berwarna hitam	Sensor END1 aktif dan kontak <i>timer pulse</i> non aktif	<i>Preset value</i> dari <i>counter</i> harus terpenuhi untuk ($C=5$)	-
Step10M (Biru6)	Mengaktifkan <i>counter</i> ($C=6$) dan <i>memory</i> untuk benda keenam berwarna biru	Sensor END1 aktif dan kontak <i>timer pulse</i> aktif	<i>Preset value</i> dari <i>counter</i> harus terpenuhi untuk ($C=6$)	-
Step10N (Hitam6)	Mengaktifkan <i>counter</i> ($C=6$) dan <i>memory</i> untuk benda keenam berwarna hitam	Sensor END1 aktif dan kontak <i>timer pulse</i> non aktif	<i>Preset value</i> dari <i>counter</i> harus terpenuhi untuk ($C=6$)	-

Dari uraian langkah kerja dan spesifikasi Tabel 3.12 di atas maka dapat dibuat rancangan *grafset* untuk pemilihan benda warna biru atau hitam sebelum memasuki *line movement module* seperti pada Gambar 3.13 di bawah ini.



Gambar 3.13 Perancangan *Grafset* Pemilihan Warna Benda

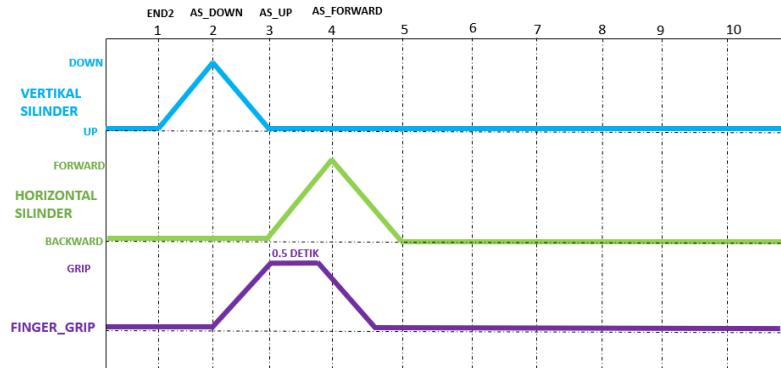
Deskripsi sistem untuk *line movement module* dibagi menjadi 2 yaitu untuk pengoperasian benda biru yang ditandai dengan jalur

berwarna biru serta pengoperasian benda hitam yang ditandai dengan jalur berwarna hijau.

Langkah kerja pengoperasian *line movement module* untuk benda berwarna biru adalah sebagai berikut:

1. Ketika sensor END2 mendeteksi benda dan memori dari Biru1,Biru2,Biru3,Biru4,Biru5 ataupun Biru6 aktif mengakibatkan *vertical cylinder* turun (*down*).
2. *Auto switch* (AS_Down) terdeteksi mengakibatkan *finger cylinder* untuk *grip* benda bersamaan dengan *vertical cylinder* yang naik. (*up*).
3. *Auto switch* (AS_Up) terdeteksi mengakibatkan *horizontal cylinder* bergerak dari kanan ke kiri (*forward*) dan *timer* 0.5 detik aktif.
4. Setelah terhitung selama 0.5 detik *finger cylinder* membuka (*open*).
5. *Auto switch* (AS_Forward) terdeteksi mengakibatkan *horizontal cylinder* bergerak kembali ke posisi awal (*backward*).

Deskripsi sistem di atas kemudian digambarkan ke dalam diagram waktu yang menunjukkan aksi dari aktuator yang aktif karena pengaruh dari transisi. Diagram waktu *line movement module* untuk benda berwarna biru digambarkan pada Gambar 3.14 berikut ini.



Gambar 3.14 Diagram Waktu *Line Movement Module* Benda Biru

Dari uraian deskripsi sistem tersebut kemudian diklasifikasikan ke dalam setiap langkah (*step*), aksi yang dihasilkan dari setiap *step*, transisi yang menjadi syarat cukup, *step* sebelumnya yang mempengaruhi *step* tersebut aktif menjadi syarat perlu serta *reset* yang membuat *step* tersebut non-aktif. Tahapan proses tersebut pada *line movement module* untuk benda berwarna biru ditinjau pada Tabel 3.13 berikut ini.

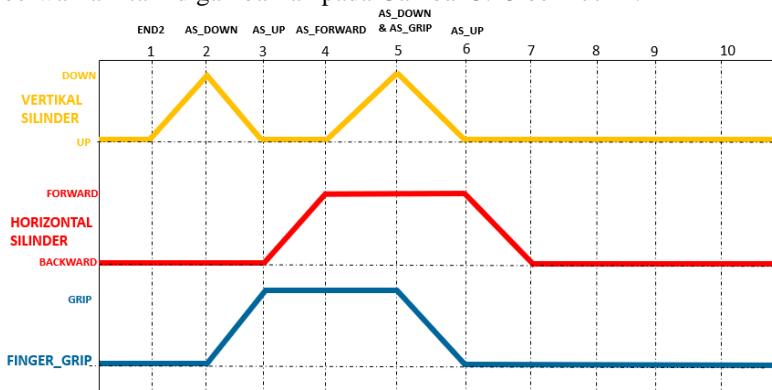
Tabel 3.13 Tahapan Proses *Line Movement Module* Benda Biru

<i>Step</i>	Aksi	Syarat Cukup	Syarat Perlu	<i>Reset</i>
<i>Step18A</i>	Mengaktifkan <i>vertical cylinder</i> untuk turun (<i>Down</i>)	Sensor END2 aktif	(B1+B2+B3+ B4+B5+B6) aktif	Tombol <i>Stop</i> , <i>Step19A</i> dan <i>Step19B</i> aktif
<i>Step19A</i>	Mengaktifkan <i>finger cylinder</i> dalam kondisi <i>grip</i>	<i>Auto Switch</i> (AS_Down) aktif	<i>Step18A</i> aktif	Tombol <i>Stop</i> dan <i>Step21A</i> aktif
<i>Step19B</i>	Mengaktifkan <i>vertical cylinder</i> naik (<i>up</i>)	<i>Auto Switch</i> (AS_Down) aktif	<i>Step18A</i> aktif	Tombol <i>Stop</i> , <i>Step20A</i> dan <i>Step20B</i> aktif
<i>Step20A</i>	Mengaktifkan <i>timer</i> 0.5s	<i>Auto Switch</i> (AS_Up) aktif	<i>Step19B</i> aktif	Tombol <i>Stop</i> dan <i>Step21A</i> aktif
<i>Step20B</i>	Mengaktifkan <i>horizontal cylinder</i> bergerak dari kanan ke kiri (<i>forward</i>)	<i>Auto Switch</i> (AS_Up) aktif	<i>Step19B</i> aktif	Tombol <i>Stop</i> dan <i>Step22A</i> aktif
<i>Step21A</i>	Mengindikasi kan bahwa <i>timer</i> 0.5s sudah terhitung	Kontak <i>timer</i> 0.5s aktif	<i>Step20A</i> dan <i>Step19A</i>	Tombol <i>Stop</i>
<i>Step22A</i>	Mengaktifkan <i>horizontal cylinder</i> bergerak dari kiri ke kanan (<i>backward</i>)	<i>Auto Switch</i> (AS_Forward) aktif	<i>Step21A</i> dan <i>Step20B</i> aktif	Tombol <i>Stop</i> , <i>Step18A</i> dan <i>Step19D</i> aktif

Langkah kerja pengoperasian *line movement module* untuk benda berwarna hitam adalah sebagai berikut:

1. Ketika sensor END2 mendeteksi benda dan memori dari Hitam1, Hitam2, Hitam3, Hitam4, Hitam5 ataupun Hitam6 aktif mengakibatkan *vertical cylinder* turun (*down*).
2. *Auto switch* (AS_Down) terdeteksi mengakibatkan *finger cylinder* untuk *grip* benda bersamaan dengan *vertical cylinder* yang naik. (*up*).
3. *Auto switch* (AS_Up) terdeteksi mengakibatkan *horizontal cylinder* bergerak dari kanan ke kiri (*forward*).
4. *Auto switch* (AS_Forward) terdeteksi mengakibatkan *vertical cylinder* bergerak turun (*down*).
5. *Auto switch* (AS_Down) terdeteksi mengakibatkan *vertical cylinder* bergerak naik (*up*).
6. *Auto switch* (AS_Up) terdeteksi mengakibatkan *horizontal cylinder* bergerak *backward* kembali ke posisi semula.

Deskripsi sistem di atas kemudian digambarkan ke dalam diagram waktu yang menunjukkan aksi dari aktuator yang aktif karena pengaruh dari transisi. Diagram waktu *line movement module* untuk benda berwarna hitam digambarkan pada Gambar 3.15 berikut ini.



Gambar 3.15 Diagram Waktu *Line Movement Module* Benda Hitam

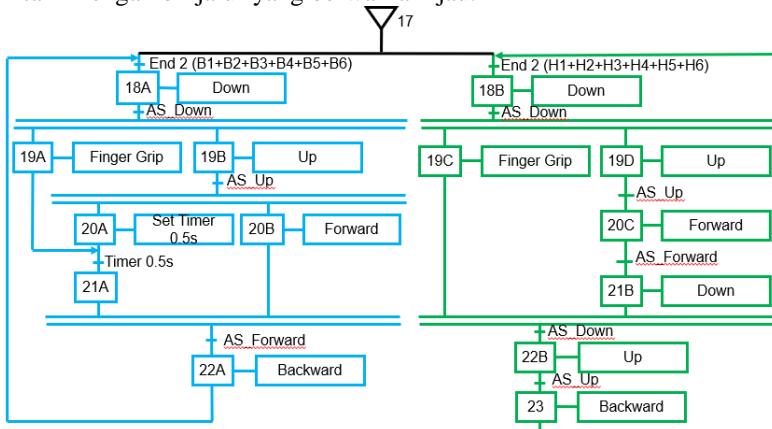
Dari uraian deskripsi sistem tersebut kemudian diklasifikasikan ke dalam setiap langkah (*step*), aksi yang dihasilkan dari setiap *step*, transisi yang menjadi syarat cukup, *step* sebelumnya yang mempengaruhi *step* tersebut aktif menjadi syarat perlu serta *reset* yang membuat *step* tersebut non-aktif. Tahapan proses tersebut pada *line movement module* untuk benda berwarna hitam ditinjau pada Tabel 3.14 berikut ini.

Tabel 3.14 Tahapan Proses *Line Movement Module* Benda Hitam

<i>Step</i>	Aksi	Syarat Cukup	Syarat Perlu	<i>Reset</i>
<i>Step18B</i>	Mengaktifkan <i>vertical cylinder</i> untuk turun (<i>Down</i>)	Sensor END1 aktif	(H1+H2+H3+H4+H5+H6) aktif	Tombol <i>Stop</i> , <i>Step19C</i> dan <i>Step19D</i> aktif
<i>Step19C</i>	Mengaktifkan <i>finger cylinder</i> dalam kondisi <i>grip</i>	<i>Auto Switch</i> (AS_Down) aktif	<i>Step18B</i> aktif	Tombol <i>Stop</i> dan <i>Step22B</i>
<i>Step19D</i>	Mengaktifkan <i>vertical cylinder</i> naik (<i>up</i>)	<i>Auto Switch</i> (AS_Down) aktif	<i>Step18B</i> aktif	Tombol <i>Stop</i> dan <i>Step20C</i>
<i>Step20C</i>	Mengaktifkan <i>horizontal cylinder</i> bergerak dari kanan ke kiri (<i>forward</i>)	<i>Auto Switch</i> (AS_Up) aktif	<i>Step19D</i> aktif	Tombol <i>Stop</i> , <i>Step21B</i> dan <i>Step23</i> aktif
<i>Step21B</i>	Mengaktifkan <i>vertical cylinder</i> untuk turun (<i>Down</i>)	<i>Auto Switch</i> (AS_Forward) aktif	<i>Step20C</i> aktif	Tombol <i>Stop</i> , <i>Step21B</i> dan <i>Step23</i> aktif
<i>Step22B</i>	Mengaktifkan <i>vertical cylinder</i> untuk naik (<i>Up</i>)	<i>Auto Switch</i> (AS_Down) aktif	<i>Step21B</i> dan <i>Step19C</i> aktif	Tombol <i>Stop</i> , <i>Step18A</i> dan <i>Step 23</i> aktif
<i>Step23</i>	Mengaktifkan <i>horizontal cylinder</i> bergerak dari kiri ke kanan (<i>backward</i>)	<i>Auto Switch</i> (AS_Up) aktif	<i>Step22B</i> aktif	Tombol <i>Stop</i> , <i>Step18A</i> dan <i>Step18B</i> aktif

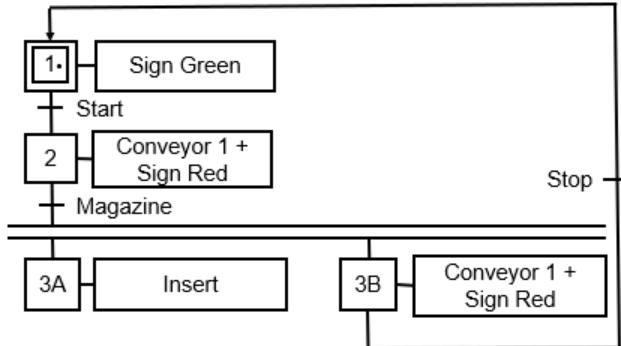
Dari uraian langkah kerja dan spesifikasi Tabel 3.13 dan Tabel 3.14 di atas maka dapat dibuat rancangan *grafset* untuk *line movement module* seperti pada Gambar 3.16 di bawah ini. Benda berwarna biru akan

mengambil jalur yang berwarna biru sedangkan benda dengan warna hitam mengambil jalur yang berwarna hijau.



lampu hijau dapat pula dengan menekan tombol *stop*. Sementara untuk menonaktifkan *step1* perlu di *reset* dengan *step* setelahnya yaitu *step2*.

Step dengan token di dalamnya menandakan bahwa *step* tersebut aktif. Pada Gambar 3.17, *step1* yang berperan mengaktifkan lampu indikator hijau aktif.



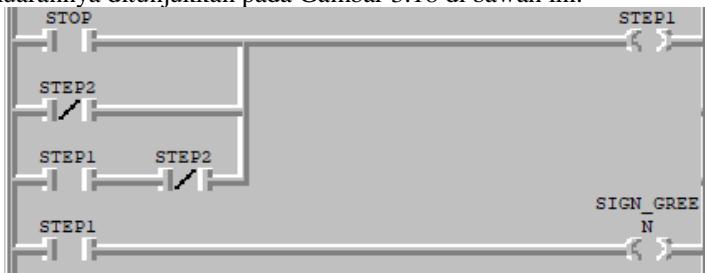
Gambar 3.17 Grafset Step1

Konversi dari *grafset* ke *switching function* untuk *step1*:

$$Step1 = Stop + Step2 + Step1 * Step2$$

$$Sign_Green = Step1$$

Konversi dari *switching function* ke *ladder diagram* untuk *step1* beserta keluarannya ditunjukkan pada Gambar 3.18 di bawah ini.

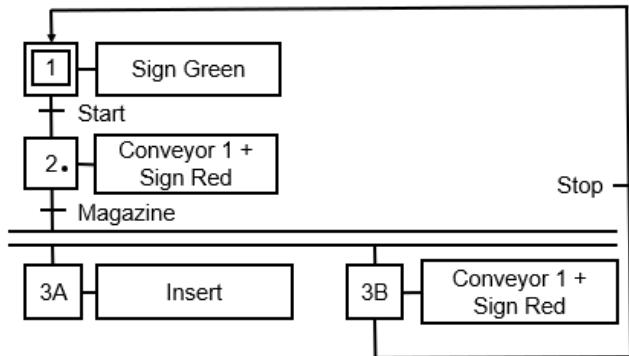


Gambar 3.18 Diagram Ladder Step1

b) *Step2*:

Step2 berfungsi untuk mengaktifkan 2 *output* yaitu *conveyor1* serta indikator lampu merah yang berarti sistem sedang beroperasi. *Step2* aktif ketika tombol *Start* ditekan. *Step2* non-aktif hanya ketika tombol *stop* ditekan.

Step dengan token di dalamnya menandakan bahwa *step* tersebut aktif. Pada Gambar 3.19, *step2* yang berperan mengaktifkan lampu indikator merah serta *conveyor1* sedang dalam kondisi aktif.



Gambar 3.19 Grafcet Step2

Konversi dari *grafcet* ke *switching function* untuk *Step2*:

$$Step2 = (Start + Step2).Stop$$

$$Conveyor1 + Sign_red = Step2$$

Konversi dari *switching function* ke *ladder diagram* untuk *Step2* beserta keluarannya ditunjukkan pada Gambar 3.20 di bawah ini.



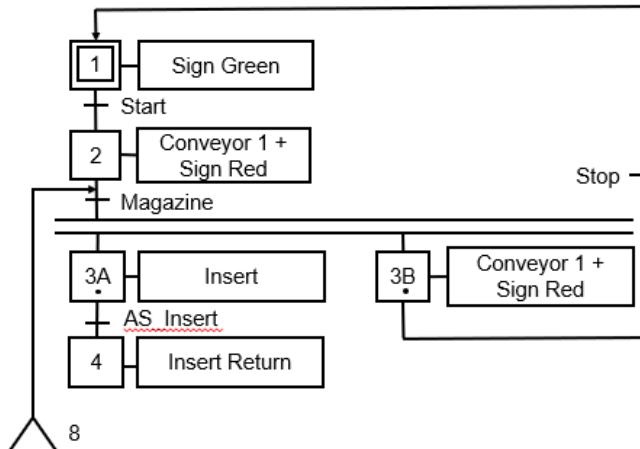
Gambar 3.20 Diagram Ladder Step2

c) *Step3A & Step3B*:

Step3A berfungsi untuk mengaktifkan silinder *insert* dan *step3B* berfungsi untuk mengaktifkan *conveyor1* dan indikator lampu merah. *Step3A* dan *step3B* aktif ketika sensor *magazine* mendekripsi benda atau kontak *timer 3s* dari *step8* aktif. Syarat perlunya adalah *step* sebelumnya

yaitu *step2* aktif. *Step3A* non-aktif ketika *step* setelahnya aktif yaitu *step4*. Sedangkan *Step3B* non-aktif ketika *step1* aktif. *Step3A* dan *Step3B* juga dapat non-aktif ketika tombol *stop* ditekan.

Step dengan token di dalamnya menandakan bahwa *step* tersebut aktif. Pada Gambar 3.21, *step3A* dan *step3B* aktif secara bersamaan diwakili dengan 2 garis horizontal di atasnya.



Gambar 3.21 Grafset Step3A dan Step3B

Konversi dari grafset ke switching function untuk *Step3A* dan *Step3B*:

$$Step3A = ((Step2 + Step8) * Magazine + Step3A) \overline{Stop} * \overline{Step4}$$

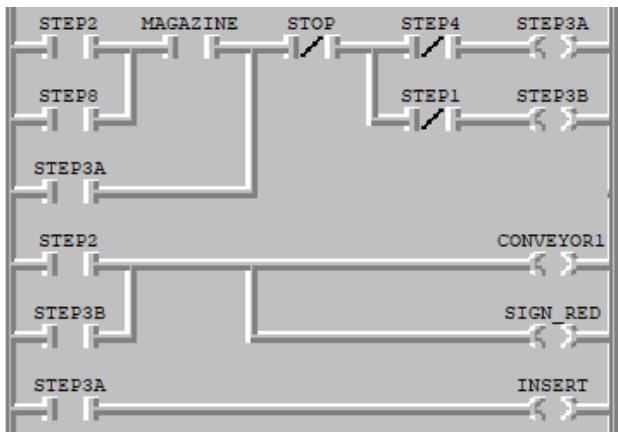
$$Step3B = (Step2 * Magazine) \overline{Stop} * \overline{Step1}$$

$$Step3A + Step3B = ((Step2 + Step8) * Magazine) \overline{Stop} (\overline{Step4} + \overline{Step1})$$

$$Insert = Step3A$$

$$Conveyor1 + Sign_red = Step3B$$

Konversi dari switching function ke ladder diagram untuk *Step3A* dan *Step3B* ditunjukkan pada Gambar 3.22 berikut ini

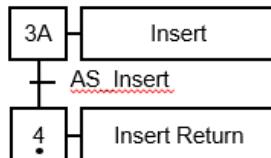


Gambar 3.22 Diagram Ladder Step3A dan Step3B

d) *Step 4:*

Step4 berfungsi untuk mengaktifkan silinder *insert* kembali ke posisi semula. Pemicu dari *step4* aktif adalah *auto switch* (*AS_Insert*) tersentuh dan *step* sebelumnya harus aktif yaitu *step3A*.

Step dengan token di dalamnya menandakan bahwa *step* tersebut aktif. Pada Gambar 3.23 ditunjukkan bahwa *step4* dengan aksi silinder *insert* kembali ke posisi awal sedang aktif.



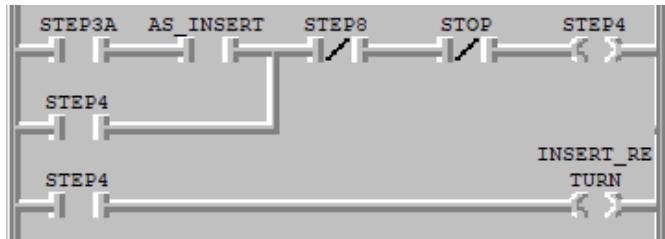
Gambar 3.23 Grafset Step4

Konversi dari *grafset* ke *switching function* untuk *Step4*:

$$Step4 = (Step3A * AS_Insert + Step4) \overline{Step8} * \overline{Stop}$$

$$Insert_return = Step4$$

Konversi dari *switching function* ke *ladder diagram* untuk *Step4* ditunjukkan pada Gambar 3.24 berikut ini.

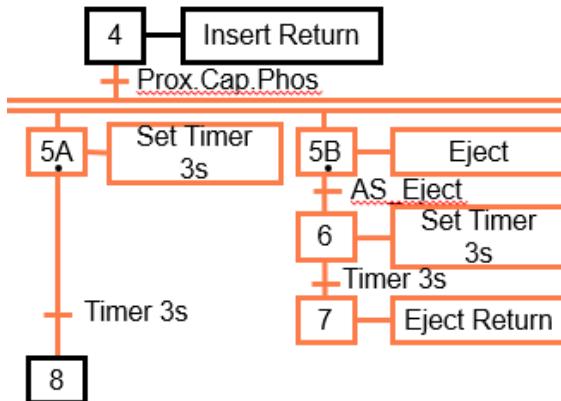


Gambar 3.24 Diagram Ladder Step4

e) *Step 5A & Step 5B:*

Step5A dan *Step5B* aktif bersama – sama ketika benda berwarna *silver* yang melintas ke sensor. *Step5A* untuk mengaktifkan *timer3s* dan *step5B* untuk mengaktifkan silinder *eject*. Syarat cukup dan syarat perlu yang dibutuhkan untuk mengaktifkan kedua *step* tersebut sama namun untuk menon-aktifkan/*resetnya* yang berbeda.

Step dengan token di dalamnya menandakan bahwa *step* tersebut aktif. Pada Gambar 3.25 ditunjukkan bahwa *step5A* dan *step5B* untuk mengaktifkan *timer3s* dan mengaktifkan silinder *eject* sedang dalam keadaan aktif. Jalur berwarna *orange* menunjukkan bahwa jalur khusus untuk benda berwarna *silver*.



Gambar 3.25 Grafset Step5A dan Step5B

Konversi dari *grafset* ke *switching fuction* untuk *Step5A* dan *Step5B*:

Step5A =

$(Step4 * proximity * capacitive * photo_sensor + Step5A) \overline{Stop} * \overline{Step8}$

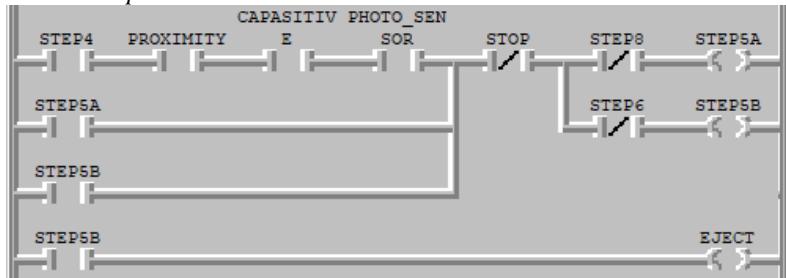
Step5B =

$(Step4.proximity.capacitive.photo_sensor+Step5B) \overline{Stop}.Step6$

Step5A + Step5B =

$\left(Step4 * proximity * capacitive * photo_sensor + Step5A + Step5B \right)$
 $Step8 \left(\overline{Step8} + \overline{Step6} \right)$

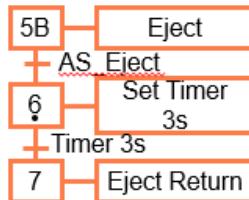
Konversi dari *switching function* ke *ladder diagram* untuk *Step4* ditunjukkan pada Gambar 3.26 di bawah ini. *Output Step8* dijelaskan pada penjelasan *Step8* dikarenakan memori *Step5A* menjadi kontak pemicu untuk mengaktifkan *timer3s* yang akan disimpan ke dalam memori *Step8*.



Gambar 3.26 Diagram Ladder *Step5A* dan *Step5B*

f) *Step6*:

Step6 digunakan untuk mengaktifkan *timer3s*. Tujuan diberikan *timer 3s* ini adalah untuk menyesuaikan waktu perjalanan benda dari letak sensor menuju silinder *eject*. *Step* dengan token di dalamnya menandakan bahwa *step* tersebut aktif. Pada Gambar 3.27 ditunjukkan bahwa *step6* untuk mengaktifkan *timer3s* sedang dalam keadaan aktif. Jalur berwarna jingga menunjukkan bahwa jalur khusus untuk benda berwarna *silver*.

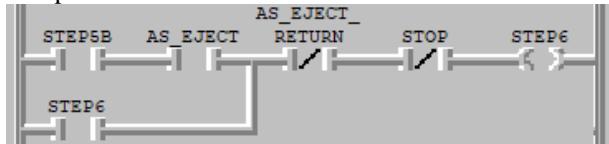


Gambar 3.27 *Grafset Step6*

Konversi dari *grafcet* ke *switching fuction* untuk *Step6*:

$$Step6 = \overline{(Step5B * AS_Eject + Step6)} AS_Eject_return * \overline{Stop}$$

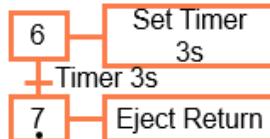
Konversi dari *switching function* ke *ladder diagram* untuk *Step6* ditunjukkan pada Gambar 3.28 di bawah ini.



Gambar 3.28 Diagram Ladder *Step6*

g) *Step 7*:

Step7 berperan untuk mengembalikan posisi silinder *eject* ke posisi semula. *Step* dengan token di dalamnya menandakan bahwa *step* tersebut aktif. Pada Gambar 3.29 ditunjukkan bahwa *step7* untuk mengembalikan posisi silinder *eject* ke posisi semula sedang aktif. Jalur berwarna *orange* menunjukkan bahwa jalur khusus untuk benda berwarna *silver*.



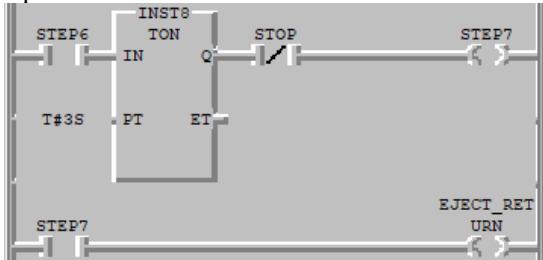
Gambar 3.29 *Grafket Step7*

Konversi dari *grafcet* ke *switching fuction* untuk *Step7*:

$$Step7 = \overline{(Step6 * Tim : 3s)} Stop$$

$$Eject_return = Step7$$

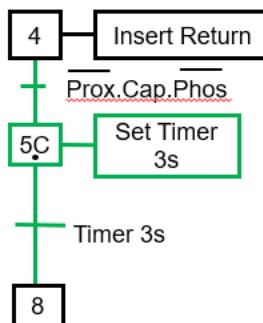
Konversi dari *switching function* ke *ladder diagram* untuk *Step7* ditunjukkan pada Gambar 3.30 berikut ini.



Gambar 3.30 Diagram Ladder *Step7*

h) *Step5C*

Step5C merupakan memori untuk mengaktifkan *timer3s*. Tujuan dari *timer3s* adalah untuk memasukkan benda kerja kembali setelah benda kerja berwarna hitam terdeteksi sensor agar sistem beroperasi secara terus-menerus. *Step* dengan token di dalamnya menandakan bahwa *step* tersebut aktif. Pada Gambar 3.31 ditunjukkan bahwa *step5C* untuk mengaktifkan *timer3s* sedang aktif. Jalur hijau menunjukkan bahwa benda berwarna hitam.

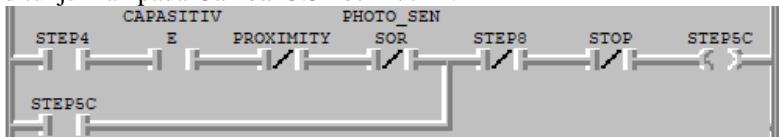


Gambar 3.31 Grafset *Step5C*

Konversi dari *grafset* ke *switching fuction* untuk *output* dari *Step5C*:

$$\text{Step5C} = (\text{Step4} * \overline{\text{Capasitive}} * \overline{\text{proximity}} * \overline{\text{photo_sensor}}) \overline{\text{Step8}} * \overline{\text{Stop}}$$

Konversi dari *switching function* ke *ladder diagram* untuk *Step5C* ditunjukkan pada Gambar 3.32 berikut ini.

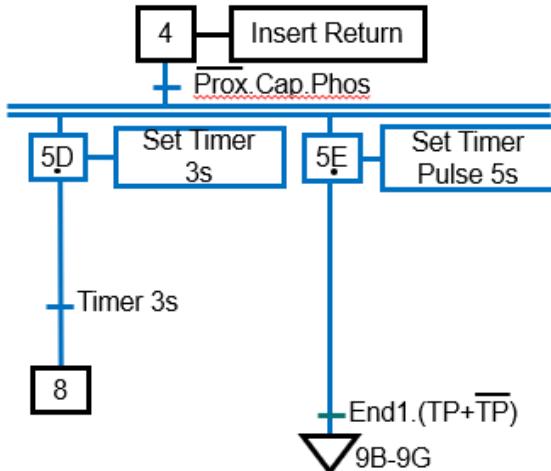


Gambar 3.32 Diagram Ladder *Step5C*

i) *Step5D* dan *Step5E*:

Step5D merupakan memori untuk mengaktifkan *timer3s*. Tujuan dari *timer3s* adalah untuk memasukkan benda kerja kembali setelah benda kerja berwarna biru terdeteksi sensor agar sistem beroperasi secara terus-menerus. *Step5E* digunakan untuk mengaktifkan *timer pulse* 5s. Tujuan dari *timer pulse* 5s adalah sebagai penanda bahwa benda berwarna biru yang sedang melintas. Ketika *timer pulse* 5s tidak aktif diindikasi bahwa benda berwarna hitam yang sedang melintas. *Step* dengan token di

dalamnya menandakan bahwa *step* tersebut aktif. Pada Gambar 3.33 ditunjukkan bahwa *step5D* untuk mengaktifkan *timer3s* sedang aktif dan *step5E* untuk mengaktifkan *timer pulse 5s* sedang aktif. Jalur dengan warna biru menunjukkan bahwa benda berwarna biru.

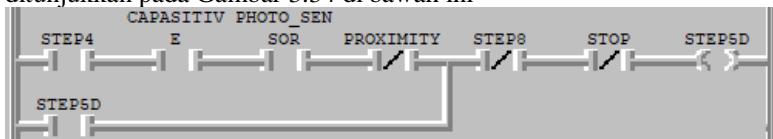


Gambar 3.33 Grafset Step5D dan Step5E

Konversi dari *grafset* ke *switching fuction* untuk *Step5D*:

$$Step5D = \left(Step4 * Capasitive * Photo_sensor * \overline{proximity} \right) \overline{Step8} * \overline{Stop}$$

Konversi dari *switching function* ke *ladder diagram* untuk *Step5D* ditunjukkan pada Gambar 3.34 di bawah ini

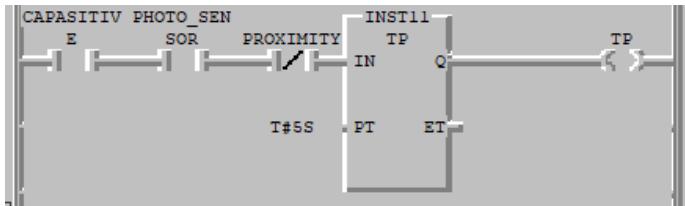


Gambar 3.34 Diagram Ladder Step5D

Konversi dari *grafset* ke *switching fuction* untuk *Step5E(TP)*:

$$TP = \left(Capasitive * Photo_sensor * \overline{proximity} \right) * TimerPulse : 5.0s$$

Konversi dari *switching function* ke *ladder diagram* untuk *Step5E(TP)* ditunjukkan pada Gambar 3.35 di bawah ini

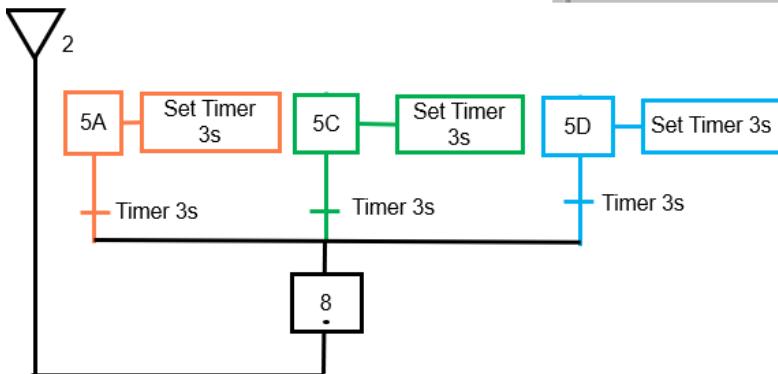


Gambar 3.35 Diagram Ladder Step5E(3s)

j) Step8:

Step8 merupakan sebuah memori sebagai indikasi bahwa timer 3s sudah terhitung. Perbedaan dengan step5A, step5C dan step5D adalah ketiga step tersebut digunakan sebagai kontak pemicu dari timer3s aktif sedangkan step8 adalah indikasi bahwa timer 3s sudah terhitung.

Step dengan token di dalamnya menandakan bahwa step tersebut aktif. Pada Gambar 3.36 ditunjukkan bahwa step8 sedang aktif. Jalur dengan warna jingga menunjukkan bahwa benda berwarna silver, jalur dengan warna hijau menunjukkan bahwa benda berwarna hitam dan jalur dengan warna biru menunjukkan bahwa benda berwarna biru,

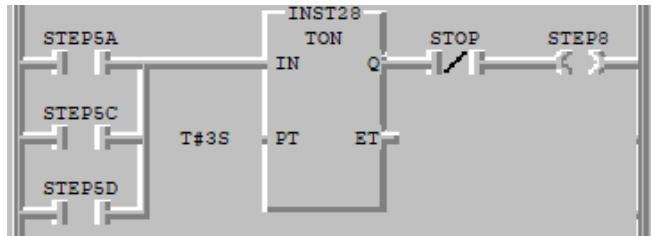


Gambar 3.36 Grafcet Step8

Konversi dari grafcet ke switching function untuk Step8:

$$Step8 = (Step5A + Step5C + Step5D) * Timer : 3.0s * Stop$$

Konversi dari switching function ke ladder diagram untuk Step8 ditunjukkan pada Gambar 3.37 berikut ini.

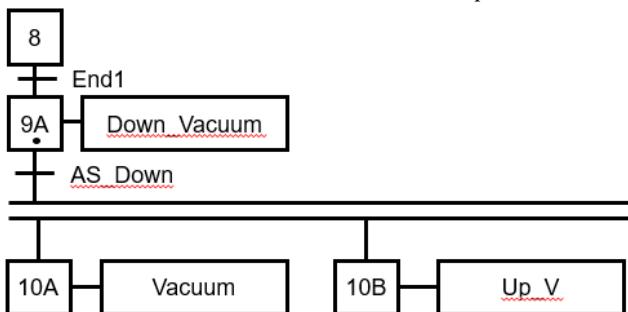


Gambar 3.37 Diagram Ladder Step8

3.3.2 Perancangan Diagram Ladder Pick and Place Module

a) Step9A:

Step9A digunakan untuk mengaktifkan silinder vertikal pada *pick and place module* ke posisi turun (*down*). Step dengan token di dalamnya menandakan bahwa *step* tersebut aktif. Pada Gambar 3.38 ditunjukkan bahwa *step9* yang berfungsi mengaktifkan *vertical cylinder* turun sedang aktif dikarenakan transisi dari sensor *END1* dan *step8* aktif.



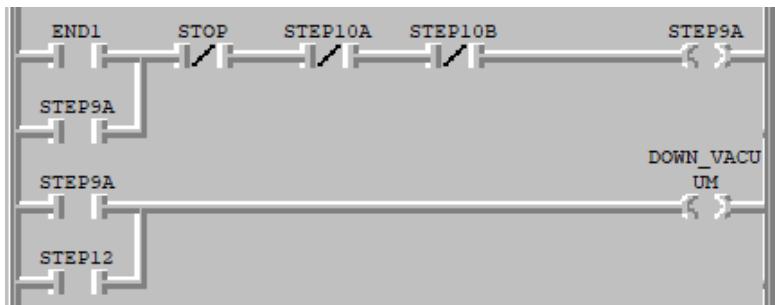
Gambar 3.38 Grafset Step9

Konversi dari grafset ke *switching function* untuk Step9A:

$$Step9A = (End1 + Step9A) \overline{Stop} * Step10A * Step10B$$

$$Down_Vacuum = Step9A$$

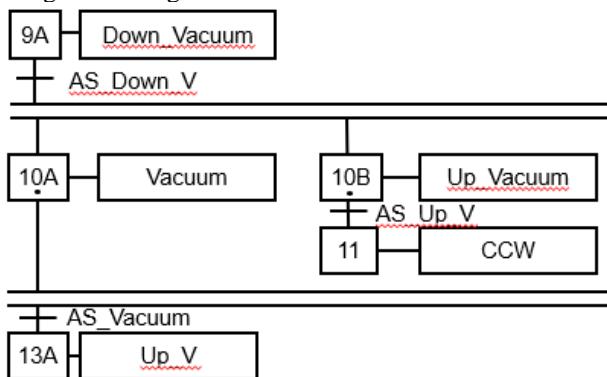
Konversi dari *switching function* ke *ladder diagram* untuk Step9A ditunjukkan pada Gambar 3.39 berikut ini.



Gambar 3.39 Diagram Ladder Step9A

b) Step10A & Step10B:

Pada saat silinder vertikal turun maka akan membuat *auto switch* (AS_Down_V) aktif. Transisi ini yang mengakibatkan step10A dan step10B aktif. Step10A digunakan untuk mengaktifkan *vacuum* dan step10B digunakan untuk mengaktifkan silinder vertikal untuk naik (*up*). Step dengan token di dalamnya menandakan bahwa step tersebut aktif. Pada Gambar 3.40 ditunjukkan bahwa step10A dan step10B aktif bersama – sama. Step yang aktif bersama – sama digambarkan pada grafset dengan 2 buah garis horizontal.



Gambar 3.40 Grafset Step10A dan Step10B

Konversi dari *grafset* ke *switching function* untuk *Step10A* dan *Step10B*:

$$Step10A = (Step9A * AS_Down_V + Step10A) \overline{Stop} * \overline{Step13A}$$

$$Step10B = (Step9A * AS_Down_V + Step10B) \overline{Stop} * \overline{Step11}$$

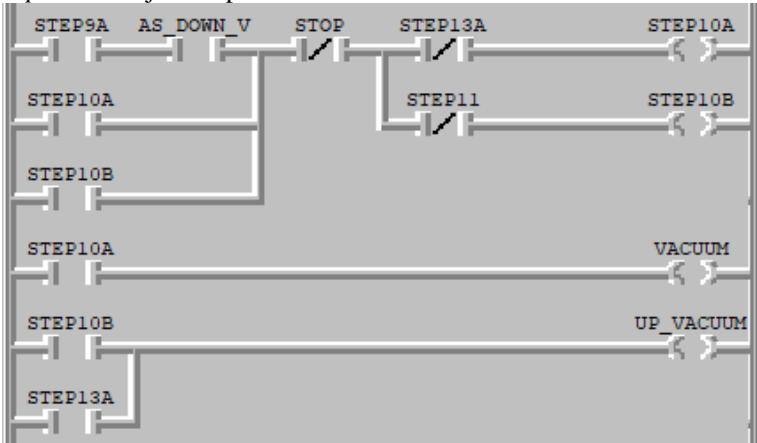
$$Step10A + Step10B =$$

$$(Step9A * AS_Down_V + Step10A + Step10B) \overline{Stop} * \overline{Step13A} * \overline{Step11}$$

$$Vacuum = Step10A$$

$$Up_Vacuum = Step10B$$

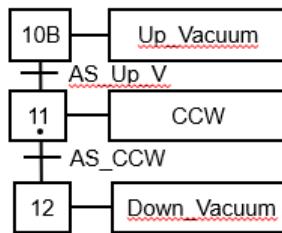
Konversi dari *switching function* ke *ladder diagram* untuk *Step10A* dan *Step10B* ditunjukkan pada Gambar 3.41 berikut ini.



Gambar 3.41 Diagram Ladder *Step10A* dan *Step10B*

c) *Step11*

Step11 digunakan untuk mengaktifkan silinder rotasi bergerak berlawanan dengan arah jarum jam (CCW). *Step* dengan token di dalamnya menandakan bahwa *step* tersebut aktif. Pada Gambar 3.42 ditunjukkan bahwa *step11* sedang aktif dikarenakan transisi dari *auto switch* (*AS_Up_V*).



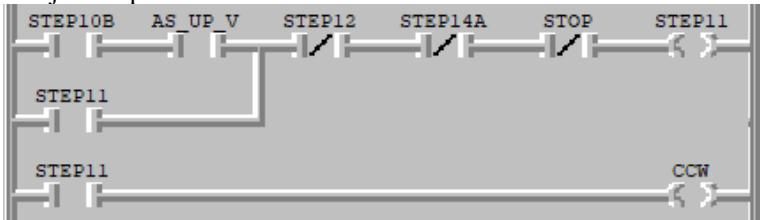
Gambar 3.42 Grafset Step11

Konversi dari *grafset* ke *switching function* untuk *Step11*:

$$Step11 = \left(Step10B * AS_Up_V + Step11 \right) \overline{Step12} * \overline{Step14A} * Stop$$

$$CCW = Step11$$

Konversi dari *switching function* ke *ladder diagram* untuk *Step11* ditunjukkan pada Gambar 3.43 berikut ini.

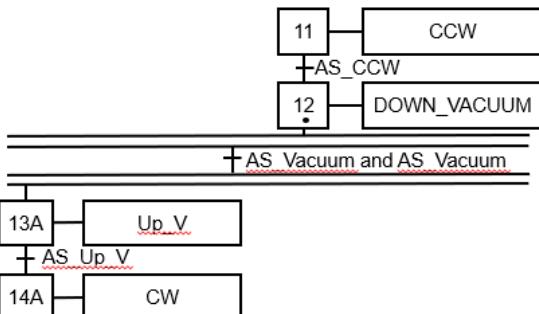


Gambar 3.43 Diagram Ladder Step11

d) *Step12*:

Pada saat silinder pada *pick and place module* bergerak berlawanan dengan arah jarum jam (CCW) membuat *auto switch* (AS_CCW) aktif. Apabila transisi ini telah terpenuhi akan mengakibatkan *step12* aktif. *Step12* dipergunakan untuk menurunkan silinder vertical.

Step dengan token di dalamnya menandakan bahwa *step* tersebut aktif. Pada Gambar 3.44 ditunjukkan bahwa *step12* sedang aktif dikarenakan transisi dari *auto switch* (AS_CCW).

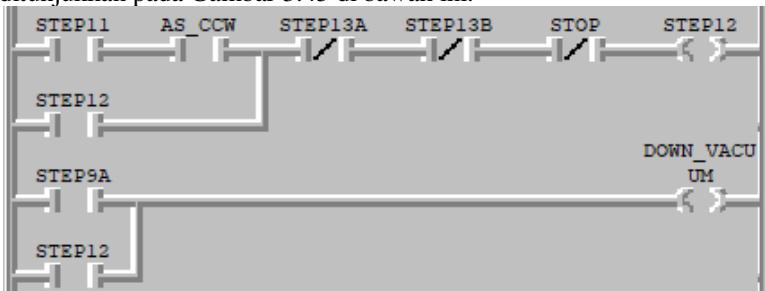


Gambar 3.44 Grafset Step12

Konversi dari grafset ke switching fuction untuk Step12:

$$\begin{aligned} \text{Step12} &= (\text{Step11} * \text{AS_CCW} + \text{Step12}) \overline{\text{Step13A}} * \overline{\text{Step14A}} * \overline{\text{Stop}} \\ \text{Down_Vacuum} &= \text{Step12} \end{aligned}$$

Konversi dari switching function ke ladder diagram untuk Step12 ditunjukkan pada Gambar 3.45 di bawah ini.



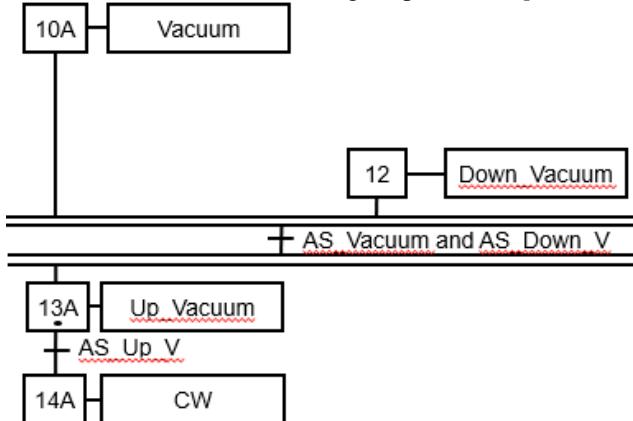
Gambar 3.45 Diagram Ladder Step12

e) Step 13A:

Silinder vertikal yang turun mengakibatkan transisi auto switch (AS_Down_V) aktif. Dengan transisi ini terpenuhi maka mengakibatkan 2 step aktif bersama yaitu step13A dan step13B. Step13A digunakan untuk menaikkan silinder vertikal. Step13B digunakan untuk mengaktifkan conveyor2. Dalam hal ini step13B akan dibahas pada sub bab perancangan diagram ladder stopper module.

Step dengan token di dalamnya menandakan bahwa step tersebut aktif. Pada Gambar 3.46 ditunjukkan bahwa step13A sedang aktif

dikarenakan transisi dari *auto switch* (AS_Vacuum) dan *auto switch* (AS_Down_V). Vacuum berhenti menghisap ketika step13A aktif.

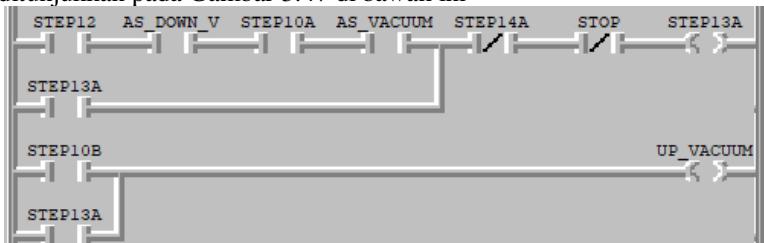


Gambar 3.46 Grafset Step13A

Konversi dari grafset ke switching function untuk Step13A:

$$\begin{aligned} \text{Step13A} &= (\text{Step12} * \text{AS_Down_V} * \text{Step10A} * \text{AS_Vacuum} + \text{Step13A}) \overline{\text{Step14A}} * \overline{\text{Stop}} \\ \text{Up_Vacuum} &= \text{Step13A} \end{aligned}$$

Konversi dari switching function ke ladder diagram untuk Step13A ditunjukkan pada Gambar 3.47 di bawah ini



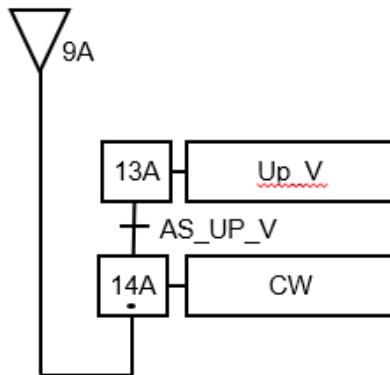
Gambar 3.47 Diagram Ladder Step13A

f) Step14A:

Ketika silinder vertikal di posisi atas maka transisi dari *auto switch* (AS_UP_V) aktif hal ini mengakibatkan step14A aktif. Step14A digunakan untuk mengaktifkan silinder rotasi pada *pick and place module* bergerak searah dengan arah jarum jam (CW).

Step dengan token di dalamnya menandakan bahwa step tersebut aktif. Pada Gambar 3.48 ditunjukkan oleh struktur dari grafset bahwa

step14A sedang aktif dikarenakan terpenuhinya transisi *auto switch* (*AS_UP_V*).



Gambar 3.48 Grafset Step14A

Konversi dari *grafset* ke *switching function* untuk *Step14A*:

$$\text{Step14A} = (\text{Step13A} * \text{AS_Up_V} + \text{Step14A}) \overline{\text{Step9}} * \overline{\text{Stop}}$$

$$CW = \text{Step14A}$$

Konversi dari *switching function* ke *ladder diagram* untuk *Step14A* ditunjukkan pada Gambar 3.49 di bawah ini.

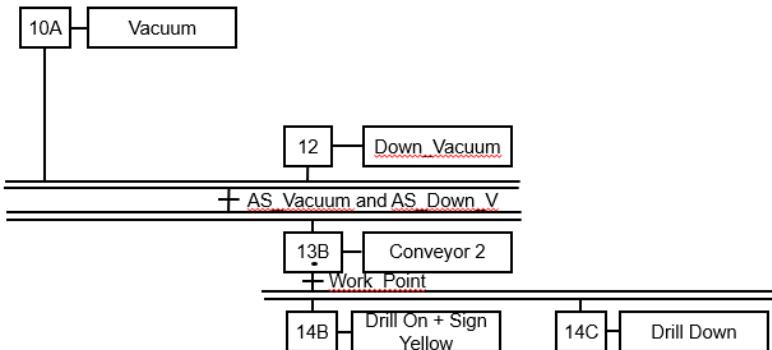


Gambar 3.49 Diagram Ladder Step14A

3.3.3 Perancangan Diagram Ladder Stopper Module

a) Step13B:

Step13B aktif bersama dengan *step13A*. *Step13B* digunakan untuk mengaktifkan *conveyor2*. *Grafset* untuk *step13B* ditunjukkan pada Gambar 3.50. *Step13B* diaktifkan karena *step* sebelumnya yaitu *step10A* dan *step12* sedangkan *reset* untuk *step13B* adalah *step* setelahnya yaitu *step14B* dan *step14C*.



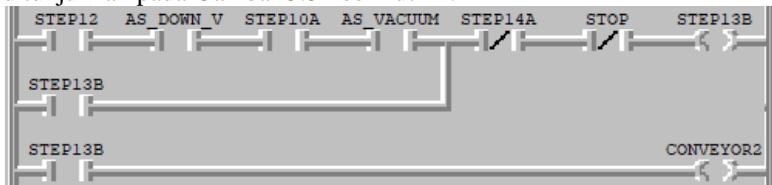
Gambar 3.50 Graf cet Step 13B

Konversi dari *graf cet* ke *switching function* untuk Step13B:

$$\begin{aligned} \text{Step13B} = & (\text{Step12*AS_Down_V*Step10A*AS_Vacuum+Step16+Step13B}) \\ & \overline{\text{Step14B}} * \overline{\text{Step14C}} * \overline{\text{Stop}} \end{aligned}$$

Conveyor2 = Step13B

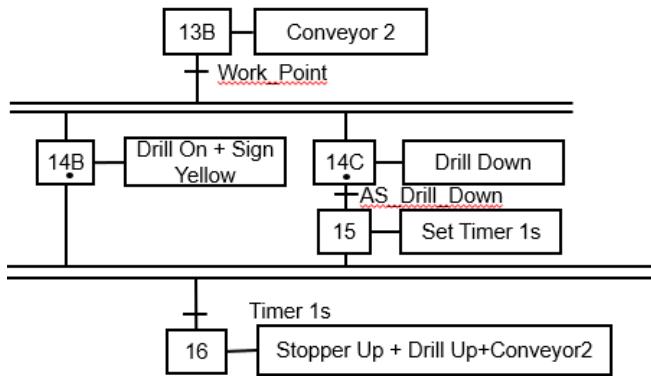
Konversi dari *switching function* ke *ladder diagram* untuk Step13B ditunjukkan pada Gambar 3.51 berikut ini.



Gambar 3.51 Diagram Ladder Step13B

b) Step14B & Step14C

Step14B digunakan untuk mengaktifkan *drilling machine* dan step14C digunakan untuk menurunkan *drilling machine*. Step dengan token di dalamnya menandakan bahwa step tersebut aktif. Pada Gambar 3.52 ditunjukkan bahwa step14B dan step14C sedang aktif. Transisi untuk mengaktifkan kedua step tersebut adalah *work point* beserta step sebelumnya yaitu step13B yang digunakan untuk mengaktifkan *conveyor2*. Sedangkan untuk *reset* dari kedua step tersebut adalah step setelahnya yaitu untuk step14B direset dengan step16 dan untuk step14C direset dengan step15. Pada saat step14B dan step14C aktif maka hal ini membuat *conveyor2* terhenti.



Gambar 3.52 Grafcet Step14B dan Step14C

Konversi dari *grafcet* ke *switching function* untuk *Step14B* dan *Step14C*:

$$\text{Step14B} = (\text{work_point} + \text{step14B}) \overline{\text{Stop}} * \overline{\text{Step16}}$$

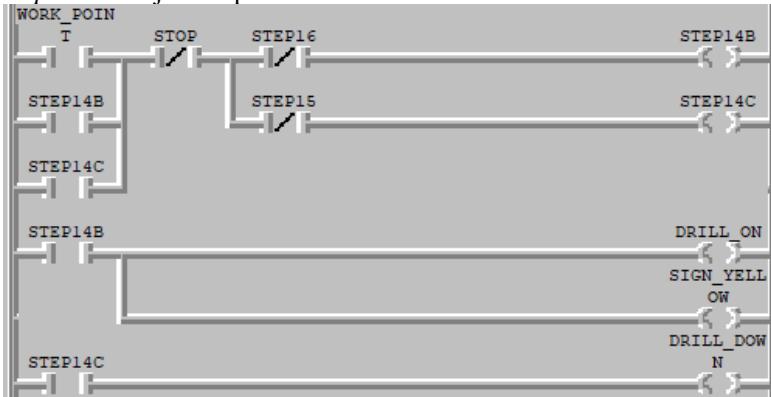
$$\text{Step14C} = (\text{work_point} + \text{step14C}) \overline{\text{Stop}} * \overline{\text{Step15}}$$

$$\text{Step14B} + \text{Step14C} = (\text{work_point} + \text{step14B} + \text{step14C}) \overline{\text{Stop}} * \overline{\text{Step16}} * \overline{\text{Step15}}$$

$$\text{Drill_on} + \text{Sign_Yellow} = \text{Step14B}$$

$$\text{Drill_Down} = \text{Step14C}$$

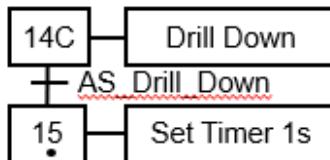
Konversi dari *switching function* ke *ladder diagram* untuk *Step14B* dan *Step14C* ditunjukkan pada Gambar 3.53 di bawah ini.



Gambar 3.53 Diagram Ladder Step14B dan Step14C

c) *Step15*

Step15 digunakan untuk mengaktifkan *timer 1* detik yang berfungsi menahan *drilling machine* tetap aktif selama 1 detik. *Step15_Tim* merupakan indikasi bahwa *timer 1* detik sudah selesai terhitung. *Step* dengan token di dalamnya menandakan bahwa *step* tersebut aktif. Pada Gambar 3.54 ditunjukkan bahwa *step15* sedang aktif. Transisi untuk mengaktifkan kedua *step* tersebut adalah *auto switch* (*AS_Drill_Down*) beserta *step* sebelumnya yaitu *step14C* yang digunakan untuk menurunkan *drilling machine*.



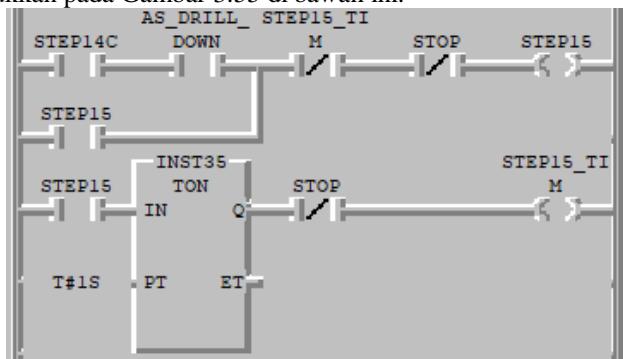
Gambar 3.54 Grafset Step15

Konversi dari *grafset* ke *switching function* untuk *Step15*:

$$Step15 = \overline{(Step14C * AS_Drill_Down + Step15)} Step15_Tim * Stop$$

$$Step15_Tim = (Step15 * Timer : 1s) Stop$$

Konversi dari *switching function* ke *ladder diagram* untuk *Step15* ditunjukkan pada Gambar 3.55 di bawah ini.

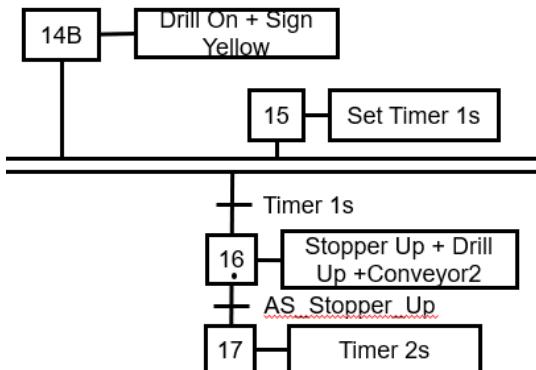


Gambar 3.55 Diagram Ladder Step15

d) *Step16*

Step16 digunakan untuk mengaktifkan 3 aksi secara bersama yaitu untuk menaikkan *stopper*, menaikkan *drilling machine* serta menjalankan

conveyor2 kembali yang sempat berhenti pada saat proses *drilling on* dan *drilling machine* turun. *Step* dengan token di dalamnya menandakan bahwa *step* tersebut aktif. Pada Gambar 3.56 ditunjukkan bahwa *step15* sedang aktif. Transisi untuk mengaktifkan kedua *step* tersebut adalah *auto switch* (*AS_Drill_Down*) beserta *step* sebelumnya yaitu *step14C* yang digunakan untuk menurunkan *drilling machine*.



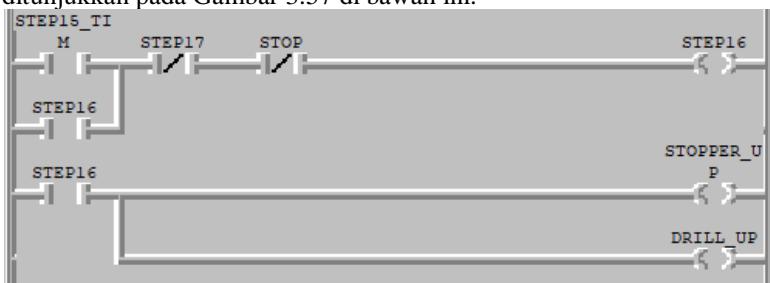
Gambar 3.56 Grafset Step16

Konversi dari *grafset* ke *switching fuction* untuk *Step15*:

$$Step16 = \overline{(Step15_Tim + Step16)} \overline{Step17} * \overline{Stop}$$

$$Stopper_Up + Drill_Up = Step16$$

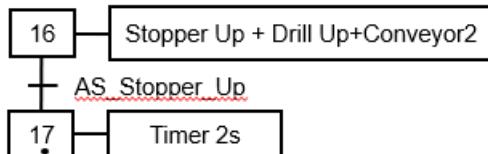
Konversi dari *switching function* ke *ladder diagram* untuk *Step15* ditunjukkan pada Gambar 3.57 di bawah ini.



Gambar 3.57 Diagram Ladder Step16

e) *Step17*

Step17 digunakan untuk mengaktifkan *timer* selama 2 detik. Tujuan dari *timer* 2 detik ini adalah untuk mempertahankan keadaan *stopper* di atas selama 2 detik. *Step* dengan token di dalamnya menandakan bahwa *step* tersebut aktif. Pada Gambar 3.58 ditunjukkan bahwa *step17* sedang aktif. *Step17* aktif dikarenakan transisi *auto switch* (*AS_Stopper_Up*) dan *step* sebelumnya yaitu *step16* aktif.

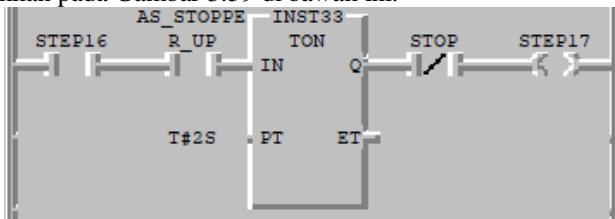


Gambar 3.58 Grafset Step17

Konversi dari *grafset* ke *switching function* untuk *Step17*:

$$Step17 = (Step16 * AS_Stopper_Up * Timer:2s) \overline{Stop}$$

Konversi dari *switching function* ke *ladder diagram* untuk *Step17* ditunjukkan pada Gambar 3.59 di bawah ini.

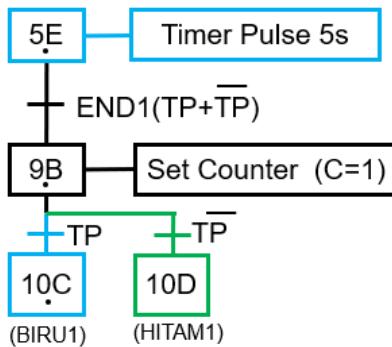


Gambar 3.59 Diagram Ladder Step17

3.3.4 Perancangan Diagram Ladder Line Movement Module

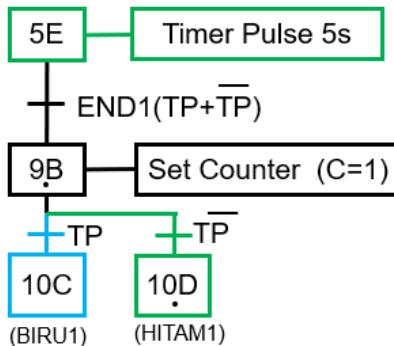
a) *Step10C* atau *Step10D*

Step10C merupakan memori untuk benda pertama yang melintas berwarna biru, sedangkan *step10D* merupakan memori untuk benda pertama yang melintas berwarna hitam. Pada Gambar 3.60 di bawah ditunjukkan bahwa benda pertama yang sedang melintas merupakan benda berwarna biru dikarenakan *step5E* (TP) aktif dan *preset value* dari *counter* untuk *c=1* sudah terpenuhi sehingga jalur yang diambil adalah jalur berwarna biru atau *step10C* aktif.



Gambar 3.60 Grafset Step10C

Pada Gambar 3.61 ditunjukkan bahwa kondisi yang terjadi adalah benda pertama yang melintas merupakan benda berwarna hitam. Pada kondisi ini step5E (TP) tidak aktif yang digambarkan dengan warna hijau, tetapi transisi untuk \overline{TP} aktif dan *preset value* pada *counter* untuk c=1 telah terpenuhi. Maka dari itu jalur hijau yang akan terpilih atau step10D dalam keadaan aktif.



Gambar 3.61 Grafset Step10D

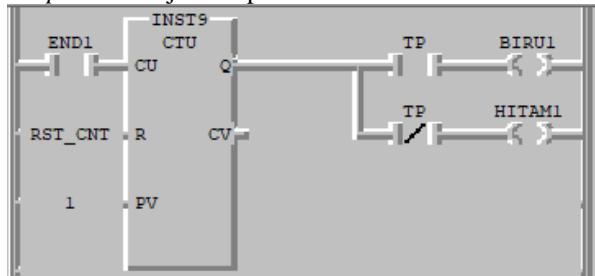
Konversi dari *grafset* ke *switching fuction* untuk *Step10C* ataupun *Step10D*:

$$Biru1 = (End1 * Counter : 1) TP$$

$$Hitam1 = (End1 * Counter : 1) \overline{TP}$$

$$Biru1 + Hitam1 = (End1 * Counter : 1) TP + \overline{TP}$$

Konversi dari *switching function* ke *ladder diagram* untuk *Step10C* ataupun *Step10D* ditunjukkan pada Gambar 3.62 di bawah ini.

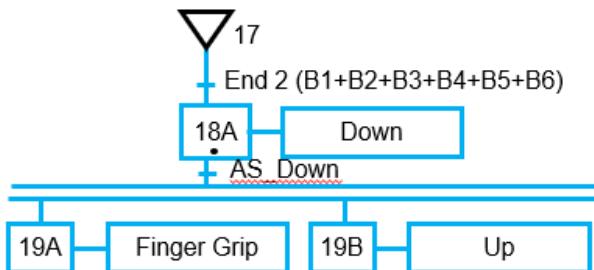


Gambar 3.62 Diagram Ladder Step10C atau Step10D

Konversi ini berlaku hingga benda biru atau hitam yang keenam. Setelah benda keenam melintas dan mengaktifkan memori *RST_CNT* maka membuat *counter* kembali *reset* menghitung benda ketujuh sebagai benda pertama sehingga sistem dapat beroperasi secara terus – menerus hingga tombol *stop* ditekan.

b) *Step18A*

Step18A merupakan memori yang dipergunakan untuk mengaktifkan *vertical cylinder* untuk turun (*down*). *Step* dengan token di dalamnya menandakan bahwa *step* tersebut aktif. Pada Gambar 3.63, *step18A* yang berperan menurunkan *vertical cylinder* pada *line movement module* untuk benda berwarna biru aktif.



Gambar 3.63 Grafset Step18A

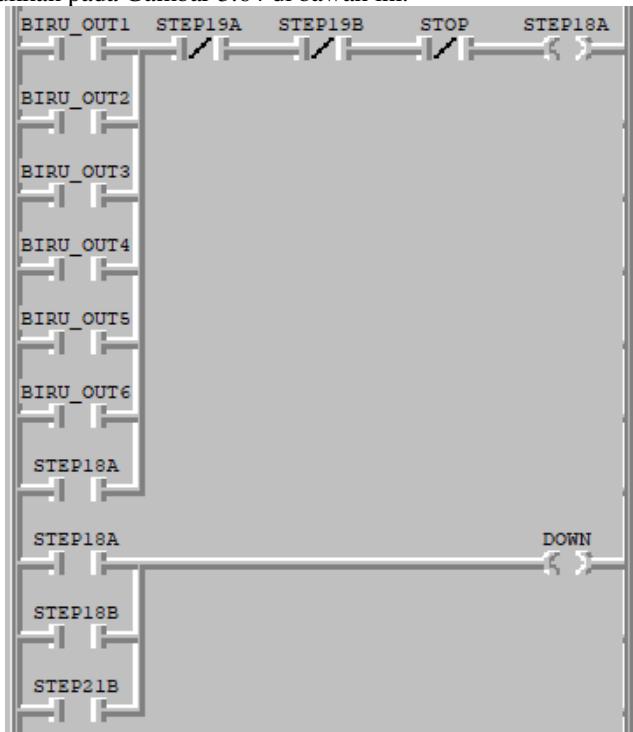
Pada saat benda biru pertama terdeteksi oleh sensor *photo fiber2* (*END2*) maka akan mengaktifkan memori *biru_out1* yang mana memori tersebut menjadi transisi untuk mengaktifkan *step18A*. Terdapat pemisahan *rung* pada pembuatan diagram *ladder* dikarenakan untuk *reset* dari *step18A* ataupun memori *biru_out1* tidaklah sama. Untuk *step18A* dinonaktifkan

oleh *step* setelahnya yaitu *step19A* dan *step19B* sedangkan untuk keluaran memori *biru_out1* hingga *biru_out6* dinonaktifkan dua benda di depannya. Konversi dari *grafset* ke *switching fuction* untuk *Step18A*:

$$Step18A = \left(\begin{array}{l} Biru_Out1 + Biru_Out2 + Biru_Out3 + \\ Biru_Out4 + Biru_Out5 + Biru_Out6 + \\ Step18A \end{array} \right) \overline{Step19A * Step19B * Stop}$$

Down = *Step18A*

Konversi dari *switching function* ke *ladder diagram* untuk *Step18A* ditunjukkan pada Gambar 3.64 di bawah ini.



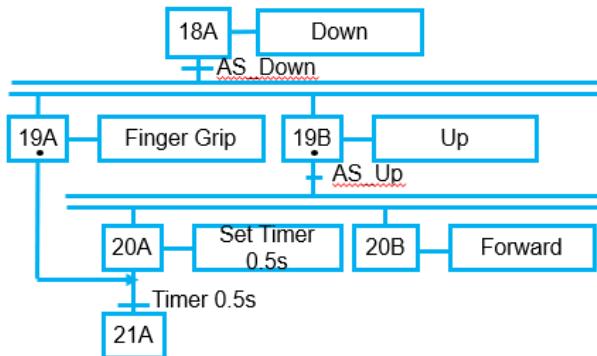
Gambar 3.64 Diagram Ladder Step18A

- c) *Step19A* dan *Step19B*:

Pada saat silinder vertikal pada *line movement module* turun mengarah ke benda biru membuat transisi *auto switch* (*AS_Down*) aktif.

Ketika transisi telah terpenuhi dapat mengakibatkan *step19A* dan *step19B* aktif bersama. *Step19A* digunakan untuk mengaktifkan *finger cylinder* dalam kondisi *grip*. *Step19B* digunakan untuk mengaktifkan *vertical cylinder* untuk naik. Pada kondisi ini benda akan dicengkram dan berada di posisi atas.

Pada *grafcet* kondisi *step* dengan token di dalamnya menandakan bahwa *step* tersebut aktif. Pada Gambar 3.65 ditunjukkan bahwa *step19A* dan *step19B* dalam keadaan aktif.



Gambar 3.65 Graf cet Step19A dan Step19B

Konversi dari *grafcet* ke *switching function* untuk *Step19A* dan *step19B*:

$$Step19A = (Step18A * AS_Down + Step19A) \overline{Stop} * \overline{Step21A}$$

$$Step19B = (Step18A * AS_Down + Step19B) \overline{Stop} * \overline{Step22A} * \overline{Step20B}$$

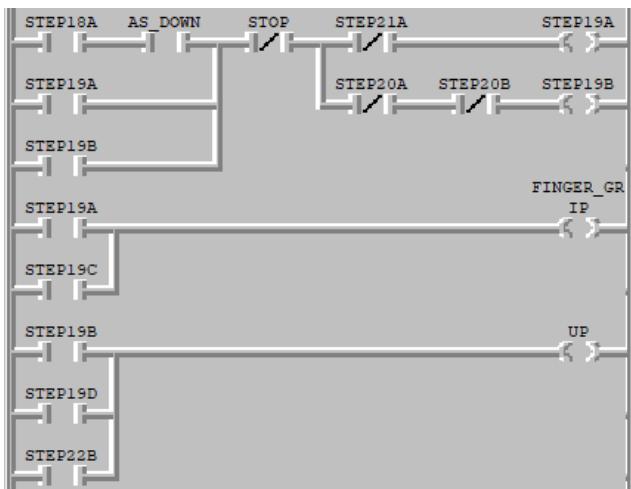
$$Step19A + Step19B =$$

$$(Step18A * AS_Down + Step19A + Step19B) \overline{Stop} (\overline{Step21A} + \overline{Step20A} + \overline{Step20B})$$

$$Finger_Grip = Step19A$$

$$Up = Step19B$$

Konversi dari *switching function* ke *ladder diagram* untuk *step19A* dan *step19B* ditunjukkan pada Gambar 3.66 berikut ini.

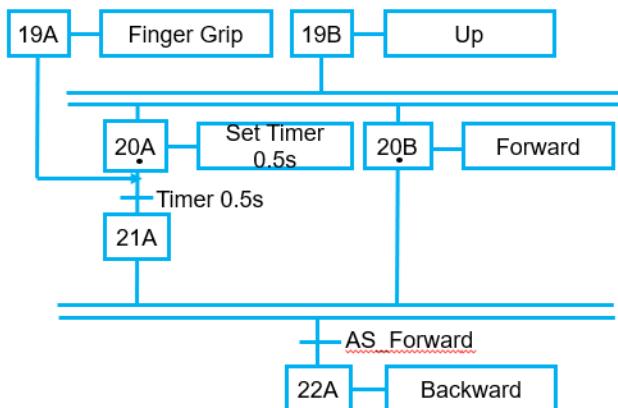


Gambar 3.66 Diagram Ladder Step19A dan Step19B

d) *Step20A* dan *Step20B*:

Pada saat silinder vertikal di atas secara otomatis transisi *auto switch* (AS_Up) terdeteksi. Hal ini mengakibatkan *step20A* dan *step20B* aktif bersama. *Step20A* digunakan untuk mengaktifkan *timer* selama 0.5 detik dan *step20B* digunakan untuk mengaktifkan *horizontal cylinder* bergerak dari *stopper module* mengarah ke *separation module* (*forward*). Tujuan *timer* 0.5 detik diberikan adalah untuk menahan *finger cylinder* dalam kondisi mencengkram selama 0.5 detik. Pada waktu 0.5 detik selesai terhitung maka *finger cylinder* akan membuka *gripper*, sehingga benda berwarna biru jatuh tepat di tengah kotak yang sudah diletakkan di tengah jalur dari *line movement module*.

Pada *grafset step* dengan token di dalamnya menandakan bahwa *step* tersebut aktif. Pada Gambar 3.67 ditunjukkan bahwa *step20A* yang beroperasi mengaktifkan *timer* selama 0.5 detik dan *step20B* yang menggerakkan *horizontal cylinder* bergerak *forward* pada *line movement module* untuk benda berwarna biru dalam keadaan aktif. Jalur berwarna biru mengindikasi bahwa jalur dikhususkan untuk aksi *line movement module* benda berwarna biru.



Gambar 3.67 Grafcet Step20A dan Step20B

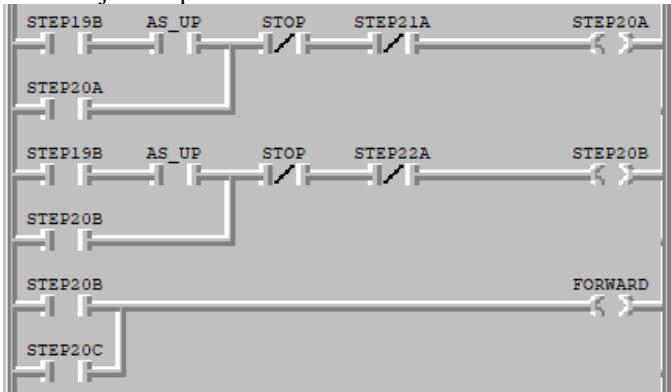
Konversi dari *grafcet* ke *switching fuction* untuk *Step20A* dan *step20B*:

$$\text{Step20A} = (\text{Step19B} * \text{AS_Up} + \text{Step20A}) \overline{\text{Stop}} * \overline{\text{Step21A}}$$

$$\text{Step20B} = (\text{Step19B} * \text{AS_Up} + \text{Step20B}) \overline{\text{Stop}} * \overline{\text{Step22A}}$$

$$\text{Forward} = \text{Step20B}$$

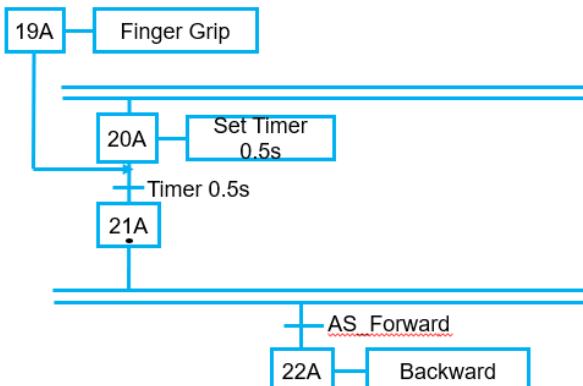
Konversi dari *switching function* ke *ladder diagram* untuk *step20A* dan *step20B* ditunjukkan pada Gambar 3.68 di bawah ini.



Gambar 3.68 Diagram Ladder Step20A dan Step20B

e) *Step21A*

Step21A digunakan untuk mengindikasikan bahwa *timer* 0.5 detik telah terhitung. *Step20A* merupakan transisi untuk membuat *timer* aktif dimana setelah *timer* menghitung akan disimpan ke dalam memori *step21A*. Pada *grafset step* dengan token di dalamnya menandakan bahwa *step* tersebut aktif. Pada Gambar 3.69, *step21A* yang menjadi indikasi bahwa *timer* 0.5 detik telah terhitung sedang aktif. *Step21A* tidak memiliki aksi atau *output*.

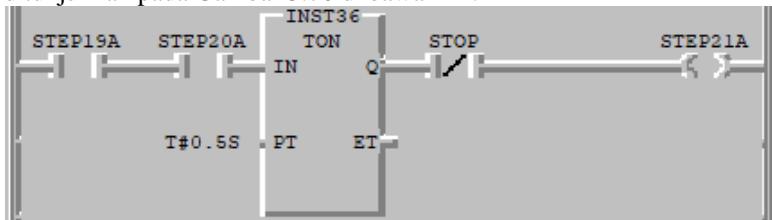


Gambar 3.69 Grafset Step21A

Konversi dari *grafset* ke *switching fuction* untuk *Step21A*:

$$\text{Step21A} = (\text{Step19A} * \text{Step20A} * \text{Timer : } 0.5\text{s}) \overline{\text{Stop}}$$

Konversi dari *switching function* ke *ladder diagram* untuk *step21A* ditunjukkan pada Gambar 3.70 di bawah ini.

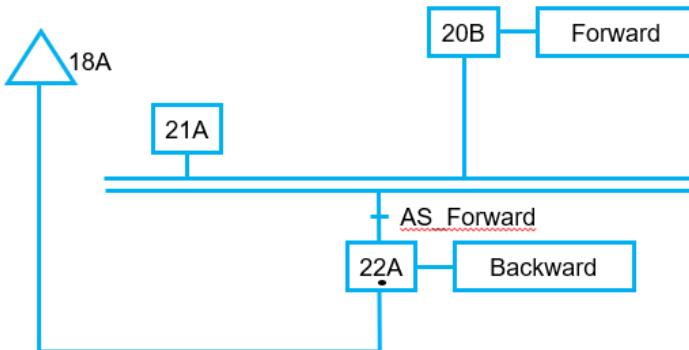


Gambar 3.70 Diagram Ladder Step21A

f) *Step22A*:

Step22A digunakan untuk mengaktifkan *horizontal cylinder* bergerak dari kiri ke kanan (*backward*) untuk benda berwarna biru. Pada

grafset step dengan token di dalamnya menandakan bahwa *step* tersebut aktif. Pada Gambar 3.71 ditunjukkan bahwa *Step22A* sedang aktif dengan transisi *auto switch* (*AS_Foward*). *Step22A* berperan mengaktifkan *horizontal cylinder* bergerak dari kiri ke kanan (*backward*) atau kembali ke posisi semula.



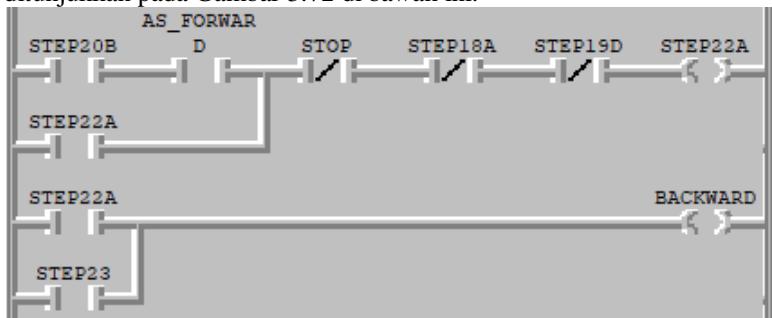
Gambar 3.71 *Grafset Step22A*

Konversi dari *grafset* ke *switching function* untuk *Step22A*:

$$\text{Step22A} = (\text{Step20B} * \text{AS_Forward} + \text{Step22A}) \overline{\text{Stop}} * \overline{\text{Step18A}} * \overline{\text{Step19D}}$$

$$\text{Backward} = \text{Step22A}$$

Konversi dari *switching function* ke *ladder diagram* untuk *step22A* ditunjukkan pada Gambar 3.72 di bawah ini.

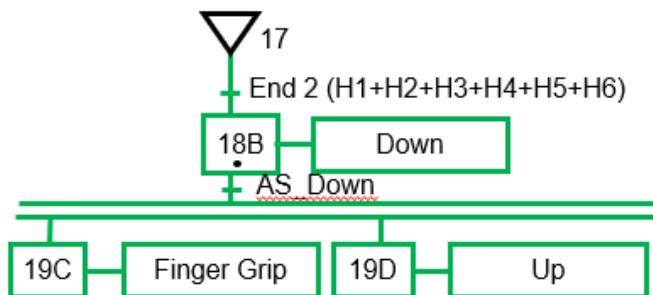


Gambar 3.72 Diagram *Ladder Step22A*

g) *Step18B:*

Ketika benda berwarna hitam melintas dan menyentuh sensor END2 yang berada di ujung *conveyor2* membuat pergerakkan silinder vertikal turun. *Step18B* berperan dalam menurunkan silinder vertikal.

Kondisi *step* dengan token di dalamnya pada *grafcet* menunjukkan bahwa *step* tersebut aktif. Hal ini ditunjukkan pada Gambar 3.82 yang memperlihatkan bahwa *step 18B* aktif dikarenakan transisinya terpenuhi yaitu sensor END2 aktif dan memori hitam1 atau sampai dengan hitam6 aktif. Jalur hijau pada *grafcet* di Gambar 3.73 mengindikasi bahwa jalur khusus digunakan untuk aksi benda berwarna hitam pada *line movement module*.



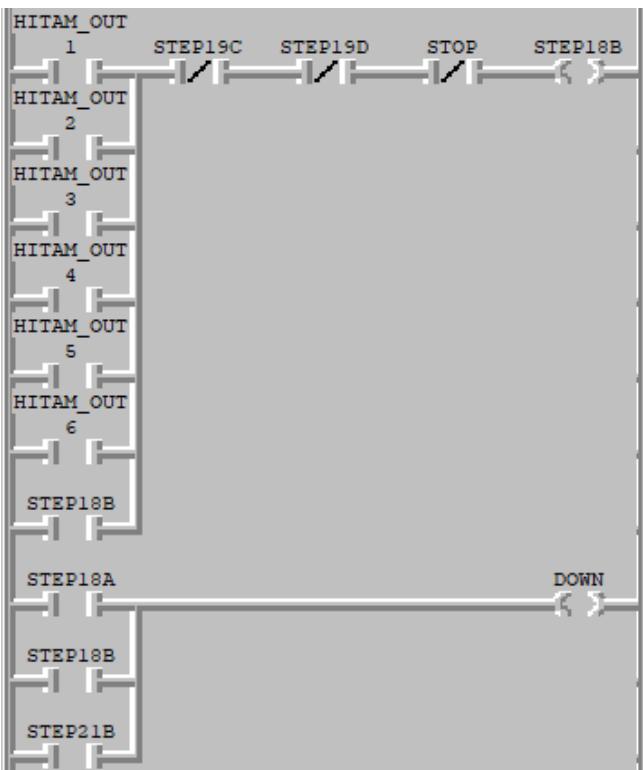
Gambar 3.73 Grafset Step18B

Konversi dari *grafcet* ke *switching fuction* untuk *Step18B*:

$$Step18B = \left(\begin{array}{l} Hitam_Out1 + Hitam_Out2 + \\ Hitam_Out3 + Hitam_Out4 + \\ Hitam_Out5 + Hitam_Out6 + \\ Step18B \end{array} \right) \overline{Step19C * Step19D * Stop}$$

Down = *Step18B*

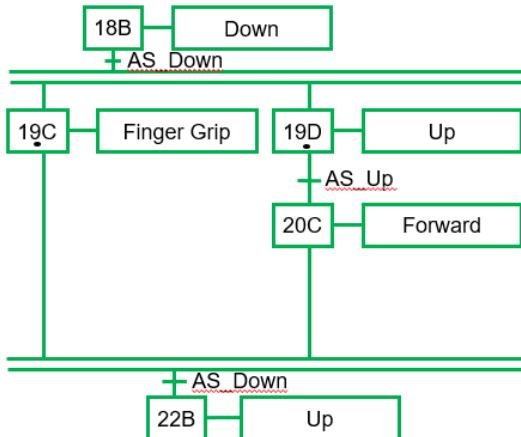
Konversi dari *switching function* ke *ladder diagram* untuk *Step18B* ditunjukkan pada Gambar 3.74 berikut ini.



Gambar 3.74 Diagram Ladder Step18B

h) Step19C dan Step19D

Step19C digunakan untuk mengaktifkan *finger cylinder* dalam kondisi mencengkeram benda (*grip*). Step19D digunakan untuk mengaktifkan *vertical cylinder* untuk naik. Kedua *step* aktif bersama. Pada *grafset step* dengan token di dalamnya menandakan bahwa *step* tersebut aktif. Pada Gambar 3.75, *step19A* yang berperan mengaktifkan *finger cylinder* dalam kondisi mencengkram (*grip*) dan *step19B* yang berperan menaikkan *vertical cylinder* pada *line movement module* untuk benda berwarna biru dalam keadaan aktif.



Gambar 3.75 Grafset Step19C dan Step19D

Konversi dari grafset ke switching fuction untuk Step19C dan Step19D:

$$\text{Step19C} = (\text{Step18B} * \text{AS_Down} + \text{Step19C}) \overline{\text{Stop}} * \overline{\text{Step22B}}$$

$$\text{Step19D} = (\text{Step18B} * \text{AS_Down} + \text{Step19D}) \overline{\text{Stop}} * \overline{\text{Step20C}}$$

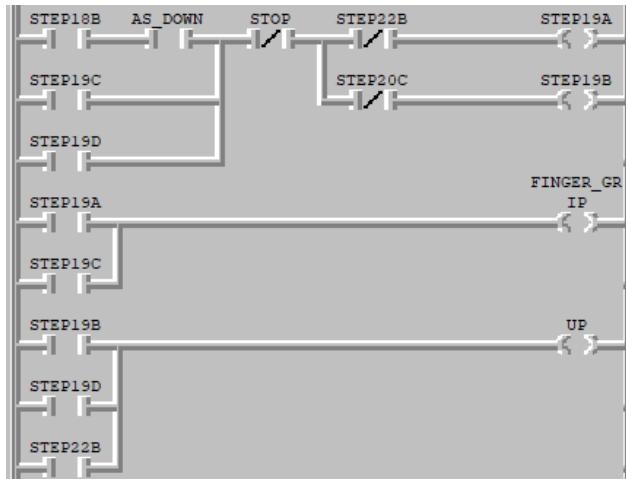
$$\text{Step19C} + \text{Step19D} =$$

$$(\text{Step18B} * \text{AS_Down} + \text{Step19C} + \text{Step19D}) \overline{\text{Stop}} (\overline{\text{Step22B}} + \overline{\text{Step20C}})$$

$$\text{Finger_Grip} = \text{Step19C}$$

$$\text{Up} = \text{Step19D}$$

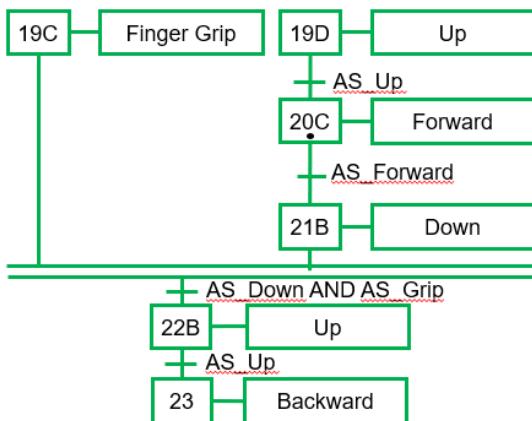
Konversi dari switching function ke ladder diagram untuk Step19C dan Step19D ditunjukkan pada Gambar 3.76 berikut ini.



Gambar 3.76 Diagram Ladder Step 19C dan Step 19D

i) *Step20C:*

Step20C digunakan untuk mengaktifkan *horizontal cylinder* bergerak dari kanan ke kiri (*forward*). Pergerakannya dari posisi semula tepat di atas sensor *photo fiber optic 2* (END2) menuju ke pangkal dari *conveyor1*. Pada *grafset step* dengan token di dalamnya menandakan bahwa *step* tersebut aktif. Pada Gambar 3.77, *step20C* yang berperan mengaktifkan *horizontal cylinder* bergerak *forward* sedang aktif.



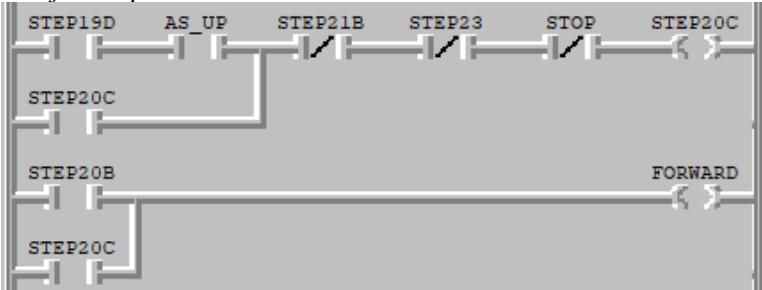
Gambar 3.77 Grafset Step20C

Konversi dari *grafset* ke *switching function* untuk *Step20C*:

$$Step20C = \overline{(Step19D * AS_Up + Step20C)} * \overline{Step21B} * \overline{Step23} * \overline{Stop}$$

Forward = *Step20C*

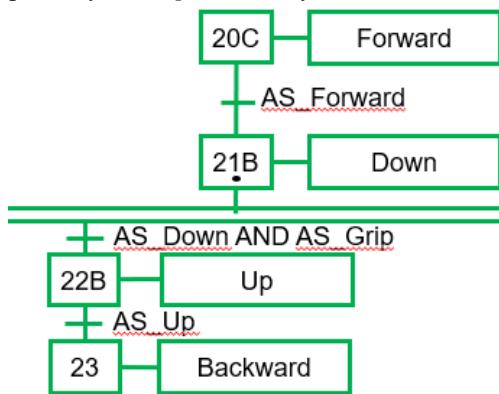
Konversi dari *switching function* ke *ladder diagram* untuk *Step20C* ditunjukkan pada Gambar 3.78 di bawah ini.



Gambar 3.78 Diagram Ladder *Step20C*

j) *Step21B*:

Step21B digunakan untuk mengaktifkan *vertical cylinder* turun khusus untuk benda dengan warna hitam. Pada *grafset step* dengan token di dalamnya merupakan indikasi bahwa *step* tersebut sedang aktif. Pada Gambar 3.79 menunjukkan bahwa *step21B* sedang aktif karena transisi dari *auto switch* (*AS_Foward*). *Step21B* aktif juga dikarenakan syarat perlunya terpenuhi *step* sebelumnya sudah aktif terlebih dahulu.

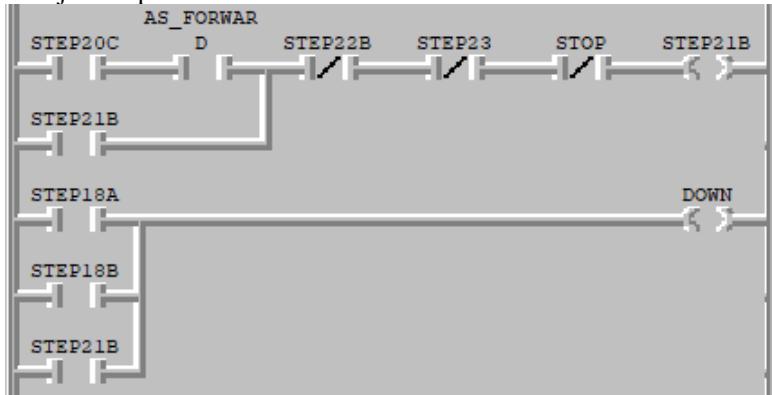


Gambar 3.79 *Grafset Step21B*

Konversi dari *grafset* ke *switching fuction* untuk *Step21B*:

$$Step21B = \left(Step20C * AS_Forward + Step21B \right) \overline{Step22B} * \overline{Step23} * \overline{Stop}$$
$$Down = Step21B$$

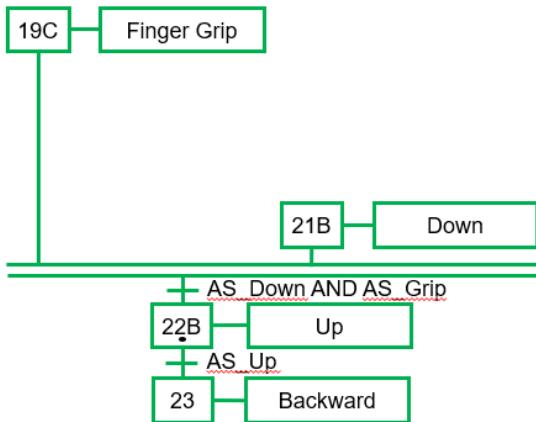
Konversi dari *switching function* ke *ladder diagram* untuk *Step20C* ditunjukkan pada Gambar 3.80 di bawah ini.



Gambar 3.80 Diagram Ladder *Step21B*

k) *Step22B*:

Pada saat silinder vertikal turun maka secara otomatis akan membuat transisi *auto switch* (*AS_Down*) aktif. Dikarenakan pada *step22B finger cylinder* akan membuka *gripper* yang semula dalam kondisi mencengkeram maka transisi untuk mengaktifkan *step* ini terdapat *auto switch* (*AS_Grip*) juga yang terdeteksi. Dari transisi yang telah terpenuhi tersebut akan membuat *step22B* aktif. *Step22B* berperan untuk menaikkan vertikal silinder. Pada Gambar 3.81 ditunjukkan oleh struktur *grafset* bahwa *step* dengan token di dalamnya adalah *step* yang sedang aktif, dalam gambar ini terlihat bahwa *step22B* aktif.



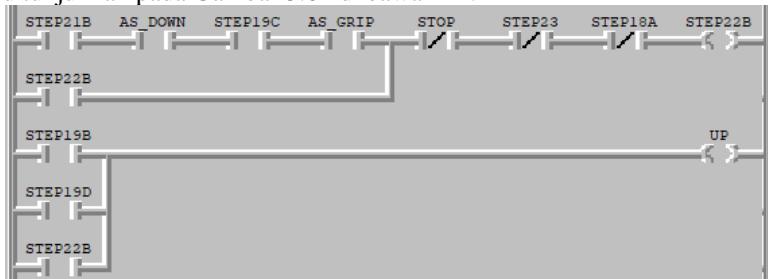
Gambar 3.81 Grafset Step22B

Konversi dari grafset ke switching fuction untuk Step22B:

$$Step22B = \frac{(Step21B * AS_Down * Step19C * AS_Grip + Step22B)}{Stop * Step23 * Step18A}$$

$$Up = Step22B$$

Konversi dari switching function ke ladder diagram untuk Step22B ditunjukkan pada Gambar 3.82 di bawah ini.

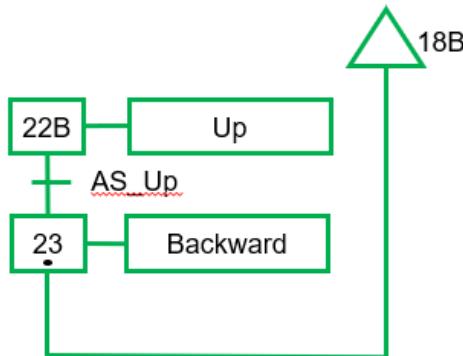


Gambar 3.82 Diagram Ladder Step22B

1) Step23:

Step23 digunakan untuk mengaktifkan *horizontal cylinder* bergerak dari kiri ke kanan (*backward*). Pergerakkan silinder ini adalah dari atas pangkal *conveyor1* menuju ke atas sensor *photo fiber optic2* (END2) atau ke posisi semula. Pada grafset step dengan token di dalamnya merupakan indikasi bahwa step tersebut sedang aktif. Pada Gambar 3.83

menunjukkan bahwa *step23* sedang aktif karena transisi dari *auto switch* (*AS_Up*). *Step23* aktif juga dikarenakan syarat perlunya terpenuhi yaitu *step* sebelumnya sudah aktif terlebih dahulu. Dapat dilihat pada *grafset* di Gambar 3.83 bahwa *step* sebelumnya yang mempengaruhi *step23* aktif adalah *step22B*.



Gambar 3.83 Grafset Step23

Konversi dari *grafset* ke *switching function* untuk *Step23*:

$$Step23 = \overline{(Step22B * AS_Up + Step23)} \overline{Step23} * \overline{Step18A} * \overline{Stop}$$

$$Backward = Step23$$

Konversi dari *switching function* ke *ladder diagram* untuk *Step23* ditunjukkan pada Gambar 3.84 di bawah ini.



Gambar 3.84 Diagram Ladder Step23

-----Halaman ini sengaja dikosongkan-----

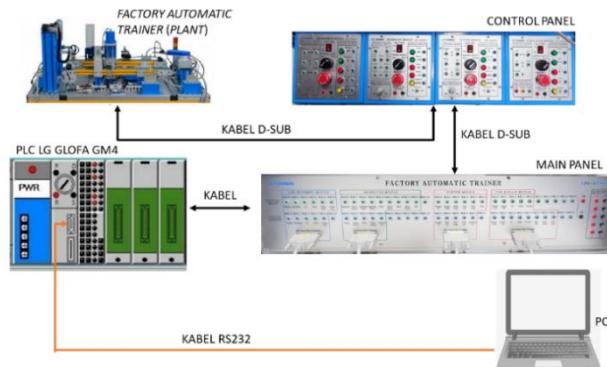
BAB 4

IMPLEMENTASI DAN ANALISA

Bab ini terdiri dari 4 sub bab dimana sub bab yang pertama menjelaskan mengenai konfigurasi pada *factory automatic trainer* yang mana menjadi acuan untuk menjalankan *factory automatic trainer*. Pada sub bab yang kedua membahas mengenai validasi sistem. Pada sub bab yang ketiga merupakan hasil implementasi sistem yang ditunjukkan dari kesesuaian diagram *ladder* yang dibuat dengan aksi aktuator pada *plant*. Pada sub bab yang keempat membahas mengenai hasil pengujian.

4.1 Konfigurasi *Factory Automatic Trainer*

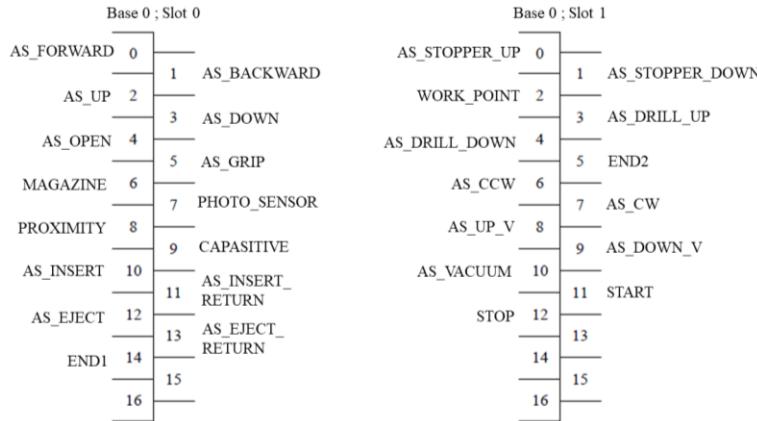
Implementasi pada *factory automatic trainer* dimulai dengan melakukan proses *wiring* seperti pada Gambar 4.1 di bawah ini. PLC LG Glofa GM4 dihubungkan dengan *main panel* dengan menggunakan kabel d-sub. *Control panel* digunakan untuk menggerakkan aktuator pada *factory automatic trainer* secara manual. Pada *control panel* disediakan tombol – tombol yang mewakili setiap aktuator pada setiap modul di *factory automatic trainer*. *Control panel* dihubungkan dengan *main panel* dan dihubungkan dengan *plant factory automatic trainer* sendiri dengan menggunakan kabel d-sub.



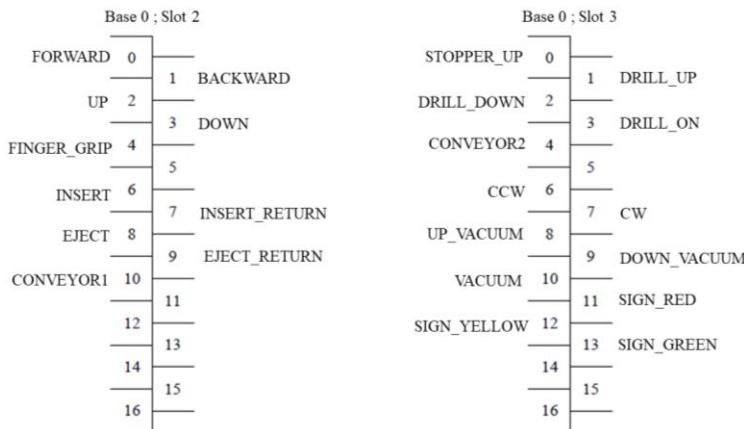
Gambar 4.1 Konfigurasi *Factory Automatic Trainer*

Komunikasi antara PLC dengan PC menggunakan kabel RS232. Komunikasi digunakan untuk melakukan proses unggah (*upload*) program diagram *ladder* dari PC ke PLC. *Main panel* terhubung dengan

I/O PLC yang tersedia 2 modul *input* (slot 0 dan slot 1) dan 2 modul *output* (slot 2 dan slot 3). Berikut merupakan *wiring I/O (input dan output)* PLC yang terhubung dengan *main panel* seperti pada Gambar 4.2 dan Gambar 4.3.



Gambar 4.2 Wiring Input PLC LG



Gambar 4.3 Wiring Output PLC LG

Berikut merupakan hasil pengkabelan terminal PLC LG yang terhubung dengan *main panel* ditunjukkan pada Gambar 4.4. G4I-D22A atau slot PLC dengan tulisan berwarna biru menunjukkan alamat *input* dari PLC LG Glofa GM4 sedangkan untuk G4Q-RY2A atau slot PLC dengan tulisan berwarna *orange* menunjukkan alamat *output* dari PLC LG Glofa GM4.



Gambar 4.4 Hasil Pengkabelan

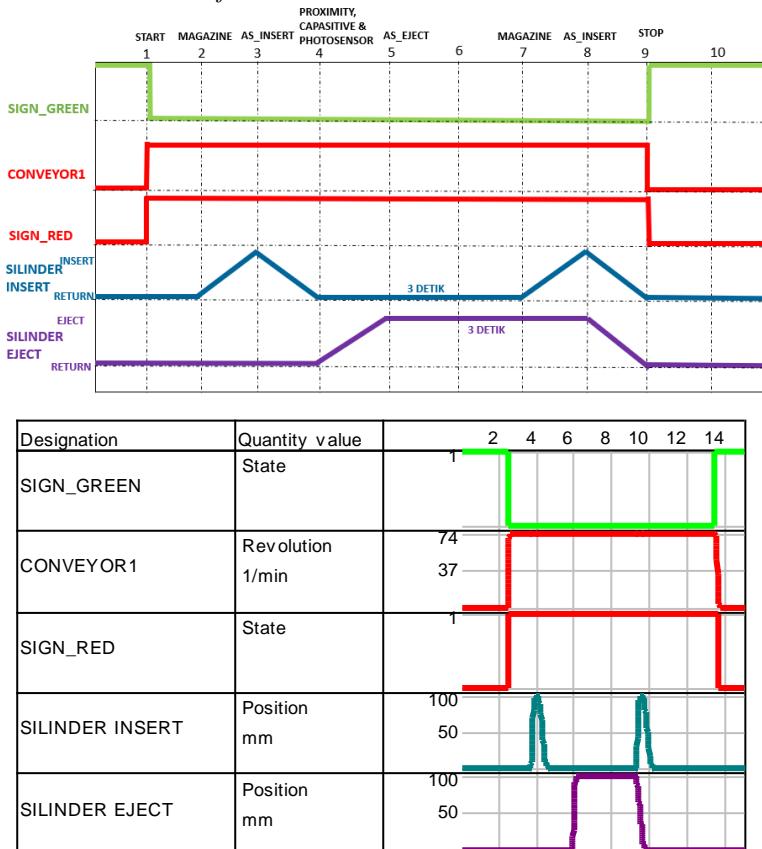
4.2 Validasi

Tujuan dari kegiatan validasi adalah membuktikan bahwa diagram *ladder* yang dirancang sesuai dengan spesifikasi atau deskripsi sistem. Dalam membutikannya dilakukan dengan menggunakan bantuan dari *software* simulasi (fluidsim) dimana diagram *ladder* dibuat kembali dan dihubungkan dengan simbol aktuator pada fluidsim yang berupa indikator lampu, *buzzer*, *solenoid* serta motor. Setelah membuat rangkaian elektrik, untuk aktuator berupa *solenoid* perlu ditambahkan rangkaian pneumatik dikarenakan keluaran dari diagram waktu untuk *solenoid* berbeda dengan yang lain. Setelah rangkaian elektrik dan pneumatik dibuat kemudian simulasi dijalankan. Hasil simulasi dapat ditampilkan berupa diagram waktu atau *timing chart* yang dibandingkan sesuai dengan rancangan diagram waktu yang telah dibuat.

4.2.1 Validasi pada *Separation Module*

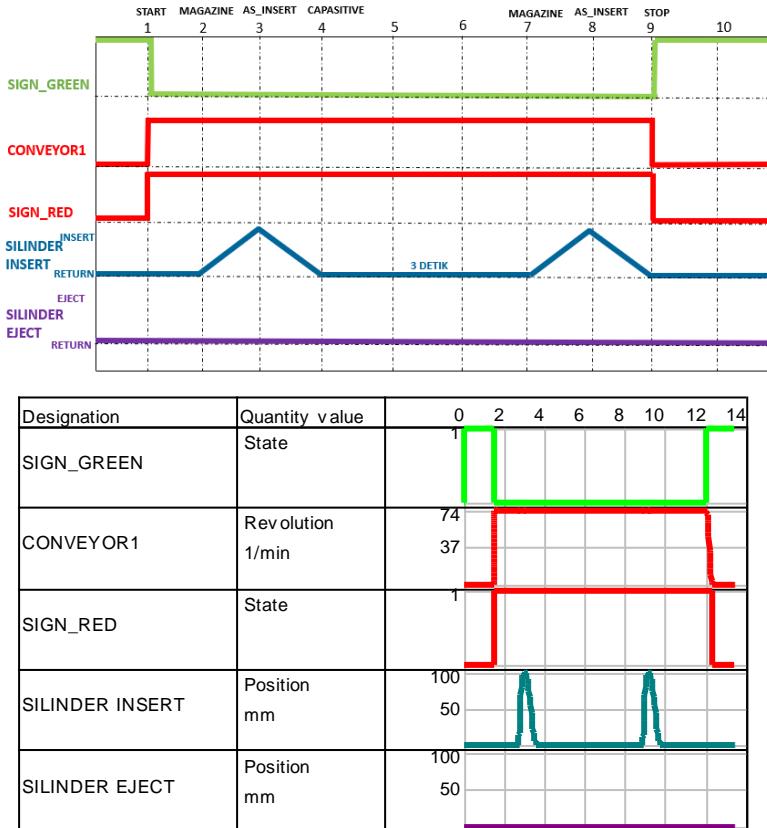
Validasi pada *separation module* dibuat dengan membuat rangkaian elektrik dan pneumatik menyesuaikan dengan hasil perancangan *grafset*

seperti pada Lampiran C.1. Pada *separation module* aktuator yang digunakan untuk bagian elektrik adalah indikator *led* untuk indikator lampu hijau dan merah serta motor untuk *conveyor1*. Sedangkan untuk silinder *insert* dan silinder *eject* menggunakan *solenoid*. Setelah program dijalankan untuk benda pertama melintas berwarna *silver* ditunjukkan diagram waktu seperti pada Gambar 4.5 di bawah ini. Gambar diagram waktu yang di atas menunjukkan rancangan deskripsi sistem yang dibuat sedangkan untuk gambar diagram waktu yang di bawah menunjukkan hasil simulasi dari *software fluidsim*.



Gambar 4.5 Validasi Separation Module Benda Silver

Pada kondisi benda pertama berwarna hitam atau biru berbeda dengan benda berwarna *silver*, perbedaannya terletak di aksi silinder *eject*. Benda biru dan hitam memiliki keluaran diagram waktu yang sama. Perbedaannya terletak pada deteksi sensor. Diagram waktu untuk benda pertama berwarna hitam ditunjukkan pada Gambar 4.6 berikut ini.



Gambar 4.6 Validasi *Separation Module* Benda Hitam

Dari hasil validasi sistem dapat dikatakan memiliki kesesuaian dengan spesifikasi yang diinginkan. Hasil validasi pada *separation module* ditunjukkan pada Tabel 4.1 berikut ini.

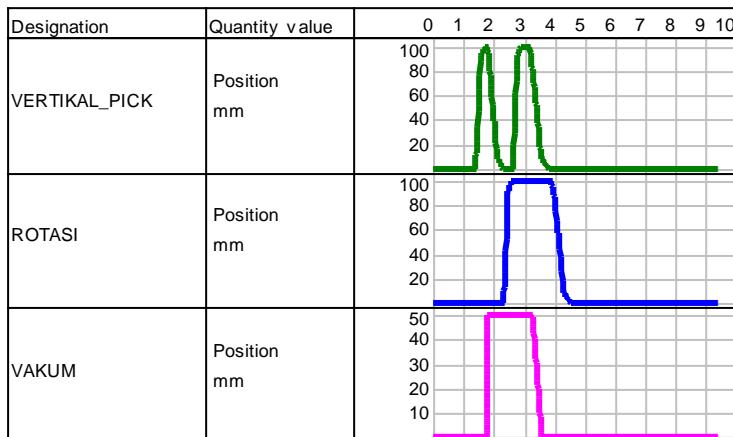
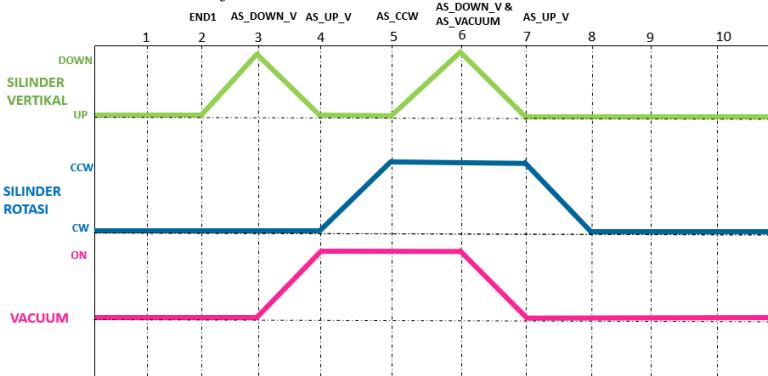
Tabel 4.1 Hasil Validasi *Separation Module*

<i>Step</i>	Aksi	Hasil
<i>Step 1</i>	Lampu Indikator dengan warna hijau menyala (<i>Sign_Green</i> aktif)	√
<i>Step 2</i>	Lampu indikator dengan warna merah menyala (<i>Sign_Red</i> aktif) + Conveyor1 aktif.	√
<i>Step 3A</i>	<i>Silinder Insert</i> mendorong benda kerja menuju conveyor1	√
<i>Step 3B</i>	Lampu indikator dengan warna merah menyala (<i>Sign_Red</i> aktif) dan Conveyor1 aktif.	√
<i>Step 4</i>	<i>Silinder Insert Return</i> mengembalikan posisi <i>silinder</i>	√
<i>Step 5A</i>	Mengaktifkan Timer 3s untuk memasukkan benda kerja kembali setelah benda kerja berwarna <i>silver</i> melewati sensor.	√
<i>Step 5B</i>	<i>Silinder Eject</i> menyisihkan benda berwarna <i>silver</i>	√
<i>Step 5C</i>	Mengaktifkan Timer 3s untuk memasukkan benda kerja kembali setelah benda kerja berwarna hitam melewati sensor.	√
<i>Step 5D</i>	Mengaktifkan Timer 3s untuk memasukkan benda kerja kembali setelah benda kerja berwarna biru melewati sensor.	√
<i>Step 5E (TP)</i>	Mengaktifkan <i>timer pulse</i> selama 5s	√
<i>Step 6</i>	Mengaktifkan Timer 3s untuk menahan silinder <i>eject</i> pada posisi <i>energize</i> selama 3 detik.	√
<i>Step 7</i>	Silinder <i>eject</i> kembali ke posisi semula (<i>eject return</i>)	√
<i>Step 8</i>	Mengindikasi timer 3s selesai menghitung.	√

4.2.2 Validasi pada *Pick and Place Module*

Validasi pada *pick and place module* dibuat dengan membuat rangkaian elektrik dan pneumatik yang menyesuaikan dengan hasil perancangan *grafset* seperti pada Lampiran C.2. Aktuator pada *pick and*

place module yang digunakan adalah *solenoid* untuk silinder vertikal, silinder rotasi serta *vacuum*. Setelah program dijalankan menghasilkan diagram waktu seperti pada Gambar 4.7 di bawah ini. Gambar diagram waktu yang di atas menunjukkan rancangan deskripsi sistem yang dibuat sedangkan untuk gambar diagram waktu yang di bawah menunjukkan hasil simulasi dari *software fluidsim*.



Gambar 4.7 Validasi *Pick and Place Module*

Dari hasil validasi sistem dapat dikatakan memiliki kesesuaian dengan spesifikasi yang diinginkan. Hasil validasi pada *pick and place module* ditunjukkan pada Tabel 4.2 berikut ini.

Tabel 4.2 Hasil Validasi *Pick and Place Module*

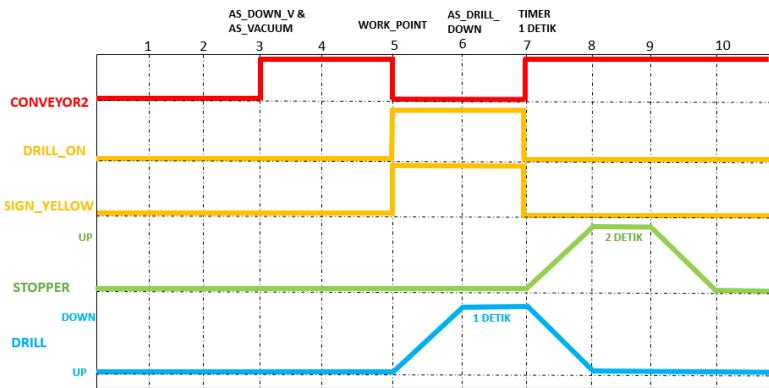
Step	Aksi	Hasil
Step 9A	Silinder Vertikal pada <i>pick and place module</i> turun (Down_Vacuum)	√
Step 10A	Vacuum Pad menghisap benda kerja	√
Step 10B	Silinder Vertikal pada <i>pick and place module</i> naik (UP_Vacuum)	√
Step 11	Silinder Rotasi pada <i>pick and place module</i> bergerak berlawanan dengan arah jarum jam (CCW)	√
Step 12	Silinder Vertikal pada <i>pick and place module</i> turun (Down_Vacuum)	√
Step 13A	Silinder Vertikal pada <i>pick and place module</i> naik (UP_Vacuum)	√
Step 13B	Conveyor 2 aktif	√
Step 14A	Silinder Rotasi pada <i>pick and place module</i> bergerak searah dengan arah jarum jam (CW)	√

4.2.3 Validasi pada *Stopper Module*

Validasi pada *stopper module* dibuat dengan membuat rangkaian elektrik dan pneumatik seperti pada Lampiran C.3. Pada *stopper module* aktuator yang digunakan untuk bagian elektrik adalah indikator led untuk indikator lampu kuning serta motor untuk *conveyor2* dan *driller*. Sedangkan untuk *stopper* dan silinder *drill* dengan pergerakan vertikal menggunakan *solenoid*. Setelah program dijalankan menghasilkan diagram waktu seperti pada Gambar 4.8 berikut ini.

Pada Gambar 4.8 berikut menggambarkan bahwa *conveyor2* aktif ketika *switch* (AS_Down_V serta AS_Vacuum) pada *pick and place module* aktif. Pergerakan aktuator pada *conveyor2* ditunjukkan dengan garis berwarna merah. Pada saat benda menyentuh sensor *work_point* mengakibatkan pergerakan *conveyor2* terhenti, *drilling machine* aktif, lampu indikator berwarna kuning (*sign_yellow*) aktif yang ditandakan dengan garis berwarna kuning serta silinder vertikal untuk *drilling machine* turun (*down*) ditunjukkan pada garis berwarna biru. Pergerakan silinder turun membuat *switch* (AS_Drill_Down) tersentuh sehingga *timer* bekerja selama 1 detik. Setelah *timer* 1 detik selesai menghitung *conveyor2* berjalan kembali, *drilling machine* dan *sign_yellow* non-aktif *stopper* naik serta *drilling machine* naik. Pergerakan *stopper* naik membuat *switch* (AS_Stopper_Up) terdeteksi membuat *timer* selama 2

detik aktif. Setelah terhitung selama 2 detik *stopper* kembali turun ke posisi semula. Pergerakan aktuator pada *stopper* ditunjukkan dengan garis berwarna hijau pada diagram waktu berikut ini.



Designation	Quantity value	0	1	2	3	4	5	6	7	8	9	10
CONVEYOR2	Rev olution 1/min	74						37				
DRILL_ON	Rev olution 1/min	74					37					
SIGN_YELLOW	State	1										
STOPPER	Position mm	50					25		50			
VERTIKAL DRILL	Position mm	100					50		100			

Gambar 4.8 Validasi Stopper Module

Dari hasil validasi sistem dapat dikatakan memiliki kesesuaian dengan spesifikasi yang diinginkan. Hasil validasi pada *stopper module* ditunjukkan pada Tabel 4.3 berikut ini.

Tabel 4.3 Hasil Validasi *Stopper Module*

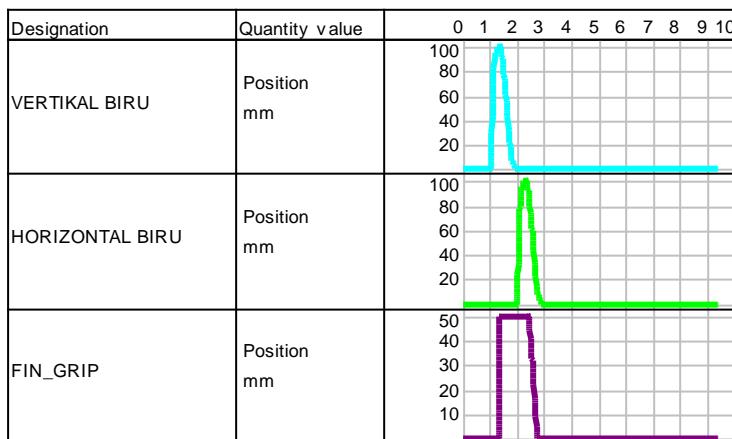
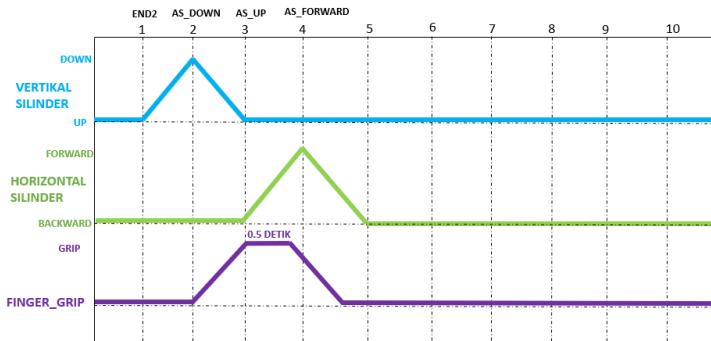
Step	Aksi	Hasil
Step14B	Mesin <i>driller</i> pada kondisi menyala (ON) + Lampu indikator berwarna kuning (<i>Sign_Yellow</i>) menyala	√
Step14C	Mesin <i>driller</i> turun (<i>Drill_Down</i>)	√
Step15	Mengaktifkan <i>timer</i> selama 1 detik	√
Step15_Tim	Mengindikasi <i>timer</i> 1s selesai menghitung	√
Step16	<i>Stopper</i> naik (<i>stopper up</i>) + mesin <i>driller</i> naik (<i>Drill_up</i>) + <i>Conveyor 2</i> aktif	√
Step17	Mengaktifkan <i>timer</i> selama 2 detik	√

4.2.4 Validasi pada *Line Movement Module*

Validasi pada *line movement module* dibuat dengan membuat rangkaian elektrik dan pneumatik menyesuaikan dengan hasil perancangan *grafset* seperti pada Lampiran C.4. Pada *line movement module* aktuator yang digunakan adalah *solenoid* untuk menggerakkan silinder vertikal, silinder horizontal dan *finger cylinder* untuk mencengkeram benda. Setelah rangkaian elektrik dan pneumatik dibuat kemudian program dijalankan untuk mendapatkan hasil diagram waktu pada fluidsim. Pada Gambar 4.9 menunjukkan hasil perbandingan diagram waktu antara hasil program dengan diagram waktu yang dibuat berdasarkan deskripsi sistem untuk aksi dari *line movement module* benda berwarna biru. Gambar diagram waktu yang di atas menunjukkan rancangan deskripsi sistem yang dibuat sedangkan untuk gambar diagram waktu yang di bawah menunjukkan hasil simulasi dari *software fluidsim*.

Pada saat benda dengan warna biru terdeteksi oleh sensor END2 membuat silinder vertikal bergerak turun (*down*). Pergerakan ini mengakibatkan *switch* (*AS_Down*) aktif. Pada saat *switch* aktif silinder vertikal kembali naik beserta *finger grip* dalam kondisi mencengkeram benda. Dikarenakan silinder vertikal naik membuat *switch* (*AS_Up*) aktif. Ketika *switch* (*AS_Up*) aktif silinder horizontal bergerak secara *forward* serta *timer* selama 0.5 detik aktif. Pada saat *timer* selesai menghitung selama 0.5 detik *finger grip* membuka cengkraman sehingga benda biru jatuh tepat di kotak yang diletakkan pada tengah pergerakan *line*

movement module. Pergerakkan silinder *horizontal* secara *forward* juga membuat *switch* (AS_Foward) aktif. Ketika *switch* (AS_Foward) aktif silinder horizontal bergerak *backward* kembali ke posisi semula dan mengulangi kembali pergerakkan ketika terdapat benda biru kembali yang terdeteksi oleh sensor END2. Aksi aktuator silinder vertikal ditunjukkan dengan garis berwarna biru, silinder horizontal dengan garis berwarna hijau serta *finger grip* dengan garis berwarna ungu.



Gambar 4.9 Validasi *Line Movement Module* Benda Biru

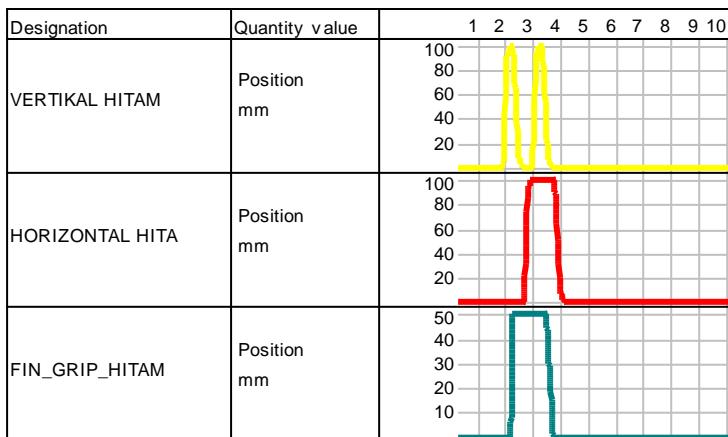
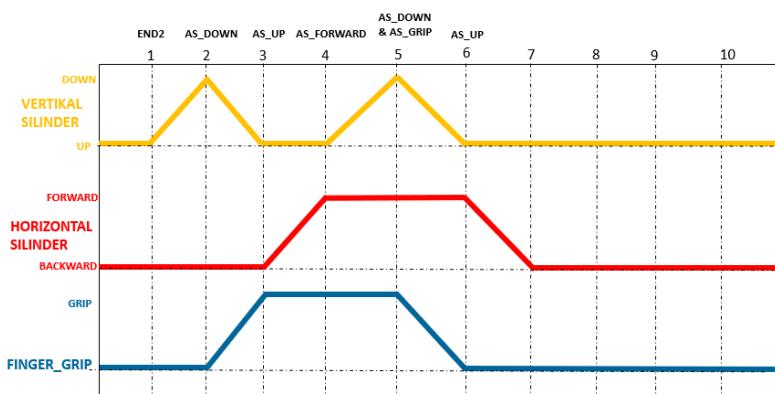
Dari hasil validasi sistem dapat dikatakan memiliki kesesuaian dengan spesifikasi yang diinginkan. Hasil validasi pada *line movement module* benda biru ditunjukkan pada Tabel 4.4 berikut ini.

Tabel 4.4 Hasil Validasi *Line Movement Module* Benda Biru

Step	Aksi	Hasil
Step18A	Mengaktifkan <i>vertical cylinder</i> untuk turun (<i>Down</i>)	√
Step19A	Mengaktifkan <i>finger cylinder</i> dalam kondisi <i>grip</i>	√
Step19B	Mengaktifkan <i>vertical cylinder</i> naik (<i>up</i>)	√
Step20A	Mengaktifkan <i>timer 0.5s</i>	√
Step20B	Mengaktifkan <i>horizontal cylinder</i> bergerak dari kanan ke kiri (<i>forward</i>)	√
Step21A	Mengindikasikan bahwa <i>timer 0.5s</i> sudah terhitung	√
Step22A	Mengaktifkan <i>horizontal cylinder</i> bergerak dari kiri ke kanan (<i>backward</i>)	√

Pada saat benda dengan warna hitam terdeteksi oleh sensor END2 membuat silinder vertikal bergerak turun (*down*). Pergerakan ini mengakibatkan *switch* (AS_Down) aktif. Pada saat *switch* aktif silinder vertikal kembali naik beserta *finger grip* dalam kondisi mencengkeram benda. Dikarenakan silinder vertikal naik membuat *switch* (AS_Up) aktif. Ketika *switch* (AS_Up) aktif silinder horizontal bergerak secara *forward*. Pergerakkan silinder *horizontal* secara *forward* juga membuat *switch* (AS_Forward) aktif. Ketika *switch* (AS_Forward) aktif, silinder vertikal bergerak turun. Pergerakan ini membuat *switch* (AS_Down serta AS_Vacuum) aktif. Pada saat *switch* tersebut aktif mengakibatkan silinder vertikal bergerak naik kembali serta *finger grip* membuka cengkeraman. Pergerakkan silinder vertikal membuat *switch* (AS_Up) tersentuh dan mengakibatkan silinder horizontal bergerak secara *backward*.

Aktuator yang digunakan untuk *line movement module* benda hitam sama dengan aktuator pada benda biru. Aksi untuk silinder vertikal ditunjukkan dengan garis berwarna kuning, aksi untuk silinder horizontal ditunjukkan dengan garis berwarna merah sedangkan aksi untuk silinder *finger grip* ditunjukkan dengan garis berwarna biru pada diagram waktu. Pada benda berwarna hitam memiliki respon diagram waktu seperti pada Gambar 4.10 berikut ini.



Gambar 4.10 Validasi *Line Movement Module* Benda Hitam

Dari hasil validasi sistem dapat dikatakan memiliki kesesuaian dengan spesifikasi yang diinginkan. Hasil validasi pada *line movement module* benda hitam ditunjukkan pada Tabel 4.5 berikut ini.

Tabel 4.5 Hasil Validasi *Line Movement Module* Benda Hitam

Step	Aksi	Hasil
Step18B	Mengaktifkan <i>vertical cylinder</i> untuk turun (<i>Down</i>)	✓
Step19C	Mengaktifkan <i>finger cylinder</i> dalam kondisi <i>grip</i>	✓

<i>Step</i>	<i>Aksi</i>	<i>Hasil</i>
<i>Step19D</i>	Mengaktifkan <i>vertical cylinder</i> naik (<i>up</i>)	✓
<i>Step20C</i>	Mengaktifkan <i>horizontal cylinder</i> bergerak dari kanan ke kiri (<i>forward</i>)	✓
<i>Step21B</i>	Mengaktifkan <i>vertical cylinder</i> untuk turun (<i>Down</i>)	✓
<i>Step22B</i>	Mengaktifkan <i>vertical cylinder</i> untuk naik (<i>Up</i>)	✓
<i>Step23</i>	Mengaktifkan <i>horizontal cylinder</i> bergerak dari kiri ke kanan (<i>backward</i>)	✓

Hasil perbandingan diagram waktu dari deskripsi sistem dengan hasil diagram waktu dari hasil simulasi *software fluidsim* menunjukkan kesesuaian proses yang diinginkan. Selisih waktu beberapa detik terjadi dikarenakan pada proses simulasi terbatasi oleh *push button* yang berperan sebagai sensor. Pada saat operator mengalami keterlambatan dalam menekan *push button* dapat mempengaruhi hasil dari aksi aktuator.

4.3 Implementasi

Tujuan dari implementasi sistem adalah untuk mengetahui apakah rancangan diagram *ladder* yang telah dibuat sudah sesuai dengan aksi dari aktuator *factory automatic trainer*. Pada implementasi sistem dijelaskan mengenai implementasi pada *separation module*, implementasi pada *pick and place module*, implementasi pada *stopper module* serta implementasi pada *line movement module*. Pada implementasi sistem dilakukan pengujian terhadap diagram *ladder* serta aksi aktuator dari *plant* yang ditunjukkan dengan dokumentasi alat.

4.3.1 Implementasi pada *Separation Module*

a) *Step1*:

Pada kondisi awal sistem non-aktif yang diindikasikan dengan lampu indikator menyala berwarna hijau. Pada saat kondisi sistem sedang beroperasi dan kemudian operator menekan tombol STOP juga mengakibatkan indikator lampu hijau ini aktif. Implementasi diagram *ladder step1* dapat dilihat pada Gambar 4.11 berikut ini.



Gambar 4.11 Implementasi Diagram *Ladder Step1*

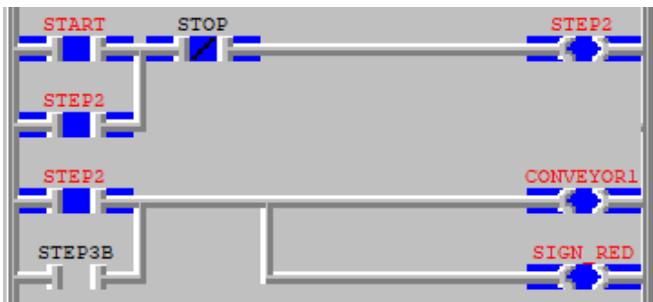
Hasil implementasi untuk aksi *step1* dapat dilihat pada Gambar 4.12 yang ditunjukkan dengan indikator lampu hijau menyala.



Gambar 4.12 Hasil Implementasi Aksi Step 1

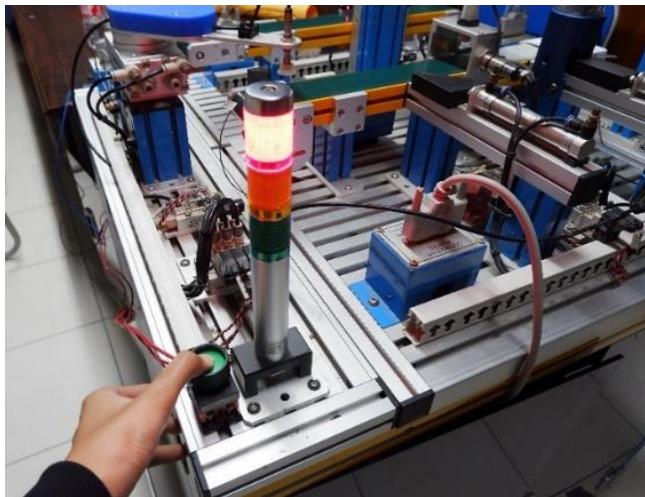
b) *Step2:*

Sistem beroperasi ketika tombol START ditekan. *Step2* mengaktifkan *conveyor1* dan indikator lampu berwarna merah. Indikator lampu merah mengindikasi bahwa sistem sedang beroperasi. Implementasi diagram *ladder step2* dapat dilihat pada Gambar 4.13 di berikut ini.



Gambar 4.13 Implementasi Diagram Ladder Step2

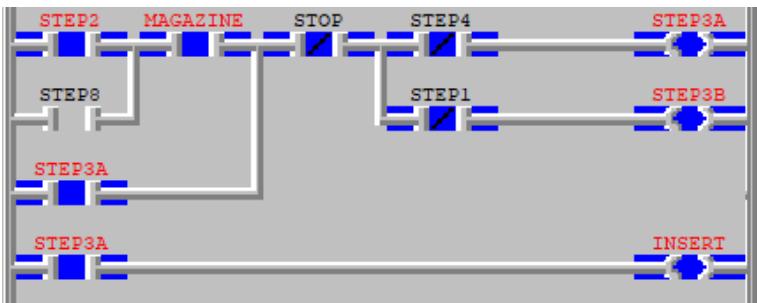
Hasil implementasi untuk aksi *step2* dapat dilihat pada Gambar 4.14. Operator menekan tombol START untuk menyalakan lampu indikator berwarna merah.



Gambar 4.14 Hasil Implementasi Aksi Step2

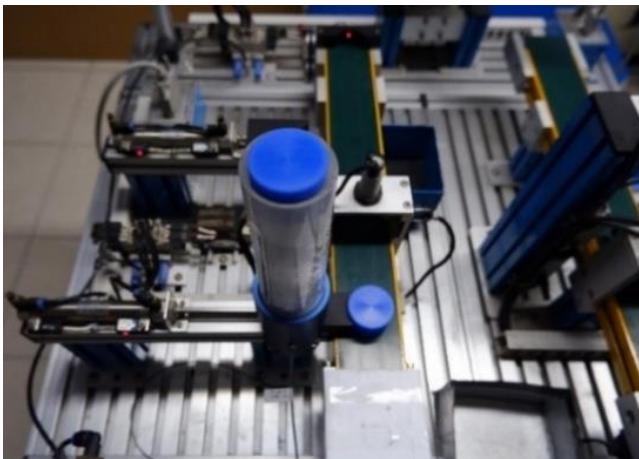
c) *Step3A & Step3B:*

Pada saat benda diletakkan di *feeding tube* akan terdeteksi oleh sensor *magazine* yang mengakibatkan silinder *insert* aktif serta *conveyor1* dan indikator lampu berwarna merah masih aktif. Implementasi diagram *ladder step3A* dan *step3B* dapat dilihat pada Gambar 4.15 berikut ini.



Gambar 4.15 Implementasi Diagram Ladder Step3A & Step3B

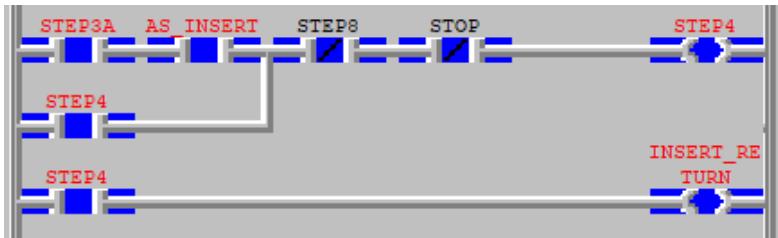
Hasil implementasi untuk aksi *step3A* dan *step3B* dapat dilihat pada Gambar 4.16 yang ditunjukkan dengan silinder *insert* mendorong benda kerja menuju *conveyor1*.



Gambar 4.16 Hasil Implementasi Aksi Step3A & Step3B

d) *Step4:*

Pada *step4*, kondisi *auto switch* (*AS_Insert*) terdeteksi dikarenakan terjadinya pergerakkan silinder *insert* hal ini mengakibatkan silinder *insert* kembali ke posisi semula (*insert return*). Implementasi diagram ladder *step4* melalui *software GMWIN* dapat dilihat pada Gambar 4.17 berikut ini ini.



Gambar 4.17 Implementasi Diagram *Ladder Step4*

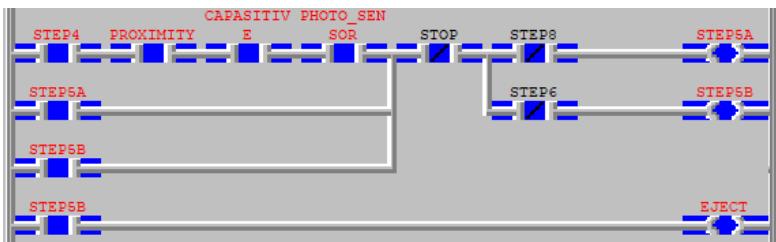
Hasil implementasi untuk aksi *step4* dapat dilihat pada Gambar 4.18 yang ditunjukkan dengan silinder *insert* kembali ke posisi semulanya.



Gambar 4.18 Hasil Implementasi Aksi *Step4*

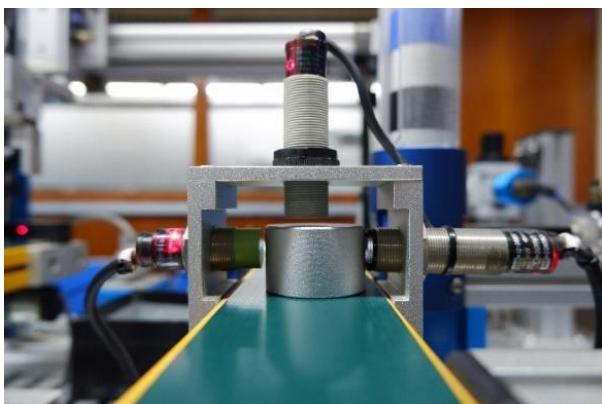
e) *Step5A & Step5B*

Pada saat benda berwarna silver yang melintas di area peletakan sensor mengakibatkan *proximity inductive sensor* mendeteksi benda berbahan logam, *proximity capacitive sensor* mendeteksi keberadaan adanya benda berbahan logam maupun non-logam dan *photoelectric sensor* mendeteksi benda berwarna terang. Hal ini mengakibatkan *step5A* yang mengaktifkan *timer 3* detik dan *step5B* yang mengakibatkan silinder *eject* aktif. Implementasi diagram *ladder* *step5A* dan *step5B* melalui *software GMWIN* dapat dilihat pada Gambar 4.19 berikut ini.



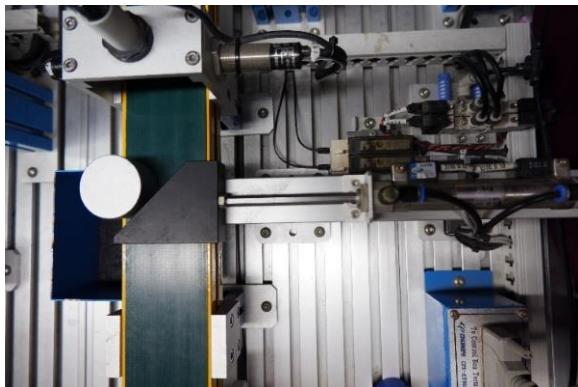
Gambar 4.19 Implementasi Diagram Ladder Step5A & Step5B

Hasil implementasi untuk *step5A* dan *step5B* dapat dilihat pada Gambar 4.20 yang ditunjukkan dengan benda berwarna *silver* melintas di letak sensor. Ditunjukkan pada Gambar 4.20 bahwa *inductive proximity sensor*, *capacitive proximity sensor* serta *photoelectric sensor* menyala.



Gambar 4.20 Hasil Implementasi Step5A & Step5B

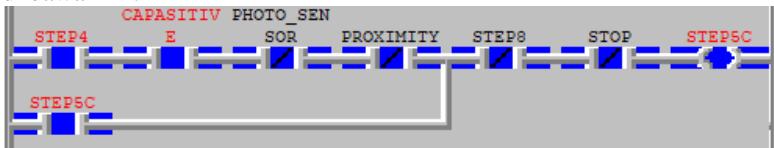
Sedangkan aksi *step5B* yang berperan dalam mengaktifkan silinder *eject* untuk membuang benda berwarna *silver* yang memiliki bahan material logam ditunjukkan pada Gambar 4.21 berikut ini. Aksi *step5A* berupa pengaktifan *timer* selama 3 detik untuk mendorong kembali benda selanjutnya sehingga sistem dapat berjalan secara terus – menerus selama benda terdeteksi sensor *magazine*. Aksi pendorongan benda selanjutnya ditunjukkan pada pembahasan *step8* yang berkesinambungan dengan *step5A*.



Gambar 4.21 Hasil Implementasi Aksi Step5B

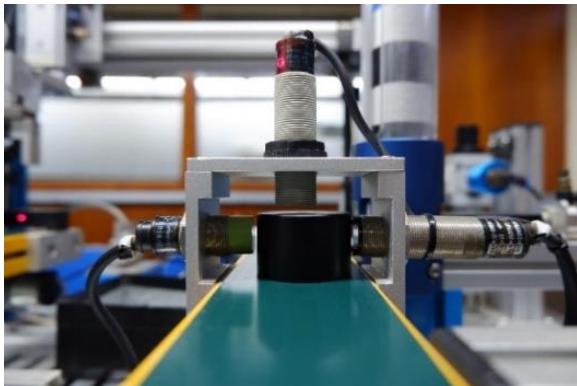
f) *Step5C*

Pada saat benda berwarna hitam melintas di area peletakkan sensor mengakibatkan *proximity capacitive sensor* mendeteksi adanya keberadaan benda berbahan non-logam. Hal ini mengakibatkan *step5C* yang berfungsi mengaktifkan *timer 3* detik aktif. Implementasi diagram *ladder step5C* melalui *software GMWIN* dapat dilihat pada Gambar 4.22 di bawah ini.



Gambar 4.22 Implementasi Diagram Ladder Step5C

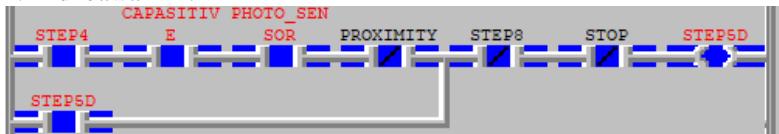
Hasil implementasi untuk *step5C* dapat dilihat pada Gambar 4.17 yang ditunjukkan dengan benda berwarna hitam melintas di letak sensor. Ditunjukkan pada Gambar 4.23 bahwa *capacitive proximity sensor* menyala. Aksi *step5C* berupa pengaktifan *timer* selama 3 detik untuk mendorong kembali benda selanjutnya sehingga sistem dapat berjalan secara terus – menerus selama benda terdeteksi sensor *magazine*. Aksi pendorongan benda selanjutnya ditunjukkan pada pembahasan *step8* yang berkesinambungan dengan *step5C*.



Gambar 4.23 Hasil Implementasi Step5C

g) *Step5D*

Pada saat benda berwarna biru melintas ke area peletakkan sensor mengakibatkan *proximity capacitive sensor* mendeteksi benda berbahan non-logam dan *photoelectric sensor* mendeteksi benda berwarna terang. Hal ini membuat *step5D* yang berfungsi untuk mengaktifkan *timer* 3 detik aktif. Implementasi diagram *ladder* *step5D* dapat dilihat pada Gambar 4.24 di bawah ini.



Gambar 4.24 Implementasi Diagram Ladder *Step5D*

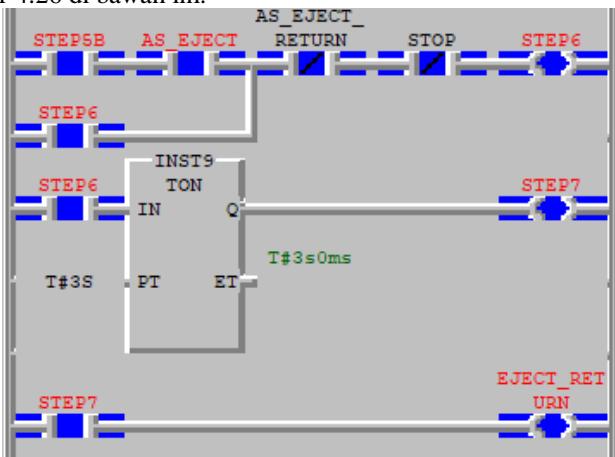
Hasil implementasi untuk *step5D* dapat dilihat pada Gambar 4.25 yang ditunjukkan dengan benda berwarna biru melintas di letak sensor dengan *capacitive proximity sensor* serta *photoelectric sensor* menyala. Aksi *step5D* berupa pengaktifan *timer* selama 3 detik untuk mendorong kembali benda selanjutnya sehingga sistem dapat berjalan secara terus – menerus selama benda terdeteksi sensor *magazine*. Aksi pendorongan benda selanjutnya ditunjukkan pada pembahasan *step8* yang berkesinambungan dengan *step5D*.



Gambar 4.25 Hasil Implementasi Step5D

h) *Step6 & Step7*

Pada saat benda berwarna silver tersisihkan oleh silinder *eject* maka *auto switch* (*AS_Eject*) akan aktif yang membuat *step6* yang mengaktifkan *timer 3* detik untuk menahan posisi silinder *eject* aktif. Setelah *timer 3* detik selesai menghitung selama 3 detik maka akan mengaktifkan *step7* yang berfungsi mengembalikan posisi silinder *eject* ke posisi semula. Implementasi diagram *ladder step7* dapat dilihat pada Gambar 4.26 di bawah ini.



Gambar 4.26 Implementasi Diagram Ladder Step6 & Step7

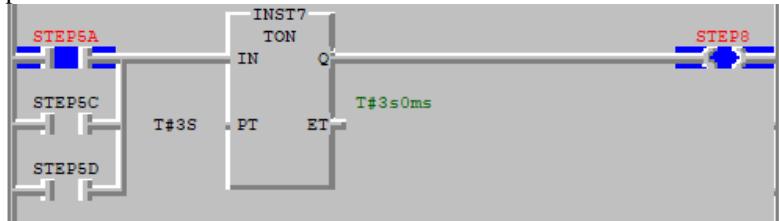
Hasil implementasi untuk *step7* dapat dilihat pada Gambar 4.27 yang ditunjukkan dengan silinder *eject* kembali ke posisi semula.



Gambar 4.27 Hasil Implementasi Aksi Step7

i) *Step8*

Pada saat *step5A*, *step5C* atau *step5D* aktif maka *timer 3* detik yang berfungsi untuk menunda waktu silinder *insert* mendorong benda selanjutnya akan aktif. Implementasi diagram *ladder step7* dapat dilihat pada Gambar 4.28 di bawah ini.



Gambar 4.28 Implementasi Diagram Ladder Step8

Hasil implementasi untuk *step8* dapat dilihat pada Gambar 4.29 yang ditunjukkan aktifnya kembali silinder *insert* untuk mendorong benda selanjutnya ke *conveyor1* setelah *timer 3* detik menghitung. *Step5A*, *step5C* atau *step5D* menjadi kontak untuk dapat mengaktifkan *timer* tersebut.

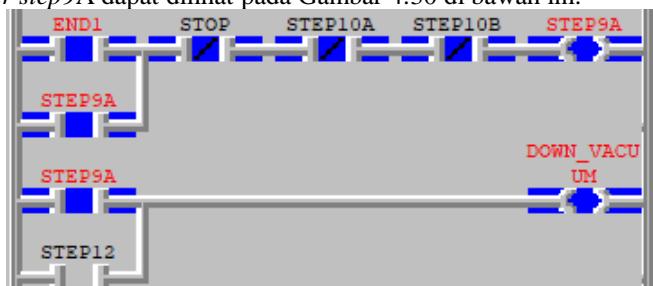


Gambar 4.29 Hasil Implementasi Step8

4.3.2 Implementasi pada *Pick and Place Module*

a) Step9A

Pada saat benda hitam maupun biru yang berhasil melintas dari *separation module* dan mengenai sensor (END1) membuat step9A aktif yang berakibat silinder vertical bergerak turun. Implementasi diagram ladder step9A dapat dilihat pada Gambar 4.30 di bawah ini.



Gambar 4.30 Implementasi Diagram Ladder Step9A

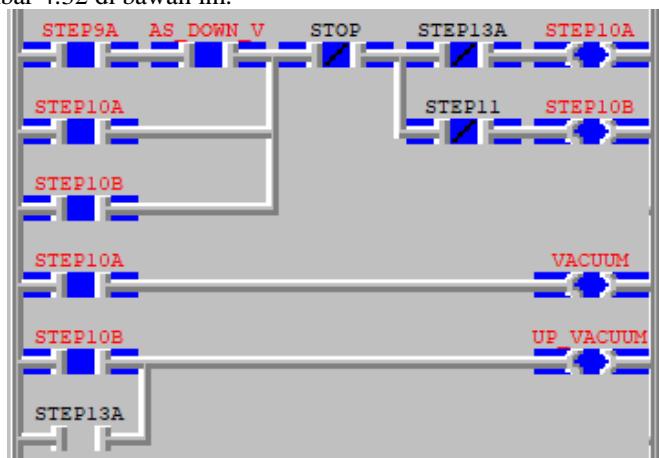
Hasil implementasi untuk step9A dapat dilihat pada Gambar 4.31 yang ditunjukkan dengan dokumentasi dari silinder vertical pada *pick & place module* turun mengarah ke benda kerja.



Gambar 4.31 Hasil Implementasi Aksi Step9A

b) *Step10A & Step10B*

Silinder vertikal yang bergerak turun akan membuat *auto switch* (AS_Down_V) aktif. Hal ini mengakibatkan *step10A* dan *step10B* aktif. *Step10A* berperan untuk mengaktifkan *vacuum* yang berfungsi menghisap benda untuk dialokasikan dari suatu tempat ke tempat yang lain. *Step10B* berperan untuk mengaktifkan silinder vertikal bergerak naik. Implementasi diagram *ladder* *step10A* dan *step10B* dapat dilihat pada Gambar 4.32 di bawah ini.



Gambar 4.32 Implementasi Diagram Ladder Step10A & Step10B

Hasil implementasi untuk *step10A* dan *step10B* dapat dilihat pada Gambar 4.33 yang ditunjukkan dengan dokumentasi dari silinder vertikal

pada *pick & place module* membawa benda kerja naik ke atas dengan *vacuum* dalam keadaan aktif.



Gambar 4.33 Hasil Implementasi Aksi Step10A & Step10B

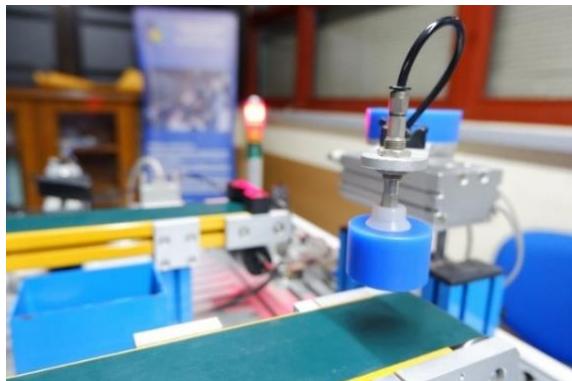
c) *Step11*

Pada saat benda terangkat ke atas oleh silinder vertikal hal ini membuat *auto switch* (AS_UP_V) aktif. Hal ini mengakibatkan *step11* yang berfungsi menggerakkan silinder rotasi bergerak berlawanan dengan arah jarum jam. Implementasi diagram *ladder step9A* dapat dilihat pada Gambar 4.34 di bawah ini.



Gambar 4.34 Implementasi Diagram Ladder Step11

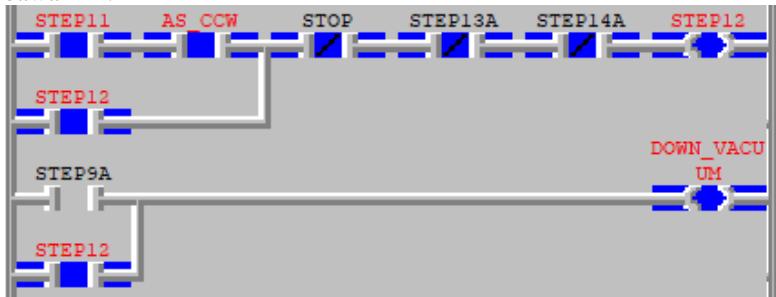
Hasil implementasi untuk *step11* dapat dilihat pada Gambar 4.35 yang ditunjukkan dengan dokumentasi dari silinder rotasi pada *pick & place module* membawa benda kerja bergerak berlawanan dengan arah jarum jam (CCW). Silinder ini membawa benda kerja mengarah ke atas *conveyor2* pada *stopper module*.



Gambar 4.35 Hasil Implementasi Aksi Step11

d) Step12

Pada saat benda bergerak berlawanan dengan arah jarum jam yang membuat *auto switch* (AS_CCW) aktif mengakibatkan step12 aktif yang berfungsi dalam menggerakkan silinder vertikal bergerak turun. Implementasi diagram *ladder* step12 dapat dilihat pada Gambar 4.36 di bawah ini.



Gambar 4.36 Implementasi Diagram Ladder Step12

Hasil implementasi untuk step12 dapat dilihat pada Gambar 4.37, ditunjukkan dengan dokumentasi dari silinder vertikal yang menurunkan benda kerja tepat di atas conveyor2.



Gambar 4.37 Hasil Implementasi Step12

e) *Step13A*

Pada saat benda berada di posisi bawah membuat *auto switch* (AS_Down_V) dan *auto switch* (AS_Vacuum) tersentuh untuk mengaktifkan *step13A*. *Step13A* berfungsi menggerakkan silinder vertikal bergerak ke atas bersamaan dengan *conveyor2* aktif yang akan dibahas di *step13B* pada pengujian sistem *stopper module*. Pada *step13A* aktif juga membuat *vacuum* non-aktif. Implementasi diagram *ladder* *step13A* dapat dilihat pada Gambar 4.38 di bawah ini.



Gambar 4.38 Implementasi Diagram Ladder Step13A

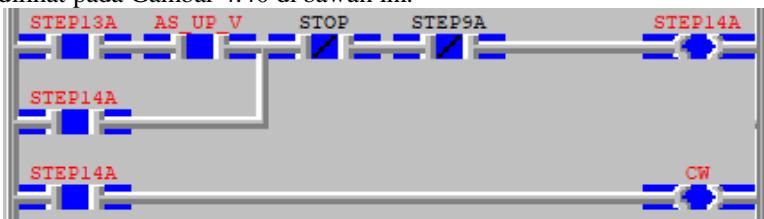
Hasil implementasi untuk *step13A* dapat dilihat pada Gambar 4.39, ditunjukkan dengan dokumentasi dari silinder vertikal yang naik bersamaan dengan *conveyor2* yang berjalan. Implementasi pada diagram *ladder* untuk berjalannya *conveyor2* ditunjukkan pada *step13B* di sub bab selanjutnya mengenai pengujian sistem pada *stopper module*.



Gambar 4.39 Hasil Implementasi Aksi Step13A

f) *Step14A*

Pada saat silinder vertikal berada di atas membuat *auto switch* (AS_UP_V) terdeteksi sehingga *step14A* aktif. *Step14A* aktif berfungsi menggerakkan silinder rotasi bergerak searah dengan arah jarum jam. Implementasi diagram *ladder* *step14A* melalui *software GMWIN* dapat dilihat pada Gambar 4.40 di bawah ini.



Gambar 4.40 Implementasi Diagram Ladder Step14A

Hasil implementasi untuk *step14A* dapat dilihat pada Gambar 4.41, ditunjukkan dengan dokumentasi dari silinder rotasi yang bergerak searah dengan arah jarum jam atau kembali ke posisi semula dari *pick and place module*.



Gambar 4.41 Hasil Implementasi Aksi Step14A

4.3.3 Implementasi pada Stopper Module

a) Step13B

Step13B merupakan memori yang digunakan untuk mengaktifkan *conveyor2* pada *stopper module*. Step13A dan step13B aktif secara bersamaan dikarenakan *auto switch* (AS_Down_V) dan *auto switch* (AS_Vacuum) aktif. Implementasi diagram *ladder* step13B melalui software GMWIN dapat dilihat pada Gambar 4.42 di bawah ini.

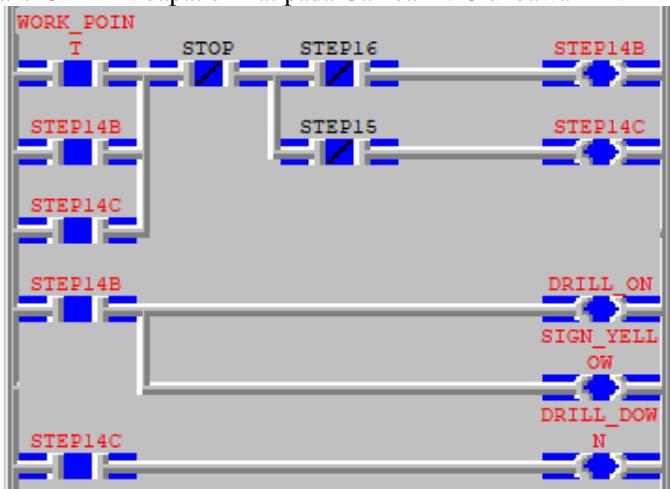


Gambar 4.42 Implementasi Diagram Ladder Step13B

b) Step14B & Step14C

Benda yang melintas pada *stopper module* menggunakan bantuan dari *conveyor2* akan terhenti pada sensor *work_point*. Pada saat benda terdeteksi sensor mengakibatkan step14B yang berperan mengaktifkan *drilling machine* serta indikator lampu warna kuning aktif dan step14C yang berperan mengaktifkan *drilling machine* bergerak ke bawah atau

turun aktif. Implementasi diagram *ladder step14B* dan *step14C* melalui software GMWIN dapat dilihat pada Gambar 4.43 di bawah ini.



Gambar 4.43 Implementasi Diagram Ladder Step14B & Step14C

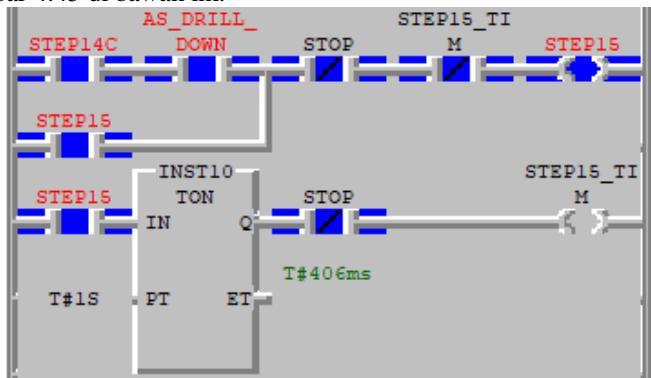
Hasil implementasi untuk *step14B* dan *step14C* ditunjukkan pada Gambar 4.44 di bawah ini. Hasil implementasi menunjukkan bahwa *drilling machine* aktif yang diindikasi dengan indikator lampu kuning menyala serta menurunnya aktuator *drilling machine* yang semula pada posisi atas.



Gambar 4.44 Hasil Implementasi Aksi Step14B & Step14C

c) *Step15*

Pada saat *auto switch* (*AS_Drill_Down*) terdeteksi mengakibatkan *step15* yang berfungsi mengaktifkan *timer 3* detik aktif. *Timer* digunakan untuk menahan proses *drilling* aktif selama 3 detik. Implementasi diagram *ladder* *step15* melalui *software GMWIN* dapat dilihat pada Gambar 4.45 di bawah ini.



Gambar 4.45 Implementasi Diagram Ladder Step15

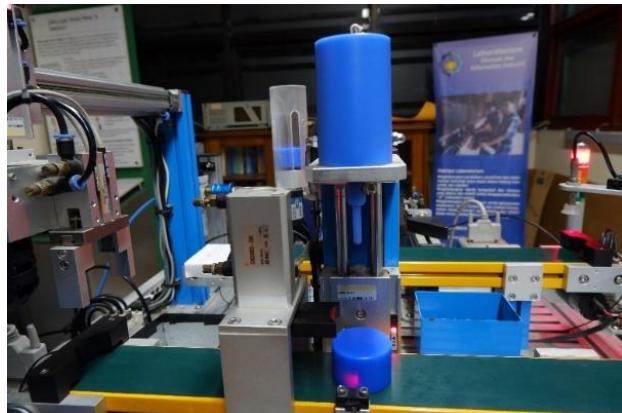
d) *Step16*

Seusai *timer* menghitung selama 1 detik maka membuat *step16* aktif. *Step16* berperan untuk membuat *stopper* dan *drilling machine* naik. Seusai *timer* menghitung selama 1 detik juga mengakibatkan *conveyor2* yang sempat terhenti menjadi beroperasi kembali. Implementasi diagram *ladder* *step16* dapat dilihat pada Gambar 4.46 di bawah ini.



Gambar 4.46 Implementasi Diagram Ladder Step16

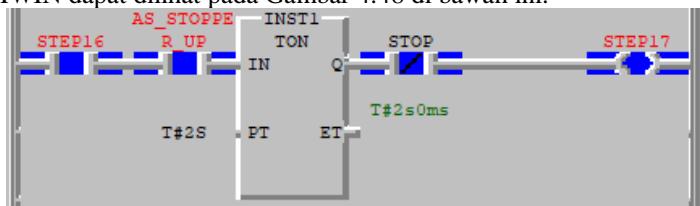
Hasil implementasi untuk aksi dari *step16* ditunjukkan pada Gambar 4.47 dengan dokumentasi *stopper* dan *drilling machine* dalam keadaan naik.



Gambar 4.47 Hasil Implementasi Aksi Step16

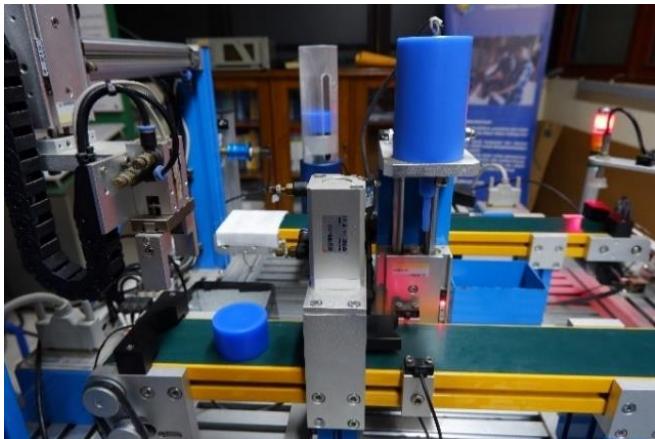
e) *Step17*

Pada saat *auto switch* (*AS_Stopper_Up*) terdeteksi juga mengakibatkan *timer 2* detik aktif untuk menahan posisi *stopper* di atas selama 2 detik. Implementasi diagram *ladder* *step17* melalui *software GMWIN* dapat dilihat pada Gambar 4.48 di bawah ini.



Gambar 4.48 Implementasi Diagram Ladder Step17

Hasil implementasi untuk *step17* ditunjukkan pada Gambar 4.49 dengan dokumentasi benda telah melintas dari *stopper* yang tertahan di posisi atas selama 2 detik. Setelah terhitung selama 2 detik *stopper* kembali ke posisi semula sedangkan *drilling machine* tetap di posisi atas selang selama benda berikutnya terdeteksi oleh sensor *work_point*.



Gambar 4.49 Hasil Implementasi Step17

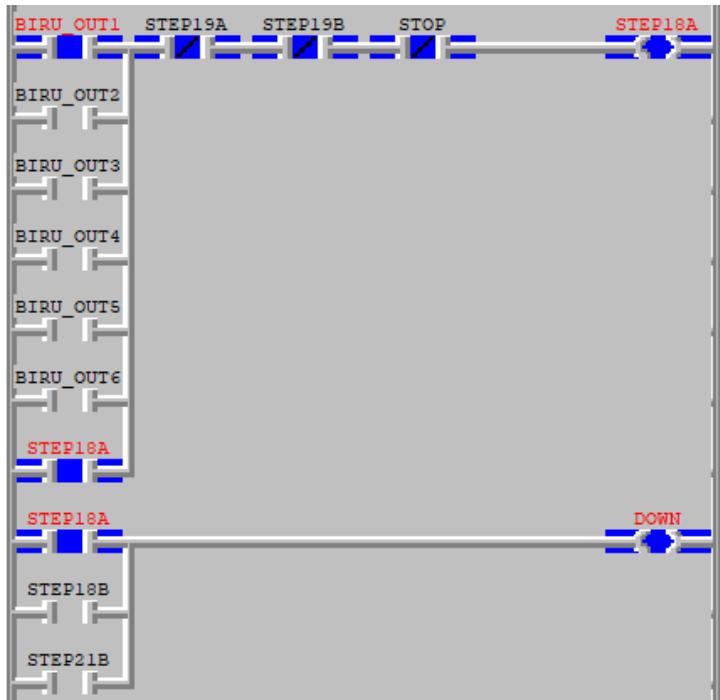
4.3.4 Implementasi pada *Line Movement Module*

Pada implementasi pada *line movement module* memiliki perbedaan aksi untuk benda dengan warna biru maupun bagi benda dengan warna hitam. Pada benda warna biru akan diletakkan di kotak yang berada di tengah jalur dari *line movement module*. *Step18A* hingga *step22A* akan menjelaskan mengenai hasil dari implementasi untuk aksi benda berwarna biru yang disertai dengan pengujian diagram *ladder* dan implementasi langsung dari aksi alat *factory automatic trainer*.

a) *Step18A*:

Ketika benda berwarna biru melintas pada *inductive proximity sensor*, *capacitive proximity sensor* dan sensor fotoelektrik di *separation module* memori yang mengindikasikan bahwa benda yang melintas merupakan benda berwarna biru dalam keadaan aktif. Pada *line movement module* tidak terdapat sensor seperti pada *separation module*, sehingga penggunaan memori untuk mengidentifikasi warna benda pada tahap ini sangat membantu.

Pada saat benda berwarna biru tersebut melintas dan terdeteksi oleh sensor (END2) maka akan mengaktifkan *step18A* yang berguna untuk menurunkan *vertical cylinder*. Implementasi diagram *ladder* *step18A* melalui *software GMWIN* dapat dilihat pada Gambar 4.50 di bawah ini.



Gambar 4.50 Implementasi Diagram Ladder Step18A

Hasil implementasi untuk *step18A* ditunjukkan pada Gambar 4.51 dengan dokumentasi silinder vertikal turun menghampiri benda kerja.



Gambar 4.51 Hasil Implementasi Aksi Step18A

b) *Step19A & Step19B*

Pada saat *vertical cylinder* di posisi bawah membuat *auto switch* (AS_Down) aktif yang mana mengakibatkan *finger cylinder* dalam posisi mencengkeram benda dan *vertical cylinder* menjadi naik mengangkat benda ke posisi atas. Implementasi diagram *ladder* *step19A* dan *step19B* melalui *software GMWIN* dapat dilihat pada Gambar 4.52 di bawah ini.



Gambar 4.52 Implementasi Diagram Ladder *Step19A & Step19B*

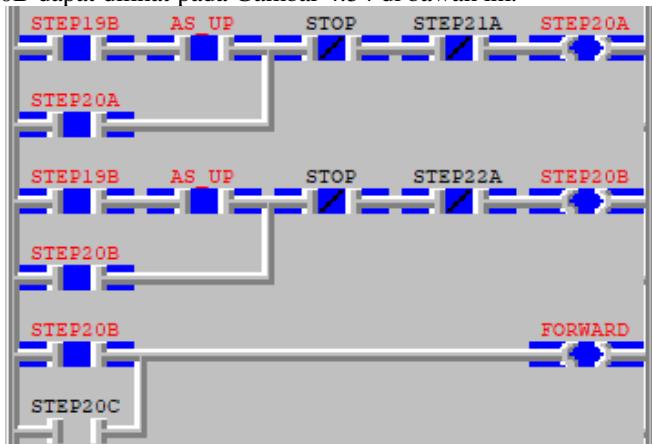
Hasil implementasi terhadap *plant* untuk aksi *step19A* dan *step19B* ditunjukkan pada Gambar 4.53 di bawah ini.



Gambar 4.53 Hasil Implementasi Aksi *Step19A & Step19B*

c) *Step20A & Step20B*

Pada saat benda di atas maka akan menyentuh *auto switch* (AS_UP) yang mengakibatkan *step20A* dan *step20B* aktif bersamaan. *Step20A* merupakan memori untuk mengaktifkan *timer* 0.5 detik. *Step20B* merupakan memori yang digunakan untuk menggerakkan *horizontal cylinder* bergerak *forward*. Implementasi diagram *ladder* *step20A* dan *step20B* dapat dilihat pada Gambar 4.54 di bawah ini.



Gambar 4.54 Implementasi Diagram *Ladder* *Step20B & Step20B*

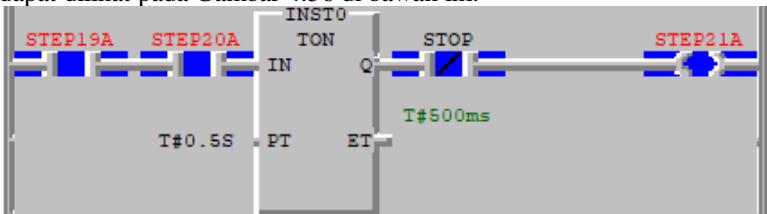
Hasil implementasi terhadap *plant* untuk aksi *step20A* dan *step20B* ditunjukkan pada Gambar 4.55 di bawah ini.



Gambar 4.55 Hasil Implementasi Aksi *Step20A & Step20B*

f) *Step21A*

Step21A digunakan untuk mengaktifkan *timer* 0.5 detik. *Timer* ini berguna untuk menahan *finger cylinder* yang tengah bergerak *forward* selama 0.5 detik. Ketika *timer* menghitung selama 0.5 detik *finger cylinder* yang berada pada posisi *grip* menjadi terbuka (*open*). Implementasi diagram *ladder* *step21A* melalui *software GMWIN* dapat dilihat pada Gambar 4.56 di bawah ini.



Gambar 4.56 Implementasi Diagram *Ladder Step21A*

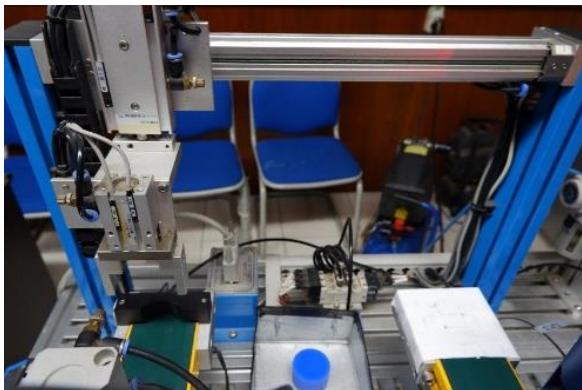
g) *Step22A*

Step22A yang berfungsi menggerakkan *horizontal cylinder* bergerak secara *backward*. *Backward* merupakan pergerakan kembali ke posisi awal dari *line movement module*. Implementasi diagram *ladder* *step22A* melalui *software GMWIN* dapat dilihat pada Gambar 4.57 di bawah ini



Gambar 4.57 Implementasi Diagram *Ladder Step22A*

Hasil implementasi terhadap *plant* untuk aksi *step20A* dan *step20B* ditunjukkan pada Gambar 4.58 berikut ini. Hasil implementasi ditunjukkan dengan pergerakan dari *horizontal cylinder* yang bergerak dari arah *separation module* mengarah ke *stopper module*. *Stopper module* dari gambar ditinjau dari adanya sensor *fiber optic* (END2). Tampak dari Gambar 4.58 berikut ini bahwa benda biru telah berada di dalam kotak.



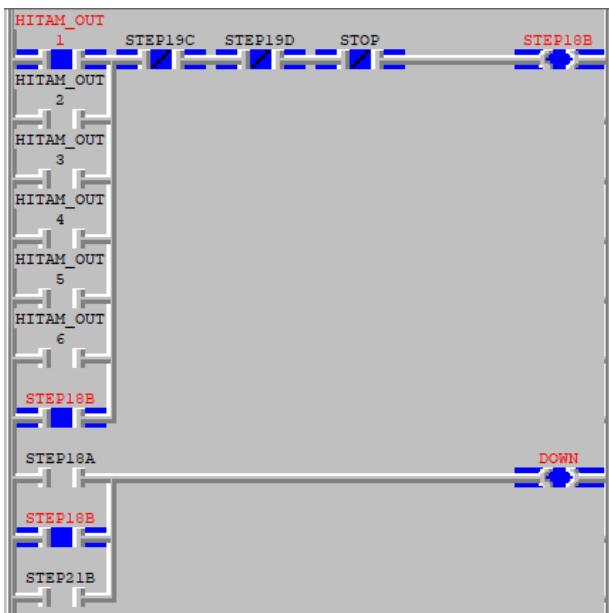
Gambar 4.58 Hasil Implementasi Aksi Step22A

Pada implementasi pada *line movement module*, benda berwarna hitam akan diletakkan di pangkal dari *conveyor1* yang terletak pada *separation module*. Hasil implementasi dari *line movement module* untuk benda berwarna hitam dijelaskan pada *step18B* sampai *step23*. Hasil implementasi ditunjukkan dengan diagram *ladder* beserta dokumentasi dari alat *factory automatic trainer*.

a) *Step18B*:

Ketika benda berwarna hitam melintas pada *inductive proximity sensor*, *capacitive proximity sensor* dan sensor fotoelektrik di *separation module* memori yang mengindikasikan bahwa benda yang melintas merupakan benda berwarna hitam dalam keadaan aktif. Pada *line movement module* tidak terdapat sensor seperti pada *separation module*, sehingga penggunaan memori untuk mengidentifikasi warna benda pada tahap ini sangat membantu.

Pada saat benda berwarna hitam tersebut melintas dan terdeteksi oleh sensor (END2) maka akan mengaktifkan *step18B* yang berguna untuk menurunkan *vertical cylinder*. Implementasi diagram *ladder* *step18B* melalui *software GMWIN* dapat dilihat pada Gambar 4.59 berikut ini.



Gambar 4.59 Implementasi Diagram Ladder Step18B

Hasil implementasi terhadap *plant* untuk aksi *step18B* ditunjukkan pada Gambar 4.60 berikut ini. Hasil implementasi disertai dengan dokumentasi dari aksi *step18B* yang berfungsi menurunkan silinder vertikal dengan *gripper* menghampiri benda berwarna hitam yang berada tepat di bawah silinder. Dalam proses ini *gripper* belum mencengkeram benda kerja.



Gambar 4.60 Hasil Implementasi Aksi Step18B

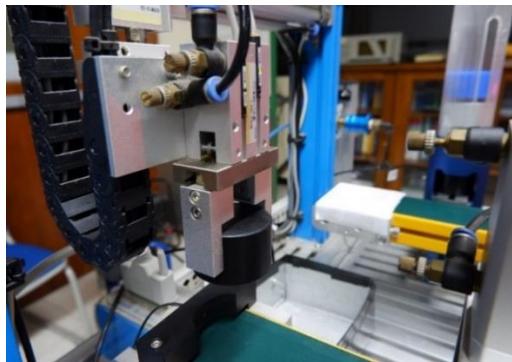
b) Step19C & Step19D

Vertical cylinder yang berada di posisi bawah mengakibatkan *auto switch* (AS_Down) menyala, dimana hal ini memicu aktifnya aksi *finger cylinder* dalam kondisi mencengkeram benda berwarna hitam (*grip*) serta vertical cylinder yang awalnya turun menjadi naik membawa benda berada di posisi atas. Implementasi diagram *ladder step19C* dan *step19D* melalui *software GMWIN* dapat dilihat pada Gambar 4.61 berikut ini.



Gambar 4.61 Implementasi Diagram Ladder Step19C & Step19D

Hasil implementasi terhadap *plant* untuk aksi *step19C* dan *step19D* ditunjukkan pada Gambar 4.62 berikut ini. Hasil implementasi disertai dengan dokumentasi dari silinder vertikal yang membawa benda berwarna hitam naik ke atas disertai dengan *finger grip* dalam keadaan mencengkeram benda.



Gambar 4.62 Hasil Implementasi Aksi Step19C & Step19D

c) *Step20C*

Vertical cylinder yang berada di atas memicu aktifnya *auto switch* (AS_UP) yang mengakibatkan *horizontal cylinder* bergerak secara *forward* menuju tepat di atas pangkal *conveyor1* yang berada di *separation module*. Implementasi diagram *ladder* step20C melalui *software GMWIN* dapat dilihat pada Gambar 4.63 di bawah ini.



Gambar 4.63 Implementasi Diagram *Ladder* Step20C

Hasil implementasi terhadap *plant* untuk aksi *step20C* ditunjukkan pada Gambar 4.64 berikut ini. Hasil implementasi disertai dengan dokumentasi dari silinder horizontal yang membawa benda berwarna hitam bergerak mengarah tepat di atas pangkal *conveyor1* pada *separation module*.



Gambar 4.64 Hasil Implementasi Aksi Step20C

d) *Step21B*

Benda yang dialokasikan ke atas pangkal *conveyor1* menyebabkan *auto switch* (*AS_Foward*) terdeteksi sehingga mengakibatkan *vertical cylinder* membawa benda turun. Implementasi diagram *ladder* *step21B* melalui *software GMWIN* dapat dilihat pada Gambar 4.65 di bawah ini.



Gambar 4.65 Implementasi Diagram Ladder *Step21B*

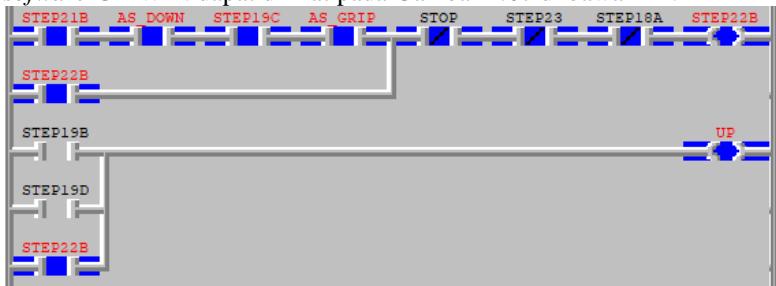
Hasil implementasi terhadap *plant* untuk aksi *step21B* ditunjukkan pada Gambar 4.66 berikut ini. Hasil implementasi disertai dengan dokumentasi dari silinder vertikal yang menurunkan benda berwarna hitam tepat di atas pangkal *conveyor1*.



Gambar 4.66 Hasil Implementasi Aksi Step21B

e) *Step22B*

Ketika *vertical cylinder* turun membuat *auto switch* (AS_Down) aktif. *Auto switch* (AS_Grip) juga aktif untuk memberikan perintah kepada *vertical cylinder* untuk naik. Pada saat *vertical cylinder* naik, *finger cylinder* yang awalnya di posisi mencengkeram benda menjadi membuka *gripper*. Implementasi diagram *ladder* step22B melalui *software GMWIN* dapat dilihat pada Gambar 4.67 di bawah ini.



Gambar 4.67 Implementasi Diagram Ladder Step22B

Hasil implementasi terhadap *plant* untuk aksi *step21B* ditunjukkan pada Gambar 4.68 berikut ini. Hasil implementasi disertai dengan dokumentasi dari silinder vertikal yang kembali naik ke atas dengan *finger cylinder* dalam kondisi membuka *gripper*.



Gambar 4.68 Hasil Implementasi Aksi Step22B

f) *Step23*

Pada saat *vertical cylinder* naik memicu aktifnya *auto switch* (AS_UP) untuk memberikan perintah kepada *horizontal cylinder* untuk bergerak secara *backward*. Pergerakan ini berpindah dari atas *conveyor1* di *separation module* menuju ke atas *conveyor2* di *stopper module* atau kembali kepada posisi semula dari *line movement module*. Implementasi diagram *ladder step23* melalui *software GMWIN* dapat dilihat pada Gambar 4.69 di bawah ini.



Gambar 4.69 Implementasi Diagram Ladder Step23

Hasil implementasi terhadap *plant* untuk aksi *step20A* dan *step20B* ditunjukkan pada Gambar 4.64 berikut ini. Hasil implementasi ditunjukkan dengan pergerakan dari *horizontal cylinder* yang bergerak dari arah *separation module* mengarah ke *stopper module*. *Stopper module* dari gambar ditinjau dari adanya sensor *fiber optic* (END2).

Tampak dari Gambar 4.70 berikut ini bahwa benda hitam telah berada di pangkal *conveyor*.



Gambar 4.70 Hasil Implementasi Aksi Step23

4.4 Hasil Pengujian

Pada *Factory Automatic Trainer* akan dilakukan proses penyortiran untuk benda berwarna hitam, biru dan *silver*. Pengujian dilakukan dengan cara menguji sebanyak 10 kali percobaan untuk masing – masing benda hitam, biru ataupun *silver*. Pengujian bertujuan untuk mengetahui keberhasilan dari sistem. Keberhasilan sistem ini dilihat dari waku proses antara pengujian benda satu dan benda selanjutnya apakah memiliki waktu proses yang sama untuk setiap warna. Hasil pengujian waktu proses untuk objek tunggal dengan benda berwarna hitam, biru dan *silver* dapat dilihat pada Tabel 4.6 berikut ini.

Tabel 4.6 Hasil Pengujian Waktu Proses Objek Tunggal

Uji Benda Ke-	Waktu Proses (detik)		
	Benda Hitam	Benda Biru	Benda Silver
1	22.44	21.45	05.89
2	21.87	20.97	05.76
3	22.05	21.52	06.08
4	22.29	21.20	05.81
5	22.43	21.73	05.76
6	21.99	21.60	05.81

Uji Benda Ke-	Waktu Proses (detik)		
	Benda Hitam	Benda Biru	Benda Silver
7	22.46	21.00	05.89
8	21.74	20.92	05.89
9	21.93	20.95	05.96
10	21.99	21.46	05.68
Rata – rata	22.12	21.28	05.85

Dari pengujian waktu proses untuk benda tunggal berwarna hitam memiliki waktu rata – rata sekitar 22.12 detik, benda biru memiliki waktu rata – rata sekitar 21.28 detik sedangkan benda *silver* memiliki waktu rata – rata sekitar 5.85 detik.

Pengujian multi proses benda berurutan dilakukan dengan melakukan pengujian sebanyak 10 kali percobaan. Pengujian dilakukan dengan memasukkan urutan benda Biru-Biru-Biru-Silver-Silver-Silver-Hitam-Hitam-Hitam dan mencatat waktu proses hingga benda terakhir selesai. Hasil pengujian multi proses untuk benda berurutan dapat dilihat pada Tabel 4.7 di bawah ini.

Tabel 4.7 Hasil Pengujian Multi Proses Benda Berurutan

Pengujian Ke-	Waktu Proses (menit)	Keterangan
1	01.08.73	Biru-Biru-Biru-Silver-Silver-Silver-Hitam-Hitam-Hitam
2	01.08.73	
3	01.08.85	
4	01.08.87	
5	01.08.72	
6	01.08.86	
7	01.09.14	
8	01.08.89	
9	01.09.09	
10	01.08.94	
Rata - Rata	01.08.85	

Dari Tabel 4.7 untuk benda berurutan memiliki waktu proses rata – rata sekitar 01.08.85 menit untuk menyelesaikan keseluruhan proses. Pengujian pertama,kedua dan selanjutnya menunjukkan bahwa waktu proses yang diperoleh tidak terlalu jauh dan masih dapat dikatakan dalam rentang waktu yang wajar. Selisih waktu dapat terjadi dikarenakan keterlambatan dalam menekan *stopwatch* ataupun tombol *start*.

Pengujian multi proses benda acak dilakukan dengan melakukan pengujian sebanyak 10 kali percobaan. Pengujian dilakukan dengan memasukkan benda secara acak (*random*). Hasil pengujian multi proses untuk benda acak dapat dilihat pada Tabel 4.8 berikut ini.

Tabel 4.8 Hasil Pengujian Multi Proses Benda Acak

Pengujian Ke-	Waktu Proses (menit)	Keterangan
	Benda Acak	
1	01.06.84	Biru-Biru-Hitam-Silver-Hitam-Silver-Silver-Hitam-Biru
2	01.05.77	Silver-Hitam-Biru-Hitam-Biru-Silver-Hitam-Silver-Biru
3	01.05.82	Silver-Biru-Silver-Hitam-Hitam-Biru-Hitam-Silver-Biru
4	01.07.16	Biru-Hitam-Silver-Silver-Biru-Hitam-Silver-Hitam-Biru
5	01.08.80	Hitam-Biru-Silver-Biru-Silver-Silver-Biru-Hitam-Hitam
6	01.08.24	Biru-Hitam-Silver-Silver-Biru-Hitam-Silver-Biru-Hitam
7	01.08.77	Silver-Biru-Biru-Hitam-Silver-Silver-Biru-Hitam-Hitam
8	01.03.67	Biru-Hitam-Silver-Silver-Biru-Hitam-Biru-Hitam-Silver
9	01.03.33	Hitam-Biru-Biru-Silver-Hitam-Silver-Hitam-Biru-Silver
10	01.03.45	Hitam-Biru-Silver-Biru-Hitam-Silver-Biru-Hitam-Silver

Dari hasil pengujian multi proses benda acak dapat disimpulkan bahwa benda dengan warna *silver* di posisi terakhir memiliki waktu proses tercepat yaitu dengan rata – rata sekitar 01.03.48 menit, diikuti dengan warna biru di posisi terakhir memiliki waktu rata – rata sekitar 01.06.39 menit dan benda dengan warna hitam di posisi terakhir memiliki waktu proses paling lama yaitu dengan rata – rata sekitar 01.08.60 menit

BAB 5

PENUTUP

Pada bab ini akan dijelaskan mengenai kesimpulan dan saran yang didapatkan dari hasil penggerjaan tugas akhir Desain Diagram *Ladder* untuk Banyak Objek pada *Factory Automatic Trainer* menggunakan Metode *Grafcet*.

5.1 Kesimpulan

Berdasarkan perancangan diagram *ladder* dengan menggunakan metode *grafcet* dapat disimpulkan beberapa hal, diantaranya adalah sebagai berikut ini.

1. Dari pembuatan diagram *ladder* dengan menggunakan metode *grafcet* terdapat beberapa *coil* yaitu terdiri dari 85 digunakan untuk memori (*step*) serta 23 digunakan untuk *output*. Sedangkan untuk kontak dari PLC yang digunakan terdapat 28 sebagai *input*.
2. Jumlah *timer* dan *counter* yang digunakan dalam pembuatan diagram *ladder* dengan menggunakan metode *grafcet* adalah sejumlah 6 *timer* serta 12 *counter*.
3. Data program *ladder* yang dihasilkan untuk keseluruhan sistem adalah sebesar 30 KB dan dengan jumlah *rung* sebanyak 72 *rung*.
4. Dari pengujian waktu proses untuk benda tunggal berwarna hitam memiliki waktu rata – rata sekitar 22.12 detik, benda biru memiliki waktu rata – rata sekitar 21.28 detik sedangkan benda *silver* memiliki waktu rata – rata sekitar 5.85 detik.
5. Hasil pengujian multi proses benda berurutan memiliki waktu proses rata – rata sekitar 01.08.85 menit.
6. Hasil pengujian multi proses benda acak benda warna *silver* di posisi terakhir memiliki waktu proses tercepat yaitu dengan rata – rata sekitar 01.03.48 menit, diikuti warna biru memiliki waktu rata – rata sekitar 01.06.39 menit dan benda warna hitam di posisi terakhir memiliki waktu proses paling lama yaitu sekitar 01.08.60 menit.

Dampak dari hasil diagram *ladder* yang dirancang dengan menggunakan metode *grafcet* apabila diterapkan ke dalam industri sortir benda membuat siklus produksi menjadi lebih cepat apabila dikerjakan secara multi proses. Dengan *single* proses untuk memproses 9 benda menghasilkan waktu 2 – 3 menit sedangkan melalui multi proses untuk 9 benda hanya membutuhkan waktu sekitar 1 menit.

5.2 Saran

1. Pengoperasian sistem pada *factory automatic trainer* alangkah baiknya apabila kedepannya terdapat penambahan *Human Machine Interface* (HMI). Penambahan HMI bertujuan agar *plant* dapat dikendalikan melalui tampilan HMI, sehingga proses berjalannya sistem dapat dipantau.
2. Urutan peletakkan benda mempengaruhi waktu dari siklus produksi diharapkan kedepannya terdapat penelitian mengenai pencarian waktu yang optimal untuk meningkatkan efisiensi waktu.

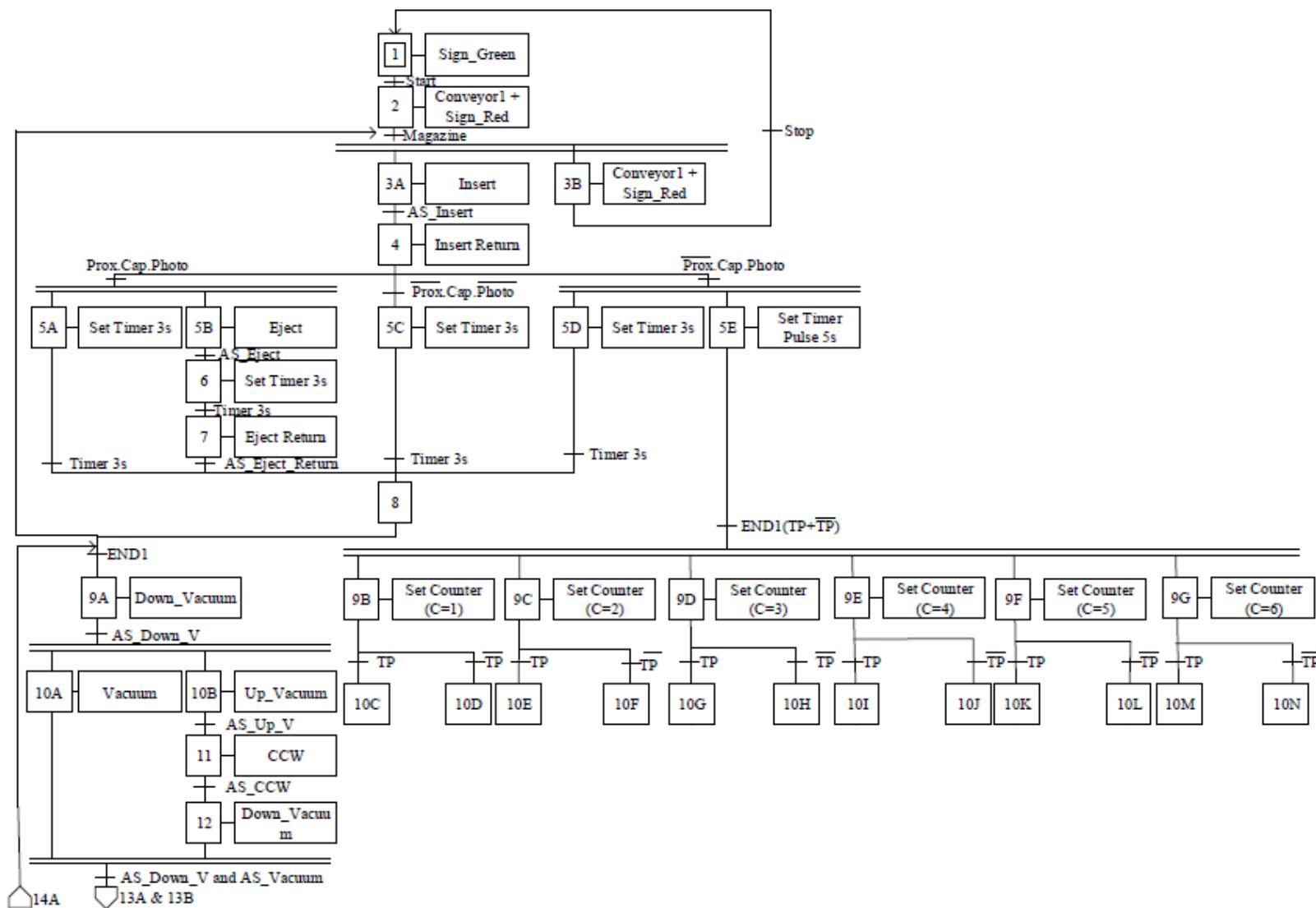
DAFTAR PUSTAKA

- [1] Witanto, Aji P., Konstruksi Diagram Ladder dengan Metode Huffman untuk Factory Automatic Trainer, Surabaya: Institut Teknologi Sepuluh Nopember, 2018.
- [2] Nugraha, Raka A., Konstruksi Diagram Ladder dengan Bantuan Grafset untuk Factory Automatic Trainer, Surabaya: Institut Teknologi Sepuluh Nopember, 2018.
- [3] Chungpa, CPE-AT8030N Factory Automatic Trainer User's Manual, Gyeonggi, Korea: Chungpa.
- [4] Autonics, "Fiber Optic Sensors Technical Description (English)," [Online]. Available: <https://www.autonics.com/service/data/view/6/216759>. [Accessed 20 4 2019].
- [5] "Sensor Induktif-Proximity," technoinlive, 12 1 2017. [Online]. Available: <https://technoinlive.wordpress.com/2017/01/12/sensor-induktif-proximity/>. [Accessed 20 4 2019].
- [6] D. H. Hansen, Programmable Logic Controllers, Hoboken, New Jersey: Wiley.
- [7] Autonics, "Photoelectric Sensors Technical Description (English)," [Online]. Available: <https://www.autonics.com/service/data/view/6/216754>. [Accessed 20 4 2019].
- [8] Alat Uji, "Photoelectric Sensor," 13 5 2019. [Online]. Available: <https://www.alatujji.com/index.php?/article/detail/694/photoelectric-sensor>. [Accessed 14 5 2019].
- [9] T. D. Suyadhi, "Sensor Optik," 6 12 2014. [Online]. Available: <http://www.robotics-university.com/2014/11/sensor-optik.html>. [Accessed 13 4 2019].
- [10] Sudaryono, Pneumatik & Hidrolik, Jakarta: Katalog Dalam Terbitan (KDT), 2013.
- [11] "Timer dan Counter," [Online]. Available: <te.ugm.ac.id/~enas/schneider/Timer%20&%20Counter.doc>. [Accessed 22 4 2019].

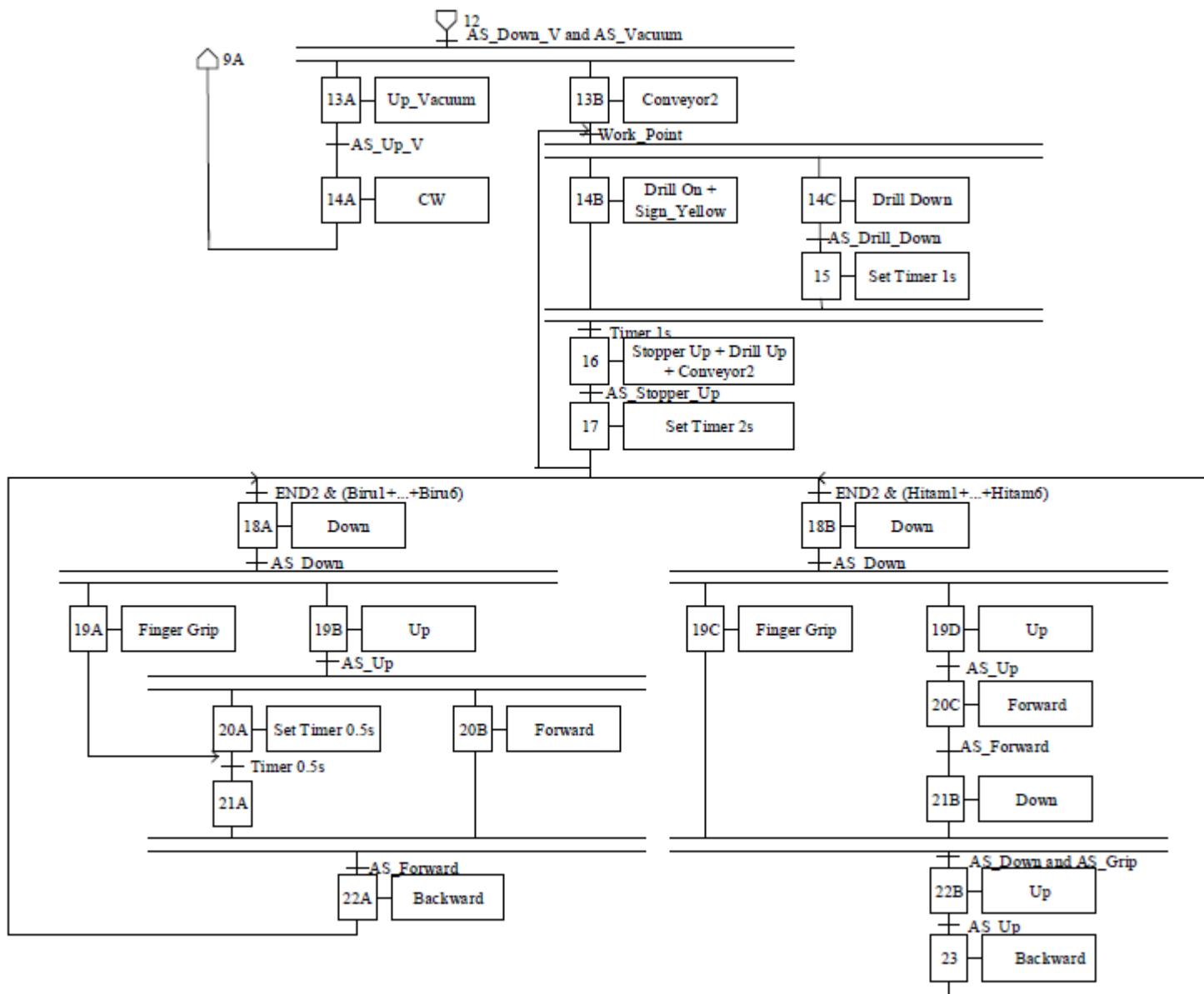
- [12] R. S. Juairiah, "Counter dengan GMWIN," Universitas Sriwijaya, Palembang, 2018.
- [13] Baracos, Paul, Grafcet Step by Step, Famic Automation, 1992.
- [14] Baracos, Paul, "GRAFCET and Ladder Diagram," in *Industrial and Electronics Control*.
- [15] D and D. D. Adhyatma, Efektivitas Penggunaan Festo Fluidsim sebagai Media Pembelajaran, Yogyakarta: Universitas Negeri Yogyakarta, 2013.
- [16] Persen, Industrial Automation : Circuit Design and Components, Israel: John Wiley & Sons, 1989.

LAMPIRAN

A. Keseluruhan Grafset

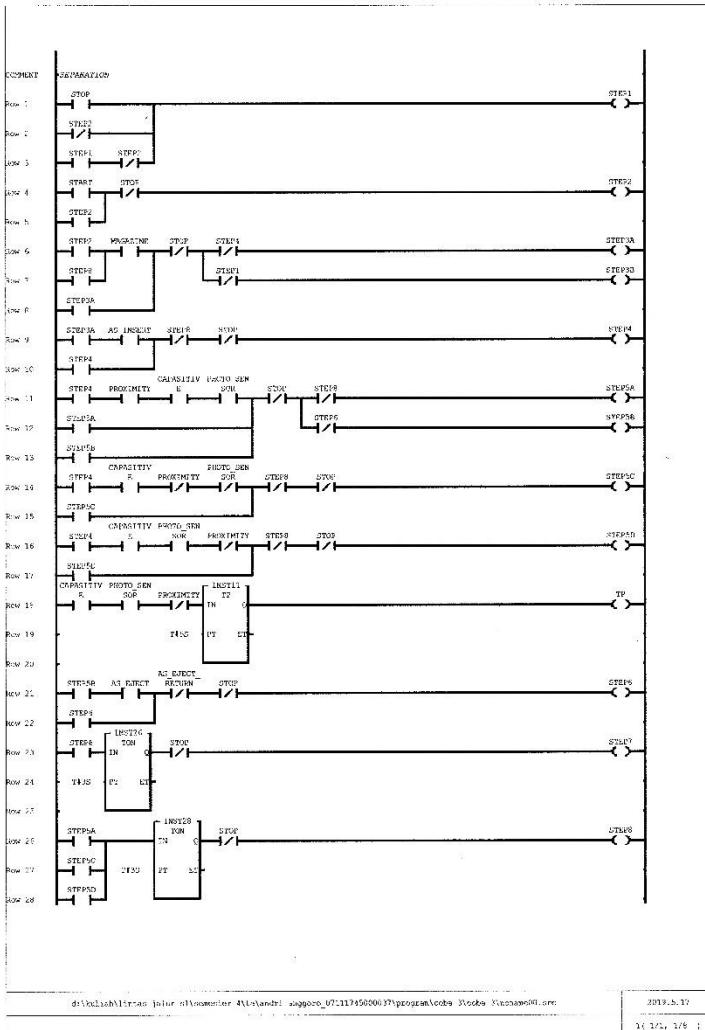


-----Halaman ini sengaja dikosongkan---



-----Halaman ini sengaja dikosongkan---

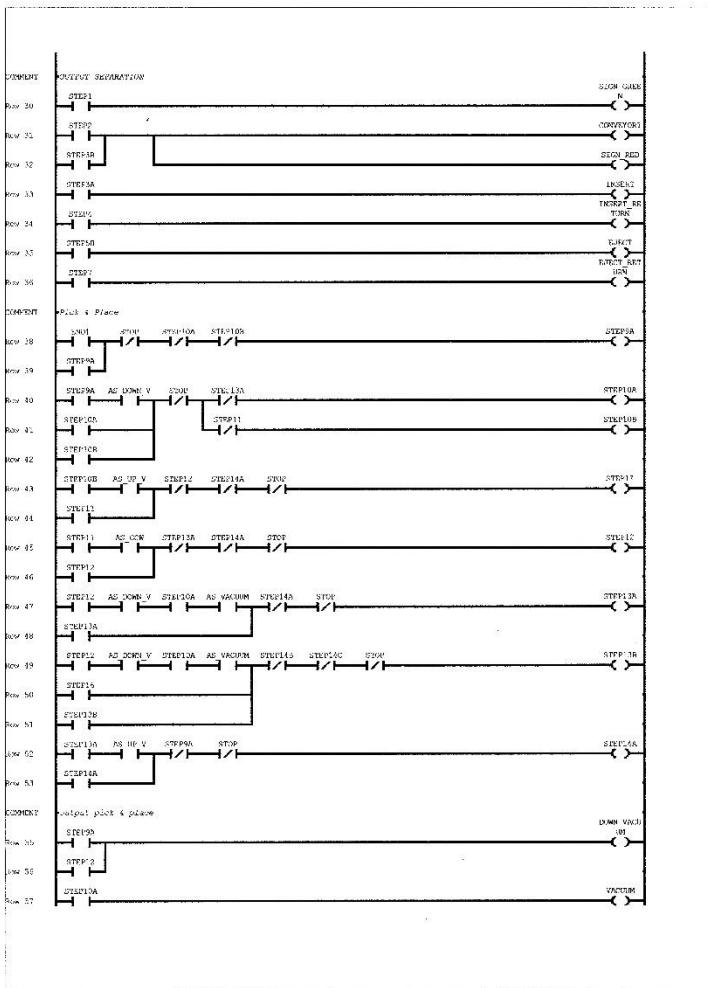
B. Keseluruhan Diagram Ladder

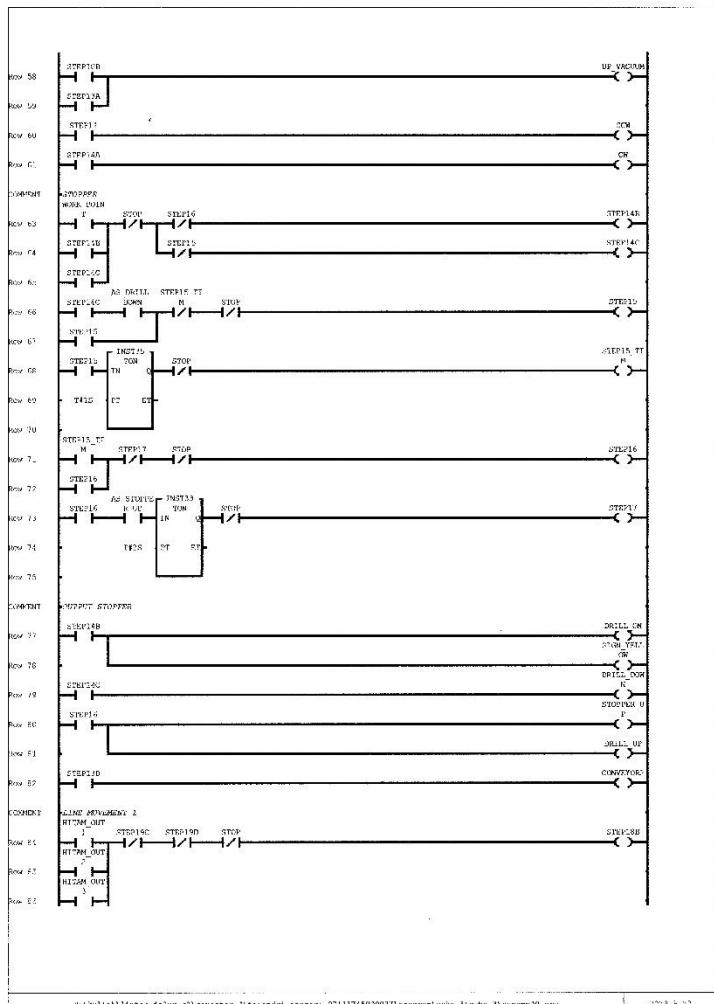


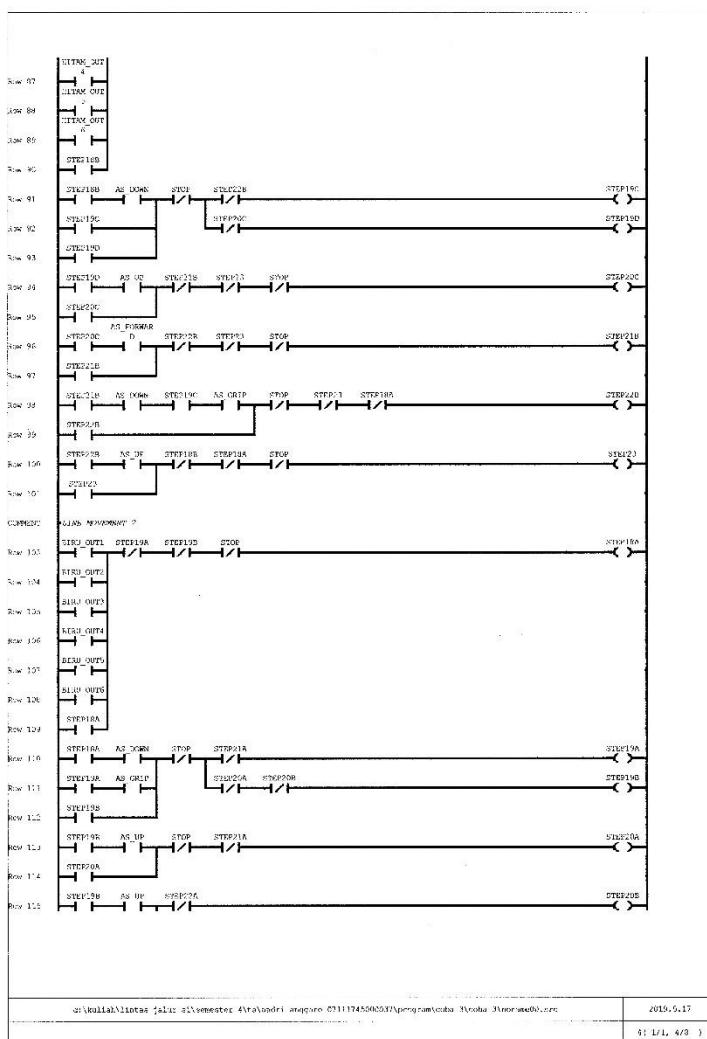
Availability: www.mechanik-studien.de Support: 0711/95620037 | programcode: Nodee_Kapazität01.htm

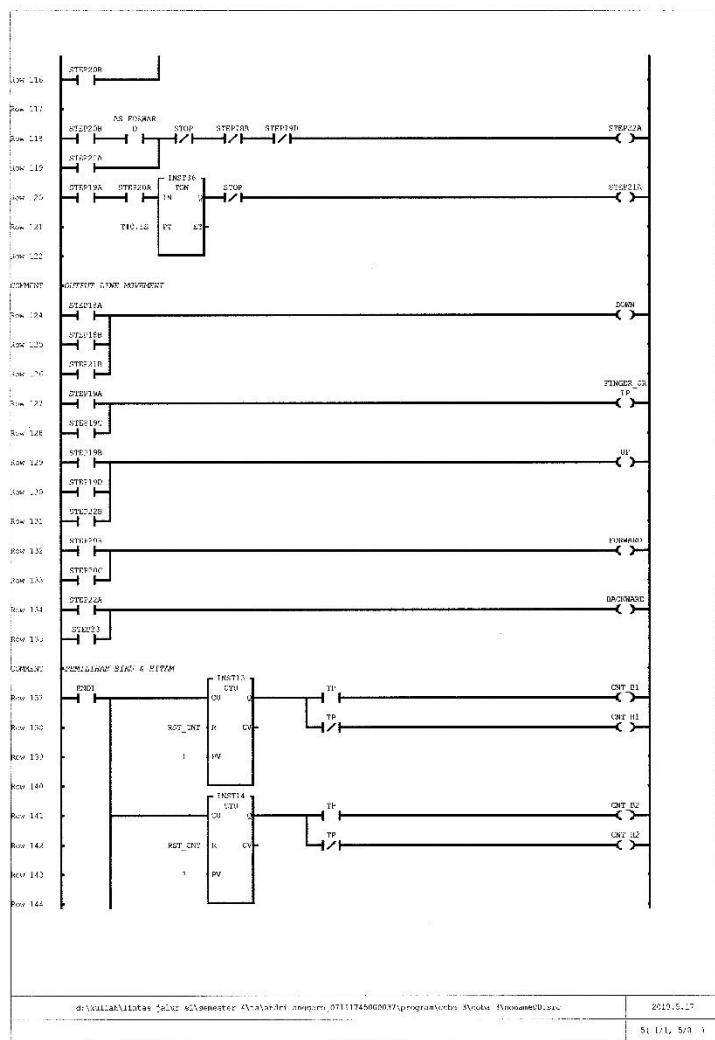
2019-5-17

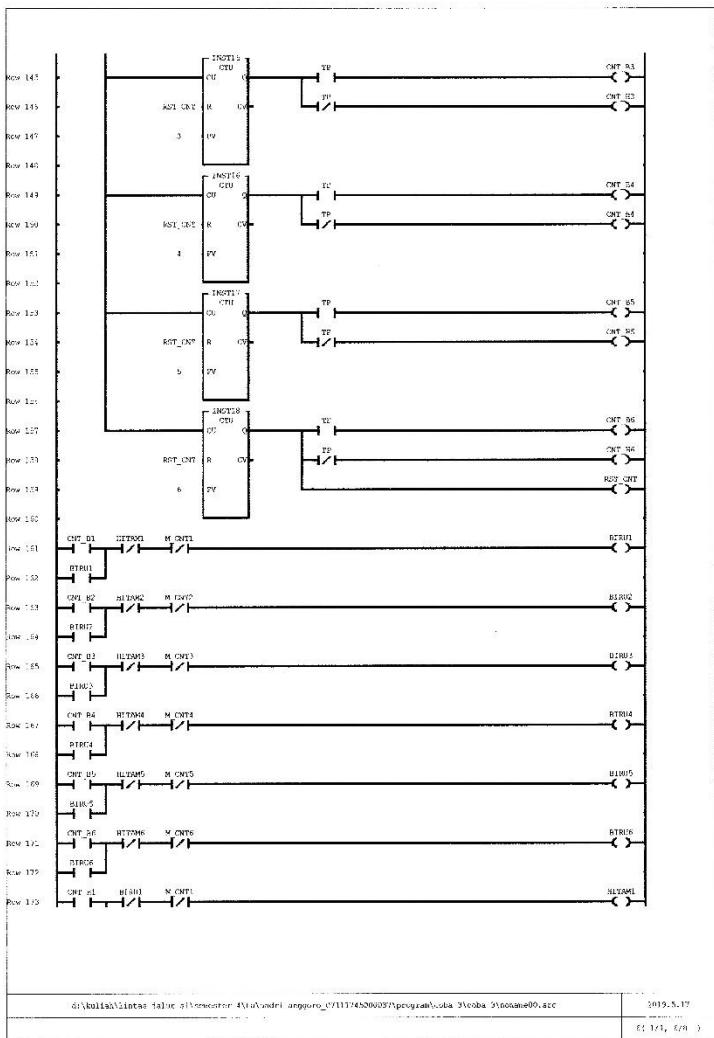
3/2/2019 3/8 1

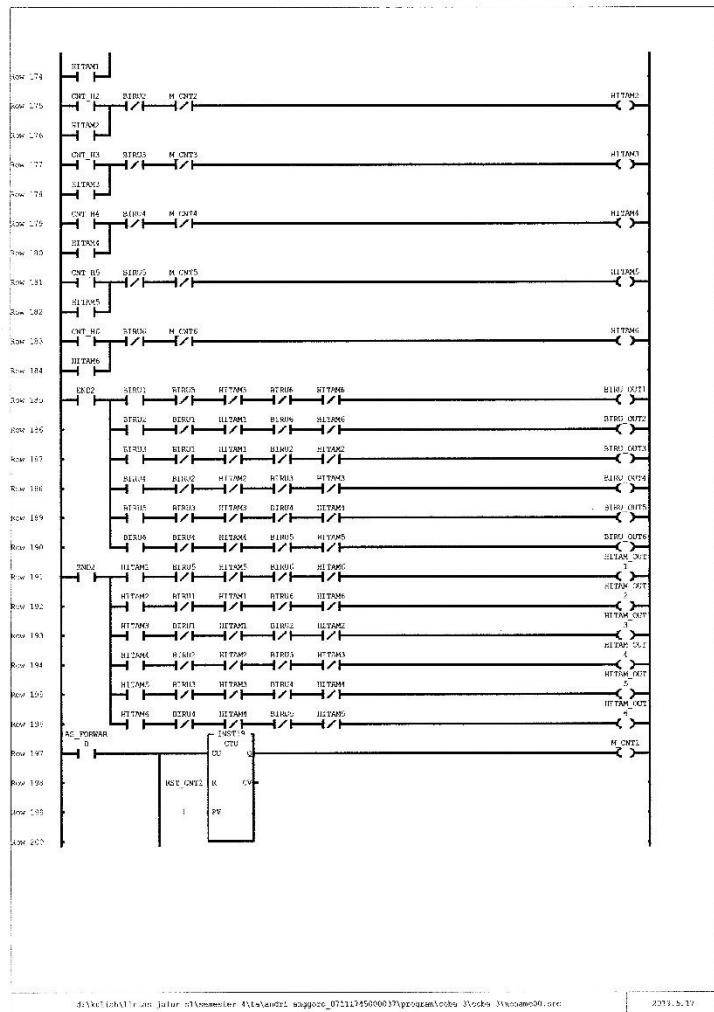


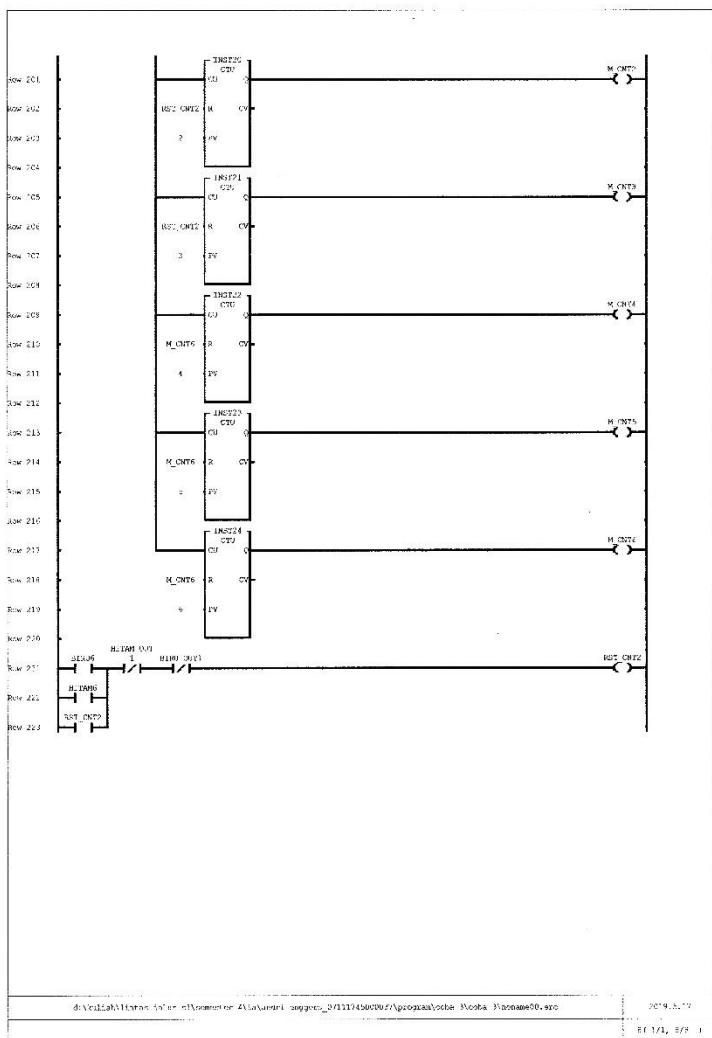






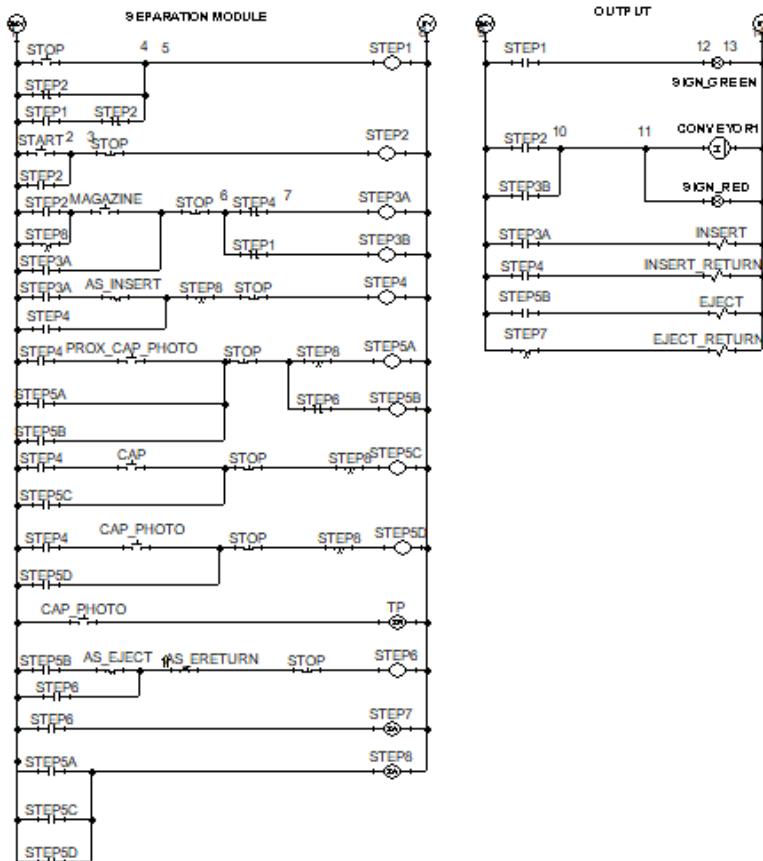


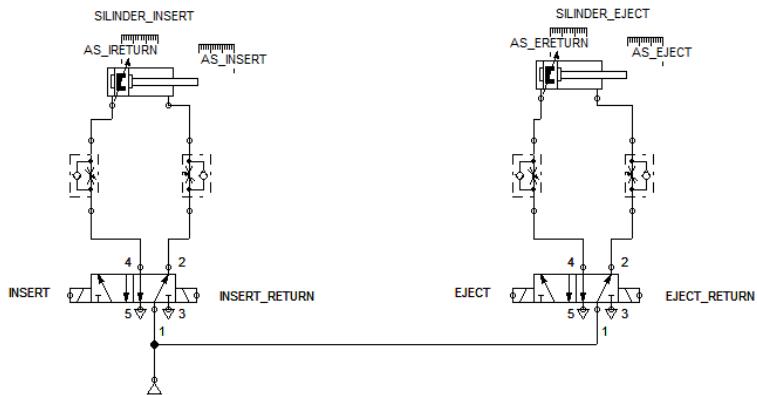




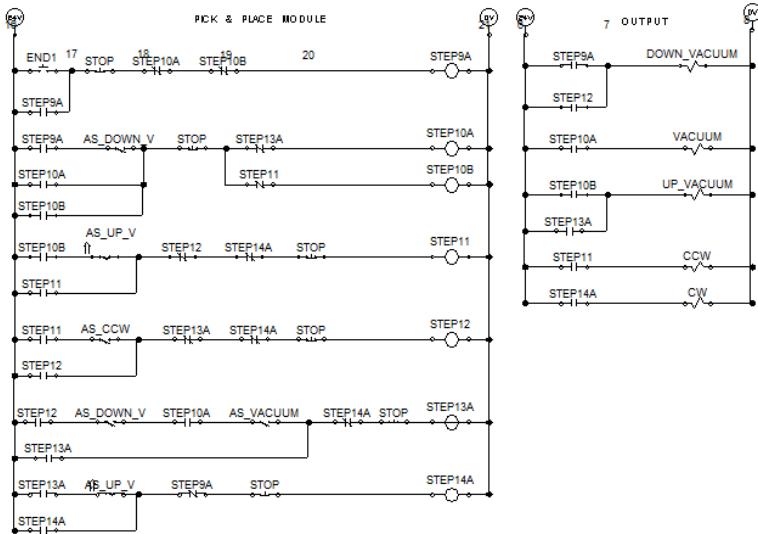
C. Hasil Rangkaian Elektrik & Pneumatik pada Fluidsim

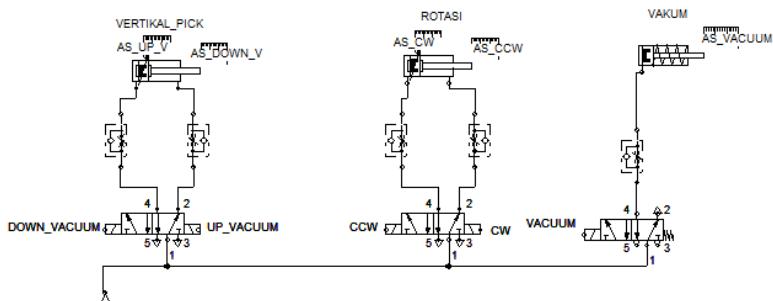
C.1 Separation Module



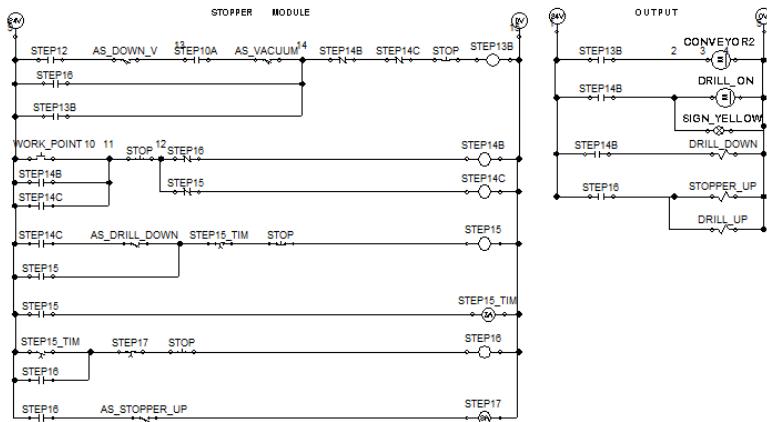


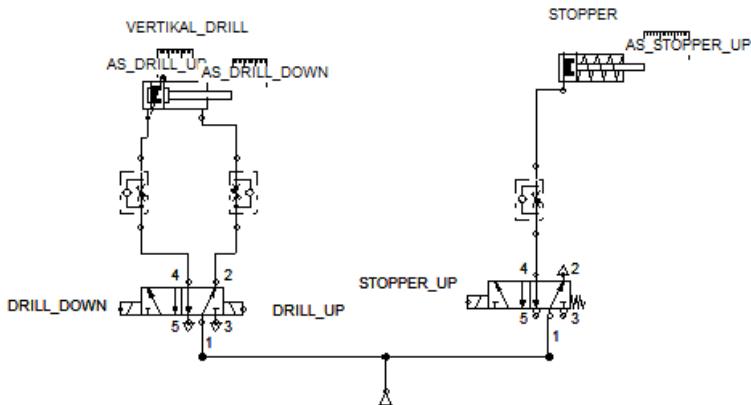
C.2 Pick and Place Module



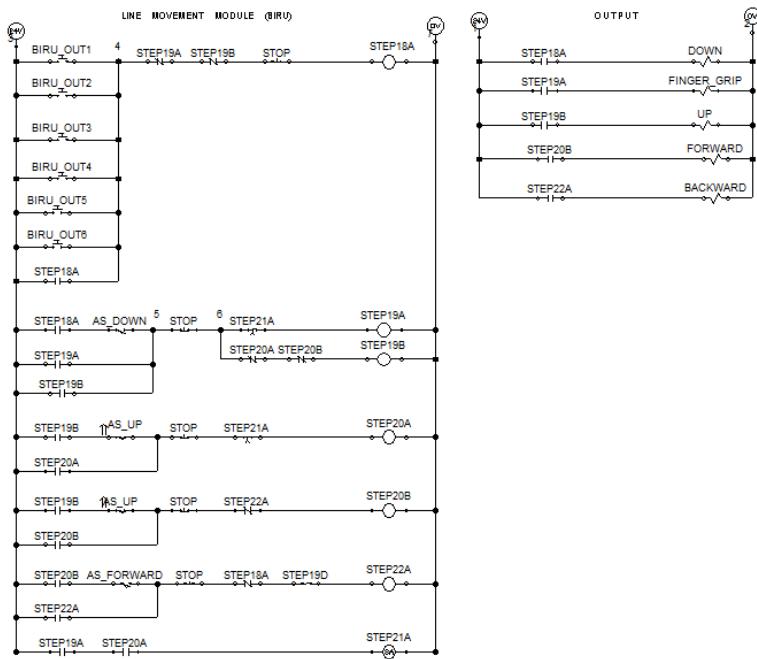


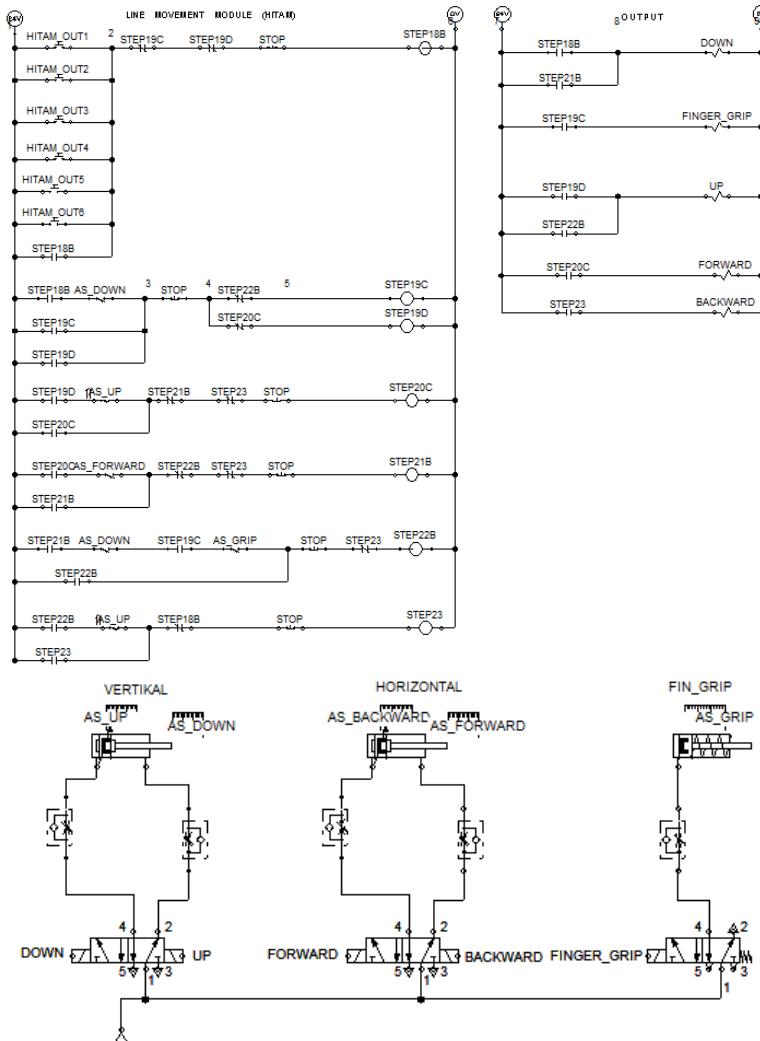
C.3 Stopper Module





C.4 Line Movement Module





-----Halaman ini sengaja dikosongkan---

DAFTAR RIWAYAT HIDUP



Nama : Innes Deviana
TTL : Mojokerto, 25 Desember 1995
Agama : Islam
Alamat : Jalan Halmahera No.9
Mojokerto
Telp/HP : 088231951000
Email : innesdeviana25@gmail.com

RIWAYAT PENDIDIKAN

1. 2002-2008 : SDN Kranggan I Mojokerto
2. 2008-2011 : SMP Negeri 4 Mojokerto
3. 2011-2014 : SMA Negeri 2 Mojokerto
4. 2014-2017 : D3 Teknik Elektro Otomasi, Program Studi Komputer Kontrol, Fakultas Vokasi, Institut Teknologi Sepuluh Nopember
5. 2017-2019 : S1 Teknik Elektro, Bidang Studi Teknik Sistem Pengaturan, Fakultas Teknologi Elektro, Institut Teknologi Sepuluh Nopember

PENGALAMAN KERJA

1. Kerja Praktek di PT Terminal Teluk Lamong
2. Kerja Praktek di PT Terminal Nilam Utara

PENGALAMAN ORGANISASI

1. Staff Magang Kewirausahaan HIMAD3TEKTRO 2014-2015
2. Staff Annisa' LDJ Salman Al-Farisi HIMAD3TEKTRO 2015-2016
3. Staff Kewirausahaan HIMAD3TEKTRO 2015-2016
4. Kepala Departemen Kewirausahaan HIMAD3TEKTRO 2016-2017

-----Halaman ini sengaja dikosongkan-----