



TUGAS AKHIR - EE 184801

**DESAIN DIAGRAM *LADDER* UNTUK BANYAK OBJEK
PADA *FACTORY AUTOMATIC TRAINER*
MENGUNAKAN METODE *CASCADE***

Izzuddin Al Qossam
NRP 07111440000151

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

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



FINAL PROJECT - EE 184801

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

Izzuddin Al Qossam
NRP 07111440000151

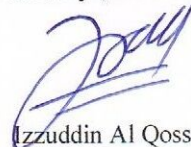
Supervisor
Dr. Ir. Mochammad Rameli
Eka Iskandar, ST., MT.

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

PERNYATAAN KEASLIAN TUGAS AKHIR

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

Surabaya, 3 Juli 2019



Izzuddin Al Qossam
NRP 0711144000151

[Halaman ini sengaja dikosongkan]

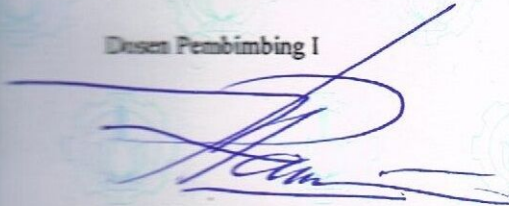
**DESAIN DIAGRAM LADDER UNTUK BANYAK OBJEK PADA
FACTORY AUTOMATIC TRAINER MENGGUNAKAN METODE
CASCADE**

TUGAS AKHIR

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

Menyetujui :

Dosen Pembimbing I



Dr. Ir. Mochammad Rameli
NIP. 19541227 1981031002

Dosen Pembimbing II



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



[Halaman ini sengaja dikosongkan]

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

Nama : Izzuddin Al Qossam
NRP : 07111440000151
Pembimbing : 1. Dr. Ir. Mochammad Rameli
2. Eka Iskandar, ST., MT.

ABSTRAK

Terdapat beberapa bahasa pemrograman dalam *Programmable Logic Controller* (PLC) namun hanya satu bahasa yang umum digunakan yaitu *Ladder Diagram* (LAD). *Ladder diagram* memodifikasi rangkaian listrik program yang diinginkan menjadi kumpulan *relay* berupa kontak dan koil. Kontak menggambarkan masukan sedangkan koil mempresentasikan keluaran *relay* pada program *ladder diagram*. Perancangan diagram *ladder* tidak dilakukan secara konservatif tetapi dibutuhkan sebuah metode agar didapatkan hasil yang efektif dan efisien serta tidak menyimpan banyak memori sehingga berdampak pada biaya. Pada topik tugas akhir ini digunakan metode *Cascade* yang dapat mengurangi jumlah *relay* tidak hanya secara fisik saja tetapi melakukan pendekatan melalui jumlah proses sehingga memudahkan untuk mencari solusi dalam masalah yang lebih kompleks.

Factory Automatic Trainer merupakan miniatur plant industri yang secara umum memiliki sistem yang mirip pada kondisi riil. Objek penelitian yang dipilih adalah banyak objek yang telah tersedia pada *Factory Automatic Trainer* dimana permasalahan terjadi saat *addressing* dilakukan berkelanjutan dan beragam. Tahapan proses *addressing* terbagi menjadi 4 yaitu *counting* benda kerja, *pairing address* benda kerja, *locking address* benda kerja dan pemutusan aliran *address* benda kerja. Dari keseluruhan diagram *ladder* sistem menggunakan metode *Cascade* diperoleh 26 input serta 113 *rung* terdiri dari 63 *relay*, 9 timer, 18 counter dan 3 output dengan besar data program sebesar 20126 Bytes.

Kata kunci : *PLC, Diagram Ladder, Cascade, Factory Automatic Trainer, Banyak Objek.*

[Halaman ini sengaja dikosongkan]

Ladder Diagram Design for Multiple Objects on Factory Automatic Trainer using Cascade Method

Name : Izzuddin Al Qossam
Registration Number : 07111440000151
Supervisor : 1. Dr. Ir. Mochammad Rameli
2. Eka Iskandar, ST., MT.

ABSTRACT

There are several programming languages in the Programmable Logic Controller (PLC), but only one language is commonly used, that is Ladder Diagram (LAD). The Ladder diagram modifies the desired program electrical circuit into a collection of relays in the form of contacts and coils. Contact describes input while the coil presents relay output on the ladder diagram program. The design of the ladder diagram is not done conservatively, but a method is needed in order to obtain effective and efficient results and not to save a lot of memory so that it has an impact on costs. In the topic of this final project, the Cascade method is used to reduce the number of relays not only physically but to approach through a number of processes so it will be easy to find solutions of more complex problems.

Factory Automatic Trainer is a miniature industrial plant that generally has a similar system in real conditions. The chosen object of the research is multiple objects, provided by Factory Automatic Trainer, where problems occur when addressing is carried out continuously and multiply. The stages of the addressing process are divided into 4, counting workpieces, pairing address of workpieces, locking address of workpieces and termination of workpiece address flow. The overall ladder diagram of the system using the Cascade method produces 26 inputs also 113 rungs consisted of 63 relays, 9 timers, 18 counters and 3 outputs with a program data size of 20126 Bytes.

Keywords : *PLC, Ladder Diagram, Cascade, Factory Automatic Trainer, Multiple Objects.*

[Halaman ini sengaja dikosongkan]

KATA PENGANTAR

Segala puji dan syukur selalu terpanjatkan kehadirat Allah SWT karena berkat karunia dan nikmat yang telah diberikan-Nya penulis dapat merampungkan buku tugas akhir dengan judul “**Desain Diagram Ladder untuk Banyak Objek pada Factory Automatic Trainer Menggunakan Metode Cascade**”. Pada bagian pengantar ini penulis ingin mengucapkan banyak terima kasih atas dukung dan bantuan dalam proses panjang pembuatan tugas akhir kepada :

1. Orang tua, keluarga, sanak saudara dan kerabat yang senantiasa memberikan doa, motivasi dan dukungan.
2. Bapak Mochamad Rameli selaku dosen pembimbing yang selalu sabar membimbing serta memberikan saran dan motivasi selama mengerjakan tugas akhir.
3. Bapak Eka Iskandar selaku dosen pembimbing yang telah berbagi banyak ilmu, bimbingan dan pengalaman sehingga tugas akhir ini dapat berjalan lancar.
4. Seluruh dosen, tenaga pendidik dan karyawan di Teknik Elektro Institut Teknologi Sepuluh Nopember.
5. Seluruh teman-teman mahasiswa reguler angkatan 2014, mahasiswa Teknik Sistem Pengaturan, penghuni laboratorium B405 dan B105.
6. Semua pihak yang telah banyak membantu dalam proses tugas akhir ini yang tidak dapat disebutkan satu per satu.

Tidak dapat dipungkiri bahwa buku tugas akhir ini masih jauh dari kata sempurna. Dengan demikian penulis berharap adanya kritik dan saran sebagai sarana perbaikan. Semoga kehadiran buku ini dapat membantu semua pihak yang tertarik mempelajarinya lebih jauh.

Surabaya, 3 Juli 2019

Izzuddin Al Qossam
NRP 07111440000151

[Halaman ini sengaja dikosongkan]

DAFTAR ISI

HALAMAN JUDUL.....	i
PERNYATAAN KEASLIAN TUGAS AKHIR.....	v
LEMBAR 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 Permasalahan	2
1.3 Batasan Masalah	2
1.4 Tujuan	2
1.5 Metodologi.....	2
1.6 Sistematika Laporan.....	4
1.7 Relevansi.....	4
BAB 2 TEORI PENUNJANG	5
2.1 <i>Factory Automatic Trainer</i>	5
2.1.1 Modul <i>Separation</i>	5
2.1.2 Modul <i>Pick & Place</i>	6
2.1.3 Modul <i>Stopper</i>	7
2.1.4 Modul <i>Line Movement</i>	8
2.1.5 <i>Control Unit</i>	8
2.2 Komponen Penunjang.....	9
2.2.1 Sensor.....	9
2.2.1.1 Sensor <i>Proximity Capacitive</i> [4].....	10
2.2.1.2 Sensor <i>Proximity Inductive</i> [6].....	11
2.2.1.3 Sensor <i>Photo</i> [6].....	13
2.2.1.4 Sensor <i>Fiber Optic</i> [6]	13
2.2.2 Aktuator	14
2.2.2.1 Konveyor.....	14
2.2.2.2 <i>Pneumatic Cylinder</i>	15
2.2.2.3 <i>Solenoid Valve</i>	17
2.2.2.4 <i>Finger</i>	17

2.2.2.5	<i>Vacuum</i>	18
2.2.2.6	<i>3-Colored Sign Tower</i>	19
2.2.3	<i>Programmable Logic Controller</i>	19
2.2.4	Benda Kerja.....	20
2.2.5	Kompresor.....	21
2.2.6	<i>Air Filter</i>	21
2.3	<i>Metode Cascade</i>	22
2.3.1	<i>Single Path Sequencing Systems without Sustained Outputs</i> 23	
2.3.2	<i>Single Path Sequencing Systems with Sustained Outputs</i>	23
2.4	<i>Fluidsim</i>	24
BAB 3	PERANCANGAN SISTEM	25
3.1	Perancangan Diagram <i>Ladder</i> dengan Metode <i>Cascade</i>	25
3.2	Input/Output <i>Factory Automatic Trainer</i>	26
3.3	Deskripsi Sistem	30
3.3.1	<i>Separation</i>	31
3.3.2	<i>Picking</i>	31
3.3.3	<i>Drilling</i>	32
3.3.4	<i>Movement</i>	33
3.4	Pemberian <i>Letter Output</i>	34
3.5	Pembagian Urutan Operasi Sistem Berdasarkan Grup.....	36
3.6	Penentuan Urutan Operasi Sistem.....	38
3.7	Pembuatan <i>Switching Function</i>	38
3.7.1	Relay Y1	39
3.7.2	Relay Y2	39
3.7.3	Relay Y3	40
3.7.4	Relay Y4	41
3.7.5	Relay Y5	41
3.7.6	Relay Y6	42
3.7.7	Relay Y7	42
3.7.8	Relay Y8	43
3.7.9	Relay Y9	43
3.7.10	Relay Y10	44
3.7.11	Relay Y11	44
3.7.12	Relay Y12	45
3.7.13	Relay Y13	45
3.7.14	Relay Y14	46

3.7.15	Relay Y15	46
3.7.16	Relay Y16	47
3.7.17	Relay Y17	47
3.7.18	Relay Y18	47
3.7.19	Relay Y19	48
3.8	Persamaan <i>Boolean Relay</i>	48
3.8.1	Relay Y1	49
3.8.2	Relay Y2	50
3.8.3	Relay Y3	51
3.8.4	Relay Y4	51
3.8.5	Relay Y5	52
3.8.6	Relay Y6	52
3.8.7	Relay Y7	53
3.8.8	Relay Y8	53
3.8.9	Relay Y9	53
3.8.10	Relay Y10	54
3.8.11	Relay Y11	54
3.8.12	Relay Y12	55
3.8.13	Relay Y13	55
3.8.14	Relay Y14	56
3.8.15	Relay Y15	56
3.8.16	Relay Y16	57
3.8.17	Relay Y17	57
3.8.18	Relay Y18	58
3.8.19	Relay Y19	58
3.9	<i>Addressing Benda Kerja</i>	59
3.9.1	<i>Counting Benda Kerja</i>	60
3.9.2	<i>Pairing Address Benda Kerja</i>	61
3.9.3	<i>Locking Address Benda Kerja</i>	63
3.9.4	Pemutusan Aliran <i>Address Benda Kerja</i>	64
BAB 4	IMPLEMENTASI DAN ANALISIS	67
4.1	Pembuatan Diagram <i>Ladder</i>	67
4.1.1	Alamat Input/Output	67
4.1.2	Diagram <i>Ladder</i>	69
4.1.2.1	Diagram <i>Ladder</i> Grup Relay	70
4.1.2.2	Diagram <i>Ladder</i> Relay Output	82

4.1.2.3	Diagram <i>Ladder Addressing</i> Benda Kerja	91
4.2	Validasi	112
4.3	Pengujian Sistem.....	119
BAB 5	PENUTUP	121
5.1	Kesimpulan	121
5.2	Saran	121
DAFTAR PUSTAKA		123
LAMPIRAN.....		125
RIWAYAT HIDUP.....		163

DAFTAR GAMBAR

Gambar 2.1 <i>Factory Automatic Trainer</i>	5
Gambar 2.2 Modul <i>separation</i>	6
Gambar 2.3 Modul <i>pick & place</i>	7
Gambar 2.4 Modul <i>stopper</i>	7
Gambar 2.5 Modul <i>line movement</i>	8
Gambar 2.6 Modul <i>control unit</i>	9
Gambar 2.7 Sensor <i>proximity capacitive</i>	10
Gambar 2.8 Cara kerja sensor <i>proximity capacitive</i>	11
Gambar 2.9 Sensor <i>proximity inductive</i>	11
Gambar 2.10 Cara kerja sensor <i>proximity inductive</i>	12
Gambar 2.11 Osilasi frekuensi pada objek metal	12
Gambar 2.12 Osilasi frekuensi pada objek <i>nonferrousmetal</i>	12
Gambar 2.13 Sensor <i>photo</i>	13
Gambar 2.14 Cara kerja sensor <i>photo</i>	13
Gambar 2.15 Sensor <i>fiber optic</i>	14
Gambar 2.16 Cara kerja sensor <i>fiber optic</i>	14
Gambar 2.17 Komponen penyusun konveyor	15
Gambar 2.18 Motor penggerak konveyor.....	15
Gambar 2.19 Cara kerja <i>single acting pneumatic cylinder</i>	16
Gambar 2.20 Cara kerja <i>double acting pneumatic cylinder</i>	16
Gambar 2.21 <i>Insert cylinder</i> dan <i>eject cylinder</i>	16
Gambar 2.22 <i>Solenoid valves double acting</i> dan <i>single acting</i>	17
Gambar 2.23 <i>Finger</i>	18
Gambar 2.24 <i>Vacuum</i>	19
Gambar 2.25 <i>3-Colored sign tower</i> dalam posisi horizontal	19
Gambar 2.26 Tiga jenis benda kerja	20
Gambar 2.27 Kompresor	21
Gambar 2.28 <i>Air filter</i> dalam posisi horizontal [15].....	21
Gambar 2.29 Rangkaian <i>RS flip flop</i> diimplementasikan dalam relay. 23	
Gambar 2.30 Contoh <i>ladder diagram Single path sequencing systems without sustained outputs</i>	24
Gambar 2.31 Contoh <i>ladder diagram Single path sequencing systems with sustained outputs</i>	24
Gambar 2.32 Logo <i>Fluidsim</i>	24

Gambar 3.1 Langkah-langkah perancangan <i>ladder diagram</i> metode <i>cascade</i>	25
Gambar 3.2 Diagram waktu perancangan aktuator modul <i>separation</i>	32
Gambar 3.3 Diagram waktu perancangan aktuator modul <i>pick & place</i>	32
Gambar 3.4 Diagram waktu perancangan aktuator modul <i>stopper</i>	33
Gambar 3.5 Diagram waktu perancangan aktuator modul <i>line movement</i> untuk benda kerja plastik hitam	34
Gambar 3.6 Diagram waktu perancangan aktuator modul <i>line movement</i> untuk benda kerja plastik biru	34
Gambar 3.7 Urutan <i>letter</i> output operasi sistem	37
Gambar 3.8 Pembagian grup urutan <i>letter</i> output operasi sistem	38
Gambar 3.9 Blok diagram urutan grup	38
Gambar 3.10 Sinyal output <i>finger horizontal cylinder</i> tidak mencapai titik maksimal.....	59
Gambar 4.1 Relay Y1	70
Gambar 4.2 Relay <i>Starting, Stopping</i> dan output <i>3-Colored Sign Tower</i>	71
Gambar 4.3 Relay Y2	71
Gambar 4.4 Relay Y3	72
Gambar 4.5 Relay deteksi sensor, timer <i>Tim_Add</i> dan timer <i>Tim_Cap</i>	73
Gambar 4.6 Relay Y4	74
Gambar 4.7 Relay Y5	74
Gambar 4.8 Relay Y6	75
Gambar 4.9 Relay Y7	75
Gambar 4.10 Relay Y8	76
Gambar 4.11 Relay Y9	76
Gambar 4.12 Relay Y10	77
Gambar 4.13 Relay Y11	77
Gambar 4.14 Relay Y12	78
Gambar 4.15 Relay R_Y12	78
Gambar 4.16 Relay Y13	79
Gambar 4.17 Relay Y14	79
Gambar 4.18 Relay Y15	80
Gambar 4.19 Relay Y16	80
Gambar 4.20 Relay Y17	81

Gambar 4.21	Relay Y18	81
Gambar 4.22	Relay Y19	82
Gambar 4.23	Output <i>Conveyor1</i>	82
Gambar 4.24	Output <i>Insert</i>	83
Gambar 4.25	Relay <i>Add</i>	83
Gambar 4.26	Output <i>Insert_R</i>	83
Gambar 4.27	Output <i>Eject</i> dan timer <i>Tim_Ej</i>	84
Gambar 4.28	Output <i>Eject_R</i>	84
Gambar 4.29	Output <i>Down</i>	85
Gambar 4.30	Output <i>Vacuum</i>	85
Gambar 4.31	Output <i>Up</i>	85
Gambar 4.32	Output <i>RotaryCCW</i>	86
Gambar 4.33	Relay <i>VacuumOff</i>	86
Gambar 4.34	Output <i>RotaryCW</i>	86
Gambar 4.35	Output <i>Conveyor2</i>	86
Gambar 4.36	Output <i>DrillDown</i>	87
Gambar 4.37	Output <i>DrillOn</i> dan timer <i>Tim_DOff</i>	87
Gambar 4.38	Output <i>DrillUp</i>	88
Gambar 4.39	Output <i>StopperUp</i> dan timer <i>Tim_STPR</i>	88
Gambar 4.40	Output <i>Finger_D</i>	88
Gambar 4.41	Output <i>FingerGrip</i>	89
Gambar 4.42	Output <i>FingerUp</i>	89
Gambar 4.43	Output <i>Forward</i> dan timer <i>Tim_GOp</i>	90
Gambar 4.44	Relay <i>FingerOpen</i>	90
Gambar 4.45	Output <i>Backward</i>	91
Gambar 4.46	Output BK1-BK5 dan output VI	92
Gambar 4.47	Relay BK6	93
Gambar 4.48	Relay <i>Reset</i>	93
Gambar 4.49	Relay M1	94
Gambar 4.50	Relay M2	94
Gambar 4.51	Relay M3	94
Gambar 4.52	Relay M4	95
Gambar 4.53	Relay M5	95
Gambar 4.54	Relay M6	96
Gambar 4.55	Relay B1	96
Gambar 4.56	Relay BC1	97

Gambar 4.57 Relay B2	97
Gambar 4.58 Relay BC2.....	97
Gambar 4.59 Relay B3	98
Gambar 4.60 Relay BC3.....	98
Gambar 4.61 Relay B4	99
Gambar 4.62 Relay BC4.....	99
Gambar 4.63 Relay B5	100
Gambar 4.64 Relay BC5.....	100
Gambar 4.65 Relay B6	100
Gambar 4.66 Relay BC6.....	101
Gambar 4.67 Output ED1-ED5 dan output E6	102
Gambar 4.68 Relay ED6.....	103
Gambar 4.69 Relay <i>Reset2</i>	103
Gambar 4.70 Relay <i>Blue1</i>	104
Gambar 4.71 Relay <i>Black1</i>	104
Gambar 4.72 Relay <i>Blue2</i>	104
Gambar 4.73 Relay <i>Black2</i>	105
Gambar 4.74 Relay <i>Blue3</i>	105
Gambar 4.75 Relay <i>Black3</i>	106
Gambar 4.76 Relay <i>Blue4</i>	106
Gambar 4.77 Relay <i>Black4</i>	106
Gambar 4.78 Relay <i>Blue5</i>	107
Gambar 4.79 Relay <i>Black5</i>	107
Gambar 4.80 Relay <i>Blue6</i>	108
Gambar 4.81 Relay <i>Black6</i>	108
Gambar 4.82 Relay FD6.....	109
Gambar 4.83 Relay <i>Reset3</i>	109
Gambar 4.84 Output FD1-FD5 dan output F6.....	110
Gambar 4.85 Relay <i>Blue</i>	111
Gambar 4.86 Relay <i>Black</i>	112
Gambar 4.87 Validasi aktuator modul <i>separation</i>	113
Gambar 4. 88 Validasi aktuator modul <i>pick & place</i>	114
Gambar 4.89 Validasi aktuator modul <i>stopper</i>	116
Gambar 4.90 Validasi aktuator modul <i>line movement</i> untuk benda kerja plastik hitam.....	117

Gambar 4.91 Validasi aktuator modul <i>line movement</i> untuk benda kerja plastik biru	118
Gambar 4.92 Grafik waktu pengujian proses setiap jenis benda kerja	120

[Halaman ini sengaja dikosongkan]

DAFTAR TABEL

Tabel 2.1 Klasifikasi jenis benda kerja.....	20
Tabel 3.1 Daftar input <i>Factory Automatic Trainer</i>	26
Tabel 3.2 Daftar output <i>Factory Automatic Trainer</i>	28
Tabel 3.3 Daftar <i>letter</i> output	35
Tabel 3.4 Urutan output operasi sistem	36
Tabel 4.1 Daftar alamat input/output.....	67
Tabel 4.2 Tabel pengecekan urutan kerja pada modul <i>separation</i>	113
Tabel 4.3 Tabel pengecekan urutan kerja pada modul <i>pick & place</i> ..	115
Tabel 4.4 Tabel pengecekan urutan kerja pada modul <i>stopper</i>	115
Tabel 4.5 Tabel pengecekan urutan kerja pada modul <i>line movement</i> untuk benda kerja plastik hitam	117
Tabel 4.6 Tabel pengecekan urutan kerja pada modul <i>line movement</i> untuk benda kerja plastik biru	119
Tabel 4.7 Pengujian sistem.....	119

[Halaman ini sengaja dikosongkan]

BAB 1

PENDAHULUAN

Pada bab pertama ini akan dijelaskan mengenai latar belakang topik tugas akhir, permasalahan, batasan masalah, tujuan, metodologi, sistematika penulisan dan relevansi atau manfaat.

1.1 Latar Belakang

Perkembangan industri dalam satu abad terakhir ini semakin maju dimana proses produksi tidak lagi dilakukan secara manual. Perusahaan-perusahaan di industri harus melakukan produksi barang secara massal, cepat dan murah. Hal itu perlu dilakukan agar industri dapat mengimbangi permintaan konsumen yang tinggi dan mempertahankan kelangsungan bisnisnya di tengah persaingan yang begitu ketat. Kemunculan teknologi otomasi sangat memudahkan kerja industri yang mulanya menggunakan mesin uap atau makhluk hidup sebagai penggerak. Dengan adanya teknologi otomasi maka dimulailah penggunaan mesin listrik dan pengendali (kontroller) otomatis. Teknologi otomasi industri mampu memperpendek waktu produksi serta menekan beban biaya karena bekerja pada mesin yang terus beroperasi dan adanya hubungan satu mesin dengan mesin lainnya yang menggerakkan satu sama lain. Penggerakan dilakukan atas perintah kontroller utama. Kontroller utama dalam dunia industri yang paling sering digunakan adalah *Programmable Logic Controller* (PLC).

PLC merupakan alat elektronika digital yang menggunakan *programmable memory* untuk menyimpan instruksi dan untuk menjalankan fungsi-fungsi khusus seperti logika, *sequence* (urutan), *timing* (pewaktuan), penghitungan dan operasi aritmatika untuk mengendalikan mesin dan proses. PLC memiliki komponen utama yang terhubung dengan input dan output perangkat. Dan PLC juga terhubung ke *Personal Computer* (PC) untuk kebutuhan pemrograman [1]. Cara penghubungannya dapat secara serial berupa kabel RS232, RS485 atau melalui *Ethernet* menggunakan TCP/IP (*Transmission Control Protocol/Internet Protocol*). Untuk mengatur kontroller PLC digunakan bahasa pemrograman. Bahasa pemrograman yang digunakan pada PLC

telah diatur oleh standar internasional IEC 1131-3 (*International Electrotechnical Commission*) yang terbit pada tahun 1993. Bahasa pemrograman IEC 1131-3 antara lain *ladder diagram* (LAD), *instruction list* (IL), *sequential function charts* (SFC), *structured text* (ST), dan *function block diagram* (FBD) [2]. Sedangkan bahasa yang umum digunakan untuk memprogram PLC adalah *ladder diagram*. *Ladder diagram* memodifikasi rangkaian listrik program yang diinginkan menjadi kumpulan relay. Relay sendiri dibagi menjadi dua macam yaitu kontak dan koil. Kontak menggambarkan masukan sedangkan koil mempresentasikan keluaran relay pada program *ladder diagram*. Setiap relay dihubungkan satu sama lain oleh sebuah garis yang mempresentasikan aliran listrik.

1.2 Permasalahan

Fokus permasalahan yang diangkat pada tugas akhir ini adalah tentang cara merancang diagram *ladder* dengan menggunakan metode *Cascade* untuk banyak objek pada *Factory Automatic Trainer*.

1.3 Batasan Masalah

Batasan masalah tugas akhir ini sebagai berikut :

1. Bahasa pemrograman PLC yang digunakan adalah diagram *ladder*.
2. Metode yang digunakan dalam pemrograman PLC adalah metode *Cascade*.
3. Pemrosesan banyak objek pada *Factory Automatic Trainer* yaitu benda kerja ditambahkan ketika benda kerja sebelumnya masih dalam proses dan tanpa memperhatikan urutan pasti dari ketiga jenis benda kerja tersebut yang merupakan bagian dari *Factory Automatic Trainer*.
4. Software perancangan diagram *ladder* pada PLC yang digunakan adalah *GMWIN 4*.

1.4 Tujuan

Tujuan tugas akhir ini adalah merancang sistem kontrol dalam bahasa diagram *ladder* dengan menggunakan metode *Cascade* untuk banyak objek pada *Factory Automatic Trainer*.

1.5 Metodologi

Metode pengerjaan tugas akhir yang dilakukan meliputi :

a. Penelitian Awal dan Studi Literatur

Pada tahap awal dilakukan identifikasi peralatan-peralatan yang digunakan pada tugas akhir berdasarkan literatur yang dikumpulkan. Studi literatur yang akan dilakukan menghasilkan data primer dan data sekunder. Data primer merupakan data yang didapatkan dari dosen pembimbing dan mahasiswa yang telah mengerjakan tugas akhir dengan judul serupa. Sedangkan data sekunder merupakan data yang didapatkan dari literatur, paper dan jurnal ilmiah demi menunjang data primer.

b. Identifikasi Input/Output *Factory Automatic Trainer*

Untuk mengetahui cara kerja *Factory Automatic Trainer* dilakukan pengoperasian terlebih dahulu serta pemahaman tentang modul-modul yang tersedia. Kemudian dilakukan identifikasi Input/Output pada PLC yang mengendalikan seluruh modul.

c. Pemrograman

Setelah mengetahui daftar Input/Output maka akan lebih mudah dalam merancang diagram *ladder* dengan metode *Cascade*. Proses pertama merancang diagram *ladder* metode *Cascade* adalah menginialisasi seluruh output menjadi huruf-huruf. Kemudian output urutan kerja berupa huruf tersebut dibagi menjadi beberapa grup. Dimana dalam satu grup tidak boleh terdapat huruf yang sama karena melakukan gerakan kontradiktif.

d. Implementasi

Rancangan diagram *ladder* yang telah selesai dirancang akan diupload ke CPU (*Central Processing Unit*) PLC menggunakan kabel RS232. Selama implementasi diagram *ladder*, hasil operasi *pada Factory Automatic Trainer* akan diamati. Apabila hasilnya belum sesuai harapan maka proses akan terus diulang.

e. Penyusunan Laporan Tugas Akhir

Tahap terakhir yang dilakukan adalah penyusunan laporan tugas akhir. Buku laporan tugas akhir merupakan laporan dan hasil dari proses-proses yang dilakukan pada tugas akhir dan disusun secara sistematis dan rinci. Terdiri dari bab pendahuluan, teori penunjang, perancangan sistem, implementasi dan analisisnya serta penutup.

1.6 Sistematika Laporan

Penulisan laporan tugas akhir terdiri dari lima bab, yang membahas keseluruhan proses tugas akhir dengan sistematika penulisan sebagai berikut :

BAB I PENDAHULUAN

Bab ini terdiri dari latar belakang, perumusan masalah, batasan masalah, tujuan, metodologi, sistematika laporan dan relevansi atau manfaat.

BAB II TEORI PENUNJANG

Bab ini membahas dasar teori yang diperlukan dalam melakukan penelitian tugas akhir. Di antaranya adalah teori otomasi sistem mengenai pemrograman diagram *ladder* PLC, teori metode *Cascade*, teori sistem *Factory Automatic Trainer*, teori instrumentasi sensor dan aktuator yang digunakan.

BAB III PERANCANGAN SISTEM

Bab ini membahas langkah-langkah perancangan diagram *ladder* pada *Fatory Automatic Trainer* dengan metode *Cascade*.

BAB IV IMPLEMENTASI DAN ANALISIS

Bab ini memaparkan hasil dari rancangan diagram *ladder*. Disertai juga penerapan pada sistem beserta analisis hasilnya.

BAB V PENUTUP

Bab ini menyajikan kesimpulan yang diperoleh selama tugas akhir serta saran dari kekurangan dan kelemahan pada hasil penelitian.

1.7 Relevansi

Penelitian tugas akhir ini diharapkan dapat digunakan sebagai referensi dalam menyusun rancangan diagram *ladder* metode *Cascade*, pengoperasian dan pengendalian *Factory Automatic Trainer*.

BAB 2

TEORI PENUNJANG

Pada bab ini akan dijelaskan tentang sistem *Factory Automatic Trainer*, penjelasan masing-masing modul, sensor beserta aktuator pada *Factory Automatic Trainer*, metode rancangan diagram *ladder* berupa metode *Cascade*.

2.1 *Factory Automatic Trainer*

Factory Automatic Trainer merupakan model miniatur plant industri. *Factory Automatic Trainer* terdiri dari 5 modul yaitu modul *separation*, modul *pick & place*, modul *stopper*, modul *line movement* dan *control unit*.



Gambar 2.1 *Factory Automatic Trainer* [3]

2.1.1 Modul *Separation*

Modul ini merupakan proses awal dari *Factory Automatic Trainer*. *Double acting pneumatic cylinder* dengan label *insert* akan menambahkan benda kerja ketika terdeteksi oleh sensor *magazine*.

Sensor *magazine* adalah sensor jenis *fiber optic*. Untuk menggerakkan *double acting pneumatic cylinder* digunakan *5/2-way solenoid valve*.

Penambahan benda kerja akan diikuti oleh identifikasi. Identifikasi dilakukan untuk mengklasifikasikan benda kerja menjadi 3 kelompok yaitu benda kerja metal, benda kerja plastik hitam dan benda kerja plastik biru. Klasifikasi menentukan proses yang akan diterapkan pada benda kerja tersebut.

Terdapat pula proses pembuangan benda kerja metal. Proses dilakukan oleh *double acting pneumatic cylinder* yang memiliki label *eject*. Gerakan *eject cylinder* juga dilakukan melalui perantara *5/2-way solenoid valve* sama seperti *insert cylinder*.

Benda kerja plastik hitam dan biru diloloskan dari proses *ejecting*. Nantinya benda kerja tersebut akan mengenai ujung konveyor dan mengaktifkan sensor *fiber optic* dengan label *end*. Sensor *end* akan memicu pengaktifan modul selanjutnya.



Gambar 2.2 Modul *separation* [3]

2.1.2 Modul *Pick & Place*

Modul ini menjelaskan proses benda kerja yang telah melewati proses modul *separation*. Sensor *end* akan memicu gerakan *vertical cylinder* ke bawah dan *vacuum pad* akan menghisap benda kerja. Benda kerja yang telah dihisap akan dibawa ke modul *stopper* oleh *rotational cylinder*. *Vertical cylinder* menggunakan *single acting pneumatic cylinder* sedangkan *rotational cylinder* adalah *double acting pneumatic cylinder*. Semua *cylinder* dikendalikan oleh *5/2-way solenoid valve*.



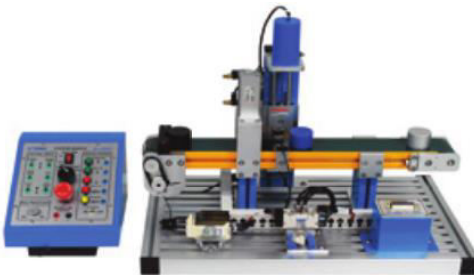
Gambar 2.3 Modul *pick & place* [3]

2.1.3 Modul *Stopper*

Modul ini menjelaskan proses benda kerja yang telah dibawa oleh modul *pick & place*. Benda kerja yang telah diletakkan akan dijalankan oleh konveyor menuju *stopper*.

Sensor *fiber optic* dengan label *workpoint* akan mendeteksi benda kerja yang datang dan mengaktifkan proses *drilling*. *Double acting pneumatic cylinder drilling* merenggang ke bawah dan melakukan proses *drilling*. Setelah proses *drilling* selesai, *single acting pneumatic cylinder stopper* akan menyusut ke atas. Konveyor kembali menjalankan benda kerja menuju ujung konveyor. *Pneumatic cylinder drilling* dan *pneumatic cylinder stopper* dikendalikan oleh *5/2-way solenoid valve*.

Sensor *fiber optic* lainnya berada di ujung konveyor dengan label *end*. Sensor ini akan aktif ketika benda kerja mengenai sensor. Sensor akan mengaktifkan modul berikutnya.



Gambar 2.4 Modul *stopper* [3]

2.1.4 Modul *Line Movement*

Modul ini menjelaskan proses benda kerja yang telah melewati proses *drilling*. Benda kerja digenggam oleh *finger grip*. *Finger* bergerak ke bawah dan ke atas mengambil benda kerja.

Benda kerja plastik hitam dibawa menuju area putih di ujung titik *forward*. Benda kerja plastik biru dibawa menuju kotak hitam di antara area putih dan modul *stopper*. Kemudian *finger* kembali bergerak mundur ke titik semula.

Gerakan ke bawah-atas dan maju-mundur dilakukan oleh *double acting pneumatic cylinder*. Sedangkan gerakan *finger grip* menggunakan *single acting pneumatic cylinder*.

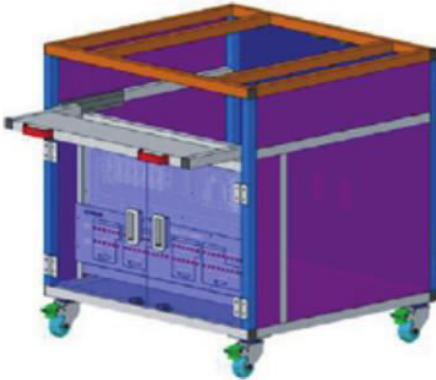


Gambar 2.5 Modul *line movement* [3]

2.1.5 *Control Unit*

Modul ini merupakan kontroller *Factory Automatic Trainer*. *Control unit* berisikan Input/Output sistem, kabel penghubung, *circuit breaker* dan PLC.

Input dan output sistem *Factory Automatic Trainer* masing-masing memiliki 32 slot. Input/Output akan disambungkan ke PLC oleh kabel penghubung. Hal ini bertujuan agar sistem *Factory Automatic Trainer* dapat berjalan secara otomatis. Kontroller PLC yang digunakan adalah merek *LG Glofa* tipe *GM4*. *Control unit* juga dilengkapi dengan *circuit breaker* sebagai pengaman sistem.



Gambar 2.6 Modul *control unit* [3]

2.2 Komponen Penunjang

Selain 5 modul *Factory Automatic Trainer*, masih terdapat komponen lain yang mendukung kerja sistem yaitu, sensor, aktuator PLC, benda kerja, kompresor dan *air filter*.

Beberapa sensor yang digunakan adalah sensor *Capacitive*, sensor *Photo*, sensor *Proximity* dan sensor *Fiber Optic*. Sedangkan aktuator yang digunakan yaitu *Konveyor*, *Solenoid Valve*, *Pneumatic Cylinder*, *Finger Grip*, *Vacuum Pad* dan *3-Colored Sign Tower*.

2.2.1 Sensor

Sensor adalah komponen elektronika yang digunakan untuk mendeteksi adanya perubahan lingkungan secara fisik atau kimia. Sensor mengubah besaran perubahan lingkungan berupa mekanik, magnetik, panas, cahaya, kelembapan, gelombang suara, level ketinggian dan kimia menjadi besaran listrik. Perubahan lingkungan yang mempengaruhi rangkaian listrik diterima sensor sebagai input dan sistem memprosesnya menjadi output sesuai kerja sistem. Pada bahasan kali ini akan dijelaskan tentang sensor-sensor yang disematkan pada *Factory Automatic Trainer*. Sensor yang digunakan ada 4 jenis yaitu, sensor *Proximity Capacitive*, sensor *Photo*, sensor *Proximity Inductive* dan sensor *Fiber Optic*.

2.2.1.1 *Sensor Proximity Capacitive* [4]

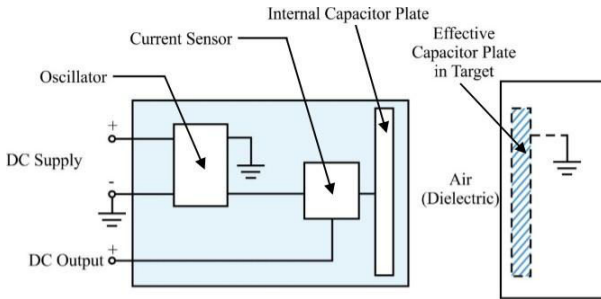
Arti *proximity* secara bahasa adalah kedekatan. Makna secara luas adalah sensor yang dapat mendeteksi suatu objek tanpa adanya kontak fisik. Sensor *proximity capacitive* berfungsi untuk mendeteksi adanya objek dengan melihat perubahan nilai kapasitansi apabila didekatkan.

Sensor *proximity capacitive* dapat merasakan objek karena kemampuan objek tersebut untuk diberi muatan secara elektrik. Bahkan benda non-konduktif dapat menahan muatan ketika didekatkan dengan sensor *proximity capacitive*. Artinya hampir semua objek dapat dideteksi oleh sensor ini.



Gambar 2.7 Sensor *proximity capacitive* [5]

Rangkaian dalam sensor menggunakan daya suplai *Direct Current* (DC) untuk menghasilkan arus *Automatic Current* (AC). Hal ini diperuntukkan dalam pengukuran arus di rangkaian dalam berupa AC dan untuk penggantian output rangkaian ketika jumlah arus AC berubah. Arus AC akan mengisi kapasitor jika terdapat dua pelat yang terpisah oleh dielektrik. Kapasitor dapat menahan muatan karena satu pelat diisi secara positif sedangkan muatan negatif ditarik ke pelat lainnya sehingga memungkinkan muatan yang lebih positif dimasukkan ke dalam pelat pertama. Jika kedua pelat tidak ada dan saling berdekatan maka sangat sulit untuk menyebabkan kedua pelat tersebut terisi daya penuh. Hanya dibutuhkan satu dari dua pelat kapasitor untuk ditanam pada sensor *proximity capacitive*.



Gambar 2.8 Cara kerja sensor *proximity capacitive* [4]

Arus AC dapat memindahkan arus masuk dan keluar ke pelat dalam sensor jika hanya ada pelat lain di dekatnya yang dapat menahan muatan berlawanan. Objek yang dideteksi bertindak sebagai pelat lain. Jika objek cukup dekat ke permukaan sensor *proximity capacitive* untuk dipengaruhi oleh muatan di pelat dalam sensor maka objek akan merespon dengan menjadi muatan yang berlawanan. Kemudian sensor dapat memindahkan arus yang signifikan masuk dan keluar pelat dalam sensor.

2.2.1.2 *Sensor Proximity Inductive* [6]

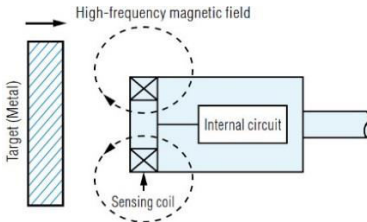
Sensor *proximity inductive* berfungsi untuk mendeteksi objek metal yang didekatkan tanpa adanya kontak fisik. Sensor *proximity inductive* menggunakan prinsip osilasi frekuensi tinggi dengan induksi elektromagnetik.



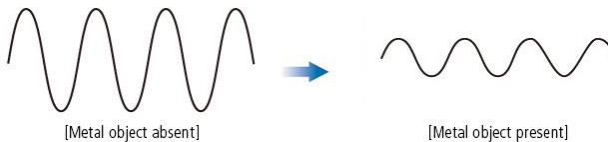
Gambar 2.9 Sensor *proximity inductive* [7]

Medan magnet frekuensi tinggi dihasilkan oleh koil yang berada di rangkaian osilasi. Ketika objek mendekati medan magnet, arus induksi

(arus *eddy*) mengalir di objek karena induksi elektromagnetik. Ketika objek mendekati sensor *proximity inductive*, aliran arus induksi meningkat yang menyebabkan peningkatan beban pada rangkaian osilasi. Kemudian osilasi melemah atau berhenti. Sensor *proximity inductive* mendeteksi perubahan ini dalam status osilasi dengan rangkaian pendeteksi amplitudo dan mengeluarkan sinyal deteksi.

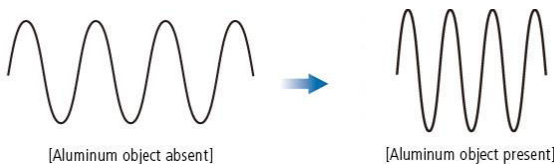


Gambar 2.10 Cara kerja sensor *proximity inductive* [6]



Gambar 2.11 Osilasi frekuensi pada objek metal [6]

Sedangkan untuk objek *nonferrousmetal* termasuk osilasi frekuensi tinggi. Jenis logam ini menggabungkan rangkaian osilasi dimana energi hilang disebabkan oleh arus induksi yang mengalir pada objek. Hal itu menyebabkan perubahan frekuensi osilasi. Misalkan aluminium atau tembaga didekatkan ke sensor *proximity inductive* maka frekuensi osilasi meningkat dan sensor mengeluarkan sinyal deteksi.



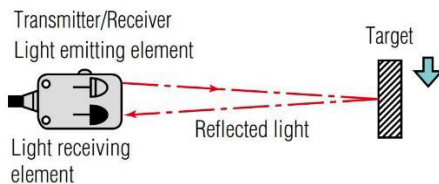
Gambar 2.12 Osilasi frekuensi pada objek *nonferrousmetal* [6]

2.2.1.3 *Sensor Photo* [6]

Sensor *photo* memancarkan sinar cahaya (cahaya terlihat atau berupa infra merah) dari komponen pemancar cahaya. Selain itu sensor *photo* juga mendeteksi sinar cahaya yang direfleksi dari objek. Komponen pemancar cahaya dan penerima cahaya berada dalam satu ruangan. Semua objek yang dapat merefleksikan cahaya dari sensor maka dapat dideteksi. Objek-objek tersebut termasuk kaca, metal, plastik, kayu dan cairan.



Gambar 2.13 Sensor *photo* [8]



Gambar 2.14 Cara kerja sensor *photo* [6]

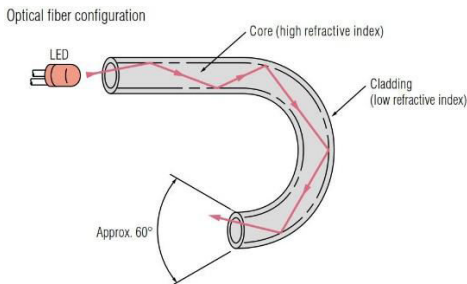
2.2.1.4 *Sensor Fiber Optic* [6]

Sensor *fiber optic* memiliki serat optik yang terhubung ke sumber cahaya untuk mengatur deteksi pada celah rapat atau sisi yang sempit. Serat optik terdiri dari inti dan pelapis yang memiliki indeks bias yang berbeda. Sinar cahaya berjalan melalui inti dengan berulung kali dipantulkan pada pelapis. Sinar cahaya yang telah melewati serat tanpa kehilangan kualitas akan tersebar pada sudut sekitar 60° dan memancarkan ke objek.

Sensor *fiber optic* dipasang untuk kebutuhan yang serba guna. Serat optik yang fleksibel dapat dipasang pada celah yang terbatas seperti celah antar mesin.



Gambar 2.15 Sensor *fiber optic* [10]



Gambar 2.16 Cara kerja sensor *fiber optic* [6]

2.2.2 Aktuator

Aktuator secara bahasa memiliki makna “penggerak”. Aktuator adalah peralatan yang mengubah sinyal masukan listrik menjadi gerakan mekanik atau keluaran lainnya berupa fisik [10]. Untuk aktuator penggerak mekanik biasanya diaktifkan oleh motor listrik (elektrik), beberapa menggunakan *solenoid valve* (pneumatik) atau aliran fluida (hidrolik). Dimana pengaktifan aktuator dikendalikan oleh kontroller diantaranya adalah PLC. Komponen aktuator yang digunakan pada bahasan tugas akhir ini adalah konveyor, *pneumatic cylinder*, *solenoid valve*, *finger*, *vacuum pad* dan *3-colored sign tower*.

2.2.2.1 Konveyor

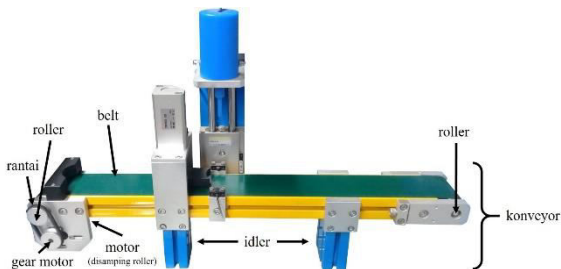
Konveyor adalah alat mekanis untuk membawa atau mengangkat barang dari suatu tempat ke tempat lain. Konveyor tersusun dari beberapa komponen antara lain *belt*, *idler*, *roller* dan motor.

Belt adalah komponen berjalan yang mengangkat barang dari titik awal konveyor menuju ujung akhir konveyor. *Belt* biasanya menggunakan bahan karet karena elastis.

Idler adalah komponen penyangga atau penahan *belt*. Fungsinya adalah dapat menahan *belt* yang mengangkat barang. Selain itu *idler* juga menjaga agar *belt* tidak keluar dari jalurnya.

Roller adalah komponen yang menggerakkan *belt* agar dapat berjalan. *Roller* dipasangkan pada kedua ujung konveyor. Pada konveyor jenis lain beberapa *roller* dapat dipasangkan di tengah konveyor sekaligus.

Motor pada konveyor berfungsi sebagai penggerak *roller*. Untuk menghubungkan motor dengan *roller* digunakan *gear* dan rantai seperti yang terlihat pada **Gambar 2.17**.



Gambar 2.17 Komponen penyusun konveyor

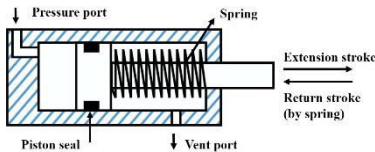


Gambar 2.18 Motor penggerak konveyor

2.2.2.2 Pneumatic Cylinder

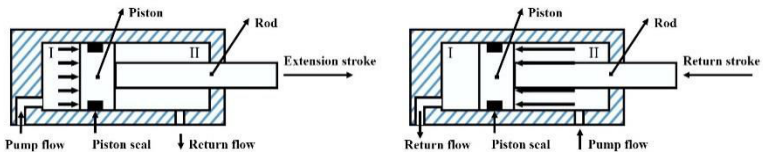
Pneumatic cylinder adalah alat mekanis yang melakukan gerakan karena diberikan tekanan gas. Suplai gas bertekanan dikendalikan oleh *solenoid valve*. Terdapat dua jenis *pneumatic cylinder* yaitu *single acting pneumatic cylinder* dan *double acting pneumatic cylinder*.

Single acting pneumatic cylinder memiliki satu suplai gas bertekanan untuk menggerakkan *cylinder*. Saat *cylinder* ingin digerakkan maju maka gas mendorong *cylinder* ke depan. Apabila *cylinder* ingin digerakkan mundur maka cukup dengan membuang gas yang terisi sebelumnya. Contoh jenis *pneumatic cylinder* ini pada *Factory Automatic Trainer* adalah *stopper cylinder*.



Gambar 2.19 Cara kerja *single acting pneumatic cylinder* [11]

Double acting pneumatic cylinder memiliki dua suplai gas bertekanan untuk menggerakkan *cylinder*. Saat *cylinder* ingin digerakkan maju maka gas akan mengisi ruangan pertama. Apabila *cylinder* ingin digerakkan mundur maka gas pada ruangan pertama akan dibuang dan diganti dengan gas baru pada ruangan kedua. Contoh jenis *pneumatic cylinder* ini pada *Factory Automatic Trainer* adalah *insert cylinder* dan *eject cylinder*.



Gambar 2.20 Cara kerja *double acting pneumatic cylinder* [11]



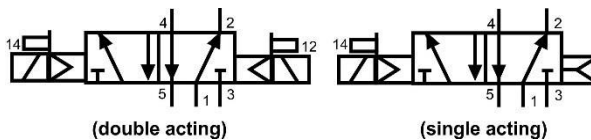
Gambar 2.21 *Insert cylinder dan eject cylinder*

2.2.2.3 Solenoid Valve

Solenoid valve biasa digunakan untuk mengendalikan gas bertekanan pada pengoperasian alat atau sistem. Jenis *solenoid valve* dibagi berdasarkan jumlah portnya yaitu *2-way valves*, *3-way valves* dan *4-way valves*. Dimana *2-way valves* mempunyai 2 port, *3-way valves* mempunyai 3 port dan *4-way valves* mempunyai 4 atau lebih port.

Selain itu setiap jenis port *solenoid valve* dibagi lagi berdasarkan jumlah lubang masuk dan keluarnya gas. Untuk *2-way valves* mempunyai berbagai jenis yaitu *3/2-way valves*, *4/2-way valves* dan *5/2-way valves*. Solenoid valve tipe *2-way* dapat dispesifikasi menjadi *normally open* (NO) atau *normally closed* (NC). Jenis NO mempertahankan kondisi katup terbuka sampai ada arus yang memerintahkan untuk menutup. Katup kembali terbuka secara otomatis ketika kembali ke kondisi normal. Berlawanan dengan NO, kondisi awal katup NC adalah tertutup sampai ada arus yang memerintahkan untuk membuka. *Solenoid valve* jenis NC adalah yang paling sering digunakan.

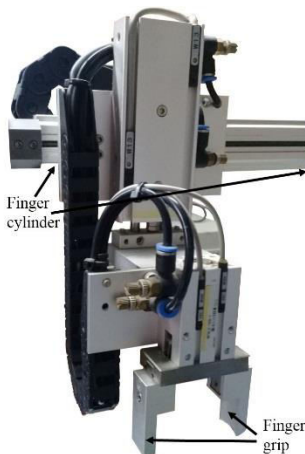
Pada sistem *Fatory Automatic Trainer* lebih banyak dijumpai solenoid tipe *5/2-way valves NC* dan *5/2-way solenoid valves NO* hanya dijumpai satu pada *stopper cylinder*. Solenoid pada *double acting pneumatic cylinder* menggunakan dua penekan elektrik. Sedangkan solenoid pada *single acting pneumatic cylinder* menggunakan satu penekan elektrik dan satu pegas pembalik.



Gambar 2.22 *Solenoid valves double acting dan single acting* [12]

2.2.2.4 *Finger*

Finger adalah alat berbentuk jari yang berfungsi sebagai penggenggam atau pengangkut barang dalam sebuah industri. *Finger* dalam sistem *Factory Automatic Trainer* menggunakan dua macam *pneumatic cylinder*. Pertama *single acting pneumatic cylinder* yang digunakan pada *finger grip*. *Finger grip* bertugas untuk menggenggam atau mencapit benda kerja. Kedua *double acting pneumatic cylinder* yang diterapkan pada *finger cylinder* untuk bergerak *forward* dan *backward* mengangkut benda kerja menuju tujuan akhir.



Gambar 2.23 *Finger*

2.2.2.5 *Vacuum*

Vacuum biasa digunakan untuk menghisap atau menyedot barang dalam proses *placing*. *Vacuum* pada *Factory Automatic Trainer* digerakkan menggunakan dua buah *double acting pneumatic cylinder* yaitu *vertical cylinder* untuk proses naik-turun membawa benda kerja dan *rotational cylinder* untuk proses berputar berlawanan arah/searah jarum jam untuk memindahkan benda kerja serta sebuah *vacuum pad* untuk menghisap benda kerja.



Gambar 2.24 Vacuum

2.2.2.6 3-Colored Sign Tower

Dalam sistem *Factory Automatic Trainer* terdapat lampu indikator penanda proses yang tengah terjadi yaitu *3-colored sign tower*. Alat ini mempunyai tiga lampu indikator yang masing-masing memiliki warna berbeda yaitu merah, kuning dan hijau. Lampu merah menandakan keseluruhan proses tengah berjalan. Lampu kuning mengisyaratkan bahwa proses *drilling* sedang berlangsung. Lampu hijau menunjukkan bahwa keseluruhan proses berhenti.



Gambar 2.25 3-Colored sign tower dalam posisi horizontal [13]

2.2.3 Programmable Logic Controller

Programmable Logic Controller (PLC) adalah salah satu kontroller yang digunakan secara luas dalam dunia otomasi. PLC sudah menjadi teknologi yang digunakan dalam dunia otomasi sejak tahun 1960-an. Program PLC adalah program pengendali gerakan yang terdiri dari instruksi-instruksi untuk menjelaskan gerakan yang diinginkan dan logika kendali secara umum untuk sebuah mesin. Semua gerakan yang diinginkan dapat diprogram menggunakan perhitungan matematika, logika, operasi mesin Input/Output dan perintah lainnya untuk tipe-tipe gerakan yang spesifik. Biasanya program PLC ditulis dalam format diagram *ladder* dan berjalan secara terus-menerus dalam mode scan.

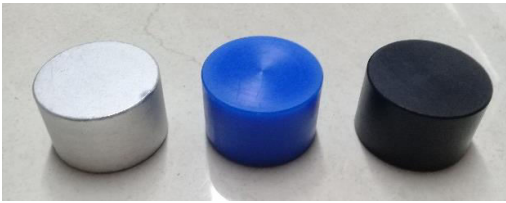
Intinya program PLC berjalan dalam *infinite loop* sekali ketika pertama kali dimulai. Pada tiap-tiap siklus scan, PLC pertama kali membaca state seluruh perangkat Input/Output. Kemudian baru dilakukan scan logika pada program dari baris atas sampai bawah untuk menentukan state output baru perangkat yang diberikan oleh state input perangkat sekarang. Terakhir, program akan mengupdate output perangkat tersebut [14].

2.2.4 Benda Kerja

Benda kerja adalah sebuah item atau objek penelitian yang diproses dalam sistem *Factory Automatic Trainer*. Benda kerja dibagi dalam tiga jenis. Klasifikasi ini berdasarkan bahan dasar pembuatan dan warnanya. Benda kerja pertama berbahan dasar metal dan berwarna abu-abu. Benda kerja metal berwarna abu-abu akan dideteksi oleh sensor *proximity inductive*, sensor *proximity capacitive* dan sensor *photo*. Kedua, benda kerja berbahan dasar plastik dan berwarna biru. Ketiga, benda kerja berbahan dasar plastik dan berwarna hitam. Benda kerja plastik biru dideteksi oleh sensor *proximity capacitive* dan sensor *photo* sedangkan benda kerja plastik hitam hanya dideteksi oleh sensor *proximity capacitive* [10]. Nantinya tiap-tiap jenis benda kerja memiliki alur proses kerja berbeda. Berikut adalah tabel jenis benda kerja.

Tabel 2.1 Klasifikasi jenis benda kerja

Benda Kerja	Warna	Bahan Dasar	Sensor		
			Proximity Capacitive	Photo	Proximity Inductive
I	Abu-abu	Metal	✓	✓	✓
II	Biru	Plastik	✓	✓	
III	Hitam	Plastik	✓		



Gambar 2.26 Tiga jenis benda kerja

2.2.5 Kompresor

Kompresor adalah alat mekanik yang memiliki fungsi memampatkan gas atau memberikan gas bertekanan. Kerja kompresor aktif ketika tuas pengaktif ditarik yang menyebabkan udara luar masuk ke dalam. Selanjutnya udara yang terkumpul dimampatkan menuju output *valve* berbentuk keran. Karena volume keran tersebut kecil maka tekanan gas yang dihasilkan tinggi dengan maksimal tekanan 6 bar. Dari keran tersebut akan dialirkan menuju Factory Automatic Trainer melalui *air filter*.



Gambar 2.27 Kompresor

2.2.6 Air Filter

Setiap mesin pneumatik memiliki *air filter* yang berfungsi sebagai penyaring. *Air filter* menyaring gas yang masuk ke dalam *Factory*

Automatic Trainer dari debu, kotoran dan mengendapkan uap air dari gas bertekanan tersebut.



Gambar 2.28 *Air filter* dalam posisi horizontal [15]

2.3 Metode *Cascade*

Metode *cascade* merupakan salah satu metode dalam pemrograman diagram *ladder*. Metode *cascade* adalah metode pengurutan yang membagi sistem menjadi beberapa grup dimana output aktif selama grup aktif. Metode *cascade* dapat mengurangi jumlah relay tidak berdasarkan secara fisik saja namun lebih kepada pendekatan jumlah proses. Hal ini dapat memudahkan dalam mendapatkan solusi dari masalah yang kompleks.

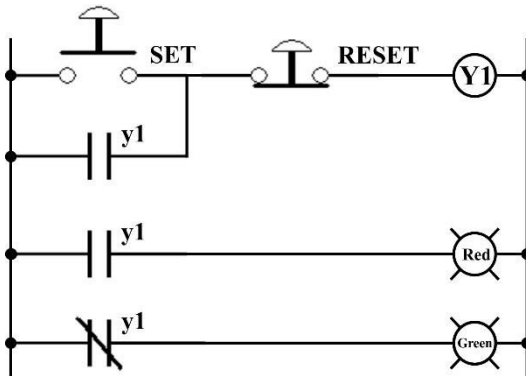
Peraturan dalam pembagian grup adalah tidak boleh terdapat huruf yang sama dalam satu grup sebab output yang dihasilkan berlawanan dan tidak dapat dijalankan. Misalkan terdapat *pneumatic cylinder* A dan B. Notasi “X+” sebagai gerakan maju atau merenggang dan notasi “X-” sebagai gerakan mundur atau menyusut. Sistem memiliki urutan kerja “START, A+, B+, B-, A-”. B+ tidak boleh berada dalam satu grup dengan B- karena merupakan huruf yang sama dan akan melakukan kerja yang berlawanan. Sehingga pembagian grup menjadi :

$$\underbrace{\text{START, A+, B+}}_I \quad \underbrace{\text{B-, A-}}_{II} \quad (2.1)$$

Setiap grup didalam urutan kerja dialokasikan dalam satu relay yang terhubung dalam ladder diagram seperti sebuah RS (Reset Set) *flip flop* pada **Gambar 2.26** [16] yang apabila ditulis dalam persamaan *Boolean* nya menjadi :

$$Y1 = (\text{SET} + y1) * \overline{\text{RESET}} \quad (2.2)$$

Tiap relay akan aktif apabila diset oleh input pertama grup bisa berupa *push button*, *limit switch*, sensor, relay lain dan sebagainya.. Kemudian relay akan non-aktif apabila direset oleh relay selanjutnya atau input terakhir grup.



Gambar 2.29 Rangkaian *RS flip flop* diimplementasikan dalam relay [16]

Metode cascade dibagi menjadi dua macam yaitu *Single path sequencing systems without sustained outputs* dan *Single path sequencing systems with sustained outputs*.

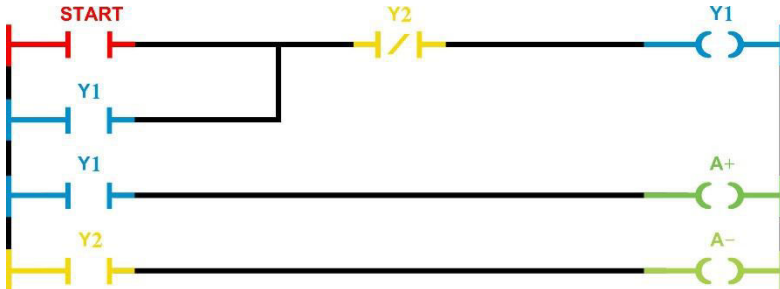
2.3.1 *Single Path Sequencing Systems without Sustained Outputs*

Diasumsikan *cylinder A* digerakkan oleh *5/2-way solenoid valve* dengan dua penekan elektrik *A+* dan *A-*. Solenoid *A+* akan dimatikan ketika *A-* bergerak dengan gerakan sistem “*A+*, *A-*”. Maka diagram *ladder* dari sistem terlihat pada **Gambar 2.30**. Metode ini digunakan pada sistem yang mempunyai komponen *double acting pneumatic cylinder*.

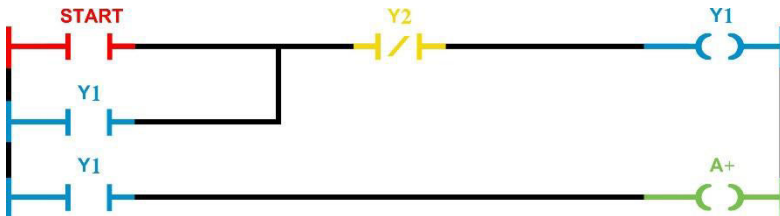
2.3.2 *Single Path Sequencing Systems with Sustained Outputs*

Dalam banyak sistem otomasi industri, sinyal penggerak yang berkelanjutan sangat dibutuhkan. Kasus ini dapat dicontohkan dengan *cylinder A* yang digerakkan oleh satu penekan elektrik dan satu pegas pembalik dengan sistem memiliki urutan kerja “*A+* *A-*”. Untuk menggerakkan *A+* dibutuhkan tegangan yang berkelanjutan karena sistem menggunakan pegas pembalik dimana apabila tegangan tidak dialirkan selama waktu yang dibutuhkan maka akan terjadi gerakan *A-*.

Dan seperti yang sudah dijelaskan sebelumnya untuk menggerakkan A- cukup dengan memutus aliran tegangan A+ sehingga pada metode ini tidak ada notasi X-. Dari penjelasan tersebut dapat dibentuk ladder diagram seperti pada **Gambar 2.31**.



Gambar 2.30 Contoh ladder diagram Single path sequencing systems without sustained outputs [17]



Gambar 2.31 Contoh ladder diagram Single path sequencing systems with sustained outputs [17]

2.4 Fluidsim

Fluidsim adalah sebuah *software* simulasi kontrol sistem pneumatik dengan komponen-komponen seperti *pneumatic cylinder*, suplai pneumatik, *valve*, kontrol elektrik, kontrol digital, port yang bersifat universal dan simulator *grafcet*. Penulis menggunakan *Fluidsim* untuk melakukan validasi diagram *ladder* pada proses pengujian sistem.



Gambar 2.32 Logo *Fluidsim*

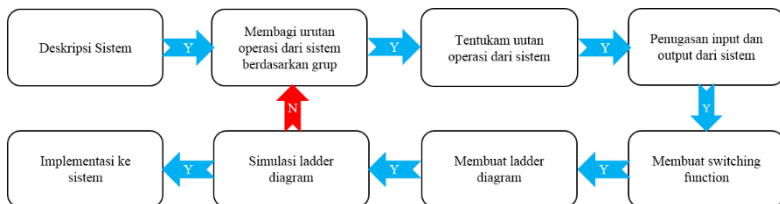
BAB 3 PERANCANGAN SISTEM

Pada bab ini akan dijelaskan tentang perancangan sistem *Factory Automatic Trainer*. Perancangan perancangan diagram *ladder* dengan metode *Cascade* dilakukan dalam beberapa langkah.

3.1 Perancangan Diagram *Ladder* dengan Metode *Cascade*

Dalam pemrograman PLC dibutuhkan metode atau teknik agar program efektif dan tidak memakan banyak memori sehingga menimbulkan *cost* yang tinggi. Dalam tugas akhir ini digunakan program diagram *ladder* yang umum digunakan dan diaplikasikan dengan metode *Cascade*.

Perancangan diagram *ladder* metode *Cascade* memiliki langkah-langkah perancangan. Langkah-langkah tersebut akan dijelaskan dalam bagian ini. Secara umum langkah-langkah perancangan diagram *ladder* metode *Cascade* digambarkan dengan *flowchart* pada **Gambar 3.1**. Sedangkan pada bab 3 ini akan menjelaskan beberapa hal yaitu deskripsi sistem, Input/Output *Factory Automatic Trainer*, pemberian *letter output*, pembagian urutan operasi sistem berdasarkan grup, penentuan urutan operasi sistem, pembuatan *switching function*, persamaan *Boolean* relay dan terakhir *addressing* benda kerja.



Gambar 3.1 Langkah-langkah perancangan *ladder diagram* metode *cascade*

3.2 Input/Output *Factory Automatic Trainer*

Sebelum melangkah lebih jauh merancang diagram *ladder*, terlebih dahulu harus diketahui input dan output sistem. *Factory Automatic Trainer* telah memiliki daftar Input/Output yang sudah dibuat oleh pabrikannya. *Factory Automatic Trainer* memiliki 26 port input dan 23 port output yang telah berlabel dari keseluruhan 32 port input dan 32 port output. Sisa port tersebut dapat digunakan untuk keperluan input maupun output tambahan. Namun salah satu port input mengalami kerusakan sehingga tidak dapat digunakan yaitu port input *insert return* pada *insert cylinder*. Untuk menyasiasi hal tersebut digunakan port input lain yang dapat menunjang kerja *insert cylinder*. Berikut adalah daftar Input/Output yang digunakan pada tugas akhir ini.

Tabel 3.1 Daftar input *Factory Automatic Trainer*

Alamat	Label	Simbol	Fungsi
%IX0.1.11	Tombol Tekan	<i>Start</i>	Memulai keseluruhan proses
%IX0.1.12	Tombol Tekan	<i>Stop</i>	Menghentikan keseluruhan proses
%IX0.0.0	<i>Forward</i>	<i>I_Forward</i>	Mendeteksi <i>finger cylinder forward</i>
%IX0.0.1	<i>Backward</i>	<i>I_Backward</i>	Mendeteksi <i>finger cylinder backward</i>
%IX0.0.2	<i>Up</i>	<i>I_Finger_U</i>	Mendeteksi <i>finger cylinder up</i>
%IX0.0.3	<i>Down</i>	<i>I_Finger_D</i>	Mendeteksi <i>finger cylinder down</i>
%IX0.0.4	<i>Finger Open</i>	<i>FingerOpen</i>	Mendeteksi <i>finger cylinder grip</i> yang terbuka
%IX0.0.5	<i>Finger Grip</i>	<i>FingerGrip</i>	Mendeteksi <i>finger cylinder grip</i>
%IX0.0.6	<i>Magazine</i>	<i>Magazine</i>	Mendeteksi adanya benda pada <i>feeder system</i>

%IX0.0.7	<i>Photo Sensor</i>	<i>Photo_S</i>	Mendeteksi benda kerja yang berwarna cerah
%IX0.0.8	<i>Proximity Sensor</i>	<i>Prox_S</i>	Mendeteksi benda kerja berbahan dasar metal
%IX0.0.9	<i>Capacitive Sensor</i>	<i>Cap_S</i>	Mendeteksi adanya benda kerja
%IX0.0.10	<i>Insert</i>	<i>I_Insert</i>	Mendeteksi posisi renggang <i>insert cylinder</i>
%IX0.0.11	<i>Insert Return</i>	-	Mendeteksi posisi awal <i>insert cylinder</i>
%IX0.0.12	<i>Eject</i>	<i>I_Eject</i>	Mendeteksi posisi renggang <i>eject cylinder</i>
%IX0.0.13	<i>Eject Return</i>	<i>I_Eject_R</i>	Mendeteksi posisi awal <i>insert cylinder</i>
%IX0.0.14	<i>End</i>	<i>End1</i>	Mendeteksi adanya benda kerja di akhir konveyor modul <i>Separation</i>
%IX0.1.0	<i>Stopper Up</i>	<i>I_STPR_U</i>	Mendeteksi naiknya <i>stopper cylinder</i>
%IX0.1.1	<i>Stopper Down</i>	<i>I_Stopper_D</i>	Mendeteksi turunnya <i>stopper cylinder</i>
%IX0.1.2	<i>Workpoint</i>	<i>Workpoint</i>	Mendeteksi adanya benda yang ditahan oleh <i>stopper</i>
%IX0.1.3	<i>Drill Up</i>	<i>I_Drill_U</i>	Mendeteksi naiknya <i>drill cylinder</i>
%IX0.1.4	<i>Drill Down</i>	<i>I_Drill_D</i>	Mendeteksi turunnya <i>drill cylinder</i>
%IX0.1.5	<i>End</i>	<i>End2</i>	Mendeteksi adanya benda kerja di akhir konveyor modul

			<i>Stopper</i>
%IX0.1.6	<i>Rotary CCW (Counter Clockwise)</i>	<i>I_Rot_CCW</i>	Mendeteksi gerakan <i>rotational cylinder</i> berlawanan arah jarum jam
%IX0.1.7	<i>Rotary CW (Clockwise)</i>	<i>I_Rot_CW</i>	Mendeteksi gerakan <i>rotational cylinder</i> searah jarum jam
%IX0.1.8	<i>Up</i>	<i>I_Up</i>	Mendeteksi gerakan <i>rotational cylinder</i> ke atas
%IX0.1.9	<i>Down</i>	<i>I_Down</i>	Mendeteksi gerakan <i>rotational cylinder</i> ke bawah
%IX0.1.10	<i>Vacuum</i>	<i>I_Vacuum</i>	Mendeteksi hisapan <i>vacuum pad</i>

Tabel 3.2 Daftar output *Factory Automatic Trainer*

Alamat	Label	Simbol	Fungsi
%QX0.2.0	<i>Forward</i>	<i>Forward</i>	<i>Finger cylinder</i> melakukan gerakan <i>forward</i>
%QX0.2.1	<i>Backward</i>	<i>Backward</i>	<i>Finger cylinder</i> melakukan gerakan <i>backward</i>
%QX0.2.2	<i>Up</i>	<i>FingerUp</i>	<i>Finger cylinder</i> melakukan gerakan ke atas
%QX0.2.3	<i>Down</i>	<i>FingerDown</i>	<i>Finger cylinder</i> melakukan gerakan ke bawah
%QX0.2.4	<i>Finger Grip</i>	<i>FingerGrip</i>	<i>Finger cylinder</i> melakukan gerakan menggenggam
%QX0.2.6	<i>Insert</i>	<i>Insert</i>	<i>Insert cylinder</i> melakukan gerakan

			merenggang menambahkan benda kerja
<i>%QX0.2.7</i>	<i>Insert Return</i>	<i>Insert_R</i>	<i>Insert cylinder</i> melakukan gerakan menyusut kembali ke posisi semula
<i>%QX0.2.8</i>	<i>Eject</i>	<i>Eject</i>	<i>Eject cylinder</i> melakukan gerakan merenggang membuang benda kerja metal
<i>%QX0.2.9</i>	<i>Eject Return</i>	<i>Eject_R</i>	<i>Eject cylinder</i> melakukan gerakan menyusut kembali ke posisi semula
<i>%QX0.2.10</i>	<i>Conveyyor On</i>	<i>Conveyor1</i>	Mengaktifkan konveyor modul <i>Separation</i>
<i>%QX0.3.0</i>	<i>Stopper Up</i>	<i>StopperUp</i>	<i>Stopper cylinder</i> melakukan gerakan ke atas
<i>%QX0.3.1</i>	<i>Drill Up</i>	<i>DrillUp</i>	<i>Drill cylinder</i> melakukan gerakan ke atas
<i>%QX0.3.2</i>	<i>Drill Down</i>	<i>DrillDown</i>	<i>Drill cylinder</i> melakukan gerakan ke bawah
<i>%QX0.3.3</i>	<i>Drill On</i>	<i>DrillOn</i>	Mengaktifkan <i>drill motor</i>
<i>%QX0.3.4</i>	<i>Conveyor On</i>	<i>Conveyor2</i>	Mengaktifkan konveyor modul <i>Stopper</i>
<i>%QX0.3.6</i>	<i>Rotary CCW</i>	<i>RotaryCCW</i>	Mengaktifkan putaran <i>rotational cylinder</i> berlawanan arah jarum

			jam
%QX0.3.7	<i>Rotary CW</i>	<i>RotaryCW</i>	Mengaktifkan putaran <i>rotational cylinder</i> searah jarum jam
%QX0.3.8	<i>Up</i>	<i>Up</i>	Mengaktifkan gerakan <i>rotational cylinder</i> ke atas
%QX0.3.9	<i>Down</i>	<i>Down</i>	Mengaktifkan gerakan <i>rotational cylinder</i> ke bawah
%QX0.3.10	<i>Vacuum</i>	<i>Vacuum</i>	Mengaktifkan penghisap <i>vacuum pad</i>
%QX0.3.11	<i>Sign Red</i>	<i>Red</i>	Mengaktifkan lampu indikator berwarna merah
%QX0.3.12	<i>Sign Yellow</i>	<i>Yellow</i>	Mengaktifkan lampu indikator berwarna kuning
%QX0.3.13	<i>Sign Green</i>	<i>Green</i>	Mengaktifkan lampu indikator berwarna hijau

3.3 Deskripsi Sistem

Pada bab sebelumnya telah dijelaskan tiap-tiap modul *Factory Automatic Trainer* serta fungsi kerjanya. Selanjutnya akan dijelaskan kerja sistem *Factory Automatic Trainer* untuk banyak objek. Kerja sistem akan dibagi ke beberapa grup sebagai bentuk implementasi metode *Cascade*.

Pada bagian ini dijelaskan terlebih dahulu skenario kerja sistem banyak objek pada *Factory Automatic Trainer*. Skenario kerja sistem akan dijabarkan tiap-tiap modul.

3.3.1 Separation

Tahap awal dari sistem *Factory Automatic Trainer* dilakukan pada modul *separation*. Terdapat tiga macam benda kerja yang akan

dimasukkan pada tahap ini yaitu benda kerja metal berwarna abu-abu, benda kerja plastik berwarna biru dan benda kerja plastik berwarna hitam. Deteksi tiga benda kerja ini dilakukan oleh tiga sensor yaitu sensor *proximity capacitive*, sensor *proximity inductive* dan sensor *photo*. Untuk lebih jelasnya dapat dilihat kembali pada **Tabel 2.1**.

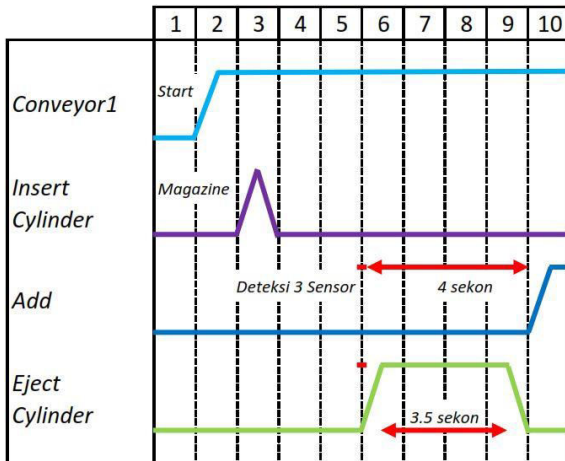
Proses dimulai ketika tombol *Start* ditekan. Selanjutnya konveyor pada modul *separation* (konveyor pertama) akan berjalan. Apabila sensor *magazine* mendeteksi benda kerja maka *insert cylinder* akan merenggang untuk memasukan benda kerja ke konveyor. Setelah selesai *insert cylinder* akan menyusut kembali ke posisi semula.

Benda kerja akan melewati tiga sensor untuk diklasifikasi. Apabila benda kerja yang terdeteksi adalah benda kerja metal berwarna abu-abu maka secara otomatis *eject cylinder* akan merenggang sampai benda kerja tersebut terbuang yaitu selama 3.5 sekon. Sedangkan benda kerja plastik berwarna biru dan benda kerja plastik berwarna hitam akan diteruskan ke ujung konveyor.

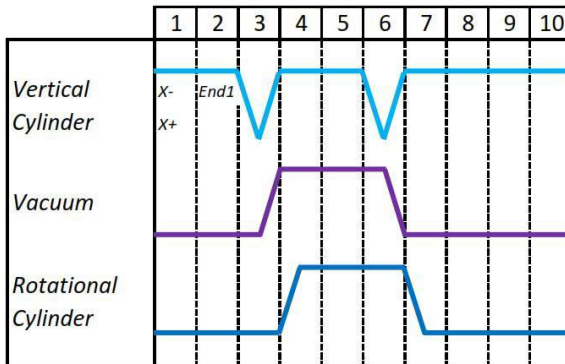
Selain dilakukan klasifikasi ketika melewati ketiga sensor, juga dilakukan penambahan benda kerja setelah 4 sekon. Dan alur kerja benda kerja lainnya sama saat benda kerja melewati ketiga sensor. Benda kerja di ujung konveyor dideteksi sensor dan menstimulasi gerakan *rotational cylinder* pada modul *pick & place*.

3.3.2 Picking

Vertical cylinder bergerak ke bawah dan memvakum (menghisap menggunakan gas) benda kerja kemudian *Rotational cylinder* membawanya ke konveyor pada modul *stopper*. *Rotational cylinder* kembali ke posisi semula dan melakukan kerja yang sama apabila mendeteksi benda kerja lain di ujung konveyor pertama.



Gambar 3.2 Diagram waktu perancangan aktuator modul *separation*



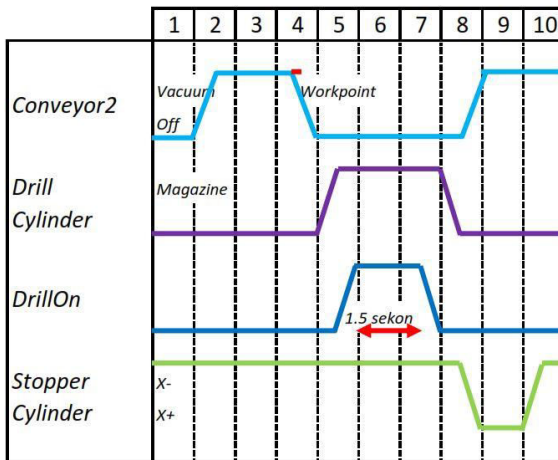
Gambar 3.3 Diagram waktu perancangan aktuator modul *pick & place*

3.3.3 Drilling

Konveyor pada modul *stopper* (konveyor kedua) akan menjalankan benda kerja menuju *stopper cylinder*. Saat sensor *workpoint* mendeteksi benda kerja pada *stopper cylinder* maka konveyor seketika mati. Kemudian *drill cylinder* merenggang ke bawah untuk melakukan proses *drilling* selama 1.5 sekon dan *drill cylinder* menyusut ke atas apabila

proses *drilling* selesai. Selanjutnya *stopper cylinder* naik ke atas dan konveyor kembali menjalankan benda kerja menuju ujung konveyor.

Ketika benda kerja telah melewati *stopper*, *stopper cylinder* turun kembali ke posisi semula dan alur kerja berulang untuk benda kerja selanjutnya yang terdeteksi oleh *workpoint*. Benda kerja di ujung konveyor dideteksi sensor dan menstimulasi gerakan *finger cylinder* pada modul *line movement*.



Gambar 3.4 Diagram waktu perancangan aktuator modul *stopper*

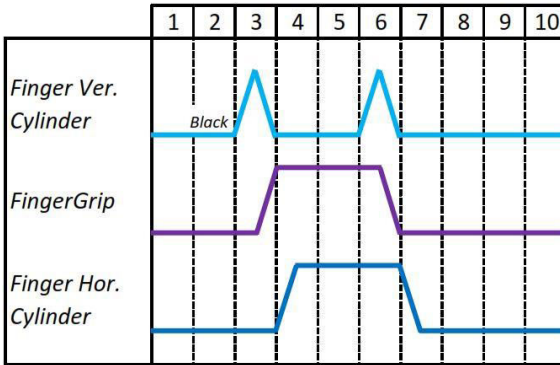
3.3.4 Movement

Benda kerja plastik berwarna hitam di ujung konveyor *stopper* yang dideteksi sensor akan memerintah gerakan *finger cylinder* ke bawah kemudian menggenggam benda kerja. Selanjutnya *finger cylinder* naik ke atas dan bergerak *forward*. Apabila telah sampai, *finger cylinder* bergerak ke bawah dan melepaskan *grip* (genggam), meletakkan benda kerja. Terakhir *finger cylinder* bergerak *backward* kembali ke posisi semula.

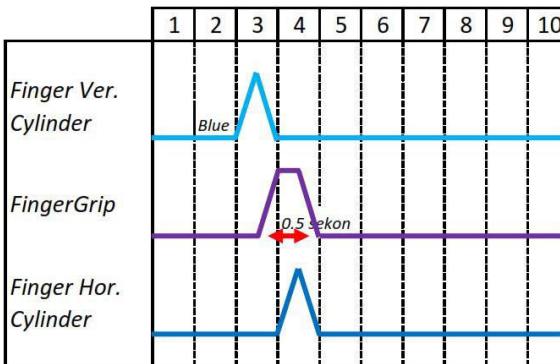
Sedangkan alur kerja untuk benda kerja plastik berwarna biru sama seperti benda kerja berwarna hitam. Namun ketika sedang melakukan gerakan *forward*, *grip* akan terbuka tepat di atas lokasi kotak hitam

berada. Setelah *finger cylinder* telah sampai di *limit switch forward* maka *finger cylinder* akan bergerak *backward* kembali ke posisi semula.

Proses yang dilakukan *finger cylinder* akan terus dilakukan selama terdapat benda kerja yang terdeteksi pada ujung konveyor *stopper* baik itu benda kerja plastik hitam maupun benda kerja plastik berwarna biru.



Gambar 3.5 Diagram waktu perancangan aktuatur modul *line movement* untuk benda kerja plastik hitam



Gambar 3.6 Diagram waktu perancangan aktuatur modul *line movement* untuk benda kerja plastik biru

3.4 Pemberian *Letter Output*

Letter adalah proses inisialisasi sebuah perangkat sistem dalam bentuk urutan abjad atau huruf untuk memudahkan penyebutan ketika merancang diagram *ladder* menggunakan metode *Cascade*. Biasanya setiap perangkat memiliki dua gerakan berlawanan. Asumsikan sebuah *pneumatic cylinder X* mempunyai label X+ dan X-. Label X+ ketika *pneumatic cylinder X* bergerak maju (merenggang) sedangkan label X- untuk gerakan mundur (menyusut) *pneumatic cylinder X*.

Pada bagian ini akan diberikan penyebutan perangkat dengan letter output pada sistem *Factory Automatic Trainer*. Dalam tabel tersebut digunakan urutan abjad yang mempresentasikan urutan perangkat yang bekerja sesuai skenario kerja sistem. Berikut adalah tabel pemberian *letter output*.

Tabel 3.3 Daftar *letter output*

Alamat	Label	Simbol	Letter
%QX0.2.10	<i>Conveveyor On</i>	<i>Conveyor1</i>	A+
%QX0.2.6	<i>Insert</i>	<i>Insert</i>	B+
%QX0.2.7	<i>Insert Return</i>	<i>Insert_R</i>	B-
%QX0.2.8	<i>Eject</i>	<i>Eject</i>	C+
%QX0.2.9	<i>Eject Return</i>	<i>Eject_R</i>	C-
%QX0.3.8	<i>Up</i>	<i>Up</i>	D+
%QX0.3.9	<i>Down</i>	<i>Down</i>	D-
%QX0.3.10	<i>Vacuum</i>	<i>Vacuum</i>	E+ (on) / E- (off)
%QX0.3.6	<i>Rotary CCW</i>	<i>RotaryCCW</i>	F+
%QX0.3.7	<i>Rotary CW</i>	<i>RotaryCW</i>	F-
%QX0.3.4	<i>Conveyor On</i>	<i>Conveyor2</i>	G+ (on) / G- (off)
%QX0.3.1	<i>Drill Up</i>	<i>DrillUp</i>	H+
%QX0.3.2	<i>Drill Down</i>	<i>DrillDown</i>	H-
%QX0.3.3	<i>Drill On</i>	<i>DrillOn</i>	I+ (on) / I- (off)
%QX0.3.0	<i>Stopper Up</i>	<i>StopperUp</i>	J+ (on) / J- (off)
%QX0.2.3	<i>Down</i>	<i>FingerDown</i>	K+
%QX0.2.2	<i>Up</i>	<i>FingerUp</i>	K-
%QX0.2.4	<i>Finger Grip</i>	<i>FingerGrip</i>	L+ (on) / L- (off)
%QX0.2.0	<i>Forward</i>	<i>Forward</i>	M+

%QX0.2.1	<i>Backward</i>	<i>Backward</i>	M-
----------	-----------------	-----------------	----

3.5 Pembagian Urutan Operasi Sistem Berdasarkan Grup

Setelah mengetahui skenario kerja sistem pada *Factory Automatic Trainer* dan daftar Input/Output nya maka akan diurutkan output operasi sistem. Output pada urutan operasi sistem ini adalah output yang telah diberi simbol. Tujuannya adalah agar dapat membedakan perangkat yang memiliki fungsi dan alamat yang berbeda namun memiliki label yang sama pada daftar Input/Output.

Tabel 3.4 Urutan output operasi sistem

START	
SEPARATION MODULE	
<i>Start</i> aktif	
<i>Conveyor1</i> aktif	
<i>Magazine</i> aktif	
<i>Insert</i> aktif	
<i>Insert_R</i> aktif	
Metal terdeteksi	Biru atau Hitam terdeteksi
<i>Prox_S</i> dan <i>Cap_S</i> aktif	<i>Cap_S</i> aktif
<i>Eject</i> aktif selama 3.5 sekon	<i>Insert</i> aktif setelah 4 sekon <i>Cap_S</i> aktif
<i>Eject_R</i> aktif	<i>Insert_R</i> aktif
<i>Insert</i> aktif setelah 4 sekon <i>Cap_S</i> aktif	<i>End1</i> aktif
<i>Insert_R</i> aktif	
LOOPING ketika <i>Magazine</i> aktif	
PICK & PLACE MODULE	
<i>Down</i> aktif	
<i>Vacuum</i> aktif	
<i>Up</i> aktif	
<i>Rot_CCW</i> aktif	
<i>Down</i> aktif	
<i>VacuumOff</i> aktif	

<i>Up</i> aktif	STOPPER MODULE
<i>Rot_CW</i> aktif	<i>Conveyor2</i> aktif
LOOPING ketika <i>End1</i> aktif	<i>Workpoint</i> aktif
	<i>Conveyor2</i> non aktif
	<i>DrillDown</i> aktif
	<i>DrillOn</i> aktif selama 1.5 sekon
	<i>DrillUp</i> aktif
	<i>StopperUp</i> aktif selama 3 sekon
	<i>Conveyor2</i> aktif setelah <i>StopperUp</i> aktif
	<i>End2</i> aktif
	LOOPING ketika <i>Workpoint</i> aktif
LINE MOVEMENT MODULE	
Benda Kerja Biru	Benda Kerja Hitam
<i>FingerDown</i> aktif	<i>FingerDown</i> aktif
<i>FingerGrip</i> aktif	<i>FingerGrip</i> aktif
<i>FingerUp</i> aktif	<i>FingerUp</i> aktif
<i>FingerForward</i> aktif	<i>FingerForward</i> aktif
<i>FingerGrip</i> non aktif setelah 0.5 sekon	<i>FingerDown</i> aktif
<i>FingerBackward</i> aktif	<i>FingerGrip</i> non aktif
	<i>FingerUp</i> aktif
	<i>FingerBackward</i> aktif
LOOPING ketika <i>End2</i> aktif	
END	

Dari tabel urutan output operasi sistem di atas dapat diubah menjadi urutan *letter* output operasi sistem sebagai berikut.

STARTING A+ B+ B- C+ C-

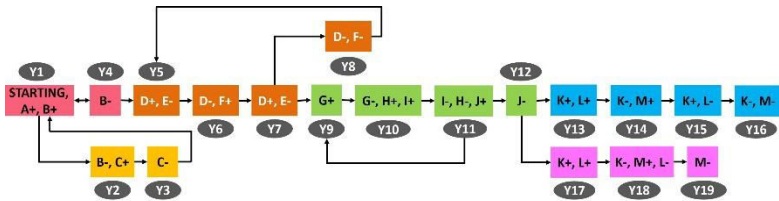
STARTING A+ B+ B- D+ E+ D- F+ D+ E- D- F-

D+ E- G+ G- H+ I+ I- H- J+ G+ J- K+ L+ K- M+ K+ L- K- M-

J- K+ L+ K- M+ L- M-

Gambar 3.7 Urutan *letter* output operasi sistem

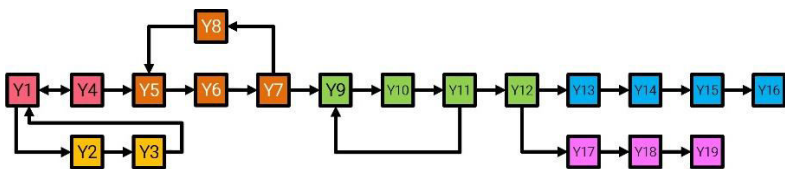
Kemudian urutan *letter* output operasi sistem akan dibagi menjadi beberapa grup. Dalam satu grup tidak boleh terdapat *letter* yang sama [18] karena hal ini akan mengakibatkan gerakan yang berlawanan sehingga sistem tidak akan berjalan. Selain itu juga dapat mengakibatkan kerusakan secara fisik pada perangkat.



Gambar 3.8 Pembagian grup urutan *letter* output operasi sistem

3.6 Penentuan Urutan Operasi Sistem

Selesai pembagian urutan operasi sistem berdasarkan grup, selanjutnya adalah penentuan urutan operasi sistem. Grup yang telah terbentuk akan diurutkan menjadi sebuah alur kerja sistem dengan anak panah sebagai arah kerjanya. Apabila terdapat arah yang kembali ke grup sebelumnya maka artinya grup-grup tersebut mengalami *looping*. Berikut adalah urutan operasi sistem digambarkan melalui blok diagram.



Gambar 3.9 Blok diagram urutan grup

3.7 Pembuatan *Switching Function*

Ini adalah langkah terakhir dalam perancangan diagram *ladder* dengan metode *Cascade*. Bagian ini menjelaskan pembuatan *Switching function* dan diagram *ladder* berdasarkan grup *Cascade* yang diimplementasikan menjadi relay. Terdapat penjelasan 19 relay yang

sebanding dengan jumlah grup *Cascade* yang terbentuk. Tiap-tiap relay memiliki beberapa output kerja yang mewakili output kerja pada urutan operasi sistem yang telah dibagi ke dalam grup-grup.

3.7.1 Relay Y1

- Input 1 : *Start*
- Output dari Input 1 : *Conveyor1*
- Input 2 : *Magazine*
- Output dari Input 2 : *Insert*
- Input 3 : *I_Eject_R*
- Output dari Input 3 : *Insert*

Relay Y1 adalah relay pertama yang memiliki tiga buah input yaitu tombol tekan *Start*, sensor *Magazine* dan *limit switch I_Eject_R*. Selain itu terdapat dua buah output yaitu *Conveyor1* dan *Insert* (gerakan maju *Insert cylinder*). Saat tombol *Start* ditekan maka akan mengaktifkan kondisi *Starting* yang memberikan suplai aliran ke seluruh sistem. *Starting* diaktifkan melalui *safe holding* dari tombol *Start* dan akan diputus oleh tombol *Stop* sehingga seluruh sistem akan berhenti bekerja. *Starting* menghidupkan relay Y1 dan seluruh outputnya dan output yang pertama kali diaktifkan adalah *Conveyor1*.

Apabila sensor *Magazine* mendeteksi adanya benda kerja maka output *Insert* aktif mendorong benda kerja menuju *Conveyor1* yang sedang berjalan. Nantinya *Conveyor1* akan membawa benda kerja menuju area deteksi sensor (deteksi dilakukan oleh sensor *proximity capacitive*, sensor *photo* dan sensor *proximity inductive*) untuk diklasifikasikan. Ketika output *Insert* telah mencapai *limit switch*, relay Y2 akan aktif dan melakukan reset terhadap relay Y1.

Relay Y1 juga aktif ketika relay Y3 aktif dan menyentuh *limit switch I_Eject_R*. Artinya relay Y1 akan kembali aktif ketika relay Y3 telah selesai memproses benda kerja metal abu-abu dimana sebelumnya aliran relay Y1 telah diputus oleh relay Y2 dan perlu kembali diaktifkan untuk memproses benda kerja lainnya.

3.7.2 Relay Y2

- Input 1 : *I_Insert*

- Output dari Input 1 : *Insert_R*
- Input 2 : *Prox_S*
- Output dari Input 2 : *Prox*, *Eject* dan *Tim_Ej*
- Input 3 : *Cap_S*
- Output dari Input 3 : *Cap*, *Tim_Add*, *Tim_Cap* dan *Add*

Relay Y2 merupakan relay yang memproses benda kerja metal abu-abu. Benda kerja metal abu-abu dideteksi saat *sensor proximity capacitive (Cap_S)*, *sensor photo (Photo_S)* dan *sensor proximity inductive (Prox_S)* aktif. Sensor *Cap_S* dirangkai *safe holding* sehingga memiliki relay *Cap* dan diputus oleh *limit switch I_Insert* begitu pula dengan sensor *Prox_S* dengan rangkaian *safe holding* relay *Prox*. Relay Y2 memiliki tiga input yaitu *I_Insert*, *Prox_S*, *Cap_S* dan *Photo_S* serta delapan output yaitu *Insert_R*, *Prox*, *Eject*, *Tim_Ej*, *Cap*, *Tim_Add*, *Tim_Cap*, dan *Add*.

Starting memberikan suplai aliran ke relay Y2 dan seluruh outputnya. Selain itu, relay Y2 harus aktif saat *Insert cylinder* mengenai *limit switch I_Insert* dan relay Y2 melakukan perintah mundur kepada *Insert cylinder* yang menandakan output *Insert_R* aktif. Sementara itu benda kerja yang terdeteksi berbahan dasar metal dan berwarna abu-abu pada area deteksi sensor diaktivasi oleh relay *Prox* akan memicu gerakan maju *Eject cylinder* atau output *Eject* aktif serta pengaktifan *Tim_Ej* setelah 3.5 sekon. Timer *Tim_Ej* merupakan jenis *Timer ON Delay (TON)*. Selanjutnya *Tim_Ej* akan mengaktifkan relay Y3. Aktifnya relay Y3 akan mereset relay Y2.

Benda kerja yang telah melewati area deteksi sensor secara otomatis mengaktifkan relay *Cap* dan dihubungkan paralel dengan TON *Tim_Add* dengan durasi 4 sekon. Aktifnya relay *Cap* dan *Tim_Add* (setelah 4 sekon) akan mengaktifkan relay *Add*. Relay *Add* akan memerintahkan *Insert cylinder* bergerak maju kembali untuk menambahkan benda kerja apabila terdeteksi oleh sensor *Magazine*. Relay *Cap* juga mengaktifkan timer *Tim_Cap* setelah 1 sekon yang berfungsi untuk *addressing* benda kerja plastik hitam dan biru.

3.7.3 Relay Y3

- Input 1 : *Tim_Ej*
- Output dari Input 1 : *Eject_R*

Relay Y3 merupakan proses pergerakan mundur *Eject cylinder* atau pengaktifan output *Eject_R*. Relay Y3 memiliki satu input yaitu *Tim_Ej* dan satu output yaitu *Eject_R*. *Starting* menyuplai aliran relay Y3 dan seluruh outputnya. Relay Y3 diaktifkan oleh timer *Tim_Ej* (aktif setelah 3.5 sekon) dan menghasilkan output *Eject_R*. Setelah *Eject cylinder* menyentuh *limit switch I_Eject_R*, relay Y3 akan kembali mengaktifkan relay Y1.

3.7.4 Relay Y4

- Input 1 : *I_Insert*
- Output dari Input 1 : *Insert_R*
- Input 2 : *Cap_S*
- Output dari Input 2 : *Cap*, *Tim_Add* dan *Add*
- Input 3 : *Photo_S*
- Output dari Input 3 : *Photo*

Relay Y4 merupakan proses dari benda kerja berbahan plastik baik berwarna biru maupun hitam yang kemudian akan dilanjutkan pada relay-relay berikutnya. Relay Y4 memiliki tiga input dengan rincian satu input merupakan pengaktif relay Y4 yaitu *limit switch I_Insert* dan dua input lain sebagai pengalamanan benda kerja plastik berwarna biru atau hitam yaitu sensor *Cap_S* dan *Photo_S*. Relay Y4 juga memiliki lima output dengan rincian tiga output sebagai aktivasi aktuatur *Insert cylinder* yaitu *Insert_R*, *Tim_Add* dan *Add*, dua output sebagai relay *self holding* dari sensor *Cap_S* dan *Photo_S* yaitu *Cap* dan *Photo* yang diputus oleh *limit switch I_Insert*. Dimana relay *Cap* dan *Photo* sebagai *addressing* benda kerja plastik hitam dan biru.

Starting memberikan suplai aliran ke relay Y4 dan seluruh outputnya. Syarat relay Y4 aktif apabila *Insert cylinder* mengenai *limit switch I_Insert*. Kasus ini hampir sama dengan relay Y2 namun Y4 tidak akan melakukan proses *ejecting* dan membiarkan benda kerja selain metal sampai ke ujung konveyor pertama mengenai sensor *End1*. Berikutnya relay Y5 aktif karena mendeteksi benda kerja pada sensor *End1* dan mereset relay Y4.

3.7.5 Relay Y5

- Input 1 : *End1*
- Output dari Input 1 : *Down*
- Input 2 : *I_Down*
- Output dari Input 2 : *Vacuum*
- Input 3 : *I_Rot_CW* dan *End1*
- Output dari Input 3 : *Down*

Relay Y5 merupakan tahap awal dari modul *pick & place*. Relay Y5 memiliki tiga input yaitu *End1*, *limit switch I_Down* serta gabungan input logika AND *I_Rot_CW* dan *End1*. Relay Y5 juga memiliki dua output yaitu *Down* dan *Vacuum*.

Starting memberikan suplai aliran ke relay Y5 dan seluruh outputnya. Syarat relay Y5 aktif adalah ketika sensor *End1* aktif karena terkena benda kerja dan menghasilkan gerakan ke bawah dari *Vertical cylinder* atau output *Down* aktif. Ketika *Vertical cylinder* mengenai *limit switch I_Down* maka akan mengaktifkan output *Vacuum*. Selain itu relay Y5 kembali aktif setelah relay Y8 selesai mengerjakan output *RotaryCW* ditandai dengan aktifnya *limit switch I_Rot_CW* dan sensor *End1* aktif. Kemudian relay Y6 aktif saat *Vacuum* aktif dan relay Y5 direset.

3.7.6 Relay Y6

- Input 1 : *I_Vacuum*
- Output dari Input 1 : *Up*
- Input 2 : *I_Up*
- Output dari Input 2 : *RotaryCCW*

Relay Y6 memiliki dua input yaitu *limit switch I_Vacuum* dan *limit switch I_Up* serta juga memiliki dua output yaitu *Up* dan *RotaryCCW*. *Starting* memberikan suplai aliran ke relay Y6 dan seluruh outputnya. Syarat relay Y6 aktif adalah saat *limit switch I_Vacuum* aktif kemudian menghasilkan output *Up* atau gerakan *Vertical cylinder* ke atas. Setelah *Vertical cylinder* menyentuh *limit switch I_Up*, relay Y6 akan mengeluarkan output *RotaryCCW* atau gerakan *Rotational cylinder* berlawanan arah jarum jam. Ketika *Rotational cylinder* menyentuh *limit switch I_Rot_CCW* maka relay Y7 akan aktif dan mereset relay Y6.

3.7.7 Relay Y7

- Input 1 : *I_Rot_CCW*
- Output dari Input 1 : *Down*
- Input 2 : *I_Down*
- Output dari Input 2 : *VacuumOff*

Relay Y7 memiliki dua input yaitu *limit switch I_Rot_CCW* dan *limit switch I_Down* serta memiliki dua output yaitu *Down* dan *VacuumOff*. *Starting* memberikan suplai aliran ke relay Y7 dan seluruh outputnya. Syarat relay Y7 aktif adalah saat *Rotational cylinder* menyentuh *limit switch I_Rot_CCW* kemudian menghasilkan output *Down* atau gerakan *Vertical cylinder* ke bawah. Setelah *Vertical cylinder* menyentuh *limit switch I_Down*, relay Y7 mengeluarkan output relay *VacuumOff* dan mengaktifkan relay Y8. Aktifnya relay Y8 akan mereset relay Y7.

3.7.8 Relay Y8

- Input 1 : *VacuumOff*
- Output dari Input 1 : *Up*
- Input 2 : *I_Up*
- Output dari Input 2 : *RotaryCW*

Relay Y8 memiliki dua input yaitu relay *VacuumOff* dan *limit switch I_Up* serta memiliki dua output yaitu *Up* dan *RotaryCW*. *Starting* memberikan suplai aliran ke relay Y8 dan seluruh outputnya. Syarat relay Y8 aktif adalah saat relay *VacuumOff* aktif kemudian menghasilkan output *Up* atau gerakan ke atas *Vertical cylinder*. Setelah *Vertical cylinder* menyentuh *limit switch I_Up*, relay Y8 mengeluarkan output *RotaryCW* yaitu gerakan *Rotational cylinder* searah jam. Setelah *Rotational cylinder* menyentuh *limit switch I_Rot_CW* dan sensor *End1* aktif, relay Y5 akan diaktifkan kembali dan mereset relay Y8.

3.7.9 Relay Y9

- Input 1 : *VacuumOff*
- Output dari Input 1 : *Conveyor2*
- Input 2 : *I_STPR_UP*

- Output dari Input 2 : *Conveyor2*

Relay Y9 memiliki dua input yaitu relay *VacuumOff* dan *limit switch I_STPR_UP* serta memiliki satu output yaitu *Conveyor2*. *Starting* memberikan suplai aliran ke relay Y9 dan seluruh outputnya. Syarat relay Y9 aktif adalah ketika *VacuumOff* pada relay Y7 memutuskan aliran *Vacuum* dan menghasilkan output *Conveyor2*. Selain itu relay Y9 juga kembali aktif ketika relay Y11 telah selesai dan *limit switch I_STPR_UP* aktif dimana *Stopper cylinder* bergerak ke atas untuk menjalankan kembali benda kerja yang telah selesai dilakukan proses *drilling* oleh *Conveyor2*.

3.7.10 Relay Y10

- Input 1 : *Workpoint*
- Output dari Input 1 : *DrillDown*
- Input 2 : *I_Drill_D*
- Output dari Input 2 : *DrillOn* dan *Tim_DOff*

Starting memberikan suplai aliran ke relay Y10 dan seluruh outputnya. Syarat relay Y10 aktif adalah saat benda kerja terdeteksi oleh sensor *Workpoint* kemudian mereset relay Y9. Relay Y10 memiliki dua input yaitu sensor *Workpoint* dan *limit switch I_Drill_D* serta memiliki tiga output yaitu *DrillDown* (gerakan *Drill cylinder* ke bawah), *DrillOn* (motor berputar melakukan proses *drilling*) dan timer *Tim_DOff*. Ketika benda kerja mengenai sensor *Workpoint* maka relay Y10 memutuskan aliran *Conveyor2*. Setelah konveyor kedua mati maka akan memerintahkan gerakan *Drill cylinder* ke bawah dan menyentuh *limit switch I_Drill_D*. *Limit switch I_Drill_D* aktif dan melakukan output *DrillOn* dan timer *Tim_DOff* aktif setelah 1.5 sekon *DrillOn* aktif. Kondisi akhir dari relay Y10 adalah motor drill berhenti berputar. Timer *Tim_DOff* juga menjadi input relay Y11 dan relay Y10 direset ketika relay Y11 aktif.

3.7.11 Relay Y11

- Input 1 : *Tim_DOff*
- Output dari Input 1 : *DrillUp*
- Input 2 : *I_Drill_U*
- Output dari Input 2 : *StopperUp* dan *Tim_STPR*

Starting memberikan suplai aliran ke relay Y11 dan seluruh outputnya. Syarat aktif relay Y11 adalah setelah timer *Tim_DOff* menghitung 1.5 sekon. Relay Y1 memiliki dua input yaitu timer *Tim_DOff* dan *limit switch I_Drill_U* serta memiliki tiga output yaitu *DrillUp*, *StopperUp* dan timer *Tim_STPR*. Timer *Tim_DOff* aktif akan memberikan perintah pada relay Y11 untuk menon-aktifkan *DrillOn*. Kemudian relay Y11 memerintahkan *Drill cylinder* bergerak ke atas atau *DrillUp* aktif. *Drill cylinder* akan menyentuh *limit switch I_Drill_U* dan mengaktifkan *StopperUp* (*Stopper cylinder* bergerak ke atas) serta mengaktifkan timer *Tim_STPR* setelah 3 sekon. Relay Y11 dan *limit switch I_STPR* yang aktif setelah *Stopper cylinder* mengenainya akan mengaktifkan kembali relay Y9 yang memberikan perintah jalan pada konveyor kedua. Aktifnya timer *Tim_STPR* akan mengaktifkan relay Y12 dan mereset relay Y11.

3.7.12 Relay Y12

- Input 1 : *Tim_STPR*
- Output dari Input 1 : -

Relay Y12 merupakan relay pengembalian posisi *Stopper cylinder* ke bawah. Relay Y12 memiliki satu input yaitu timer *Tim_STPR*. *Starting* memberikan suplai aliran ke relay Y12. Syarat aktif relay Y12 adalah setelah *Tim_STPR* menghitung 3 sekon. Relay Y12 mereset relay Y11 sehingga aliran output *StopperUp* terputus atau bisa dikatakan bahwa gerakan *Stopper cylinder* kembali ke bawah. Pada langkah berikutnya, relay Y12 menjadi input bagi relay Y13 dan Y17. Kemudian relay Y12 direset oleh relay *R_Y12* (lihat detailnya di bagian **3.8.12**).

3.7.13 Relay Y13

- Input 1 : *Black*
- Output dari Input 1 : *Finger_D*
- Input 2 : *I_Finger_D*
- Output dari Input 2 : *FingerGrip*

Relay Y13 merupakan relay untuk memproses benda kerja plastik berwarna hitam. *Starting* memberikan suplai aliran ke relay Y13 dan seluruh outputnya. Syarat aktif relay Y13 adalah ketika benda kerja

menyentuh sensor *End2* dan dideteksi menjadi *Black*. Relay Y13 memiliki dua input yaitu *Black* dan *I_Finger_D* serta memiliki dua output yaitu *Finger_D* dan *Finger_Grip*. Setelah benda dideteksi sebagai *Black*, *Finger vertical cylinder* bergerak ke bawah atau *Finger_D* aktif. Ketika *Finger vertical cylinder* menyentuh *limit switch I_Finger_D*, output *FingerGrip* diaktifkan yang bekerja untuk menggenggam benda kerja tersebut. Setelah *limit switch I_Finger_G* aktif, akan diaktifkan relay Y14 sedangkan relay Y13 direset.

3.7.14 Relay Y14

- Input 1 : *I_Finger_G*
- Output dari Input 1 : *FingerUp*
- Input 2 : *I_Finger_U*
- Output dari Input 2 : *Forward*

Relay Y14 memiliki dua input yaitu *limit switch I_Finger_G* dan *limit switch I_Finger_Up* serta memiliki dua output yaitu *FingerUp* dan *Forward*. *Starting* memberikan suplai aliran ke relay Y14 dan seluruh outputnya. *Limit switch I_Finger_G* aktif akan memberikan perintah output *FingerUp* yaitu *Finger cylinder* bergerak ke atas. Ketika *Finger cylinder* telah menyentuh *limit switch I_Finger_U*, relay Y14 akan mengeluarkan output *Forward* atau *Finger horizontal cylinder* bergerak maju. Apabila *Finger horizontal cylinder* menyentuh *limit switch I_Forward* maka relay Y15 diaktifkan sedangkan relay Y14 direset.

3.7.15 Relay Y15

- Input 1 : *I_Forward*
- Output dari Input 1 : *FingerDown*
- Input 2 : *I_Finger_D*
- Output dari Input 2 : *FingerOpen*

Relay Y15 memiliki dua input yaitu *limit switch I_Forward* dan *limit switch I_Finger_D* serta memiliki dua output yaitu *FingerDown* dan *FingerGrip*. *Starting* memberikan suplai aliran ke relay Y15 dan seluruh outputnya. Syarat aktif relay Y15 adalah setelah *limit switch I_Forward* telah disentuh, kemudian menghasilkan output *FingerDown* atau gerakan ke bawah *Finger vertical cylinder*. Selanjutnya saat *limit*

switch I_Finger_D aktif, relay Y15 akan memberikan perintah untuk membuka genggaman *Finger* atau output *FingerOpen* aktif. Relay *FingerOpen* akan mengaktifkan relay Y16. Setelah itu relay Y16 mereset relay Y15.

3.7.16 Relay Y16

- Input 1 : *FingerOpen*
- Output dari Input 1 : *FingerUp*
- Input 2 : *I_Finger_U*
- Output dari Input 2 : *Backward*

Relay Y16 memiliki dua input yaitu relay *FingerOpen* dan *limit switch I_Finger_U* serta memiliki dua output yaitu *FingerUp* dan *Backward*. *Starting* memberikan suplai aliran ke relay Y16 dan seluruh outputnya. Syarat aktif relay Y16 adalah ketika relay *FingerOpen* aktif, kemudian mengeluarkan output *FingerUp* atau gerakan ke atas *Finger vertical cylinder*. Apabila *limit switch I_Finger_U* aktif maka relay Y16 akan mengeluarkan output *Backward* atau gerakan mundur *Finger horizontal cylinder*. Saat *limit switch I_Backward* telah ditekan maka relay Y16 telah berakhir.

3.7.17 Relay Y17

- Input 1 : *Blue*
- Output dari Input 1 : *Finger_D*
- Input 2 : *I_Finger_D*
- Output dari Input 2 : *FingerGrip*

Relay Y17 merupakan relay untuk memproses benda kerja plastik berwarna biru. *Starting* memberikan suplai aliran ke relay Y17 dan seluruh outputnya. Syarat aktif relay Y17 adalah ketika benda kerja menyentuh sensor *End2* dan dideteksi menjadi *Blue*. Relay Y17 memiliki dua input yaitu *Blue* dan *I_Finger_D* serta memiliki dua output yaitu *Finger_D* dan *FingerGrip*. Setelah benda dideteksi sebagai *Blue*, *Finger vertical cylinder* bergerak ke bawah atau *Finger_D* aktif. Ketika *Finger vertical cylinder* menyentuh *limit switch I_Finger_D*, output *FingerGrip* diaktifkan yang bekerja untuk menggenggam benda kerja

tersebut. Setelah *limit switch I_Finger_G* aktif, akan diaktifkan relay Y18 sedangkan relay Y17 direset.

3.7.18 Relay Y18

- Input 1 : *I_Finger_G*
- Output dari Input 1 : *FingerUp*
- Input 2 : *I_Finger_U*
- Output dari Input 2 : *Forward* dan *Tim_GOp*
- Input 3 : *Tim_GOp*
- Output dari Input 3 : *FingerOpen*

Relay Y18 memiliki tiga input yaitu *limit switch I_Finger_G*, *limit switch I_Finger_Up* dan timer *Tim_GOp* serta memiliki empat output yaitu *FingerUp*, *Forward*, timer *Tim_GOp* dan relay *FingerOpen*. *Starting* memberikan suplai aliran ke relay Y18 dan seluruh outputnya. *Limit switch I_Finger_G* aktif akan memberikan perintah output *FingerUp* yaitu *Finger cylinder* bergerak ke atas. Ketika *Finger cylinder* telah menyentuh *limit switch I_Finger_U*, relay Y18 akan mengeluarkan output *Forward* (*Finger horizontal cylinder* bergerak maju) dan timer *Tim_GOp* aktif setelah 0.5 sekon. Aktifnya timer *Tim_GOp* akan menghasilkan relay *FingerOpen* untuk memutus aliran *FingerGrip* (genggaman *finger* terlepas). Relay *FingerOpen* juga menjadi input yang mengaktifkan relay Y19. Apabila relay Y19 aktif maka relay Y18 direset.

3.7.19 Relay Y19

- Input 1 : *FingerOpen*
- Output dari Input 1 : *Backward*
- Input 2 : *I_Forward*
- Output dari Input 2 : *Backward*

Relay 19 memiliki dua input yaitu relay *FingerOpen* dan *limit switch I_Forward* serta memiliki satu output yaitu *Backward* atau gerakan mundur *Finger horizontal cylinder*. *Starting* memberikan suplai aliran ke relay Y2 dan seluruh outputnya. Syarat aktif relay Y19 adalah ketika relay *FingerOpen* aktif. Apabila *Finger horizontal cylinder* menyentuh *limit switch I_Forward* maka *cylinder* akan bergerak mundur

atau output *Backward* aktif. Saat *limit switch I_Backward* telah ditekan maka relay Y19 telah berakhir.

3.8 Persamaan *Boolean* Relay

Setelah menentukan *Switching function* maka langkah selanjutnya adalah merumuskan persamaan *Boolean* relay. Dengan menggunakan persamaan *Boolean* relay akan memudahkan dalam merancang diagram *ladder*.

3.8.1 Relay Y1

Bila dilihat dari *Switching function* relay Y1 akan didapatkan beberapa persamaan *Boolean* relay.

Starting

$$(Start + Starting) * \overline{Stop} = Starting \quad (3.1)$$

Relay Y1

$$Starting * [Start + Y1 + (Y3 * I_Eject_R)] * \overline{Y2} = Y1 \quad (3.2)$$

Output relay Y1

$$Starting * (Y1 + Conveyor1) = Conveyor1 \quad (3.3)$$

$$Starting * (Y1 + Add) * Magazine = Insert \quad (3.4)$$

Relay *Starting* pada persamaan *Boolean* relay 3.1 digunakan untuk menghidupkan keseluruhan sistem mulai dari relay Y1-Y19 sampai output-outputnya. Saat ingin menghentikan kerja sistem cukup menekan tombol tekan *Stop* dan aliran relay *Starting* terputus.

Relay Y1 dimulai ketika tombol *Start* ditekan dan terputus oleh relay Y2. Output yang dihasilkan oleh relay Y1 adalah *Conveyor1* atau konveyor pertama berjalan dan *Insert* atau gerakan *Insert cylinder* bergerak maju untuk menambahkan benda kerja. Output *Insert* aktif saat benda kerja terdeteksi oleh sensor *Magazine* pada kondisi relay Y1 menyala. Namun output *Insert* akan kembali aktif ketika relay *Add* aktif yaitu ketika sensor *Cap_S* mendeteksi benda kerja yang melewatinya melalui konveyor pertama dan dihitung oleh timer *Tim_Add* setelah 4 sekon. Untuk mengaktifkan timer *Tim_Add* selama 4 sekon diperlukan *safe holding* dari sensor *Cap_S* berupa relay *Cap* (lihat persamaan 3.5 dan 3.6). Relay *Add* memiliki persamaan *Boolean* relay 3.8. Relay *Add* sendiri berfungsi menambahkan benda kerja secara otomatis setelah 4

sekon benda kerja pertama atau sebelumnya terdeteksi di area pendeteksi sensor. Relay *Cap* juga mengaktifkan timer *Tim_Cap* untuk *addressing* benda kerja plastik hitam dan biru.

$$\mathbf{Starting * (Cap_S + Cap) = Cap} \quad (3.5)$$

$$\mathbf{Starting * (Cap_S + Cap) = Tim_Add} \quad (3.6)$$

$$\mathbf{Starting * (Cap_S + Cap) = Tim_Cap} \quad (3.7)$$

$$\mathbf{Starting * Tim_Add = Add} \quad (3.8)$$

Relay Y1 akan kembali aktif ketika relay Y3 aktif dan mengenai *limit switch I_Eject_R* (sehingga memiliki persamaan 3.2). Sebelumnya relay Y1 telah diputus oleh relay Y2 sehingga perlu diaktifkan kembali untuk memproses benda kerja setelahnya.

3.8.2 Relay Y2

Bila dilihat dari *Switching function* relay Y2 akan didapatkan beberapa persamaan *Boolean* relay.

Relay Y2

$$\mathbf{Starting * [(Y1 * I_Insert) + Y2] * \overline{Y3} = Y2} \quad (3.9)$$

Output relay Y2

$$\mathbf{Starting * Y2 * I_Insert = Insert_R} \quad (3.10)$$

$$\mathbf{Starting * Y2 * Prox = Eject} \quad (3.11)$$

$$\mathbf{Starting * Y2 * Prox * (TON t\#3.5s) = Tim_Ej} \quad (3.12)$$

Relay Y2 merupakan relay proses benda kerja metal abu-abu. Relay Y2 aktif ketika *Insert cylinder* menyentuh *limit switch I_Insert* dan terputus ketika relay Y3 aktif. Output relay Y2 adalah *Insert_R*, *Eject*, *Tim_Ej*. Output *Insert_R* menggerakkan *Insert cylinder* kembali ke posisi semula. Pada persamaan *Boolean* relay 3.10 diperlukan kembali *limit switch I_Insert* (sebelumnya digunakan sebagai aktivasi relay Y2, lihat persamaan 3.9) dikarenakan outputnya yaitu *Insert_R* digunakan juga pada relay Y4 (lihat persamaan 3.18) sehingga membutuhkan memori yang lebih besar.

Output *Eject* menggerakkan *Eject cylinder* maju ke depan saat sensor *Prox_S* aktif. Namun pada persamaan *Boolean* relay 3.11 digunakan input relay *Prox* dimana relay *Prox* adalah relay *latching* dari sensor *Prox_S* seperti yang dijelaskan pada persamaan 3.13 dan 3.14.

$$\mathbf{Starting * Y2 * Prox_S * \overline{Y3} = Eject} \quad (3.13)$$

$$\text{Starting} * (\text{Prox}_S + \text{Prox}) * \overline{I_Insert} = \text{Prox} \quad (3.14)$$

Pada sensor *Prox_S* diperlukan relay *latching (safe holding) Prox* agar dapat melakukan output *Eject*. Dimana *Prox_S* akan mengaktifkan *Prox* dan *Prox* akan mengaktifkan dirinya sendiri. Kemudian diperlukan pengaturan ulang *Prox* agar *Prox* dapat membaca sinyal selanjutnya dari sensor *Prox_S*. Untuk itu diperlukan relay NC *I_Insert* yang berfungsi mereset nilai *Prox*.

Output timer *Tim_Ej* pada persamaan 3.12 digunakan untuk mengatur durasi output *Eject* dari *Eject Cylinder* (lihat persamaan 3.11) yaitu 3.5 sekon. Ketika output timer *Tim_Ej* aktif, relay Y3 akan aktif dan relay Y2 direset.

3.8.3 Relay Y3

Bila dilihat dari *Switching function* relay Y3 akan didapatkan beberapa persamaan *Boolean* relay.

Relay Y3

$$\text{Starting} * [(Y2 * \text{Tim}_Ej) + (Y3)] * \overline{Y1} = Y3 \quad (3.15)$$

Output relay Y3

$$\text{Starting} * Y3 = \text{Eject}_R \quad (3.16)$$

Seperti yang telah dijelaskan pada persamaan *Boolean* relay Y2, relay Y3 aktif ketika timer *Tim_Ej* aktif. Relay Y3 akan mengaktifkan kembali relay Y1 apabila prosesnya telah selesai dikerjakan (lihat persamaan 3.2). Setelah relay Y1 aktif, relay Y3 akan direset. Relay Y3 memiliki satu output yaitu *Eject_R* dimana *Eject cylinder* akan bergerak mundur kembali ke posisi semula.

3.8.4 Relay Y4

Bila dilihat dari *Switching function* relay Y4 akan didapatkan beberapa persamaan *Boolean* relay.

Relay Y4

$$\text{Starting} * [(Y1 * I_Insert) + Y4] * \overline{Y5} = Y4 \quad (3.17)$$

Output relay Y4

$$\text{Starting} * Y4 * I_Insert = \text{Insert}_R \quad (3.18)$$

Relay Y4 merupakan relay proses untuk benda kerja plastik hitam dan biru. Relay Y4 aktif saat *limit switch I_Insert* aktif dan direset

ketika relay Y5 aktif. Relay Y4 memiliki satu output yaitu *Insert_R* dimana *Insert cylinder* akan bergerak mundur kembali ke posisi semula. Pada persamaan *Boolean* relay **3.18** diperlukan kembali *limit switch I_Insert* (sebelumnya digunakan sebagai aktivasi relay Y4, lihat persamaan **3.17**) dikarenakan outputnya yaitu *Insert_R* digunakan juga pada relay Y2 (lihat persamaan **3.10**) sehingga membutuhkan memori yang lebih besar.

3.8.5 Relay Y5

Bila dilihat dari *Switching function* relay Y5 akan didapatkan beberapa persamaan *Boolean* relay.

Relay Y5

$$\textit{Starting} * \{[Y4 + (Y8 * I_{Rot_CW}) * End1] + Y5\} \quad (3.19)$$

$$* \overline{Y6} = Y5$$

Output relay Y5

$$\textit{Starting} * Y5 = \textit{Down} \quad (3.20)$$

$$\textit{Starting} * [(Y5 * I_{Down}) + Vacuum] * \quad (3.21)$$

$$\overline{\textit{VacuumOff}} = \textit{Vacuum}$$

Relay Y5 aktif ketika benda kerja terdeteksi oleh sensor *End1* dan kembali aktif setelah relay Y8 selesai melakukan kerjanya. Relay Y5 memiliki dua output yaitu *Down* atau *Vertical cylinder* bergerak ke bawah serta *Vacuum* atau proses menghisap benda kerja menggunakan gas. Pada persamaan *Boolean* relay **3.21** diperlukan relay *latching Vacuum* agar *Vacuum pad* tetap terus bekerja sampai pada waktu non-aktivasinya yaitu relay *VacuumOff*. Setelah *Vacuum* aktif maka relay Y6 aktif dan relay Y5 direset.

3.8.6 Relay Y6

Bila dilihat dari *Switching function* relay Y6 akan didapatkan beberapa persamaan *Boolean* relay.

Relay Y6

$$\textit{Starting} * [(Y5 * I_{Vacuum}) + Y6] * \overline{Y7} = Y6 \quad (3.22)$$

Output relay Y6

$$\textit{Starting} * Y6 = \textit{Up} \quad (3.23)$$

$$\textit{Starting} * Y6 * I_{Up} = \textit{RotaryCCW} \quad (3.24)$$

Relay Y6 aktif setelah *Vacuum* mengaktifkan *limit switch I_Vacuum* dan direset oleh relay Y7. Relay Y6 memiliki dua output yaitu *Up* dimana *Vertical cylinder* bergerak ke atas serta *RotaryCCW* dimana *Rotational cylinder* bergerak berlawanan arah jarum jam.

3.8.7 Relay Y7

Bila dilihat dari *Switching function* relay Y7 akan didapatkan beberapa persamaan *Boolean* relay.

Relay Y7

$$\text{Starting} * [(Y6 * I_{Rot_CCW}) + Y7] * \overline{Y8} = Y7 \quad (3.25)$$

Output relay Y7

$$\text{Starting} * Y7 = \text{Down} \quad (3.26)$$

$$\text{Starting} * Y7 * I_{Down} = \text{VacuumOff} \quad (3.27)$$

Relay Y7 diaktifkan ketika *limit switch I_Rot_CCW* aktif dan direset oleh relay Y8. Relay Y7 memiliki dua output yaitu *Down* dimana *Vertical cylinder* bergerak ke atas serta relay *VacuumOff* yang menonaktifkan *Vacuum*.

3.8.8 Relay Y8

Bila dilihat dari *Switching function* relay Y8 akan didapatkan beberapa persamaan *Boolean* relay.

Relay Y8

$$\text{Starting} * [(Y7 * \text{VacuumOff}) + Y8] * \overline{Y5} = Y8 \quad (3.28)$$

Output relay Y8

$$\text{Starting} * Y8 = \text{Up} \quad (3.29)$$

$$\text{Starting} * Y8 * I_{Up} = \text{RotaryCW} \quad (3.30)$$

Relay Y8 diaktifkan ketika relay *VacuumOff* aktif dan direset oleh relay Y5 dimana pada **Gambar 3.4** relay Y8 dibentuk paralel dari relay Y7 menuju relay Y5. Relay Y8 memiliki dua output yaitu *Up* dimana *Vertical cylinder* bergerak ke bawah serta *RotaryCW* dimana *Rotational cylinder* bergerak memutar searah jarum jam.

3.8.9 Relay Y9

Bila dilihat dari *Switching function* relay Y9 akan didapatkan beberapa persamaan *Boolean* relay.

Relay Y9

$$\text{Starting} * [(Y7 * \text{VacuumOff}) + (Y11 * I_STPR_UP) + Y9] * \overline{Y10} = Y9 \quad (3.31)$$

Output relay Y9

$$\text{Starting} * Y9 = \text{Conveyor2} \quad (3.32)$$

Relay Y9 merupakan relay lanjutan dari relay Y7. Relay Y9 aktif ketika benda kerja dilepaskan dari *Vacuum pad* atau output relay *VacuumOff* aktif. Relay Y9 merupakan relay penggerak konveyor kedua. Relay Y9 mempunyai satu output yaitu *Conveyor2*. *Conveyor2* merupakan output konveyor kedua yang berada di modul *stopper*. Relay Y9 kembali aktif ketika relay Y11 dan *limit switch I_STPR_UP* aktif yaitu ketika *Stopper cylinder* telah berada di atas selama 3 sekon.

3.8.10 Relay Y10

Bila dilihat dari *Switching function* relay Y10 akan didapatkan beberapa persamaan *Boolean* relay.

Relay Y10

$$\text{Starting} * [(Y9 * \text{Workpoint}) + Y10] * \overline{Y11} = Y10 \quad (3.33)$$

Output relay Y10

$$\text{Starting} * Y10 = \text{DrillDown} \quad (3.34)$$

$$\text{Starting} * Y10 * I_Drill_D = \text{DrillOn} \quad (3.35)$$

$$\text{Starting} * Y10 * I_Drill_D * (\text{TON t\#1.5s}) = \text{Tim_DOff} \quad (3.36)$$

Relay Y10 aktif ketika benda kerja yang dijalankan oleh konveyor kedua terdeteksi pada sensor *Workpoint* dan direset saat relay Y11 aktif. Relay Y10 mereset relay Y9 yang merupakan relay penggerak konveyor kedua sehingga *Conveyor2* mati. Kemudian *DrillDown* memerintahkan *Drill cylinder* bergerak ke bawah dan mengaktifkan proses *drilling* bernama *DrillOn* selama 3 sekon. Durasi ini diatur oleh timer *Tim_DOff* yang apabila aktif maka akan mengaktifkan relay Y11. Timer *Tim_DOff* sendiri dipasang paralel dengan *DrillOn*.

3.8.11 Relay Y11

Bila dilihat dari *Switching function* relay Y11 akan didapatkan beberapa persamaan *Boolean* relay.

Relay Y11

$$\text{Starting} * [(Y10 * Tim_DOff) + Y11] * \overline{Y12} = Y11 \quad (3.37)$$

Output relay Y11

$$\text{Starting} * Y11 = DrillUp \quad (3.38)$$

$$\text{Starting} * Y11 * I_Drill_U = StopperUp \quad (3.39)$$

$$\text{Starting} * Y11 * I_Drill_U * (TON\ t\#3s) = Tim_STPR \quad (3.40)$$

Relay Y11 aktif ketika perhitungan timer *Tim_DOff* telah selesai dan direset ketika relay Y12 aktif. Ketika relay Y11 telah aktif maka aliran *DrillOn* yang merupakan output relay Y10 akan diputus sehingga motor *drilling* mati. Kemudian relay Y11 akan mengaktifkan *DrillUp* yang menggerakkan *Drill cylinder* kembali ke atas. Selanjutnya *limit switch I_Drill_U* yang aktif akan memerintahkan *Stopper cylinder* naik ke atas atau *StopperUp* aktif selama 3 sekon. Durasi lamanya *StopperUp* aktif diatur oleh timer *Tim_STPR*. Relay Y11 juga mengaktifkan kembali relay Y9 agar *Conveyor2* kembali berjalan melewati *Stopper cylinder*.

3.8.12 Relay Y12

Bila dilihat dari *Switching function* relay Y12 akan didapatkan beberapa persamaan *Boolean* relay.

Relay Y12

$$\text{Starting} * [(Y11 * Tim_STPR) + Y12] * \overline{R_Y12} = Y12 \quad (3.41)$$

$$\text{Starting} * (Y13 + Y17 + R_Y12) = R_Y12 \quad (3.42)$$

Relay Y12 merupakan relay pengembali *Stopper cylinder* ke bawah. Relay Y12 aktif ketika perhitungan timer *Tim_STPR* selesai dan direset oleh *R_Y12*. Relay *R_Y12* adalah relay *latching* yang memiliki input relay Y13 dan relay Y17 dengan logika OR (lihat persamaan 3.42). Relay Y12 mereset relay Y11 yang memiliki output *StopperUp* sehingga *Stopper cylinder* bergerak ke bawah kembali ke posisi semula.

3.8.13 Relay Y13

Bila dilihat dari *Switching function* relay Y13 akan didapatkan beberapa persamaan *Boolean* relay.

Relay Y13

$$\text{Starting} * [(Y12 * \text{Black}) + Y13] * \overline{Y14} = Y13 \quad (3.43)$$

Output relay Y13

$$\text{Starting} * Y13 = \text{Finger}_D \quad (3.44)$$

$$\text{Starting} * [(Y13 * I_Finger_D) + \text{FingerGrip}] \quad (3.45)$$

$$* \overline{\text{FingerOpen}} = \text{FingerGrip}$$

Relay Y13-Y16 merupakan relay proses penempatan benda kerja plastik hitam dengan melanjutkan proses dari relay Y12. Relay Y13 aktif ketika benda kerja terdeteksi di sensor *End2* dan diidentifikasi sebagai benda kerja plastik hitam dimana relay *Black* aktif. Relay Y13 direset ketika relay Y14 aktif. Kemudian relay Y13 akan memberikan perintah output *Finger_D* atau menggerakkan *Finger vertical cylinder* ke bawah. Setelah mengenai *limit switch I_Finger_D*, output selanjutnya yang diperintahkan adalah *FingerGrip* atau gerakan finger menggenggam benda kerja. Pada persamaan *Boolean* relay 3.45 output *FingerGrip* menggunakan relay *latching* karena relay Y13 akan diputus apabila relay Y14 aktif sedangkan *FingerGrip* masih dibutuhkan sampai relay Y16.

3.8.14 Relay Y14

Bila dilihat dari *Switching function* relay Y14 akan didapatkan beberapa persamaan *Boolean* relay.

Relay Y14

$$\text{Starting} * [(Y13 * I_Finger_G) + Y14] * \overline{Y15} = Y14 \quad (3.46)$$

Output relay Y14

$$\text{Starting} * Y14 = \text{FingerUp} \quad (3.47)$$

$$\text{Starting} * Y14 * I_Finger_Up = \text{Forward} \quad (3.48)$$

Relay Y14 aktif ketika *limit switch I_Finger_G* aktif dan melakukan output *FingerUp*. Ketika *limit switch I_Finger_Up* tersentuh maka relay Y14 akan memerintahkan output *Forward* atau gerakan maju *Finger horizontal cylinder*. Terakhir relay Y14 direset ketika relay Y15 aktif.

3.8.15 Relay Y15

Bila dilihat dari *Switching function* relay Y15 akan didapatkan beberapa persamaan *Boolean* relay.

Relay Y15

$$\textit{Starting} * [(Y14 * I_Forward) + Y15] * \overline{Y16} = Y15 \quad (3.49)$$

Output relay Y15

$$\textit{Starting} * Y15 = \textit{FingerDown} \quad (3.50)$$

$$\textit{Starting} * Y15 * I_Finger_D = \textit{FingerOpen} \quad (3.51)$$

Relay Y15 aktif setelah *Finger horizontal cylinder* menyentuh *limit switch I_Forward*, kemudian mengaktifkan output *FingerDown*. Selanjutnya setelah *limit switch I_Finger_D* disentuh, relay Y15 akan menghasilkan output *FingerOpen*. Terakhir relay Y15 direset setelah relay Y16 aktif.

3.8.16 Relay Y16

Bila dilihat dari *Switching function* relay Y16 akan didapatkan beberapa persamaan *Boolean* relay.

Relay Y16

$$\textit{Starting} * [(Y15 * \textit{FingerOpen}) + Y16] \quad (3.52)$$

$$* \overline{I_Backward} = R16$$

Output relay Y16

$$\textit{Starting} * Y16 = \textit{FingerUp} \quad (3.53)$$

$$\textit{Starting} * Y16 * I_Forward = \textit{Backward} \quad (3.54)$$

Relay Y16 aktif ketika relay *FingerOpen* aktif dan direset ketika *limit switch I_Backward* disentuh. Relay Y16 akan memberikan perintah *FingerUp* atau gerakan *Finger vertical cylinder* ke atas. Setelah *limit switch I_Forward* disentuh maka relay Y16 akan memerintahkan output *Backward* atau gerakan *Finger vertical cylinder* ke bawah.

3.8.17 Relay Y17

Bila dilihat dari *Switching function* relay Y17 akan didapatkan beberapa persamaan *Boolean* relay.

Relay Y17

$$\textit{Starting} * [(Y12 * \textit{Blue}) + Y17] * \overline{Y18} = Y17 \quad (3.55)$$

Output relay Y17

$$\mathbf{Starting * Y17 = Finger_D} \quad (3.56)$$

$$\mathbf{Starting * [(Y17 * I_Finger_D) + FingerGrip]} \quad (3.57)$$

$$\mathbf{* FingerOPen = FingerGrip}$$

Relay Y17-Y19 merupakan relay proses benda kerja plastik biru dengan melanjutkan proses dari relay Y12. Relay Y17 aktif ketika sensor *End2* mendeteksi benda kerja yang teridentifikasi sebagai benda kerja plastik biru atau relay *Blue* aktif. Relay Y17 akan direset setelah relay Y18 aktif. Output dari relay Y17 sama dengan relay Y13. Walaupun outputnya sama namun tidak akan dijadikan satu relay karena akan lebih memudahkan PLC dalam pembacaan daripada membaginya menjadi dua relay dan dalam prakteknya sendiri menjadi lebih cepat. Output dari relay Y17 yaitu *Finger_D* atau gerakan *Finger vertical cylinder* ke bawah dan *FingerGrip* atau gerakan *finger* menggenggam. Pada persamaan *Boolean* relay 3.57 output *FingerGrip* menggunakan relay *latching* karena relay Y17 akan diputus apabila relay Y18 aktif sedangkan *FingerGrip* masih dibutuhkan pada relay Y18.

3.8.18 Relay Y18

Bila dilihat dari *Switching function* relay Y18 akan didapatkan beberapa persamaan *Boolean* relay.

Relay Y18

$$\mathbf{Starting * [(Y17 * I_Finger_G) + Y18] * \overline{Y19} = Y18} \quad (3.58)$$

Output relay Y18

$$\mathbf{Starting * Y18 = FingerUp} \quad (3.59)$$

$$\mathbf{Starting * Y18 * I_Finger_Up = Forward} \quad (3.60)$$

$$\mathbf{Starting * Y18 * I_Finger_Up * (TON t\#0.5s)} \quad (3.61)$$

$$\mathbf{= Tim_GOp}$$

$$\mathbf{Starting * Y18 * Tim_GOp = FingerOpen} \quad (3.62)$$

Relay Y18 aktif ketika *limit switch I_Finger_G* disentuh dan direset jika relay Y19 aktif. Relay Y18 mengaktifkan output *FingerUp* dimana *Finger vertical cylinder* bergerak ke atas. Setelah *limit switch I_Finger_Up* ditekan maka akan mengeluarkan output *Forward* atau gerakan *Finger horizontal cylinder* bergerak maju. Sementara itu, aktifnya output *Forward* juga menghidupkan timer *Tim_GOp* yang bekerja sebagai penghitung aktifnya relay *FingerOpen* untuk memutus

aliran output *FingerGrip*. Durasi yang digunakan pada timer *Tim_GOp* adalah 0.5 sekon.

3.8.19 Relay Y19

Bila dilihat dari *Switching function* relay Y19 akan didapatkan beberapa persamaan *Boolean* relay.

Relay Y19

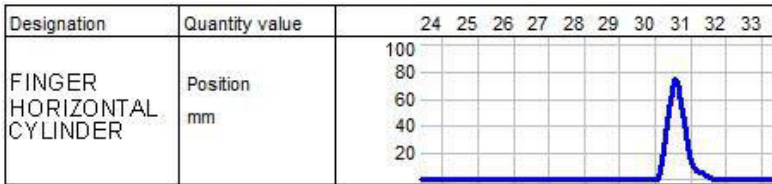
$$\mathbf{Starting * [(Y18 * FingerOpen) + Y19]} \quad (3.63)$$

$$\mathbf{* \overline{I_Backward} = Y19}$$

Output relay Y19

$$\mathbf{Starting * Y19 * I_Forward = Backward} \quad (3.64)$$

Relay Y19 aktif ketika relay *FingerOpen* aktif. Relay Y19 akan direset apabila *Finger horizontal cylinder* menyentuh *limit switch I_Backward*. Relay Y19 hanya memiliki satu output saja yaitu output *Backward* atau gerakan *Finger horizontal cylinder* mundur ke posisi semula. Walaupun relay Y19 sudah aktif ketika relay *FingerOpen* aktif, relay Y19 baru akan menjalankan outputnya ketika *limit switch I_Forward* aktif. Pada persamaan *Boolean* relay 3.64 penggunaan *limit switch I_Forward* pada perintah *Backward* dilakukan untuk pengamanan *solenoid valve* pada *Finger horizontal cylinder* dimana relay Y19 aktif setelah aktifnya *FingerOpen* namun masih melakukan perintah *Forward*. Untuk lebih jelasnya diberikan simulasi sinyal output pada **Gambar 3.10** yang dilakukan pada *Fluidsim*. Maka alangkah baiknya perintah *Backward* dilakukan setelah perintah *Forward* telah selesai dikerjakan atau ketika *limit switch I_Forward* aktif. Cara ini diharapkan dapat mengurangi kerusakan pada *solenoid valve* ketika kedua penekan elektrik ditekan bersamaan dan dilakukan secara terus-menerus serta pada jangka waktu yang panjang.



Gambar 3.10 Sinyal output *finger horizontal cylinder* tidak mencapai titik maksimal

3.9 Addressing Benda Kerja

Pada judul tugas akhir ini digunakan banyak objek pada proses *Factory Automatic Trainer* sehingga sangat penting untuk membahas *addressing* benda kerja. Bahasan yang paling penting disini adalah *addressing* benda kerja plastik hitam dan biru dimana material metal abu-abu akan dibuang ketika proses *ejecting* dan tidak akan berada di seluruh modul. Subjudul ini akan membahas beberapa hal yaitu, *counting* benda kerja, *pairing address* benda kerja, *locking address* benda kerja dan pemutusan aliran *address* benda kerja.

3.9.1 Counting Benda Kerja

Proses pertama *addressing* benda kerja adalah menghitung benda kerja plastik hitam dan biru. Dibutuhkan dua input dengan hubungan logika AND yaitu relay NC *Prox* dan timer *Tim_Cap*. Relay NC *Prox* memfilter *address* dari sensor *Prox_S* yang tidak terdeteksi pada benda kerja plastik hitam dan biru namun hanya terdeteksi pada benda kerja metal abu-abu. Relay NC *Prox* dihubungkan AND dengan timer *Tim_Cap* memberikan sinyal bahwa benda kerja plastik hitam dan biru terdeteksi. Timer *Tim_Cap* merupakan TON dari sensor *Cap_S* dengan waktu aktif selama 1 sekon. Penggunaan timer *Tim_Cap* ditentukan karena deteksi sensor *Cap_S* lebih cepat beberapa milisekon dibandingkan deteksi sensor *Prox_S* sehingga hasil akhir yang didapatkan adalah benda kerja terdeteksi oleh sensor *Cap_S* saja. Maka deteksi sensor *Cap_S* harus didelay selama 1 sekon oleh timer *Tim_Cap*. Kedua input dihubungkan dengan 6 Counter UP (CTU). Setiap CTU memberikan *Preset Value* (PV) berbeda-beda yaitu dari 1-6 dan mengeluarkan output *address* berupa urutan benda kerja berdasarkan PV

yang telah dihitung yaitu output BK1-BK5 dan output VI. Selanjutnya output VI dirangkai *safe holding* sehingga menghasilkan relay baru yaitu BK6 serta dihubungkan secara OR dengan timer on *Reset* selama 1 sekon. *Reset* juga diaktifkan ketika relay *Stopping* aktif. Fungsi dari *Reset* sendiri adalah mereset hitungan 6 CTU dan memtus aliran relay BK6 agar hitungan dapat dimulai kembali. Berikut adalah persamaan *Boolean* relay dari proses *counting* benda kerja.

$$\overline{Starting} * \overline{Prox} * \overline{Tim_Cap} * (CTU == 1) * \overline{BK2} * \overline{BK3} * \overline{BK4} * \overline{BK5} * \overline{BK6} = BK1 \quad (3.65)$$

$$\overline{Starting} * \overline{Prox} * \overline{Tim_Cap} * (CTU == 2) * \overline{BK3} * \overline{BK4} * \overline{BK5} * \overline{BK6} = BK2 \quad (3.66)$$

$$\overline{Starting} * \overline{Prox} * \overline{Tim_Cap} * (CTU == 3) * \overline{BK4} * \overline{BK5} * \overline{BK6} = BK3 \quad (3.67)$$

$$\overline{Starting} * \overline{Prox} * \overline{Tim_Cap} * (CTU == 4) * \overline{BK5} * \overline{BK6} = BK4 \quad (3.68)$$

$$\overline{Starting} * \overline{Prox} * \overline{Tim_Cap} * (CTU == 5) * \overline{BK6} = BK5 \quad (3.69)$$

$$\overline{Starting} * \overline{Prox} * \overline{Tim_Cap} * (CTU == 5) = VI \quad (3.70)$$

$$\overline{Starting} * (VI + BK6) * \overline{Reset} = BK6 \quad (3.71)$$

$$\overline{Starting} * VI * (TON t\#1s) + \overline{Stopping} = \overline{Reset} \quad (3.72)$$

3.9.2 *Pairing Address* Benda Kerja

Proses kedua dari *addressing* benda kerja adalah *pairing address* benda kerja. Proses *pairing address* benda kerja adalah memasangkan output *address* urutan benda kerja BK1-BK6 dengan deteksi benda kerja plastik hitam dan biru yang kemudian dirangkai *safe holding*. Deteksi benda kerja plastik biru menggunakan input logika AND relay NC *Photo* dan relay M1-M6 (deteksi benda kerja metal) dengan hasil berupa output *address* benda kerja plastik yaitu relay B1-B6. Sedangkan deteksi benda kerja plastik hitam dilakukan dengan input logika AND relay NO *Photo* dan relay M1-M6 menghasilkan output *address* benda kerja plastik biru yaitu relay BC1-BC6. Relay M1-M6 juga merupakan proses

pasangan dari output BK1-BK6 dengan input logika AND relay *Cap*, *Photo* dan *Proxy* yang dirangkai *safe holding*.

Aliran relay B1-B6 diputus oleh relay *Blue1-Blue6* yang aktif ketika benda kerja plastik biru mengenai sensor *End2* serta relay BC1-BC6. Pemutusan oleh relay BC1-BC6 berfungsi untuk menstabilkan aliran relay BC1-BC6 yang aktif ketika benda kerja hitam terdeteksi terlebih dahulu kemudian benda kerja yang terdeteksi selanjutnya adalah metal abu-abu. Relay BC1-BC6 juga akan diputus alirannya oleh relay *Black1-Black6* yang aktif ketika benda kerja biru menyentuh sensor *End2* serta relay B1-B6. Pemutusan yang dilakukan oleh relay B1-B6 berguna untuk mengunci aliran relay B1-B6 yang aktif apabila nantinya benda kerja yang terdeteksi adalah benda kerja plastik biru dimana relay NO *Photo* terlambat aktif dari relay BC1-BC6. Pada kasus pemasangan relay M1-M6 pada relay B1-B6 dan relay BC1-BC6 bertujuan untuk menghindari sinyal gangguan yang berasal dari *address* benda kerja metal abu-abu saat benda kerja tersebut berada pada posisi kedua dibelakang benda kerja plastik hitam maupun biru. Pemutusan relay M1-M6 dilakukan oleh *Reset* benda kerja saat PVnya bernilai 6. Berikut adalah persamaan *Boolean* relay dari proses *pairing address* benda kerja.

$$\text{Starting} * [(BK1 * Cap * Photo * Prox) + M1] \quad (3.73)$$

$$* \overline{Reset} = M1$$

$$\text{Starting} * [(BK2 * Cap * Photo * Prox) + M2] \quad (3.74)$$

$$* \overline{Reset} = M2$$

$$\text{Starting} * [(BK3 * Cap * Photo * Prox) + M3] \quad (3.75)$$

$$* \overline{Reset} = M3$$

$$\text{Starting} * [(BK4 * Cap * Photo * Prox) + M4] \quad (3.76)$$

$$* \overline{Reset} = M4$$

$$\text{Starting} * [(BK5 * Cap * Phot * Prox) + M5] \quad (3.77)$$

$$* \overline{Reset} = M5$$

$$\text{Starting} * [(BK6 * Cap * Photo * Prox) + M6] \quad (3.78)$$

$$* \overline{Reset} = M6$$

$$\text{Starting} * [(BK1 * Photo * \overline{M1}) + B1] * \overline{Blue1} * \overline{BC1} \quad (3.79)$$

$$= B1$$

$$\text{Starting} * [(BK1 * \overline{Photo} * \overline{M1}) + BC1] * \overline{Black1} \quad (3.80)$$

$$\begin{aligned}
& * \overline{B1} = BC1 \\
& \textit{Starting} * [(BK2 * \overline{Photo} * \overline{M2}) + B2] * \overline{Blue2} * \overline{BC2} \quad (3.81) \\
& = B2
\end{aligned}$$

$$\begin{aligned}
& \textit{Starting} * [(BK2 * \overline{Photo} * \overline{M2}) + BC2] * \overline{Black2} \quad (3.82) \\
& * \overline{B2} = BC2
\end{aligned}$$

$$\begin{aligned}
& \textit{Starting} * [(BK3 * \overline{Photo} * \overline{M3}) + B3] * \overline{Blue3} * \overline{BC3} \quad (3.83) \\
& = B3
\end{aligned}$$

$$\begin{aligned}
& \textit{Starting} * [(BK3 * \overline{Photo} * \overline{M3}) + BC3] * \overline{Black3} \quad (3.84) \\
& * \overline{B3} = BC3
\end{aligned}$$

$$\begin{aligned}
& \textit{Starting} * [(BK4 * \overline{Photo} * \overline{M4}) + B4] * \overline{Blue4} * \overline{BC4} \quad (3.85) \\
& = B4
\end{aligned}$$

$$\begin{aligned}
& \textit{Starting} * [(BK4 * \overline{Photo} * \overline{M4}) + BC4] * \overline{Black4} \quad (3.86) \\
& * \overline{B4} = BC4
\end{aligned}$$

$$\begin{aligned}
& \textit{Starting} * [(BK5 * \overline{Photo} * \overline{M5}) + B5] * \overline{Blue5} * \overline{BC5} \quad (3.87) \\
& = B5
\end{aligned}$$

$$\begin{aligned}
& \textit{Starting} * [(BK5 * \overline{Photo} * \overline{M5}) + BC5] * \overline{Black5} \quad (3.88) \\
& * \overline{B5} = BC5
\end{aligned}$$

$$\begin{aligned}
& \textit{Starting} * [(BK6 * \overline{Photo} * \overline{M6}) + B6] * \overline{Blue6} * \overline{BC6} \quad (3.89) \\
& = B6
\end{aligned}$$

$$\begin{aligned}
& \textit{Starting} * [(BK6 * \overline{Photo} * \overline{M6}) + BC6] * \overline{Black6} \quad (3.90) \\
& * \overline{B6} = BC6
\end{aligned}$$

3.9.3 Locking Address Benda Kerja

Setelah proses *pairing address* benda kerja selesai, dilakukan proses *locking address* benda kerja. *Locking address* benda kerja adalah proses mengunci *address* benda kerja dari relay BC1-BC6 dan relay B1-B6 menjadi relay *address* baru *Black1-Black6* dan *Blue1-Blue6* yang akan dieksekusi pada relay Y13 dan relay Y17 melalui relay *Black* dan relay *Blue* (lihat persamaan 3.111 dan 3.112). Proses penguncian *address* terjadi saat benda kerja hitam atau biru mengenai sensor *End2* sehingga relay *Black1-Black6* atau relay *Blue1-Blue6* yang dirangkai *safe holding* menjadi aktif. Input dari relay *Black1-Black6* dan relay *Blue1-Blue6* adalah relay BC1-BC6, relay B1-B6 dan output ED1-ED6,

dimana output ED6 merupakan relay *latching* dari output E6. Relay ED1-ED5 dan E6 adalah output PV dari hasil *counting* sensor *End2* dengan 6 CTU yang masing-masing memiliki nilai PV berbeda dari 1-6. Selain mengaktifkan relay *latching* ED6, output E6 mengaktifkan TON *Reset2* selama 1 sekon untuk mereset hitungan 6 CTU dan memutus relay *latching* ED6. *Reset2* juga diaktifkan ketika relay *Stopping* aktif. Terakhir, aliran relay *Black1-Black6* dan relay *Blue1-Blue6* akan diputus oleh relay FD1-FD6 yang akan dibahas pada proses terakhir. Berikut adalah persamaan *Boolean* pada proses *locking address* benda kerja.

$$\text{Starting} * \text{End2} * (\text{CTU} == 1) * \overline{\text{ED2}} * \overline{\text{ED3}} * \overline{\text{ED4}} * \overline{\text{ED5}} * \overline{\text{ED6}} = \text{ED1} \quad (3.91)$$

$$\text{Starting} * \text{End2} * (\text{CTU} == 2) * \overline{\text{ED3}} * \overline{\text{ED4}} * \overline{\text{ED5}} * \overline{\text{ED6}} = \text{ED2} \quad (3.92)$$

$$\text{Starting} * \text{End2} * (\text{CTU} == 3) * \overline{\text{ED4}} * \overline{\text{ED5}} * \overline{\text{ED6}} = \text{ED3} \quad (3.93)$$

$$\text{Starting} * \text{End2} * (\text{CTU} == 4) * \overline{\text{ED5}} * \overline{\text{ED6}} = \text{ED4} \quad (3.94)$$

$$\text{Starting} * \text{End2} * (\text{CTU} == 5) * \overline{\text{ED6}} = \text{ED5} \quad (3.95)$$

$$\text{Starting} * \text{End2} * (\text{CTU} == 6) = \text{E6} \quad (3.96)$$

$$\text{Starting} * (\text{E6} + \text{ED6}) * \overline{\text{Reset2}} = \text{ED6} \quad (3.97)$$

$$\text{Starting} * \text{E6} * (\text{TON t\#1s}) + \text{Stopping} = \text{Reset2} \quad (3.98)$$

$$\text{Starting} * [(\text{B1} * \text{ED1}) + \text{Blue1}] * \overline{\text{FD1}} = \text{Blue1} \quad (3.99)$$

$$\text{Starting} * [(\text{BC1} * \text{ED1}) + \text{Black1}] * \overline{\text{FD1}} = \text{Black1} \quad (3.100)$$

$$\text{Starting} * [(\text{B2} * \text{ED2}) + \text{Blue}] * \overline{\text{FD2}} = \text{Blue2} \quad (3.101)$$

$$\text{Starting} * [(\text{BC2} * \text{ED2}) + \text{Black2}] * \overline{\text{FD2}} = \text{Black2} \quad (3.102)$$

$$\text{Starting} * [(\text{B3} * \text{ED3}) + \text{Blue3}] * \overline{\text{FD3}} = \text{Blue3} \quad (3.103)$$

$$\text{Starting} * [(\text{BC3} * \text{ED3}) + \text{Black3}] * \overline{\text{FD3}} = \text{Black3} \quad (3.104)$$

$$\text{Starting} * [(\text{B4} * \text{ED4}) + \text{Blue4}] * \overline{\text{FD4}} = \text{Blue4} \quad (3.105)$$

$$\text{Starting} * [(\text{BC4} * \text{ED4}) + \text{Black4}] * \overline{\text{FD4}} = \text{Black4} \quad (3.106)$$

$$\text{Starting} * [(\text{B5} * \text{ED5}) + \text{Blue5}] * \overline{\text{FD5}} = \text{Blue5} \quad (3.107)$$

$$\text{Starting} * [(\text{BC5} * \text{ED5}) + \text{Black5}] * \overline{\text{FD5}} = \text{Black5} \quad (3.108)$$

$$\text{Starting} * [(\text{B6} * \text{ED6}) + \text{Blue6}] * \overline{\text{FD6}} = \text{Blue6} \quad (3.109)$$

$$\text{Starting} * [(\text{BC6} * \text{ED6}) + \text{Black6}] * \overline{\text{FD6}} = \text{Black6} \quad (3.110)$$

$$\text{Starting} * (\text{Blue1} + \text{Blue2} + \text{Blue3} + \text{Blue4}) \quad (3.111)$$

$$\begin{aligned}
&+Blue5 + Black6) = Blue \\
\mathbf{Starting * (Black1 + Black2 + Black3 + Black4} & \quad (3.112) \\
&+Black5 + Black6) = Black
\end{aligned}$$

3.9.4 Pemutusan Aliran *Address* Benda Kerja

Proses terakhir dari *addressing* benda kerja adalah pemutusan aliran *address* benda kerja pada relay *Black1-Black6* dan relay *Blue1-Blue6*. Pemutusan aliran *address* diperlukan untuk mereset ulang *address* sehingga dapat digunakan untuk beberapa benda kerja selanjutnya. Pada proses ini *limit switch I_Forward* dihubungkan dengan 6 CTU dimana setiap CTU memiliki nilai PV berbeda dari 1-6 dan menghasilkan output urutan benda kerja yang telah mengenai *limit switch* tersebut yaitu relay FD1-FD5 dan F6. Output F6 akan mengaktifkan relay *latching* FD6 dan TON *Reset3* selama 1 sekon. *Reset3* juga diaktifkan ketika relay *Stopping* aktif. Fungsi dari *Reset3* adalah mereset hitungan 6 CTU dan memutus aliran relay *latching* FD6. Dari output FD1-FD6 yang dihasilkan akan memutus aliran relay *Black1-Black6* dan *Blue1-Blue6*. Berikut adalah persamaan *Boolean* relay dari proses pemutusan aliran *address* benda kerja.

$$\begin{aligned}
\mathbf{Starting * I_Forward * (CTU == 1) * \overline{FD2} * \overline{FD3}} & \quad (3.111) \\
&* \overline{FD4} * \overline{FD5} * \overline{FD6} = FD1
\end{aligned}$$

$$\begin{aligned}
\mathbf{Starting * I_Forward * (CTU == 2) * \overline{FD3} * \overline{FD4}} & \quad (3.112) \\
&* \overline{FD5} * \overline{FD6} = ED2
\end{aligned}$$

$$\begin{aligned}
\mathbf{Starting * I_Forward * (CTU == 3) * \overline{FD4} * \overline{FD5}} & \quad (3.113) \\
&* \overline{FD6} = FD3
\end{aligned}$$

$$\begin{aligned}
\mathbf{Starting * I_Forard * (CTU == 4) * \overline{FD5} * \overline{FD6}} & \quad (3.114) \\
= FD4
\end{aligned}$$

$$\mathbf{Starting * I_Forward * (CTU == 5) * \overline{FD6} = FD5} \quad (3.115)$$

$$\mathbf{Starting * I_Forward * (CTU == 6) = F6} \quad (3.116)$$

$$\mathbf{Starting * (F6 + FD6) * \overline{Reset3} = FD6} \quad (3.117)$$

$$\mathbf{Starting * F6 * (TON t\#1s) + Stopping = Reset3} \quad (3.118)$$

[Halaman ini sengaja dikosongkan]

BAB 4

IMPLEMENTASI DAN ANALISIS

Setelah berhasil merancang *Switching function* relay dan persamaan *Boolean* relay, langkah selanjutnya adalah merancang diagram *ladder* berdasarkan dua komponen tersebut serta mengujinya. Tujuan dari pengujian ini adalah memastikan apakah rancangan diagram *ladder* berhasil diimplementasikan terhadap plant. Dari hasil implementasi tersebut akan dianalisis apakah diagram *ladder* berjalan sesuai dengan sistem yang diinginkan. Pada bab ini akan dibahas rancangan diagram *ladder* serta analisis keberhasilannya.

4.1 Pembuatan Diagram Ladder

Setelah selesai merancang *Switching function* dan persamaan *Boolean* relay pada bab sebelumnya, dapat dilakukan rancangan diagram *ladder* yang akan diimplementasikan pada PLC. *Software* yang digunakan oleh penulis untuk merancang diagram *ladder* adalah *GMWIN 4*. Sebelum membahas diagram *ladder*, penulis perlu untuk menjelaskan Input/Output beserta alamat yang digunakan pada topik kali ini.

4.1.1 Alamat Input/Output

Langkah pertama dalam pembuatan diagram *ladder* adalah mendaftarkan alamat input/output pada program PLC. Salah satu yang berada dalam pendaftaran alamat input/output adalah daftar input/output yang telah dicantumkan pada pabrikan *Factory Automatic Trainer*. Kemudian ditambahkan beberapa relay memori tambahan sesuai dengan yang dibutuhkan dalam perancangan sistem. Berikut adalah alamat input/output yang didaftarkan.

Tabel 4.1 Daftar alamat input/output

No	Input		Output		Memori	
	Alamat	Simbol	Alamat	Simbol	Alamat	Simbol
1.	%IX0.1.11	<i>Start</i>	%QX0.2.0	<i>Forward</i>	%MX0	<i>Starting</i>
2.	%IX0.1.12	<i>Stop</i>	%QX0.2.1	<i>Backward</i>	%MX1	Y1
3.	%IX0.0.0	<i>L_Forward</i>	%QX0.2.2	<i>FingerUp</i>	%MX2	Y2

4.	%IX0.0.1	<i>L_Backward</i>	%QX0.2.3	<i>FingerDown</i>	%MX3	Y3
5.	%IX0.0.2	<i>L_Finger_U</i>	%QX0.2.4	<i>FingerGrip</i>	%MX4	Y4
No	<i>Input</i>		<i>Output</i>		<i>Memori</i>	
	<i>Alamat</i>	<i>Simbol</i>	<i>Alamat</i>	<i>Simbol</i>	<i>Alamat</i>	<i>Simbol</i>
6.	%IX0.0.3	<i>L_Finger_D</i>	%QX0.2.6	<i>Insert</i>	%MX5	Y5
7.	%IX0.0.4	<i>FingerOpen</i>	%QX0.2.7	<i>Insert_R</i>	%MX6	Y6
8.	%IX0.0.5	<i>FingerGrip</i>	%QX0.2.8	<i>Eject</i>	%MX7	Y7
9.	%IX0.0.6	<i>Magazine</i>	%QX0.2.9	<i>Eject_R</i>	%MX8	Y8
10.	%IX0.0.7	<i>Photo_S</i>	%QX0.2.10	<i>Conveyor1</i>	%MX9	Y9
11.	%IX0.0.8	<i>Prox_S</i>	%QX0.3.0	<i>StopperUp</i>	%MX10	Y10
12.	%IX0.0.9	<i>Cap_S</i>	%QX0.3.1	<i>DrillUp</i>	%MX11	Y11
13.	%IX0.0.10	<i>L_Insert</i>	%QX0.3.2	<i>DrillDown</i>	%MX12	Y12
14.	%IX0.0.12	<i>L_Eject</i>	%QX0.3.3	<i>DrillOn</i>	%MX13	Y13
15.	%IX0.0.13	<i>L_Eject_R</i>	%QX0.3.4	<i>Conveyor2</i>	%MX14	Y14
16.	%IX0.0.14	<i>End1</i>	%QX0.3.6	<i>RotaryCCW</i>	%MX15	Y15
17.	%IX0.1.0	<i>L_STPR_U</i>	%QX0.3.7	<i>RotaryCW</i>	%MX16	Y16
18.	%IX0.1.2	<i>Workpoint</i>	%QX0.3.8	<i>Up</i>	%MX17	Y17
19.	%IX0.1.3	<i>L_Drill_U</i>	%QX0.3.9	<i>Down</i>	%MX18	Y18
20.	%IX0.1.4	<i>L_Drill_D</i>	%QX0.3.10	<i>Vacuum</i>	%MX19	Y19
21.	%IX0.1.5	<i>End2</i>	%QX0.3.11	<i>Red</i>	%MX21	<i>Tim_Ej</i>
22.	%IX0.1.6	<i>L_Rot_CCW</i>	%QX0.3.12	<i>Yellow</i>	%MX22	<i>Prox</i>
23.	%IX0.1.7	<i>L_Rot_CW</i>	%QX0.3.13	<i>Green</i>	%MX23	<i>Photo</i>
24.	%IX0.1.8	<i>L_Up</i>			%MX24	<i>Cap</i>
25.	%IX0.1.9	<i>L_Down</i>			%MX25	<i>Tim_Add</i>
26.	%IX0.1.10	<i>L_Vacuum</i>			%MX26	<i>Add</i>
27.					%MX27	<i>VacummOff</i>
28.					%MX29	<i>Tim_Doff</i>
29.					%MX30	<i>Tim_STPR</i>
30.					%MX31	<i>Tim_GOp</i>
31.					%MX32	<i>FingerOpen</i>
32.					%MX33	BK1
33.					%MX34	BK2
34.					%MX35	BK3
35.					%MX36	BK4
36.					%MX37	BK5
37.					%MX38	BK6
38.					%MX39	VI
39.					%MX40	<i>Reset</i>
40.					%MX41	B1
41.					%MX42	B2
42.					%MX43	B3
43.					%MX44	B4
44.					%MX45	B5
45.					%MX46	B6
46.					%MX47	BC1
47.					%MX48	BC2
48.					%MX49	BC3
49.					%MX50	BC4
50.					%MX51	BC5

51.					%MX52	BC6
52.					%MX53	ED1
No	Input		Output		Memori	
	Alamat	Simbol	Alamat	Simbol	Alamat	Simbol
53.					%MX54	ED2
54.					%MX55	ED3
55.					%MX56	ED4
56.					%MX57	ED5
57.					%MX58	ED6
58.					%MX59	E6
59.					%MX60	Reset2
60.					%MX61	Blue1
61.					%MX62	Blue2
62.					%MX63	Blue3
63.					%MX64	Blue4
64.					%MX65	Blue5
65.					%MX66	Blue6
66.					%MX67	Black1
67.					%MX68	Black2
68.					%MX69	Black3
69.					%MX70	Black4
70.					%MX71	Black5
71.					%MX72	Black6
72.					%MX73	FD1
73.					%MX74	FD2
74.					%MX75	FD3
75.					%MX76	FD4
76.					%MX77	FD5
77.					%MX78	FD6
78.					%MX79	F6
79.					%MX80	Reset3
80.					%MX81	Blue
81.					%MX82	Black
82.					%MX83	R_Y12
83.					%MX84	Tim_Cap
84.					%MX85	Stopping
85.					%MX86	M1
86.					%MX87	M2
87.					%MX88	M3
88.					%MX89	M4
89.					%MX90	M5
90.					%MX91	M6

4.1.2 Diagram Ladder

Dalam memudahkan penjelasan, pembuatan diagram *ladder* akan dibagi menjadi tiga bagian yaitu diagram *ladder* grup relay Y1 sampai

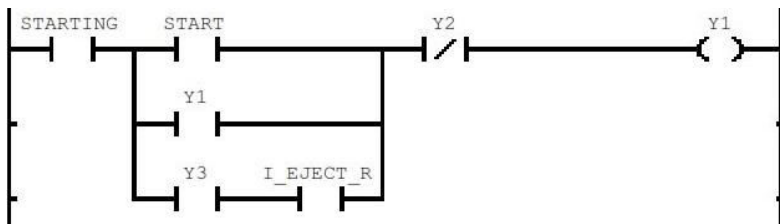
Y19, diagram *ladder* relay output aktuator yang dihasilkan dan diagram *ladder* relay *addressing* benda kerja.

4.1.2.1 Diagram Ladder Grup Relay

Relay pertama adalah Y1 yang memiliki persamaan sebagai berikut.

Urutan dan grup		
Y1	Y2	Y3
Starting, Conveyor1, Insert	Insert_R, Eject	Eject_R
Relay Y1		
$Y1 = \text{Starting} * [\text{Start} + Y1 + (Y3 * I_Eject_R)] * \overline{Y2}$		

Dari persamaan tersebut maka diagram *ladder* yang dibentuk sebagai berikut.

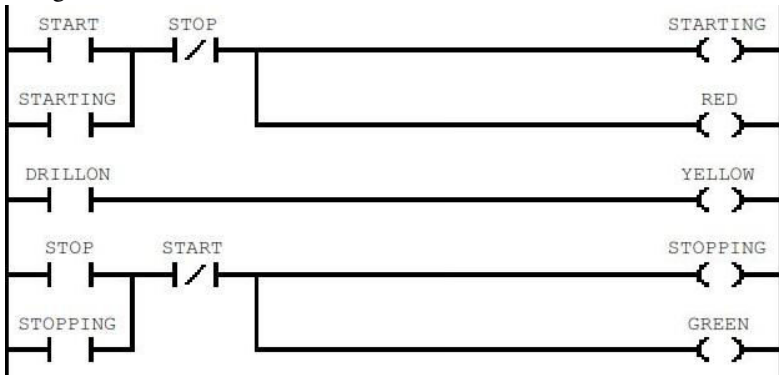


Gambar 4.1 Relay Y1

Sebelum mengaktifkan relay Y1 diperlukan suplai aliran untuk keseluruhan sistem yaitu relay *Starting* dan diputus oleh tombol tekan *Stop*. Selain itu kondisi sistem juga akan diperlihatkan melalui output 3-*Colored Sign Tower*.

Relay Starting
$\text{Starting} = (\text{Start} + \text{Starting}) * \overline{\text{Stop}}$
Relay Stopping
$\text{Stopping} = (\text{Stop} + \text{Stopping}) * \overline{\text{Start}}$
Output 3-Colored Sign Tower
$\text{Red} = (\text{Start} + \text{Starting}) * \overline{\text{Stop}}$
$\text{Yellow} = \text{DrillOn}$
$\text{Green} = (\text{Stop} + \text{Stopping}) * \overline{\text{Start}}$

Dari beberapa persamaan diatas, diagram *ladder* yang dibentuk sebagai berikut.

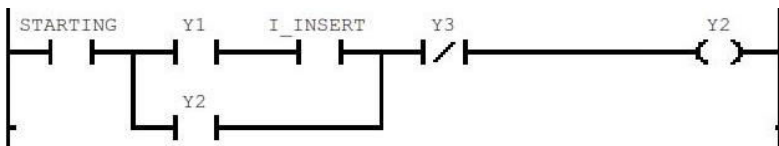


Gambar 4.2 Relay *Starting*, *Stopping* dan output *3-Colored Sign Tower*

Relay kedua adalah Y2 yang memiliki persamaan sebagai berikut.

Urutan dan grup						
<table style="width: 100%; border: none;"> <tr> <td style="text-align: center; color: blue; font-weight: bold;">Y1</td> <td style="text-align: center; color: purple; font-weight: bold;">Y2</td> <td style="text-align: center; color: blue; font-weight: bold;">Y3</td> </tr> <tr> <td style="text-align: center; border: 1px solid blue; border-radius: 10px; padding: 2px;">Starting, Conveyor1, Insert</td> <td style="text-align: center; border: 1px solid red; border-radius: 10px; padding: 2px;">Insert_R, Eject</td> <td style="text-align: center; border: 1px solid red; border-radius: 10px; padding: 2px;">Eject_R</td> </tr> </table>	Y1	Y2	Y3	Starting, Conveyor1, Insert	Insert_R, Eject	Eject_R
Y1	Y2	Y3				
Starting, Conveyor1, Insert	Insert_R, Eject	Eject_R				
Relay Y2						
$Y2 = Starting * [(Y1 * I_Insert) + Y2] * \overline{Y3}$						

Dari persamaan tersebut maka diagram *ladder* yang dibentuk sebagai berikut.



Gambar 4.3 Relay Y2

Pada relay Y2 terdapat area deteksi sensor yang digunakan untuk *addressing* benda kerja dan penambahan benda kerja. Berikut adalah persamaan dari ketiga sensor tersebut.

Deteksi 3 sensor

$$Prox = Starting * (Prox_S + Prox) * \overline{I_Insert}$$

$$Photo = Starting * (Photo_S + Photo) * \overline{I_Insert}$$

$$Cap = Starting * (Cap_S + Cap) * \overline{I_Insert}$$

Penambahan benda kerja

$$Tim_Add = Starting * (Cap_S + Cap) * \overline{I_Insert}$$

* (TON t#4s)

Addressing benda kerja

$$Tim_Cap = Starting * (Cap_S + Cap) * \overline{I_Insert}$$

* (TON t#1s)

Dari persamaan tersebut maka diagram *ladder* yang dibentuk seperti pada **Gambar 4.5**.

Relay ketiga adalah Y3 yang memiliki persamaan sebagai berikut.

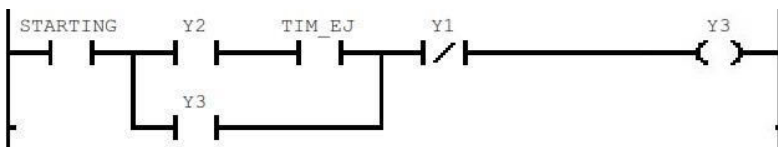
Urutan dan grup



Relay Y3

$$Y3 = Starting * [(Y2 * Tim_Ej) + Y3] * \overline{Y1}$$

Dari persamaan tersebut maka diagram *ladder* yang dibentuk sebagai berikut.



Gambar 4.4 Relay Y3

Relay keempat adalah Y4 yang memiliki persamaan sebagai berikut.

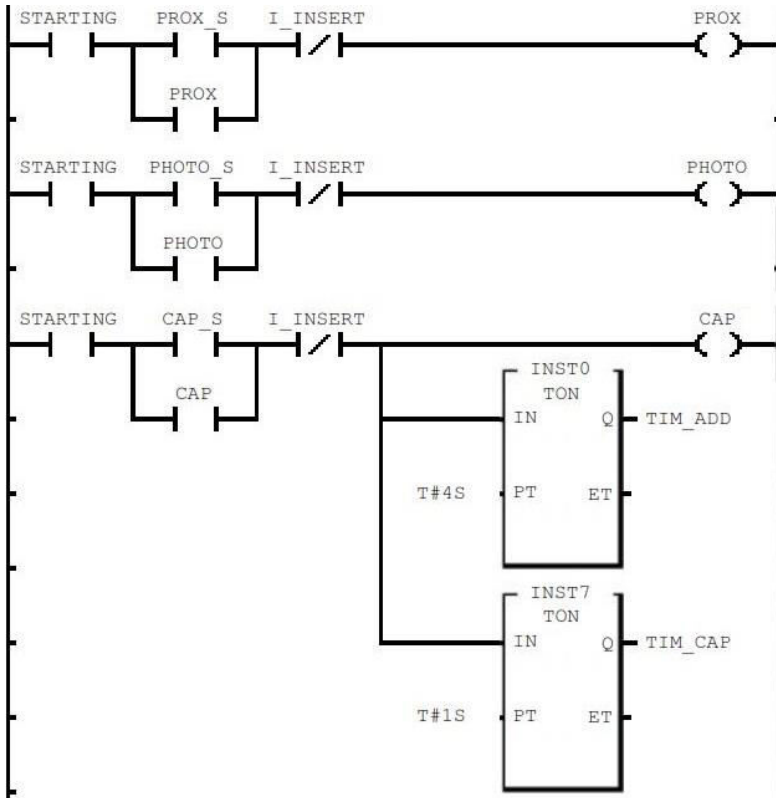
Urutan dan grup



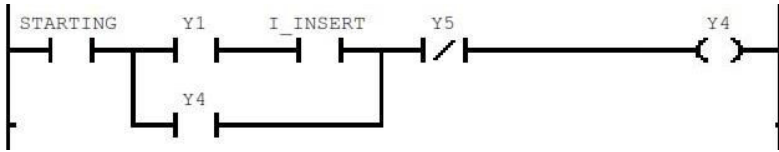
Relay Y4

$$Y4 = Starting * [(Y1 * I_Insert) + Y4] * \overline{Y5}$$

Dari persamaan tersebut maka diagram *ladder* yang dibentuk seperti pada **Gambar 4.6**.



Gambar 4.5 Relay deteksi sensor, timer *Tim_Add* dan timer *Tim_Cap*



Gambar 4.6 Relay Y4

Relay kelima adalah Y5 yang memiliki persamaan sebagai berikut.

Urutan dan grup

Y4
Insert_R
Up, RotCW

Y5
Down, Vacuum

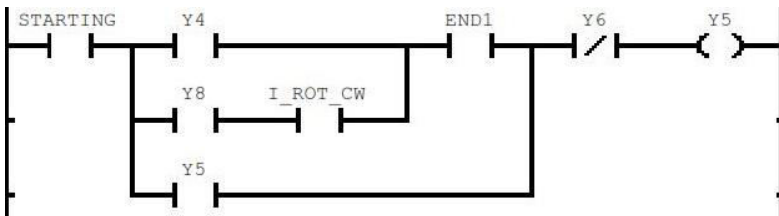
Y6
Up, RotCCW

Y8

Relay Y5

$$Y5 = Starting * \{ \{ [Y4 + (Y8 * I_{Rot_CW})] * End1 \} + Y5 \} * \overline{Y6}$$

Dari persamaan tersebut maka diagram *ladder* yang dibentuk sebagai berikut.



Gambar 4.7 Relay Y5

Relay keenam adalah Y6 yang memiliki persamaan sebagai berikut.

Urutan dan grup

Y5
Down, Vacuum

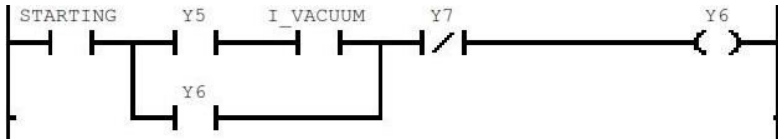
Y6
Up, RotCCW

Y7
Down, VacuumOff

Relay Y6

$$Y6 = Starting * [(Y5 * I_{Vacuum}) + Y6] * \overline{Y7}$$

Dari persamaan tersebut maka diagram *ladder* yang dibentuk sebagai berikut.



Gambar 4.8 Relay Y6

Relay ketujuh adalah Y7 yang memiliki persamaan sebagai berikut.

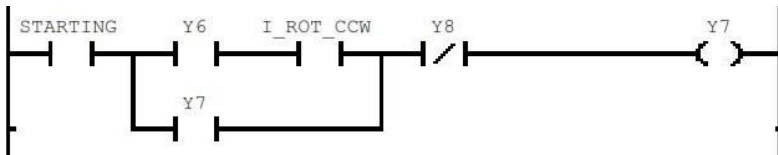
Urutan dan grup

Y6	Y7	Y8
Up, RotCCW	Down, VacuumOff	Up, RotCW

Relay Y7

$$Y7 = Starting * [(Y6 * I_Rot_CCW) + Y7] * \overline{Y8}$$

Dari persamaan tersebut maka diagram *ladder* yang dibentuk sebagai berikut.



Gambar 4.9 Relay Y7

Relay kedelapan adalah Y8 yang memiliki persamaan sebagai berikut.

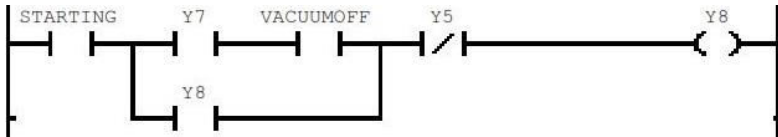
Urutan dan grup

Y7	Y8	Y5
Down, VacuumOff	Up, RotCW	Down, Vacuum

Relay Y8

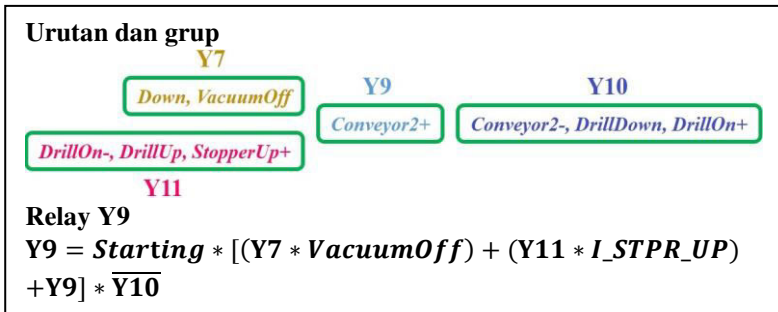
$$Y8 = Starting * [(Y7 * VacuumOff) + Y8] * \overline{Y5}$$

Dari persamaan tersebut maka diagram *ladder* yang dibentuk sebagai berikut.

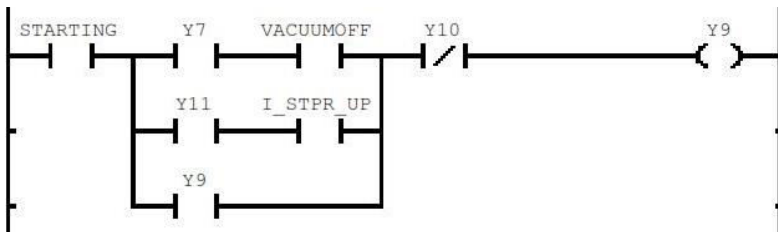


Gambar 4.10 Relay Y8

Relay kesembilan adalah Y9 yang memiliki persamaan sebagai berikut.

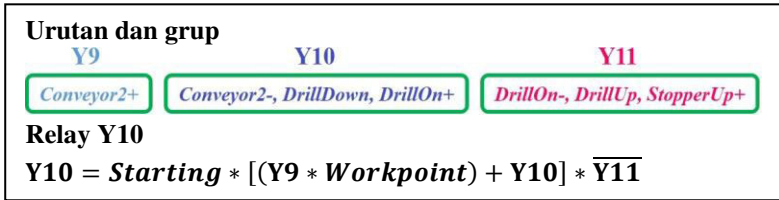


Dari persamaan tersebut maka diagram *ladder* yang dibentuk sebagai berikut.

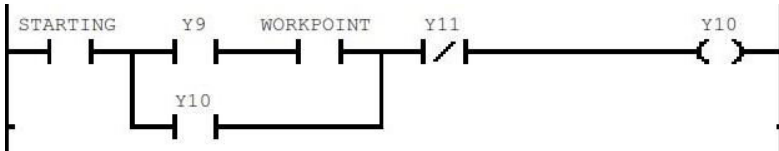


Gambar 4.11 Relay Y9

Relay kesepuluh adalah Y10 yang memiliki persamaan sebagai berikut.



Dari persamaan tersebut maka diagram *ladder* yang dibentuk sebagai berikut.

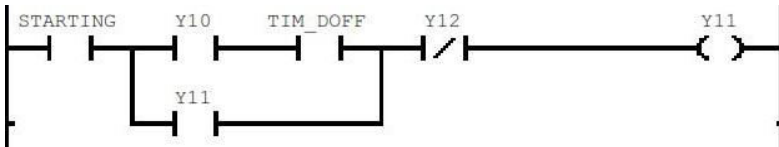


Gambar 4.12 Relay Y10

Relay kesebelas adalah Y11 yang memiliki persamaan sebagai berikut.



Dari persamaan tersebut maka diagram *ladder* yang dibentuk sebagai berikut.



Gambar 4.13 Relay Y11

Relay kedua belas adalah Y12 yang memiliki persamaan sebagai berikut.

Urutan dan grup

Y11: DrillOn-, DrillUp, StopperUp+

Y12: StopperUp-

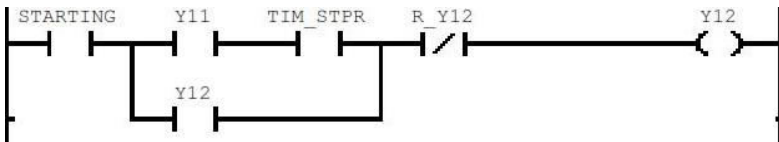
Y13: FingerDown, FingerGrip

Y17: FingerDown, FingerGrip

Relay Y12

$$Y12 = \text{Starting} * [(Y11 * \text{Tim_STPR}) + Y12] * \overline{R_Y12}$$

Dari persamaan tersebut maka diagram *ladder* yang dibentuk sebagai berikut.



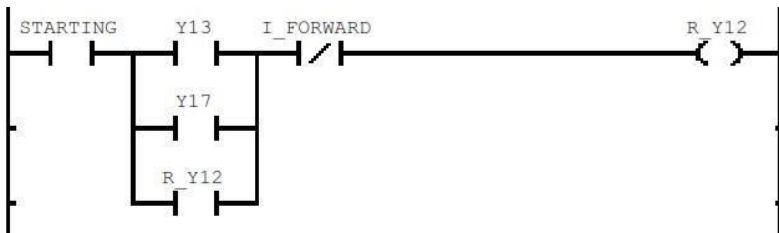
Gambar 4.14 Relay Y12

Reset relay Y12 adalah relay R_Y12 yang memiliki persamaan.

Relay R_Y12

$$R_Y12 = \text{Starting} * (Y13 + Y17 + R_Y12) * \overline{I_Forward}$$

Dari persamaan tersebut dibentuk diagram *ladder* sebagai berikut.



Gambar 4.15 Relay R_Y12

Relay ketigabelas adalah Y13 yang memiliki persamaan sebagai berikut.

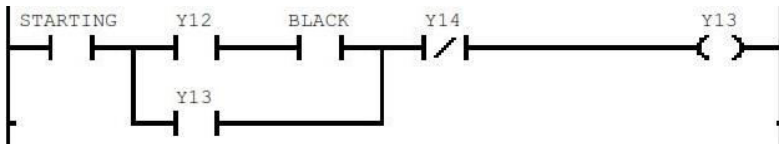
Urutan dan grup

Y12	Y13	Y14
StopperUp-	FingerDown, FingerGrip	FingerUp, Forward

Relay Y13

$$Y13 = \text{Starting} * [(Y12 * \text{Black}) + Y13] * \overline{Y14}$$

Dari persamaan tersebut maka diagram *ladder* yang dibentuk sebagai berikut.



Gambar 4.16 Relay Y13

Relay keempatbelas adalah Y14 yang memiliki persamaan sebagai berikut.

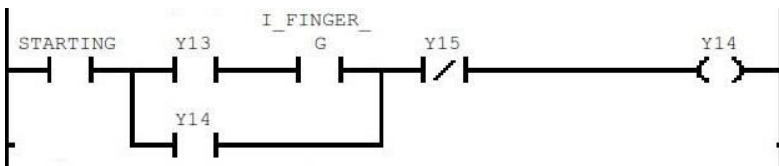
Urutan dan grup

Y13	Y14	Y15
FingerDown, FingerGrip	FingerUp, Forward	FingerDown, FingerOpen

Relay Y14

$$Y14 = \text{Starting} * [(Y13 * I_Finger_G) + Y14] * \overline{Y15}$$

Dari persamaan tersebut maka diagram *ladder* yang dibentuk sebagai berikut.



Gambar 4.17 Relay Y14

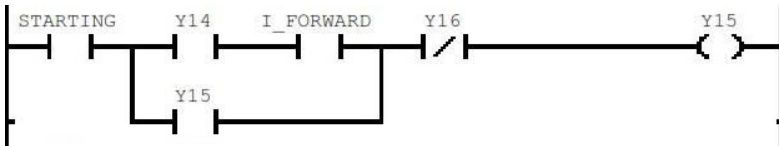
Relay kelimabelas adalah Y15 yang memiliki persamaan sebagai berikut.

Urutan dan grup Y14	Y15	Y16
<i>FingerUp, Forward</i>	<i>FingerDown, FingerOpen</i>	<i>FingerUp, Backward</i>

Relay Y15

$$Y15 = \textit{Starting} * [(Y14 * I_Forward) + Y15] * \overline{Y16}$$

Dari persamaan tersebut maka diagram *ladder* yang dibentuk sebagai berikut.



Gambar 4.18 Relay Y15

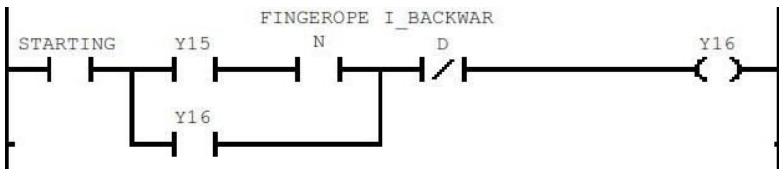
Relay keenambelas adalah Y16 yang memiliki persamaan sebagai berikut.

Urutan dan grup Y15	Y16
<i>FingerDown, FingerOpen</i>	<i>FingerUp, Backward</i>

Relay Y16

$$Y16 = \textit{Starting} * [(Y15 * \textit{FingerOpen}) + Y16] * \overline{\textit{Backward}}$$

Dari persamaan tersebut maka diagram *ladder* yang dibentuk sebagai berikut.



Gambar 4.19 Relay Y16

Relay ketujuhbelas adalah Y17 yang memiliki persamaan sebagai berikut.

Urutan dan grup

Y12 Y17 Y18

StopperUp-

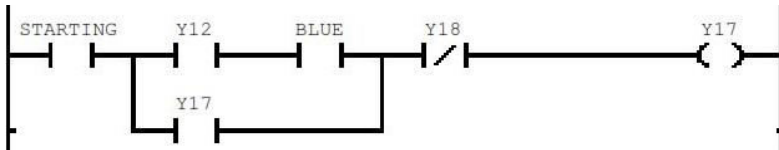
FingerDown, FingerGrip

FingerUp, Forward, FingerOpen

Relay Y17

$$Y17 = \textit{Starting} * [(Y12 * \textit{Blue}) + Y17] * \overline{Y18}$$

Dari persamaan tersebut maka diagram *ladder* yang dibentuk sebagai berikut.



Gambar 4.20 Relay Y17

Relay kedelapanbelas adalah Y18 yang memiliki persamaan sebagai berikut.

Urutan dan grup

Y17 Y18 Y19

FingerDown, FingerGrip

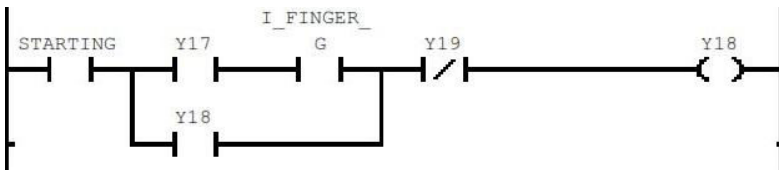
FingerUp, Forward, FingerOpen

Backward

Relay Y18

$$Y18 = \textit{Starting} * [(Y17 * \textit{I_Finger_F})] + Y18 * \overline{Y19}$$

Dari persamaan tersebut maka diagram *ladder* yang dibentuk sebagai berikut.

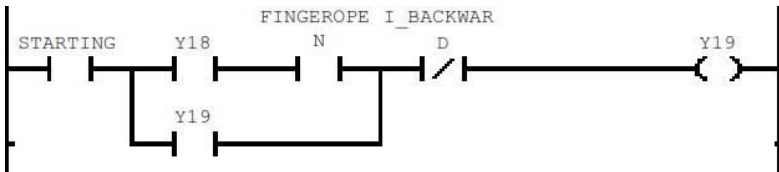


Gambar 4.21 Relay Y18

Relay kesembilanbelas adalah Y19 yang memiliki persamaan sebagai berikut.

<p>Urutan dan grup</p> <p style="text-align: center;">Y18 Y19</p> <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="border: 1px solid black; border-radius: 10px; padding: 5px; background-color: #e0e0ff;"> <i>FingerUp, Forward, FingerOpen</i> </div> <div style="border: 1px solid black; border-radius: 10px; padding: 5px; background-color: #800080; color: white;"> <i>Backward</i> </div> </div> <p>Relay Y19</p> <p>Y19 = <i>Starting</i> * [(Y18 * <i>FingerOpen</i>)] + Y19</p> <p>* <i>Backward</i></p>
--

Dari persamaan tersebut maka diagram *ladder* yang dibentuk sebagai berikut.



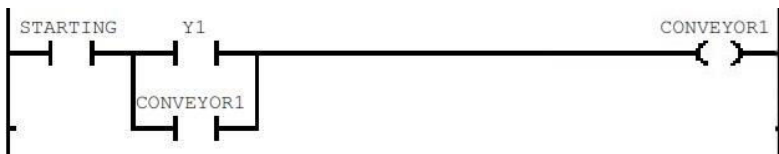
Gambar 4.22 Relay Y19

4.1.2.2 Diagram Ladder Relay Output

Output pertama adalah *Conveyor1* yang memiliki persamaan.

<p>Conveyor1</p> <p>Conveyor1 = <i>Starting</i> * (Y1 + <i>Conveyor1</i>)</p>

Dari persamaan tersebut dapat dibentuk diagram *ladder* sebagai berikut.



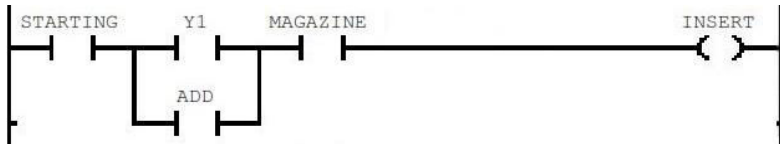
Gambar 4.23 Output *Conveyor1*

Output kedua adalah *Insert* yang memiliki persamaan.

Insert

$$Insert = Starting * (Y1 + Add) * Magazine$$

Dari persamaan tersebut dapat dibentuk diagram *ladder* sebagai berikut.



Gambar 4.24 Output *Insert*

Output ketiga adalah relay *Add* yang memiliki persamaan.

Add

$$Add = Starting * Tim_Add$$

Dari persamaan tersebut dapat dibentuk diagram *ladder* sebagai berikut.



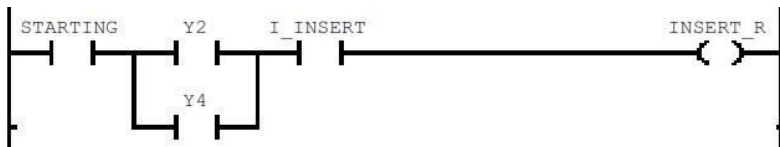
Gambar 4.25 Relay *Add*

Output keempat adalah *Insert_R* yang memiliki persamaan.

Insert_R

$$Insert_R = Starting * (Y1 + Y4) * I_Insert$$

Dari persamaan tersebut dapat dibentuk diagram *ladder* sebagai berikut.

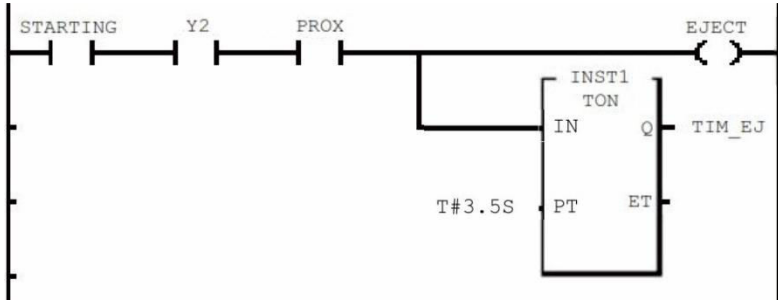


Gambar 4.26 Output *Insert_R*

Output kelima adalah *Eject* yang memiliki persamaan.

Eject
 $Eject = Starting * Y2 * Prox$
 $Tim_{Ej} = Starting * Y2 * Prox * (TON t\#3.5s)$

Dari persamaan tersebut dapat dibentuk diagram *ladder* sebagai berikut.



Gambar 4.27 Output *Eject* dan timer *Tim_Ej*

Output keenam adalah *Eject_R* yang memiliki persamaan.

Eject_R
 $Eject_R = Starting * Y3$

Dari persamaan tersebut dapat dibentuk diagram *ladder* sebagai berikut.



Gambar 4.28 Output *Eject_R*

Output ketujuh adalah *Down* yang memiliki persamaan.

Down
 $Down = Starting * (Y5 + Y7)$

Dari persamaan tersebut dapat dibentuk diagram *ladder* sebagai berikut.



Gambar 4.29 Output *Down*

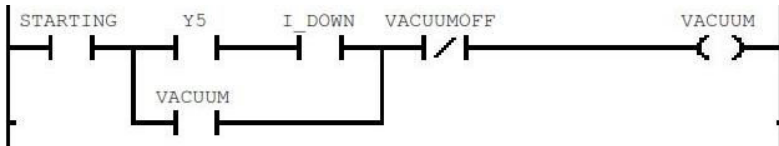
Output delapan adalah *Down* yang memiliki persamaan.

Vacuum

$$\underline{Vacuum} = \text{Starting} * [(Y5 * I_{Down}) + Vacuum]$$

$$* \overline{VacuumOff}$$

Dari persamaan tersebut dapat dibentuk diagram *ladder* sebagai berikut.



Gambar 4.30 Output *Vacuum*

Output kesembilan adalah *Up* yang memiliki persamaan.

Up

$$\underline{Up} = \text{Starting} * (Y6 + Y8)$$

Dari persamaan tersebut dapat dibentuk diagram *ladder* sebagai berikut.



Gambar 4.31 Output *Up*

Output kesepuluh adalah *RotaryCCW* yang memiliki persamaan.

RotaryCCW

$$\underline{RotaryCCW} = \text{Starting} * Y6 * I_{Up}$$

Dari persamaan tersebut dapat dibentuk diagram *ladder* sebagai berikut.



Gambar 4.32 Output *RotaryCCW*

Output kesebelas adalah *VacuumOff* yang memiliki persamaan.

VacuumOff

$$\mathbf{VacuumOff = Starting * Y7 * I_Down}$$

Dari persamaan tersebut dapat dibentuk diagram *ladder* sebagai berikut.



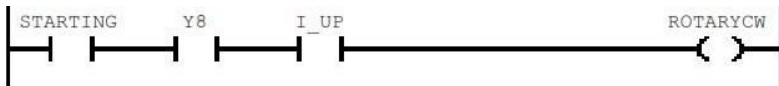
Gambar 4.33 Relay *VacuumOff*

Output keduabelas adalah *RotaryCW* yang memiliki persamaan.

RotaryCW

$$\mathbf{RotaryCW = Starting * Y8 * I_Up}$$

Dari persamaan tersebut dapat dibentuk diagram *ladder* sebagai berikut.



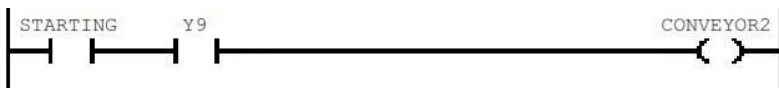
Gambar 4.34 Output *RotaryCW*

Output ketigabelas adalah *Conveyor2* yang memiliki persamaan.

Conveyor2

$$\mathbf{Conveyor2 = Starting * Y9}$$

Dari persamaan tersebut dapat dibentuk diagram *ladder* sebagai berikut.

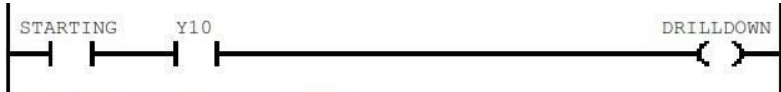


Gambar 4.35 Output *Conveyor2*

Output keempatbelas adalah *DrillDown* yang memiliki persamaan.

<p>DrillDown $DrillDown = Starting * Y10$</p>
--

Dari persamaan tersebut dapat dibentuk diagram *ladder* sebagai berikut.

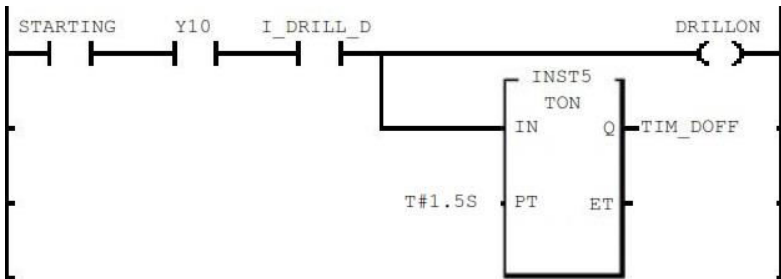


Gambar 4.36 Output *DrillDown*

Output kelimabelas adalah *DrillOn* yang memiliki persamaan.

<p>DrillOn $DrillOn = Starting * Y10 * I_Drill_D$ $Tim_DOff = Starting * Y10 * I_Drill_D * (TON\ t\#1.5s)$</p>
--

Dari persamaan tersebut dapat dibentuk diagram *ladder* sebagai berikut.

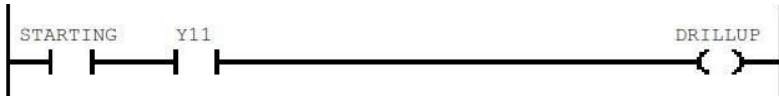


Gambar 4.37 Output *DrillOn* dan timer *Tim_DOff*

Output keenambelas adalah *DrillUp* yang memiliki persamaan.

<p>DrillUp $DrillUp = Starting * Y11$</p>
--

Dari persamaan tersebut dapat dibentuk diagram *ladder* sebagai berikut.

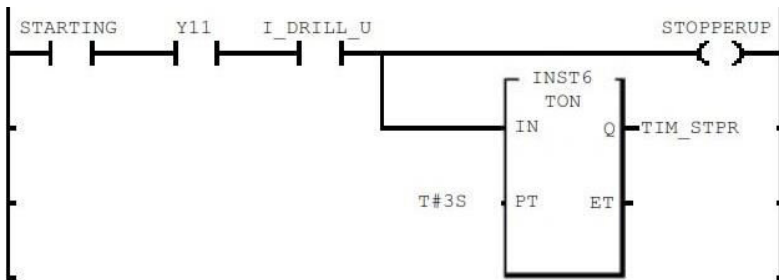


Gambar 4.38 Output *DrillUp*

Output ketujuhbelas adalah *StopperUp* yang memiliki persamaan.

<p>StopperUp $StopperUp = Starting * Y11 * I_Drill_U$ $Tim_STPR = Starting * Y11 * I_Drill_U * (TON t\#3s)$</p>

Dari persamaan tersebut dapat dibentuk diagram *ladder* sebagai berikut.



Gambar 4.39 Output *StopperUp* dan timer *Tim_STPR*

Output kedelapanbelas adalah *Finger_D* yang memiliki persamaan.

<p>Finger_D $Finger_D = Starting * (Y13 + Y15 + Y17)$</p>

Dari persamaan tersebut dapat dibentuk diagram *ladder* sebagai berikut.

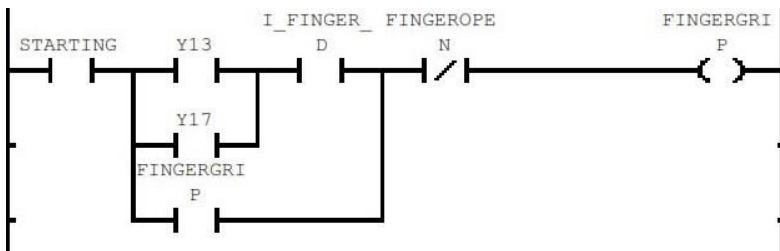


Gambar 4.40 Output *Finger_D*

Output kesembilanbelas adalah *FingerGrip* yang memiliki persamaan.

<p>FingerGrip $FingerGrip = Starting * \{[(Y13 + Y17) * I_Finger_D] + FingerGrip\} * \overline{FingerOpen}$</p>
--

Dari persamaan tersebut dapat dibentuk diagram *ladder* sebagai berikut.

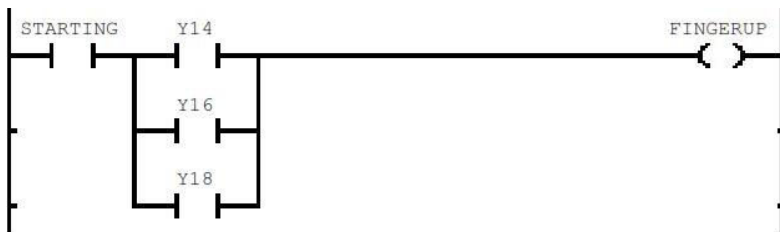


Gambar 4.41 Output *FingerGrip*

Output kedua puluh adalah *FingerUp* yang memiliki persamaan.

<p>FingerUp $FingerUp = Starting * (Y14 + Y16 + Y18)$</p>
--

Dari persamaan tersebut dapat dibentuk diagram *ladder* sebagai berikut.



Gambar 4.42 Output *FingerUp*

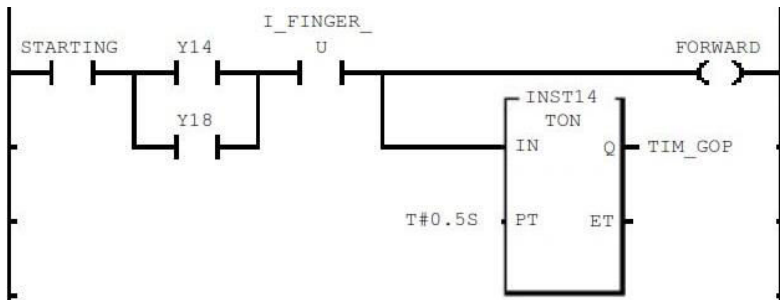
Output kedua puluh satu adalah *Forward* yang memiliki persamaan.

Forward

$$Forward = Starting * (Y14 + Y18) * I_Finger_U$$

$$Tim_Gop = Starting * (Y14 + Y18) * I_Finger_U * (TON\#0.5s)$$

Dari persamaan tersebut dapat dibentuk diagram *ladder* sebagai berikut.



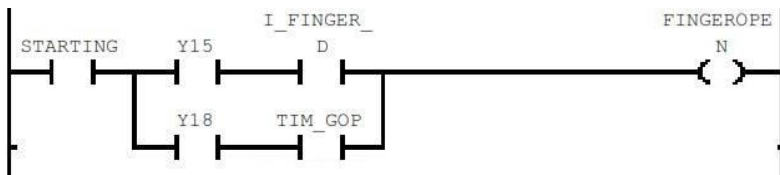
Gambar 4.43 Output *Forward* dan timer *Tim_Gop*

Output kedua puluh dua adalah *FingerOpen* yang memiliki persamaan.

FingerOpen

$$FingerOpen = Starting * [(Y15 * I_Finger_D) + (Y18 * Tim_Gop)]$$

Dari persamaan tersebut dapat dibentuk diagram *ladder* sebagai berikut.



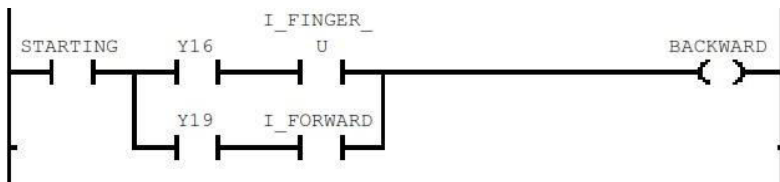
Gambar 4.44 Relay *FingerOpen*

Output kedua puluh tiga adalah *Backward* yang memiliki persamaan.

Backward

$$Backward = Starting * [(Y16 * I_Finger_U) + (Y19 * I_Forward)]$$

Dari persamaan tersebut dapat dibentuk diagram *ladder* sebagai berikut.



Gambar 4.45 Output *Backward*

4.1.2.3 Diagram Ladder Addressing Benda Kerja

Addressing benda kerja pertama adalah output 6 CTU yang menghitung benda kerja plastik hitam dan biru dengan persamaan sebagai berikut.

BK1-BK5 dan VI

$$BK1 = Starting * \overline{Prox} * Tim_Cap * (CTU == 1) * \overline{BK2} * \overline{BK3} * \overline{BK4} * \overline{BK5} * \overline{BK6}$$

$$BK2 = Starting * \overline{Prox} * Tim_Cap * (CTU == 2) * \overline{BK3} * \overline{BK4} * \overline{BK5} * \overline{BK6}$$

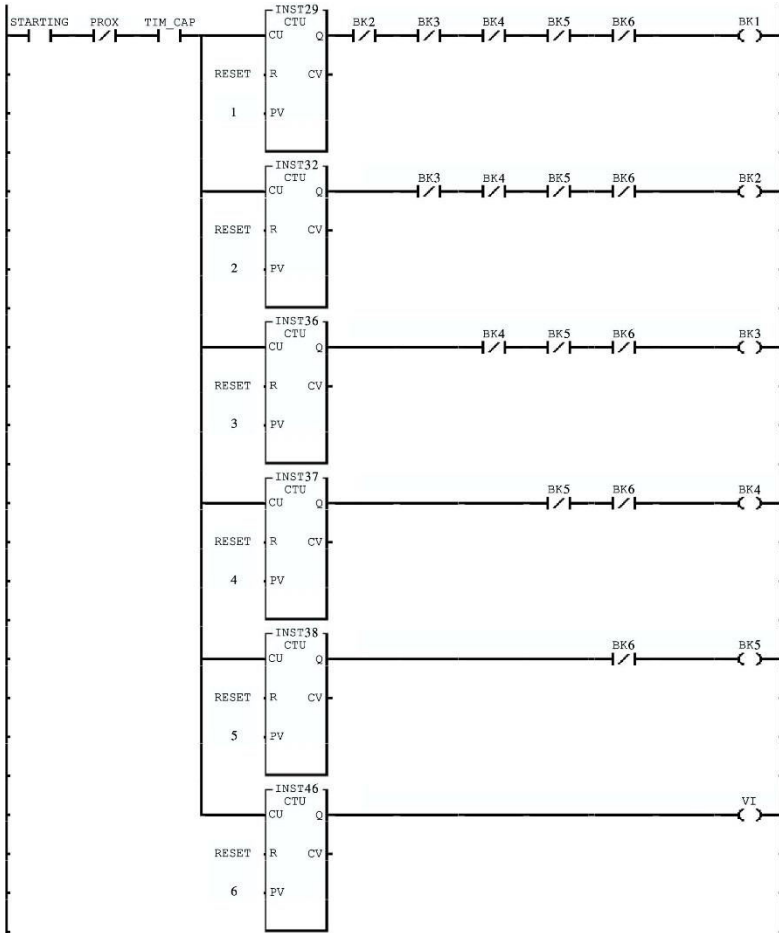
$$BK3 = Starting * \overline{Prox} * Tim_Cap * (CTU == 3) * \overline{BK4} * \overline{BK5} * \overline{BK6}$$

$$BK4 = Starting * \overline{Prox} * Tim_Cap * (CTU == 4) * \overline{BK5} * \overline{BK6}$$

$$BK5 = Starting * \overline{Prox} * Tim_Cap * (CTU == 5) * \overline{BK6}$$

$$VI = Starting * \overline{Prox} * Tim_Cap * (CTU == 6)$$

Dari beberapa persamaan tersebut dapat dibentuk diagram *ladder* sebagai berikut.

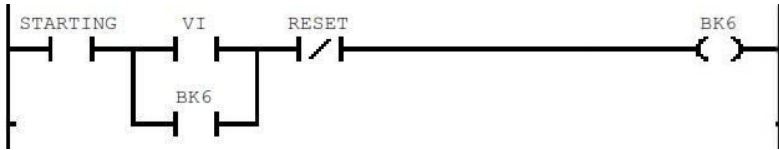


Gambar 4.46 Output BK1-BK5 dan output VI

Kedua adalah relay BK6 yang memiliki persamaan sebagai berikut.

$$\mathbf{BK6} = \mathbf{Starting} * (\mathbf{VI} + \mathbf{BK6}) * \overline{\mathbf{Reset}}$$

Dari persamaan tersebut dapat dibentuk diagram *ladder* sebagai berikut.

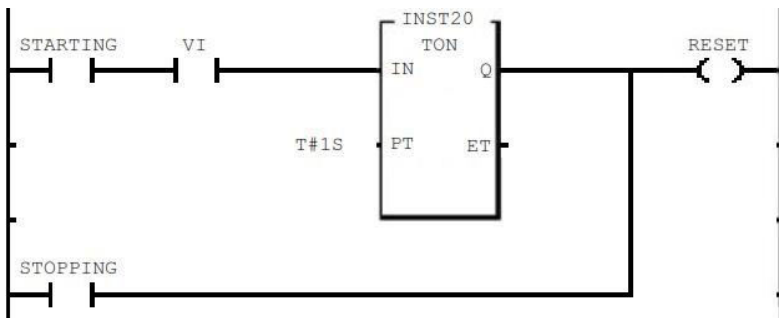


Gambar 4.47 Relay BK6

Ketiga adalah relay *Reset* yang memiliki persamaan sebagai berikut.

<p>Reset</p> $Reset = [Starting * VI * (TON t\#1s)] + Stopping$
--

Dari persamaan tersebut dapat dibentuk diagram *ladder* sebagai berikut.

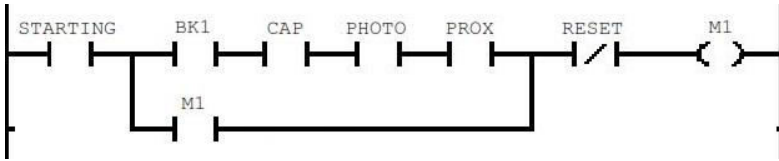


Gambar 4.48 Relay *Reset*

Keempat adalah relay M1 yang memiliki persamaan sebagai berikut.

<p>M1</p> $M1 = Starting * [(BK1 * Cap * Photo * Prox) + M1] * \underline{Reset}$
--

Dari persamaan tersebut dapat dibentuk diagram *ladder* sebagai berikut.

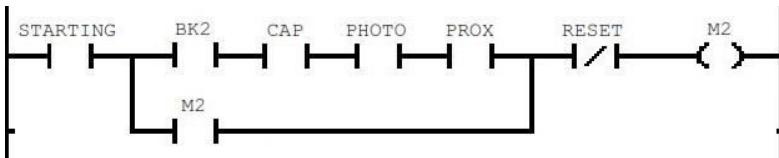


Gambar 4.49 Relay M1

Kelima adalah relay M2 yang memiliki persamaan sebagai berikut.

$$\begin{aligned}
 & \mathbf{M2} \\
 & \mathbf{M2 = Starting * [(BK2 * Cap * Photo * Prox) + M2]} \\
 & \mathbf{* Reset}
 \end{aligned}$$

Dari persamaan tersebut dapat dibentuk diagram *ladder* sebagai berikut.

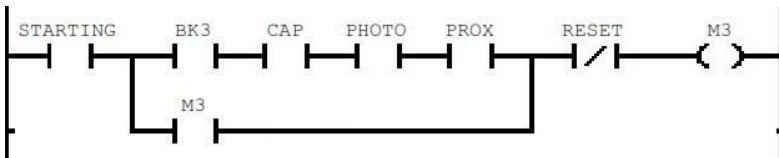


Gambar 4.50 Relay M2

Keenam adalah relay M3 yang memiliki persamaan sebagai berikut.

$$\begin{aligned}
 & \mathbf{M3} \\
 & \mathbf{M3 = Starting * [(BK3 * Cap * Photo * Prox) + M3]} \\
 & \mathbf{* Reset}
 \end{aligned}$$

Dari persamaan tersebut dapat dibentuk diagram *ladder* sebagai berikut.

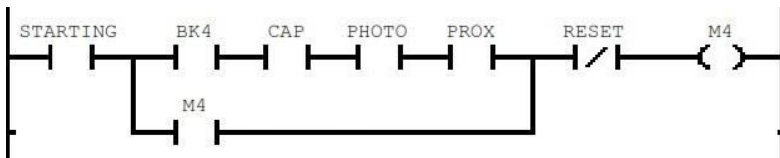


Gambar 4.51 Relay M3

Ketujuh adalah relay M4 yang memiliki persamaan sebagai berikut.

$$\begin{aligned}
 & \mathbf{M4} \\
 & \mathbf{M4 = Starting * [(BK4 * Cap * Photo * Prox) + M4]} \\
 & \mathbf{* Reset}
 \end{aligned}$$

Dari persamaan tersebut dapat dibentuk diagram *ladder* sebagai berikut.

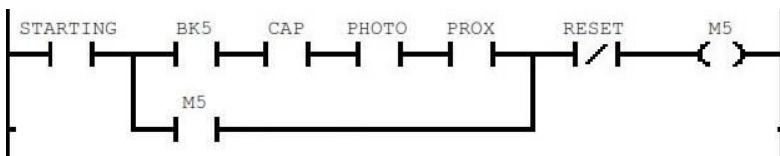


Gambar 4.52 Relay M4

Kedelapan adalah relay M5 yang memiliki persamaan sebagai berikut.

$$\begin{aligned}
 & \mathbf{M5} \\
 & \mathbf{M5 = Starting * [(BK5 * Cap * Photo * Prox) + M5]} \\
 & \mathbf{* Reset}
 \end{aligned}$$

Dari persamaan tersebut dapat dibentuk diagram *ladder* sebagai berikut.



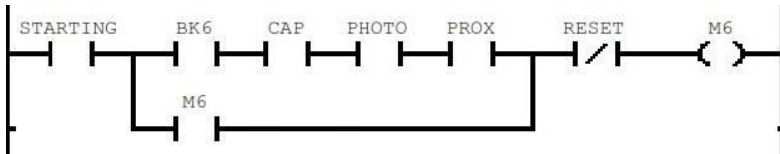
Gambar 4.53 Relay M5

Kesembilan adalah relay M6 yang memiliki persamaan sebagai berikut.

M6

$$M6 = Starting * [(BK6 * Cap * Photo * Prox) + M6] * \overline{Reset}$$

Dari persamaan tersebut dapat dibentuk diagram *ladder* sebagai berikut.



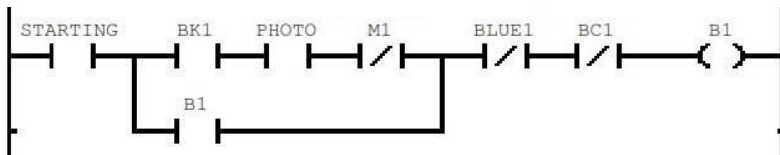
Gambar 4.54 Relay M6

Kesepuluh adalah relay B1 yang memiliki persamaan sebagai berikut.

B1

$$B1 = Starting * [(BK1 * Photo * \overline{M1}) + B1] * \overline{Blue1} * \overline{BC1}$$

Dari persamaan tersebut dapat dibentuk diagram *ladder* sebagai berikut.



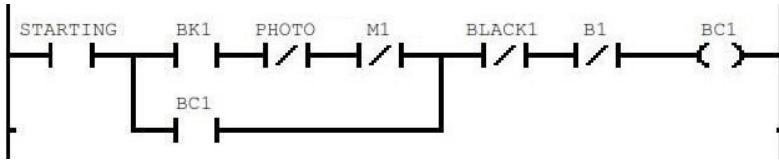
Gambar 4.55 Relay B1

Kesebelas adalah relay BC1 yang memiliki persamaan sebagai berikut.

BC1

$$BC1 = Starting * [(BK1 * \overline{Photo} * \overline{M1}) + BC1] * \overline{Black1} * \overline{B1}$$

Dari persamaan tersebut dapat dibentuk diagram *ladder* sebagai berikut.

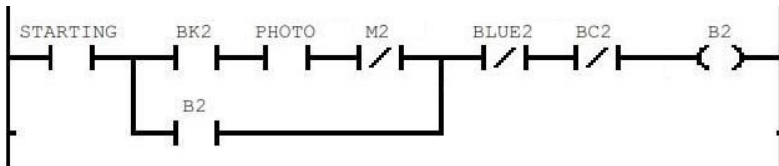


Gambar 4.56 Relay BC1

Keduabelas adalah relay B2 yang memiliki persamaan sebagai berikut.

$$B2 = Starting * [(BK2 * Photo * \overline{M2}) + B2] * \overline{Blue2} * \overline{BC2}$$

Dari persamaan tersebut dapat dibentuk diagram *ladder* sebagai berikut.

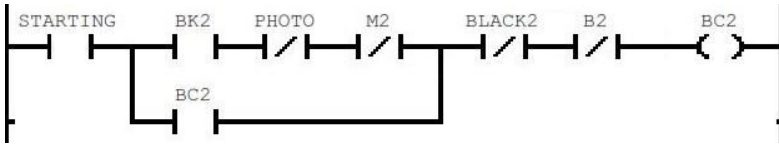


Gambar 4.57 Relay B2

Ketigabelas adalah relay BC2 yang memiliki persamaan sebagai berikut.

$$BC2 = Starting * [(BK2 * \overline{Photo} * \overline{M2}) + BC2] * \overline{Black2} * \overline{B2}$$

Dari persamaan tersebut dapat dibentuk diagram *ladder* sebagai berikut.

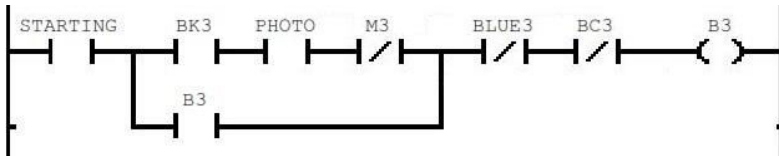


Gambar 4.58 Relay BC2

Keempatbelas adalah relay B3 yang memiliki persamaan sebagai berikut.

$$\begin{aligned}
 & \mathbf{B3} \\
 & \mathbf{B3 = Starting * [(BK3 * Photo * \overline{M3}) + B3] * \overline{Blue3} * \overline{BC3}}
 \end{aligned}$$

Dari persamaan tersebut dapat dibentuk diagram *ladder* sebagai berikut.

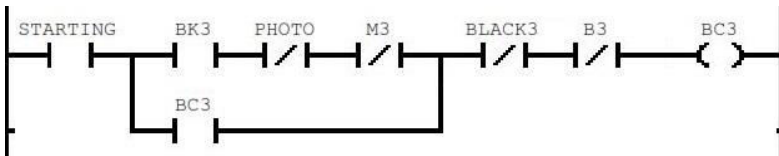


Gambar 4.59 Relay B3

Kelimabelas adalah relay BC3 yang memiliki persamaan sebagai berikut.

$$\begin{aligned}
 & \mathbf{BC3} \\
 & \mathbf{BC3 = Starting * [(BK3 * \overline{Photo} * \overline{M3}) + BC3] * \overline{Black3} * \overline{B3}}
 \end{aligned}$$

Dari persamaan tersebut dapat dibentuk diagram *ladder* sebagai berikut.

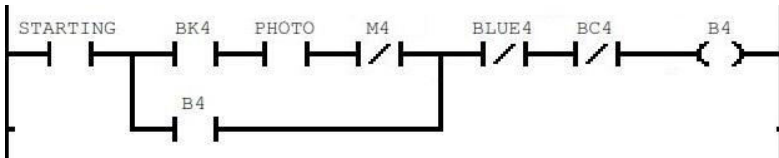


Gambar 4.60 Relay BC3

Keenambelas adalah relay B4 yang memiliki persamaan sebagai berikut.

$$B4 = Starting * [(BK4 * Photo * \overline{M4}) + B4] * \overline{Blue4} * \overline{BC4}$$

Dari persamaan tersebut dapat dibentuk diagram *ladder* sebagai berikut.

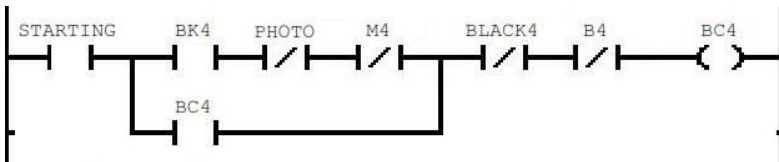


Gambar 4.61 Relay B4

Ketujuhbelas adalah relay BC4 yang memiliki persamaan sebagai berikut.

$$BC4 = Starting * [(BK4 * \overline{Photo} * \overline{M4}) + BC4] * \overline{Black4} * \overline{B4}$$

Dari persamaan tersebut dapat dibentuk diagram *ladder* sebagai berikut.

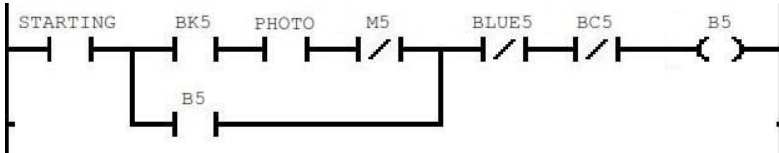


Gambar 4.62 Relay BC4

Kedelapanbelas adalah relay B5 yang memiliki persamaan sebagai berikut.

$$B5 = Starting * [(BK5 * Photo * \overline{M5}) + B5] * \overline{Blue5} * \overline{BC5}$$

Dari persamaan tersebut dapat dibentuk diagram *ladder* sebagai berikut.

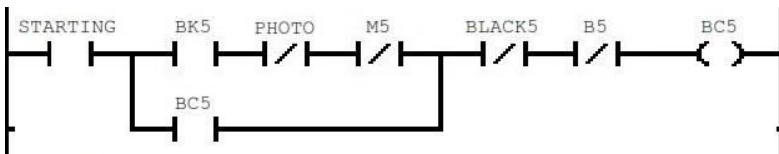


Gambar 4.63 Relay B5

Kesembilanbelas adalah relay BC5 yang memiliki persamaan sebagai berikut.

$$\begin{aligned}
 & \mathbf{BC5} \\
 & \mathbf{BC5 = Starting * [(BK5 * \overline{Photo} * \overline{M5}) + BC5] * \overline{Black5}} \\
 & \mathbf{* \overline{B5}}
 \end{aligned}$$

Dari persamaan tersebut dapat dibentuk diagram *ladder* sebagai berikut.

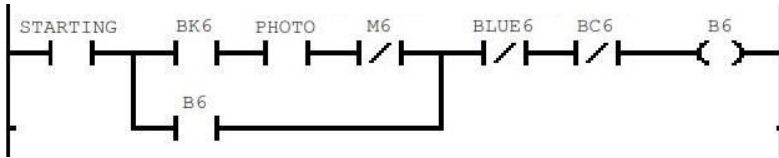


Gambar 4.64 Relay BC5

Keduapuluh adalah relay B6 yang memiliki persamaan sebagai berikut.

$$\begin{aligned}
 & \mathbf{B6} \\
 & \mathbf{B6 = Starting * [(BK6 * Photo * \overline{M6}) + B6] * \overline{Blue6} * \overline{BC6}}
 \end{aligned}$$

Dari persamaan tersebut dapat dibentuk diagram *ladder* sebagai berikut.



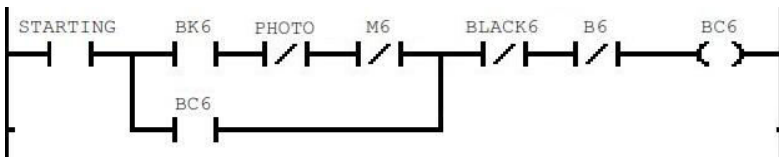
Gambar 4.65 Relay B6

Keduapuluhatsu adalah relay BC6 yang memiliki persamaan sebagai berikut.

BC6

$$BC6 = Starting * [(BK6 * \overline{Photo} * \overline{M6}) + BC6] * \overline{Black6} * \overline{B6}$$

Dari persamaan tersebut dapat dibentuk diagram *ladder* sebagai berikut.



Gambar 4.66 Relay BC6

Keduapuluhdua adalah ouput 6 CTU yang menghitung benda kerja plastik hitam dan biru terdeteksi pada sensor *End2* dengan persamaan sebagai berikut.

ED1-ED5 dan E6

$$ED1 = Starting * End2 * (CTU == 1) * \overline{ED2} * \overline{ED3} * \overline{ED4} * \overline{ED5} * \overline{ED6}$$

$$ED2 = Starting * End2 * (CTU == 2) * \overline{ED3} * \overline{ED4} * \overline{ED5} * \overline{ED6}$$

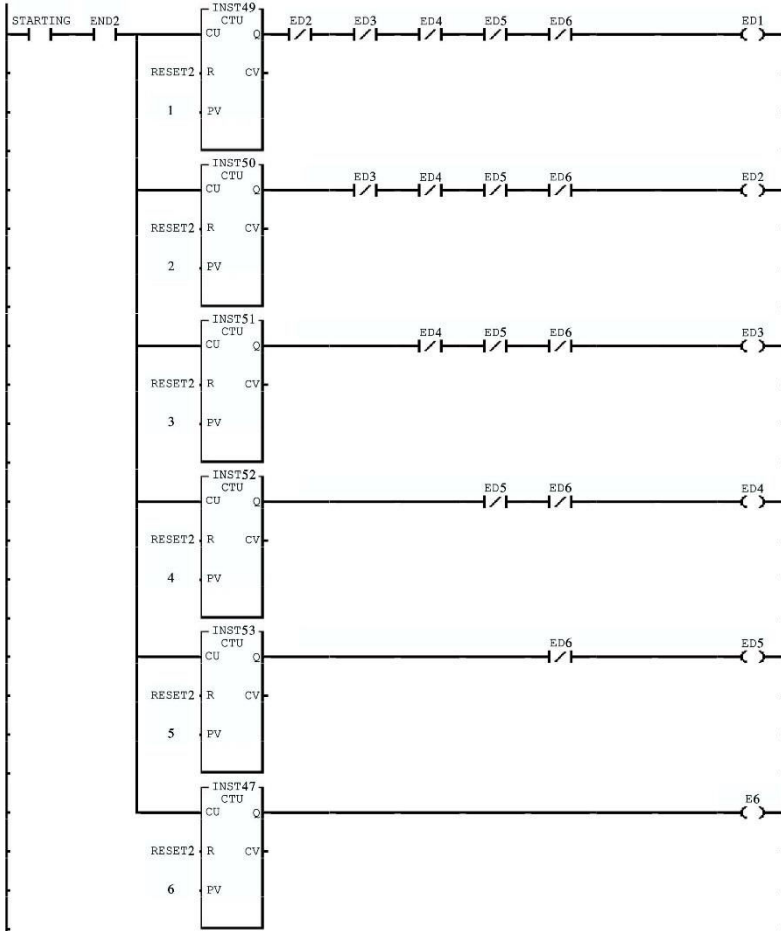
$$ED3 = Starting * End2 * (CTU == 3) * \overline{ED4} * \overline{ED5} * \overline{ED6}$$

$$ED4 = Starting * End2 * (CTU == 4) * \overline{ED5} * \overline{ED6}$$

$$ED5 = Starting * End2 * (CTU == 5) * \overline{ED6}$$

$$E6 = Starting * End2 * (CTU == 6)$$

Dari beberapa persamaan tersebut dapat dibentuk diagram *ladder* sebagai berikut.



Gambar 4.67 Output ED1-ED5 dan output E6

Keduapuluhtiga adalah relay ED6 yang memiliki persamaan sebagai berikut.

ED6

$$ED6 = Starting * (E6 + ED6) * \overline{Reset2}$$

Dari persamaan tersebut dapat dibentuk diagram *ladder* sebagai berikut.



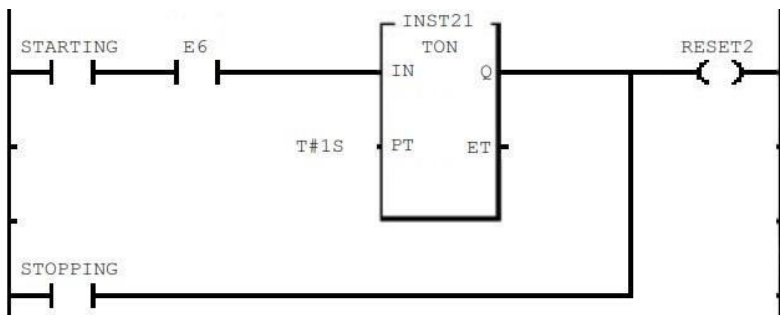
Gambar 4.68 Relay ED6

Keduapuluhempat adalah relay *Reset2* yang memiliki persamaan sebagai berikut.

Reset2

$$Reset2 = [Starting * E6 * (TON t\#1s)] + Stopping$$

Dari persamaan tersebut dapat dibentuk diagram *ladder* sebagai berikut.



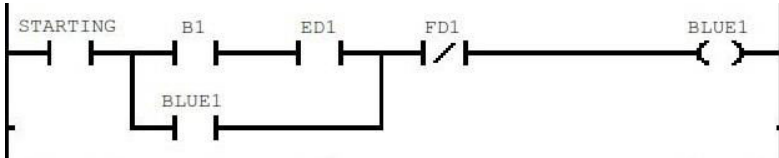
Gambar 4.69 Relay *Reset2*

Keduapuluhlima adalah relay *Blue1* yang memiliki persamaan sebagai berikut.

Blue1

$$Blue1 = Starting * [(B1 * ED1) + Blue1] * \overline{FD1}$$

Dari persamaan tersebut dapat dibentuk diagram *ladder* sebagai berikut.

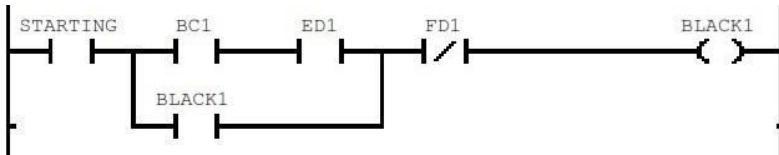


Gambar 4.70 Relay *Blue1*

Keduapuluhenam adalah relay *Black1* yang memiliki persamaan sebagai berikut.

<p>Black1</p> $Black1 = Starting * [(BC1 * ED1) + Black1] * \overline{FD1}$
--

Dari persamaan tersebut dapat dibentuk diagram *ladder* sebagai berikut.

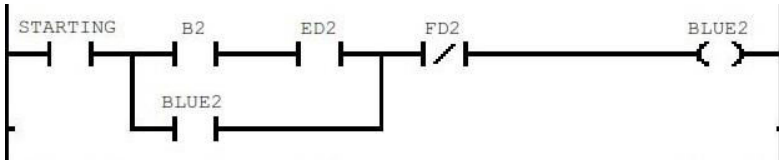


Gambar 4.71 Relay *Black1*

Keduapuluh tujuh adalah relay *Blue2* yang memiliki persamaan sebagai berikut.

<p>Blue2</p> $Blue2 = Starting * [(B2 * ED2) + Blue2] * \overline{FD2}$
--

Dari persamaan tersebut dapat dibentuk diagram *ladder* sebagai berikut.



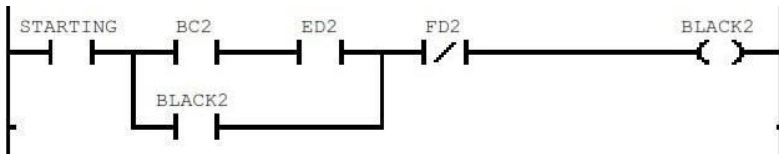
Gambar 4.72 Relay *Blue2*

Keduapuluhdelapan adalah relay *Black2* yang memiliki persamaan sebagai berikut.

Black2

$$Black2 = Starting * [(BC2 * ED2) + Black2] * \overline{FD2}$$

Dari persamaan tersebut dapat dibentuk diagram *ladder* sebagai berikut.



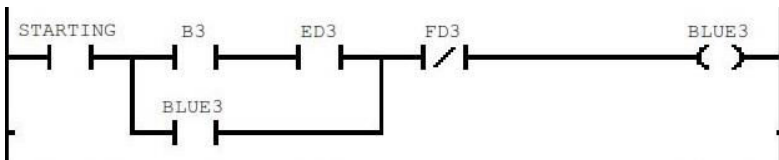
Gambar 4.73 Relay *Black2*

Keduapuluhsembilan adalah relay *Blue3* yang memiliki persamaan sebagai berikut.

Blue3

$$Blue3 = Starting * [(B3 * ED3) + Blue3] * \overline{FD3}$$

Dari persamaan tersebut dapat dibentuk diagram *ladder* sebagai berikut.



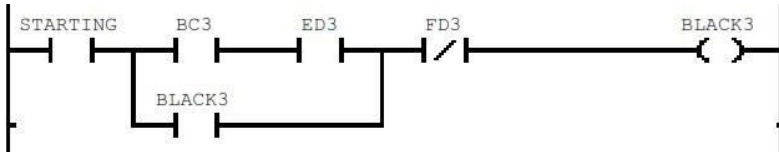
Gambar 4.74 Relay *Blue3*

Ketigapuluh adalah relay *Black3* yang memiliki persamaan sebagai berikut.

Black3

$$Black3 = Starting * [(BC3 * ED3) + Black3] * \overline{FD3}$$

Dari persamaan tersebut dapat dibentuk diagram *ladder* sebagai berikut.



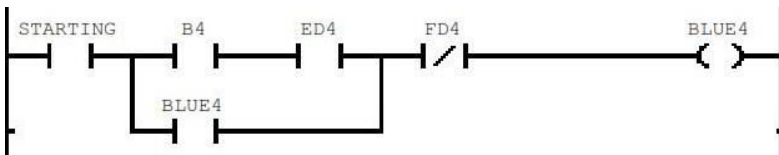
Gambar 4.75 Relay *Black3*

Ketigapuluhsatu adalah relay *Blue4* yang memiliki persamaan sebagai berikut.

Blue4

$$Blue4 = Starting * [(B4 * ED4) + Blue4] * \overline{FD4}$$

Dari persamaan tersebut dapat dibentuk diagram *ladder* sebagai berikut.



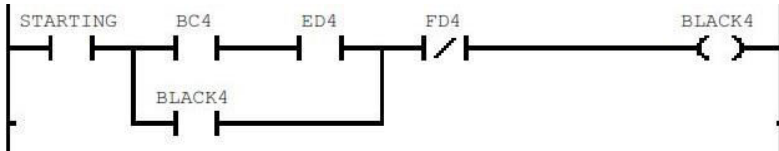
Gambar 4.76 Relay *Blue4*

Ketigapuluhdua adalah relay *Black4* yang memiliki persamaan sebagai berikut.

Black4

$$Black4 = Starting * [(BC4 * ED4) + Black4] * \overline{FD4}$$

Dari persamaan tersebut dapat dibentuk diagram *ladder* sebagai berikut.



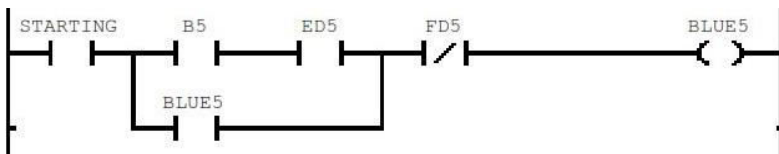
Gambar 4.77 Relay *Black4*

Ketigapuluh tiga adalah relay *Blue5* yang memiliki persamaan sebagai berikut.

Blue5

$$Blue5 = Starting * [(B5 * ED5) + Blue5] * \overline{FD5}$$

Dari persamaan tersebut dapat dibentuk diagram *ladder* sebagai berikut.



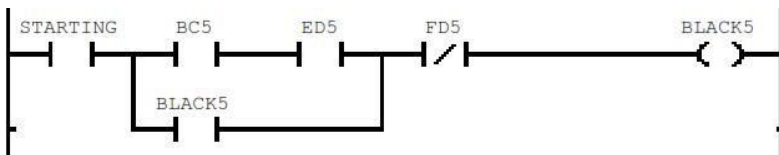
Gambar 4.78 Relay *Blue5*

Ketigapuluh empat adalah relay *Black5* yang memiliki persamaan sebagai berikut.

Black5

$$Black5 = Starting * [(BC5 * ED5) + Black5] * \overline{FD5}$$

Dari persamaan tersebut dapat dibentuk diagram *ladder* sebagai berikut.

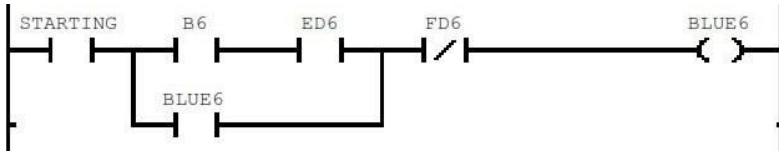


Gambar 4.79 Relay *Black5*

Ketigapuluhlima adalah relay *Blue6* yang memiliki persamaan sebagai berikut.

$$\mathbf{Blue6 = Starting * [(B6 * ED6) + Blue6] * \overline{FD6}}$$

Dari persamaan tersebut dapat dibentuk diagram *ladder* sebagai berikut.

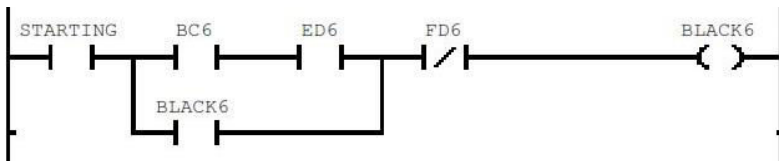


Gambar 4.80 Relay *Blue6*

Ketigapuluhenam adalah relay *Black6* yang memiliki persamaan sebagai berikut.

$$\mathbf{Black6 = Starting * [(BC6 * ED6) + Black6] * \overline{FD6}}$$

Dari persamaan tersebut dapat dibentuk diagram *ladder* sebagai berikut.



Gambar 4.81 Relay *Black6*

Ketigapuluhtujuh adalah ouput 6 CTU yang menghitung aktifnya *limit switch I_Forward* dengan persamaan sebagai berikut.

FD1-FD5 dan F6

$$\mathbf{FD1 = Starting * I_Forward * (CTU == 1) * \overline{FD2} * \overline{FD3} * \overline{FD4} * \overline{FD5} * \overline{FD6}}$$

$$\mathbf{FD2 = Starting * I_Forward * (CTU == 2) * \overline{FD3} * \overline{FD4} * \overline{FD5} * \overline{FD6}}$$

$$\mathbf{FD3 = Starting * I_Forward * (CTU == 3) * \overline{FD4} * \overline{FD5} * \overline{FD6}}$$

$$\mathbf{FD4 = Starting * I_Forward * (CTU == 4) * \overline{FD5} * \overline{FD6}}$$

$$\mathbf{FD5 = Starting * I_Forward * (CTU == 5) * \overline{FD6}}$$

$$\mathbf{F6 = Starting * I_Forward * (CTU == 6)}$$

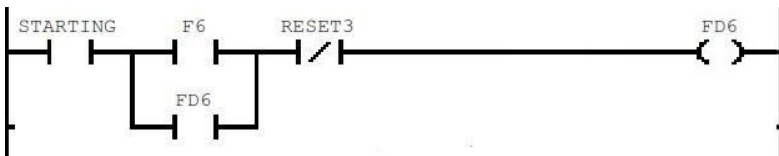
Dari beberapa persamaan tersebut dapat dibentuk diagram *ladder* sesuai pada **Gambar 4.84**.

Ketigapuluhdelapan adalah relay FD6 yang memiliki persamaan sebagai berikut.

ED6

$$\mathbf{FD6 = Starting * (F6 + FD6) * \overline{Reset3}}$$

Dari persamaan tersebut dapat dibentuk diagram *ladder* sebagai berikut.



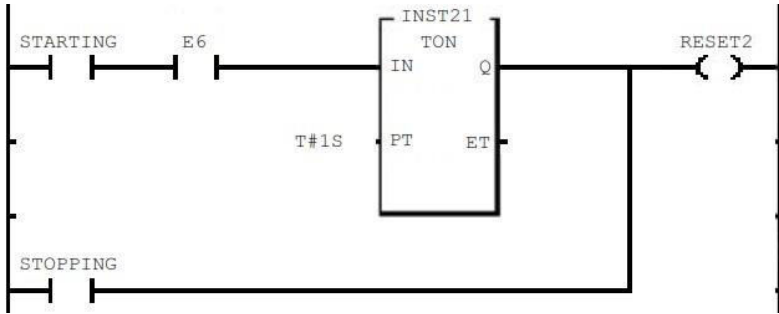
Gambar 4.82 Relay FD6

Ketigapuluhsembilan adalah relay *Reset3* yang memiliki persamaan sebagai berikut.

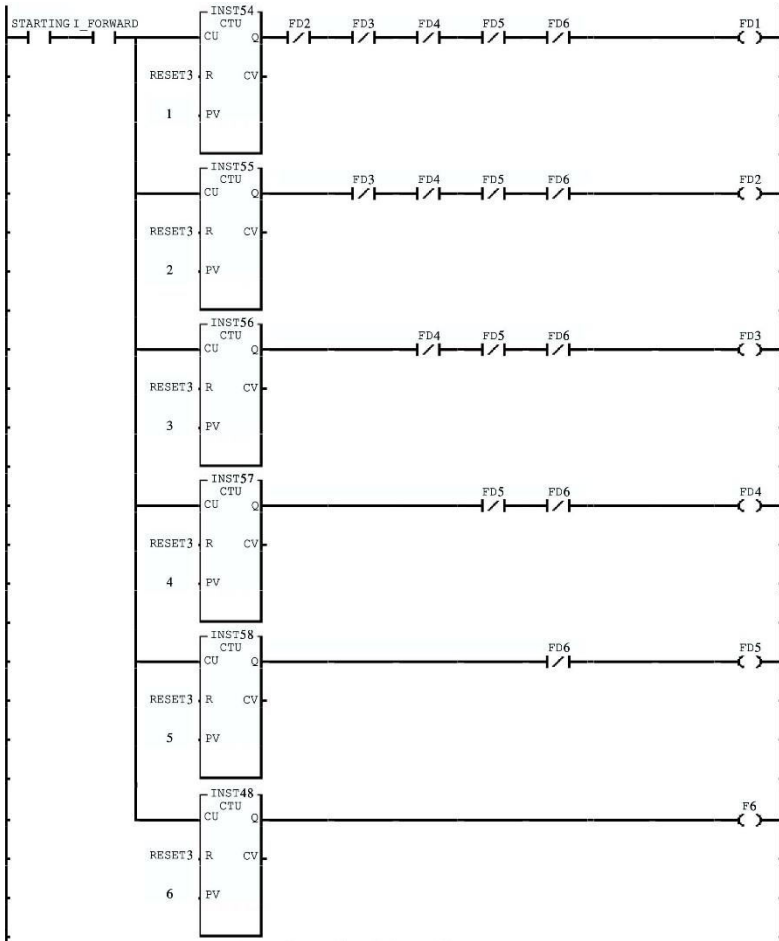
Reset3

$$\mathbf{Reset3 = [Starting * F6 * (TON t\#1s)] + Stopping}$$

Dari persamaan tersebut dapat dibentuk diagram *ladder* sebagai berikut.



Gambar 4.83 Relay *Reset3*



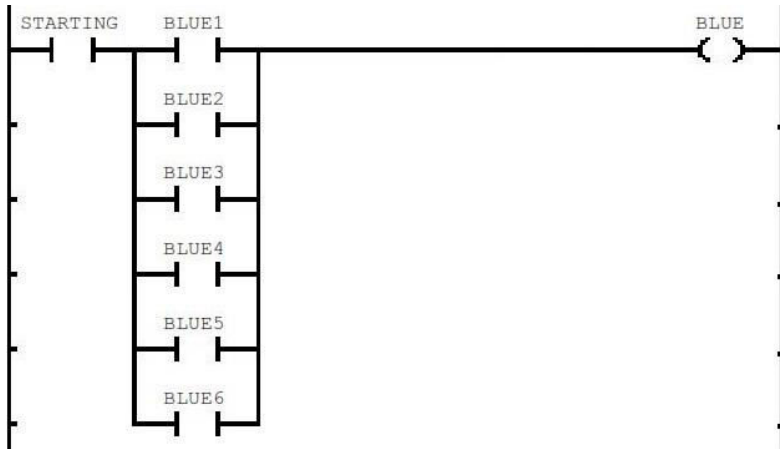
Gambar 4.84 Output FD1-FD5 dan output F6

Keempatpuluh adalah relay *Blue* yang memiliki persamaan sebagai berikut.

Blue

$$Blue = Starting * (Blue1 + Blue2 + Blue3 + Blue4 + Blue5 + Blue6)$$

Dari persamaan tersebut dapat dibentuk diagram *ladder* sebagai berikut.

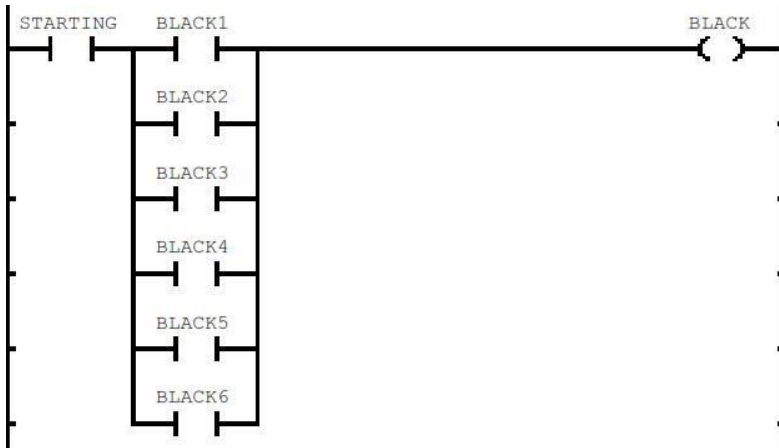


Gambar 4.85 Relay *Blue*

Keempatpuluhsatu adalah relay *Black* yang memiliki persamaan sebagai berikut.

<p>Black $\mathbf{Black = Starting * (Black1 + Black2 + Black3 + Black4 + Black5 + Black6)}$</p>

Dari persamaan tersebut dapat dibentuk diagram *ladder* sebagai berikut.

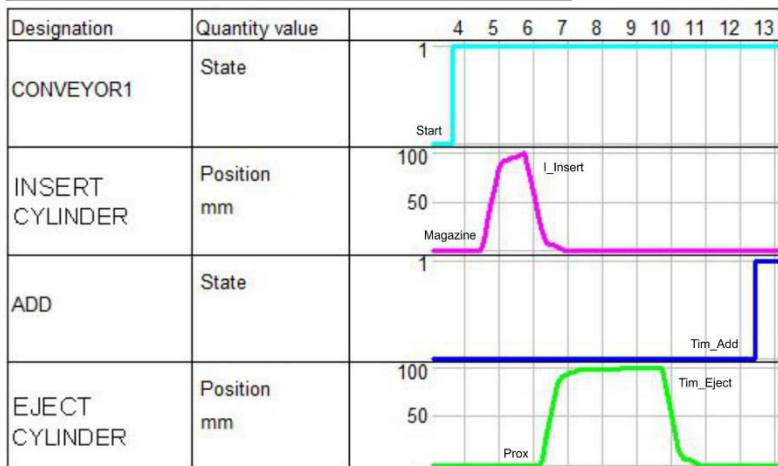
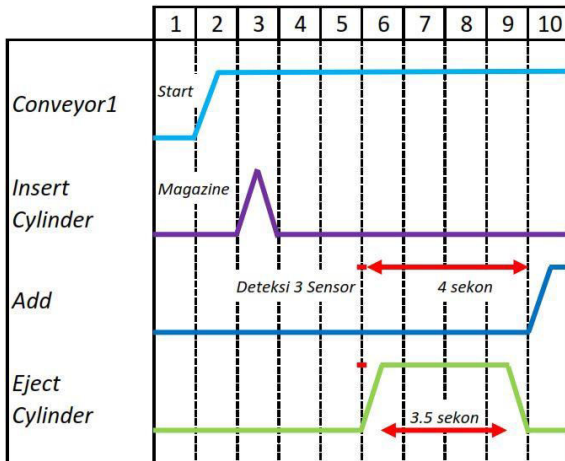


Gambar 4.86 Relay Black

4.2 Validasi

Setelah perancangan diagram *ladder* selesai, langkah selanjutnya yaitu validasi. Tujuan validasi adalah memastikan kesesuaian diagram *ladder* dengan skenario kerja sebelum diimplementasikan ke sistem sebenarnya. Pada langkah validasi ini digunakan *software Fluidsim* untuk menampilkan hasil sinyal output aktuator ketika diagram *ladder* dijalankan.

Validasi pertama dilakukan pada modul *separation* dengan cara membandingkan diagram waktu perancangan diagram *ladder* dengan hasil simulasi kemudian mengecek urutan kerja.



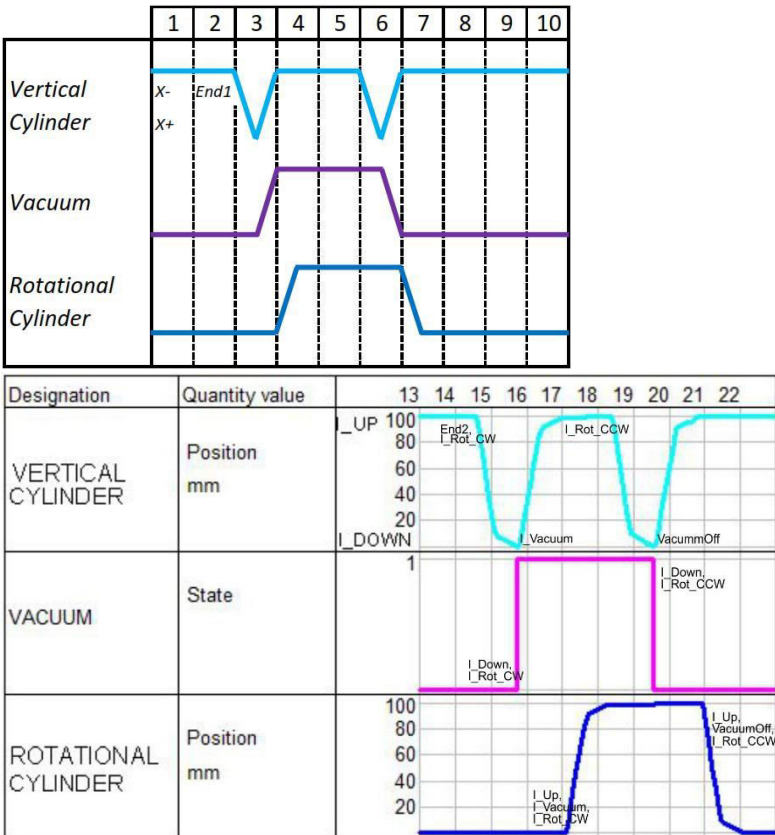
Gambar 4.87 Validasi aktuator modul *separation*

Tabel 4.2 Tabel pengecekan urutan kerja pada modul *separation*

Daftar urutan kerja	Check
<i>Conveyor1</i> berjalan setelah tombol <i>Start</i> ditekan.	✓
<i>Insert cylinder</i> bergerak maju setelah sensor <i>Magazine</i> mendeteksi benda kerja.	✓
<i>Insert cylinder</i> bergerak mundur kembali ke posisi semula.	✓

Penambahan benda kerja setelah sensor mendeteksi benda kerja sebelumnya selama 4 detik.	✓
<i>Eject cylinder</i> bergerak maju setelah <i>Prox</i> aktif.	✓
<i>Eject cylinder</i> mempertahankan posisinya selama 3.5 detik.	✓
<i>Eject cylinder</i> bergerak mundur kembali ke posisi semula.	✓

Validasi kedua dilakukan pada modul *pick & place* dengan cara membandingkan diagram waktu perancangan diagram *ladder* dengan hasil simulasi kemudian mengecek urutan kerja.



Gambar 4. 88 Validasi aktuator modul *pick & place*

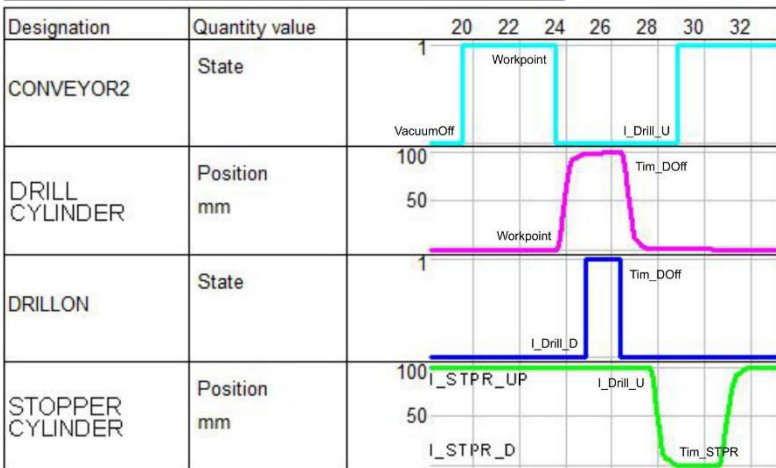
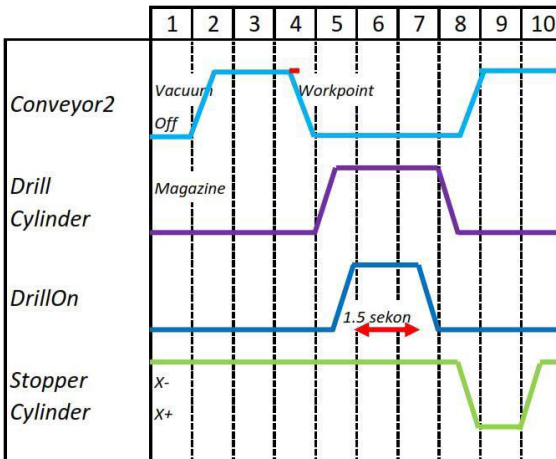
Tabel 4.3 Tabel pengecekan urutan kerja pada modul *pick & place*

Daftar urutan kerja	Check
<i>Vertical cylinder</i> bergerak ke bawah ketika sensor <i>End1</i> mendeteksi benda kerja.	✓
<i>Vacuum pad</i> memvakum benda kerja.	✓
<i>Vertical cylinder</i> bergerak ke atas.	✓
<i>Rotational cylinder</i> bergerak berlawanan arah jarum jam.	✓
<i>Vertical cylinder</i> bergerak ke bawah.	✓
<i>Vacuum pad</i> berhenti memvakum benda kerja.	✓
<i>Vertical cylinder</i> bergerak ke atas.	✓
<i>Rotational cylinder</i> bergerak searah jarum jam kembali ke posisi semula.	✓

Validasi ketiga dilakukan pada modul *stopper* dengan cara membandingkan diagram waktu perancangan diagram *ladder* dengan hasil simulasi dan mengecek urutan kerja.

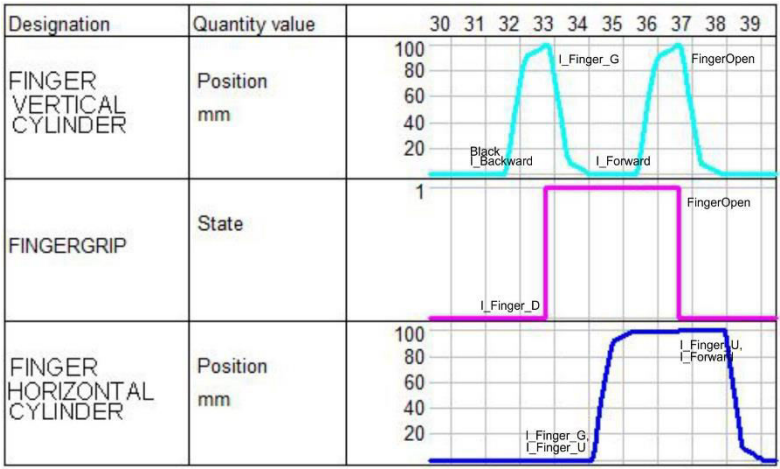
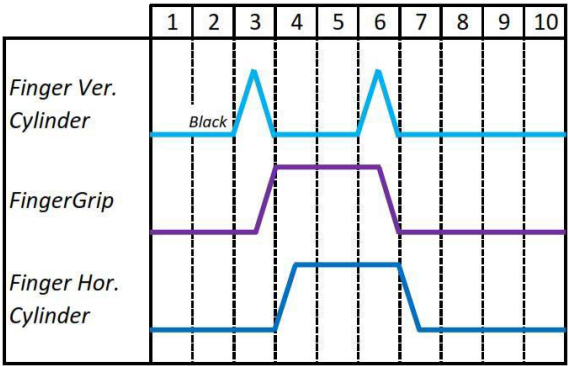
Tabel 4.4 Tabel pengecekan urutan kerja pada modul *stopper*

Daftar urutan kerja	Check
<i>Conveyor2</i> berjalan setelah <i>Vacuum</i> dinon-aktifkan.	✓
<i>Drill cylinder</i> bergerak ke bawah setelah sensor <i>Workpoint</i> mendeteksi benda kerja.	✓
<i>Motor drill</i> berputar melakukan proses <i>drilling</i> selama 1.5 sekon.	✓
<i>Drill cylinder</i> bergerak ke atas kembali ke posisi semula setelah proses <i>drilling</i> selesai.	✓
<i>Stopper cylinder</i> bergerak ke atas.	✓
<i>Conveyor2</i> kembali berjalan.	✓
<i>Stopper cylinder</i> bergerak ke bawah ke posisi semula setelah benda kerja telah dilewatkan oleh <i>Conveyor2</i> .	✓



Gambar 4.89 Validasi aktuator modul *stopper*

Validasi keempat dilakukan pada modul *line movement* dengan cara membandingkan diagram waktu perancangan diagram *ladder* dengan hasil simulasi kemudian mengecek urutan kerja untuk benda kerja plastik hitam dan benda kerja plastik biru.

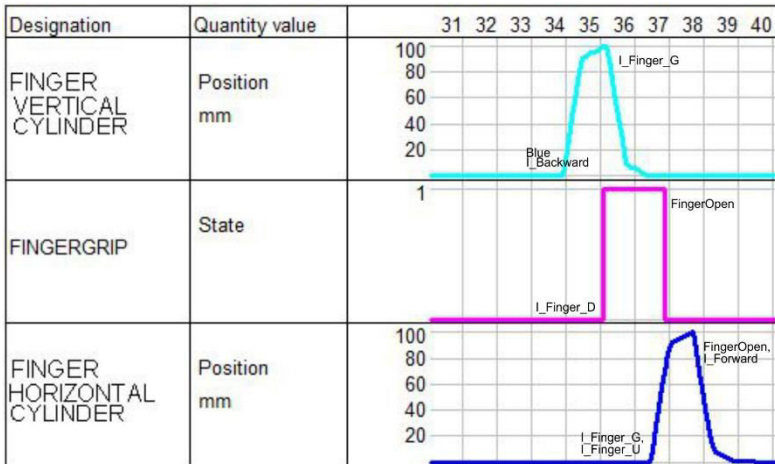
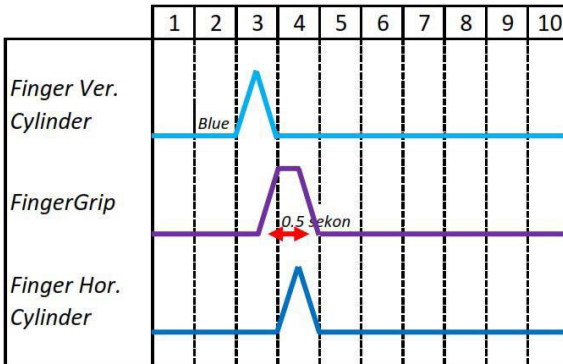


Gambar 4.90 Validasi aktuator modul *line movement* untuk benda kerja plastik hitam

Tabel 4.5 Tabel pengecekan urutan kerja pada modul *line movement* untuk benda kerja plastik hitam

Daftar urutan kerja	Check
<i>Finger vertical cylinder</i> bergerak ke bawah ketika benda kerja plastik hitam berada di ujung <i>Conveyor2</i> .	✓
<i>Fingergrip</i> menggenggam benda kerja.	✓
<i>Finger vertical cylinder</i> bergerak ke atas.	✓
<i>Finger horizontal cylinder</i> bergerak maju ke depan.	✓

<i>Finger vertical cylinder</i> bergerak ke bawah.	✓
<i>Fingergrip</i> melepas genggaman pada benda kerja.	✓
<i>Finger vertical cylinder</i> bergerak ke atas.	✓
<i>Finger horizontal cylinder</i> bergerak mundur ke belakang kembali ke posisi semula.	✓



Gambar 4.91 Validasi aktuator modul *line movement* untuk benda kerja plastik biru

Tabel 4.6 Tabel pengecekan urutan kerja pada modul *line movement* untuk benda kerja plastik biru

Daftar urutan kerja	Check
<i>Finger vertical cylinder</i> bergerak ke bawah ketika benda kerja plastik biru berada di ujung <i>Conveyor2</i> .	✓
<i>Fingergrip</i> menggenggam benda kerja.	✓
<i>Finger vertical cylinder</i> bergerak ke atas.	✓
<i>Finger horizontal cylinder</i> bergerak maju ke depan.	✓
<i>Fingergrip</i> melepas genggamannya pada benda kerja tepat di atas kotak hitam.	✓
<i>Finger horizontal cylinder</i> bergerak mundur ke belakang kembali ke posisi semula.	✓

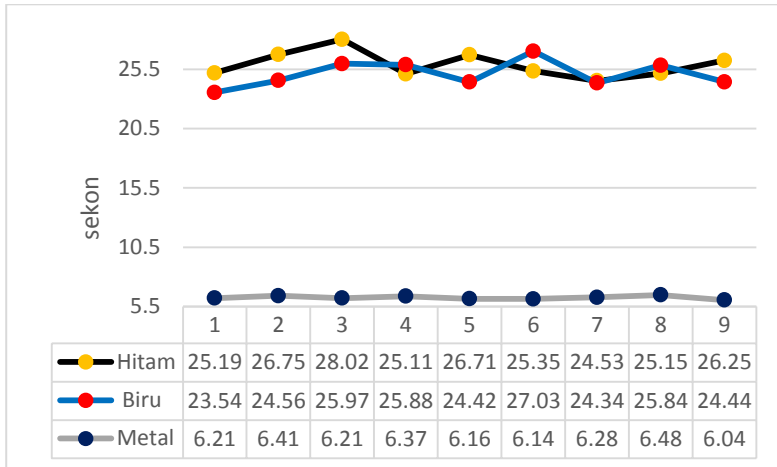
4.3 Pengujian Sistem

Pengujian sistem dilakukan untuk mengetahui keberhasilan hasil implementasi diagram *ladder* pada *Factory Automatic Trainer*. Pengujian dilakukan secara beragam antara benda kerja metal abu-abu, benda kerja plastik hitam dan benda kerja plastik biru sebanyak 27 kali dengan jumlah masing-masing benda kerja uji berjumlah sama yaitu 9 buah (detailnya lihat **Tabel 4.7**) dimana nomor menunjukkan urutan benda kerja yang diproses. Penghitungan waktu dimulai dari tombol tekan *Start* ditekan sampai proses benda kerja selesai.

Tabel 4.7 Pengujian sistem

No	Benda Kerja menurut Warna	Waktu (sekon)	No	Benda Kerja menurut Warna	Waktu (sekon)
1.	Hitam	25.19	15.	Biru	23.54
2.	Hitam	26.75	16.	Metal	6.21
3.	Hitam	28.02	17.	Hitam	25.11
4.	Biru	24.56	18.	Biru	25.97
5.	Biru	25.88	19.	Biru	24.42
6.	Biru	27.03	20.	Hitam	26.71
7.	Metal	6.41	21.	Metal	6.21
8.	Metal	6.37	22.	Biru	24.34
9.	Metal	6.16	23.	Metal	6.14
10.	Hitam	25.35	24.	Hitam	24.53

11.	Biru	25.84	25.	Metal	6.28
12.	Metal	6.48	26.	Biru	24.44
13.	Hitam	25.15	27.	Hitam	26.25
14.	Metal	6.04			



Gambar 4.92 Grafik waktu pengujian proses setiap jenis benda kerja

Dari pengujian sebanyak 27 kali didapatkan grafik waktu setiap jenis benda kerja seperti pada **Gambar 4.92**. Grafik waktu proses yang paling stabil diperoleh benda kerja metal abu-abu dengan rata-rata waktu proses 6.25 sekon. Hal ini dikarenakan benda kerja metal abu-abu harus berhenti di proses *ejecting*. Sedangkan waktu proses terpanjang adalah benda kerja plastik hitam dengan waktu proses rata-rata 25.89 sekon. Penyebabnya adalah proses benda kerja plastik hitam merupakan proses yang terlama. Apabila dibandingkan dengan benda kerja plastik biru maka selisih rata-rata waktu proses sebesar 0.78 sekon lebih lama.

BAB 5

PENUTUP

Pada bab terakhir ini penulis memberikan kesimpulan dari hasil penelitian serta saran dari kendala yang telah dihadapi agar dapat dilakukan perbaikan pada penelitian selanjutnya.

5.1 Kesimpulan

Setelah dilakukan proses perancangan dan pengujian sistem maka dapat ditarik beberapa kesimpulan, antara lain :

1. Perancangan diagram *ladder* menggunakan metode *Cascade* untuk banyak objek pada *Factory Automatic Trainer* menghasilkan 26 input serta 113 *rung* dengan detail 63 relay, 9 timer, 18 counter dan 23 output.
2. Waktu proses rata-rata benda kerja dalam 27 kali pengujian adalah 25.89 sekon untuk material plastik berwarna hitam, 25.11 sekon untuk material plastik berwarna biru dan 6.25 sekon untuk material metal berwarna abu-abu.
3. Nilai jangkauan waktu proses rata-rata benda kerja plastik berwarna hitam dan biru adalah sama yaitu 3.49 sekon.
4. Hasil *buid all* dari diagram *ladder GMWIN4* yaitu besar program 20126 Bytes (16%, Max. 128KB), besar data 4452 Bytes (9%, Max. 52KB) dan besar *upload file* 17376 Bytes (14%, Max. 128KB).

5.2 Saran

Dari beberapa kendala yang dialami selama penelitian maka perlu adanya saran agar dapat dilakukan perbaikan untuk topik selanjutnya :

2. Pastikan benda kerja terdeteksi oleh sensor *Magazine* agar berjalan lancar dengan mengubah letak posisi benda kerja.
3. Perlu dilakukan penggantian pada *limit switch Insert Return* karena sudah tidak dapat bekerja.
4. Kompresor yang digunakan saat ini memiliki sedikit kebocoran pada valve keran maka lakukan perbaikan terlebih dahulu atau gunakan kompresor yang lebih baik agar modul berjalan dengan maksimal.

[Halaman ini sengaja dikosongkan]

DAFTAR PUSTAKA

- [1] Wicaksono, Handy, “SCADA *Software* dengan *Wonderware InTouch* : Dasar-dasar Pemrograman”, Graha Ilmu, Yogyakarta, 2011.
- [2] Bolton, W., “*Programmable Logic Controller*” Elsevier, USA, 2009.
- [3] Anonim, “*CPE-AT8030N Factory Automatic Trainer*”, Chungpa.
- [4] Pc-control.co.uk, “*Capacitive Proximity Sensors*” <URL:<https://pc-control.co.uk/Capacitive.htm>>, 2008.
- [5] Anonim, “*Electrical Capacitive Type Proximity Sensor : CR18-8DN*”, Autonics.
- [6] Anonim. “*What is Sensor? Sensor Basics : Understanding Fundamental Principles and Features*” Keyence, USA, 2015.
- [7] Anonim, “*Cylindrical Type Proximity Sensor : PR18-8DN*”, Autonics.
- [8] Anonim, “*DC Cylindrical Housing Type Photoelectric Sensor : BR100-DDT*”, Autonics.
- [9] Anonim, “*Fiber Optic Cable Diffuse Reflective Type : Model FD-320-05*”, Autonics.
- [10] Witanto, Aji Prasetyo, “Konstruksi Diagram *Ladder* Dengan Metode *Huffman* untuk *Factory Automatic Trainer*” Institut Teknologi Sepuluh Nopember, Surabaya, 2018.
- [11] Vcomp Inc, “*What are Single Acting & Double Acting Hydraulic Cylinders*” <URL:<https://medium.com/@vcompmississaug/what-are-single-acting-double-acting-hydraulic-cylinders-817d6e6f30a8>>, Desember, 2017.
- [12] Festo, “*Symbols of Fluid Power : Pneumatics*” <URL:<https://www.festo-didactic.com/int-en/services/symbols/fluid-power-pneumatics/?fbid=aW50LmVuLjU1Ny4xNy4zNC44NDg&page=1&offset=0&showitems=64>>, 2019.
- [13] Anonim, “*LED Sign Tower*”, Hynux.

- [14] Gurocak, Hakan, "*Industrial Motion Control : Motor Drives Controller Tuning Applications*", Wiley, USA, 2016.
- [15] Anonim, "*Modular Type Air Combination : Series PP2*", TPC.
- [16] Pessen, David W., "*Industrial Automation : Circuit Design and Components*", Haifa, Israel, 1989.
- [17] Putra, Gilang Permana, "Konstruksi Diagram *Ladder* Menggunakan Metode *Cascade* untuk *Factory Automatic Trainer*", Institut Teknologi Sepuluh Nopember, Surabaya :. 2018.
- [18] Iskandar, E., Rameli, M., Adiyat, R., "*Ladder Diagram Design Based On Cascade Method For Selection And Assembling Part On Dual Conveyor Plant*", Journal on Advanced Research in Electrical Engineering, Vol.2, pp.37-43, Oktober 2018.

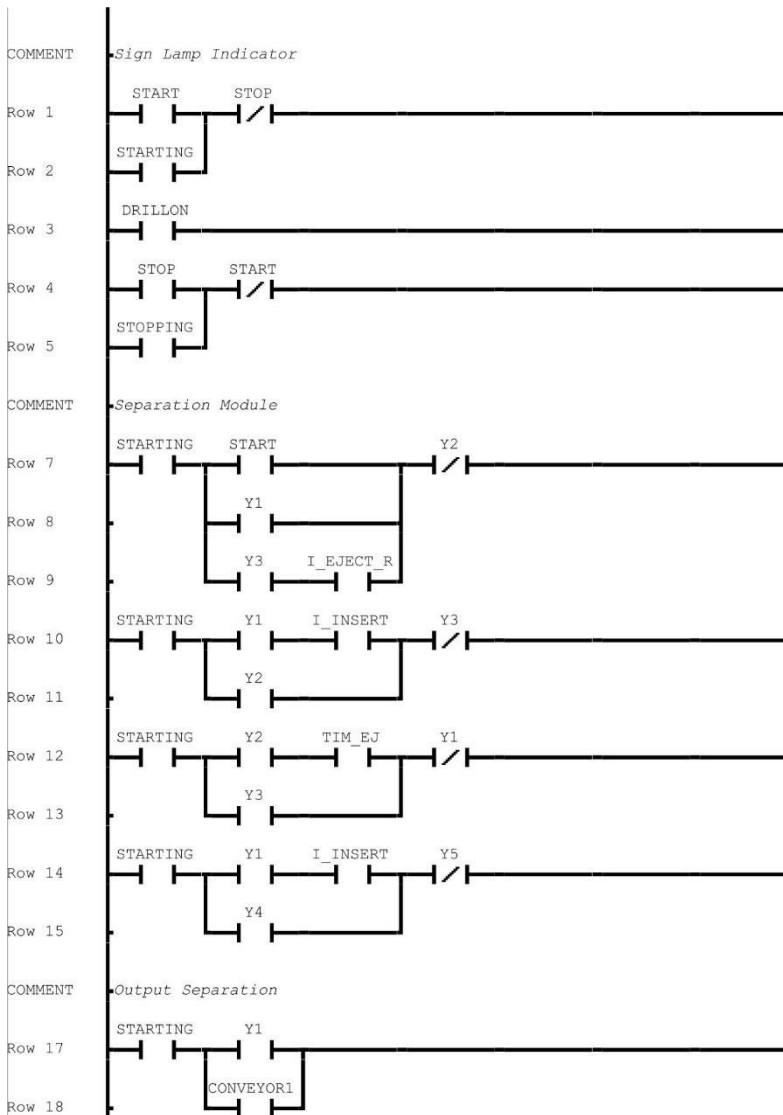
LAMPIRAN

Lampiran 1

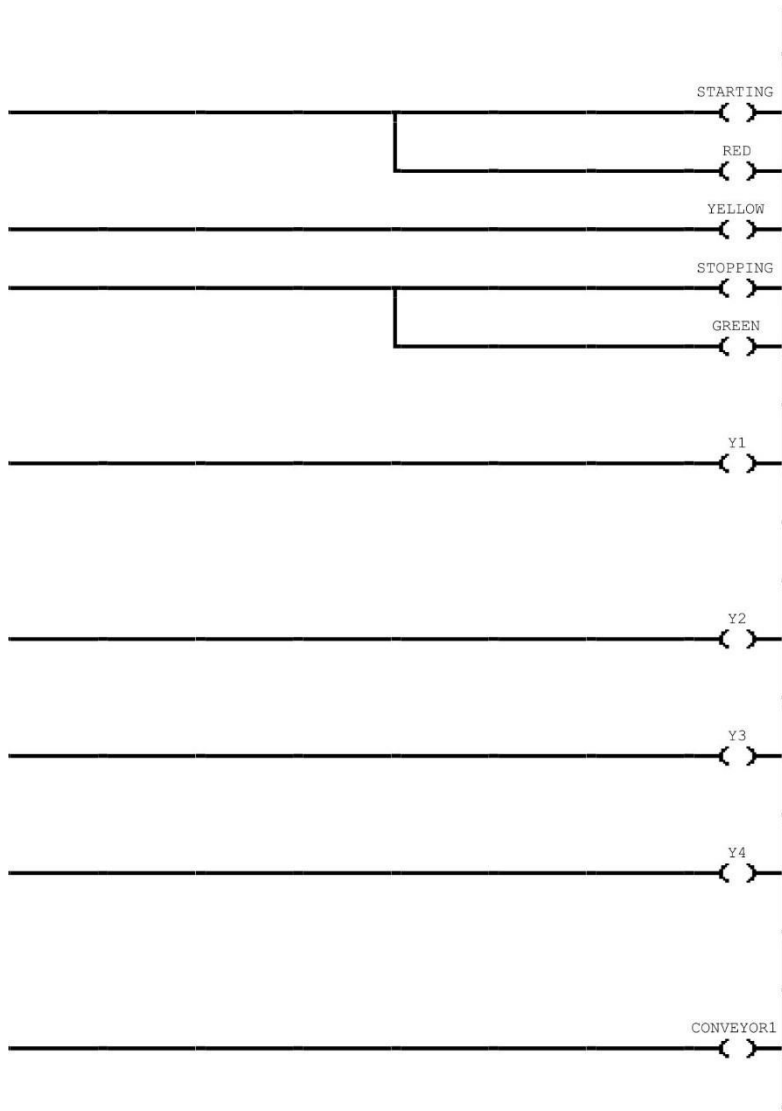
Program

noname00.src

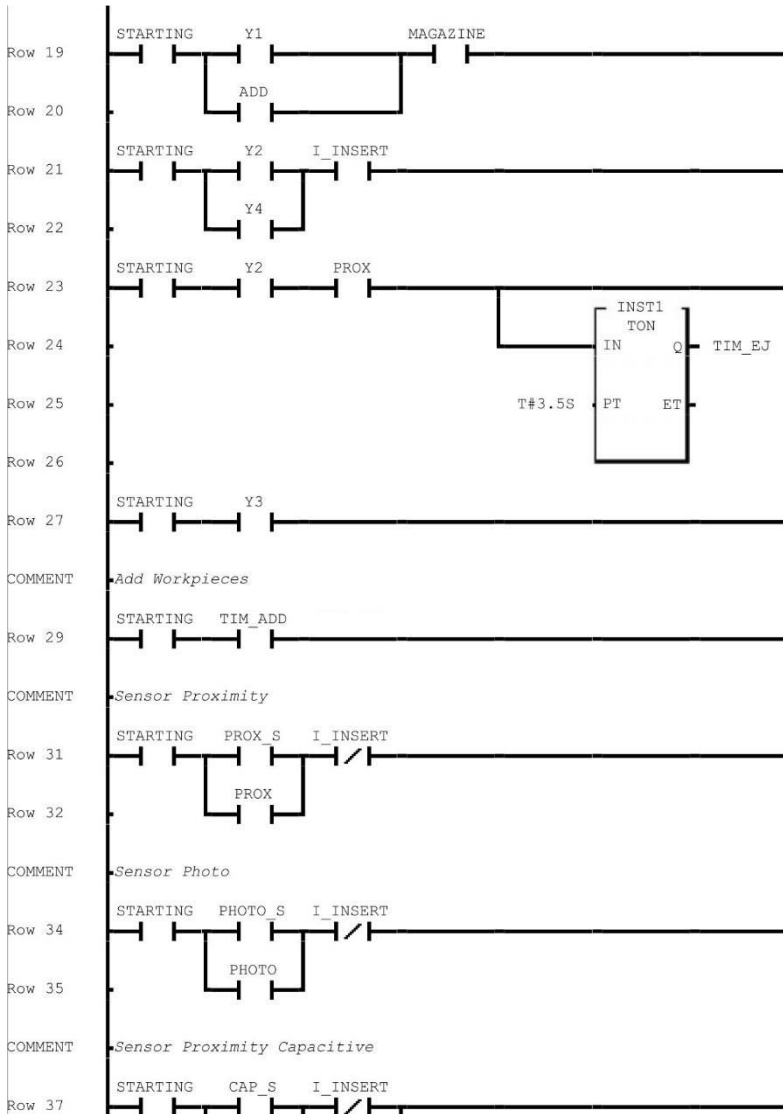
Lampiran 2



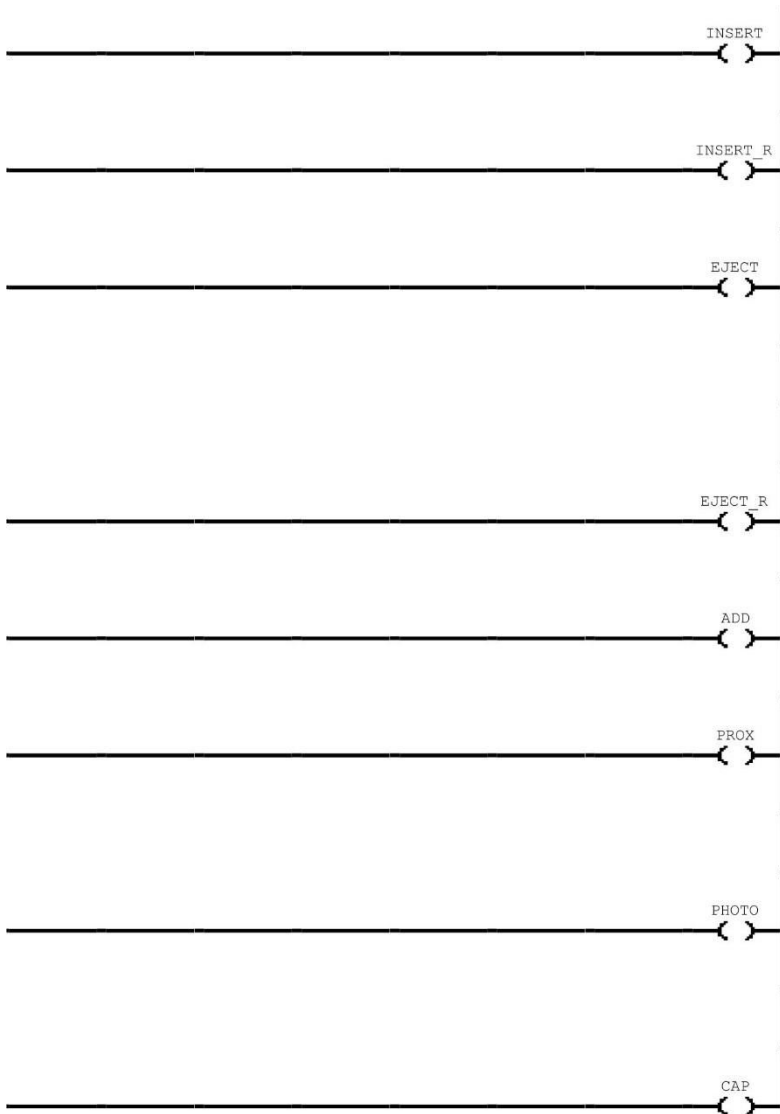
Lampiran 3



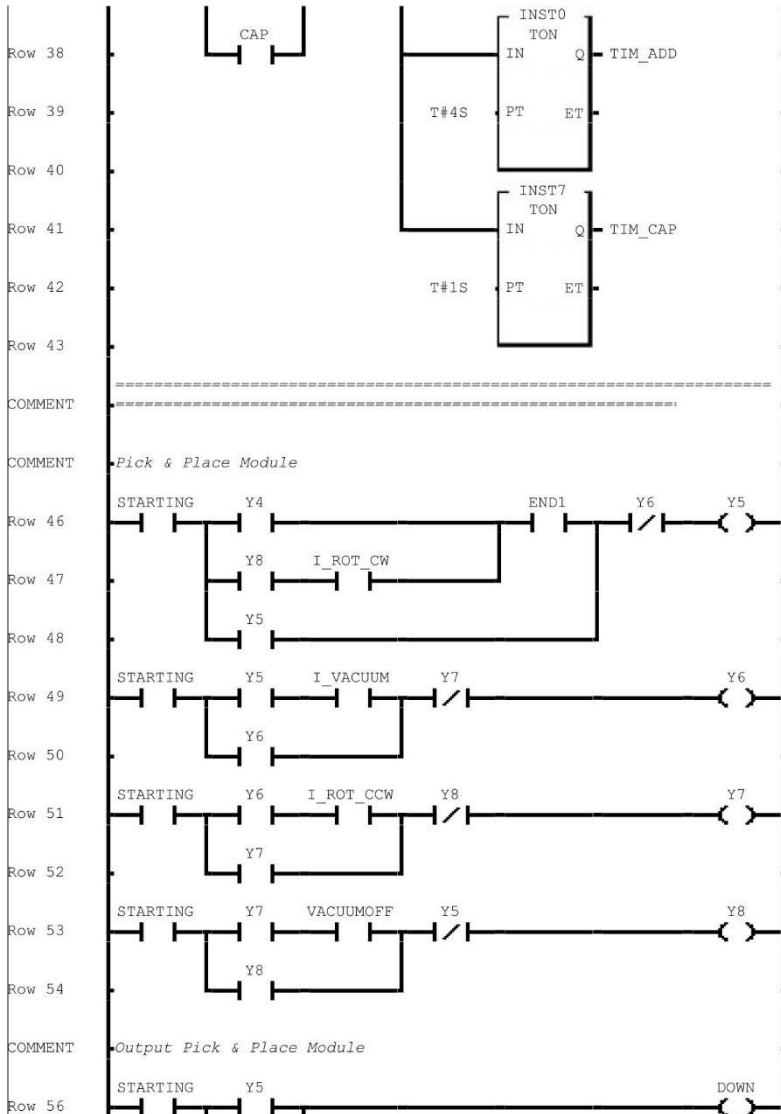
Lampiran 4



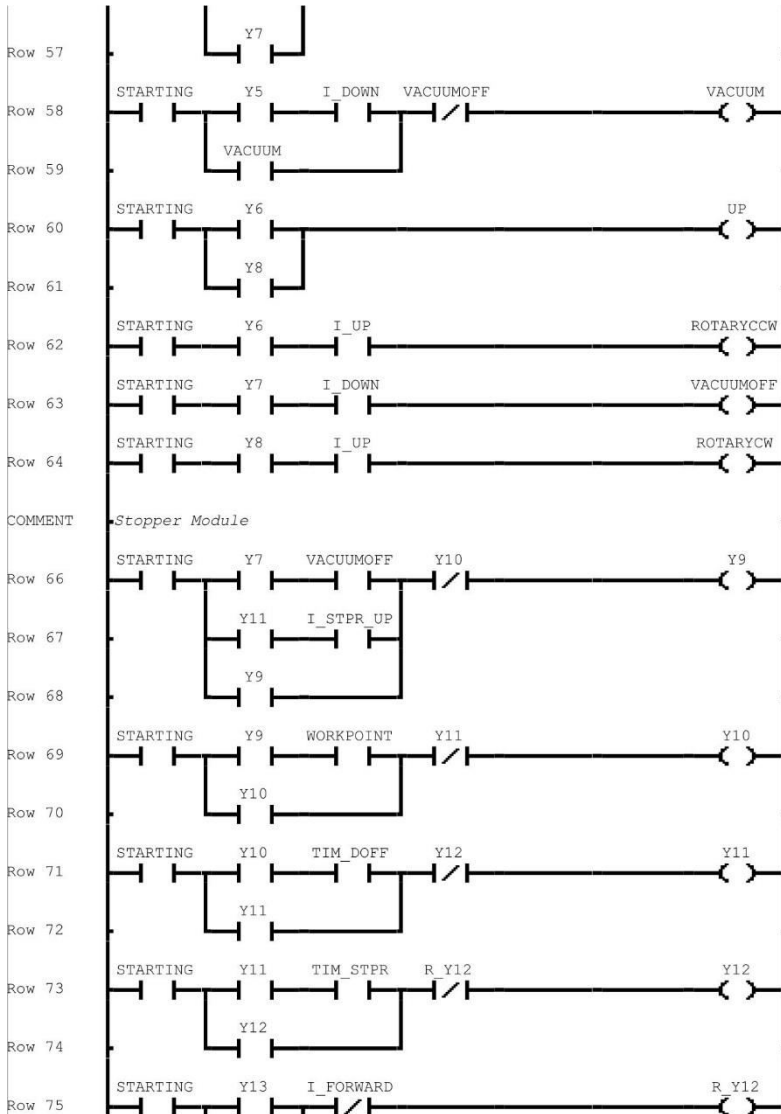
Lampiran 5



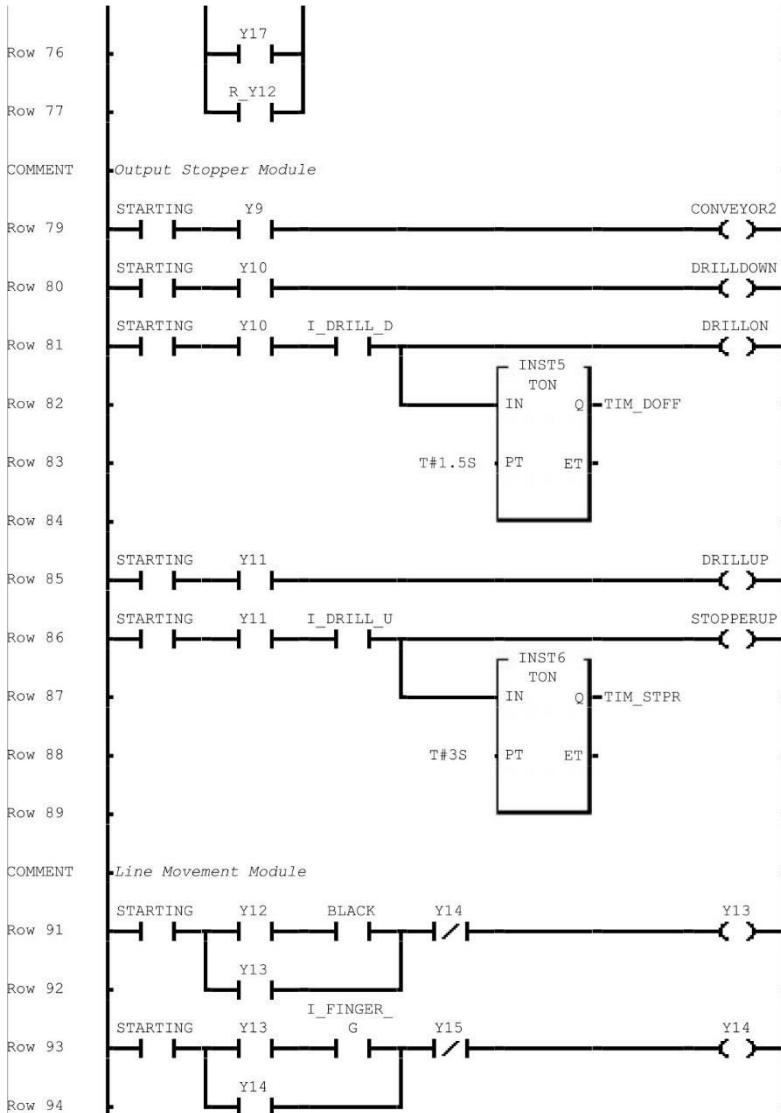
Lampiran 6



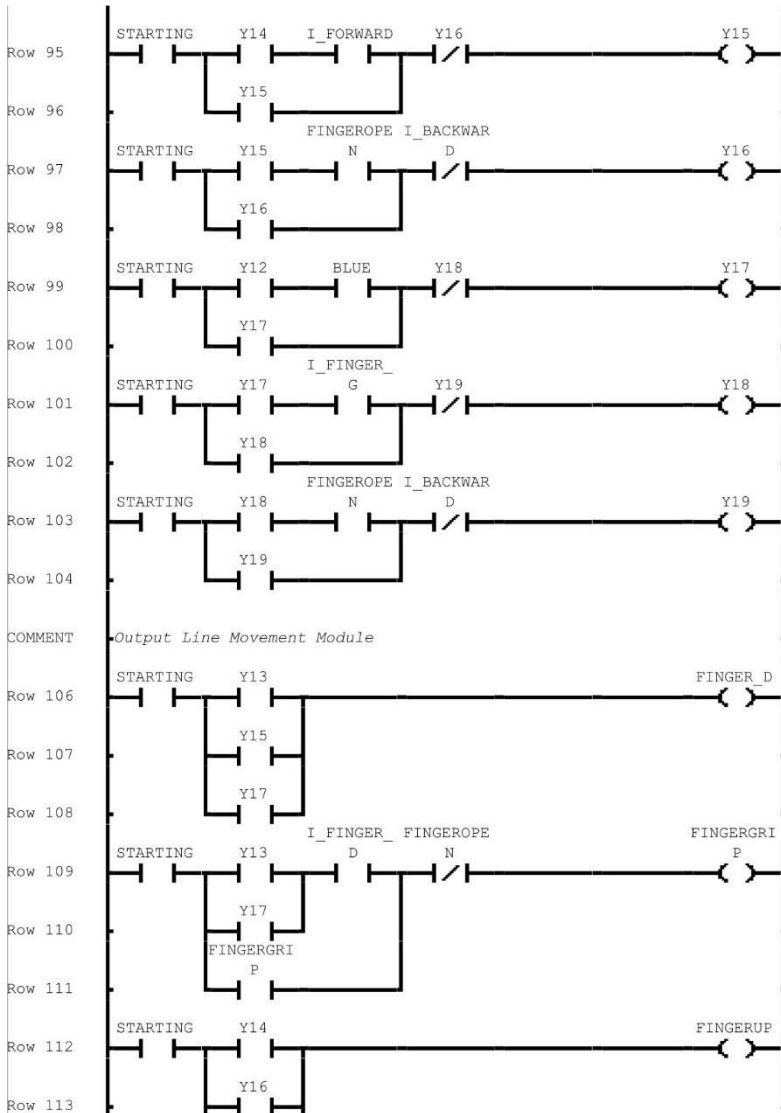
Lampiran 7



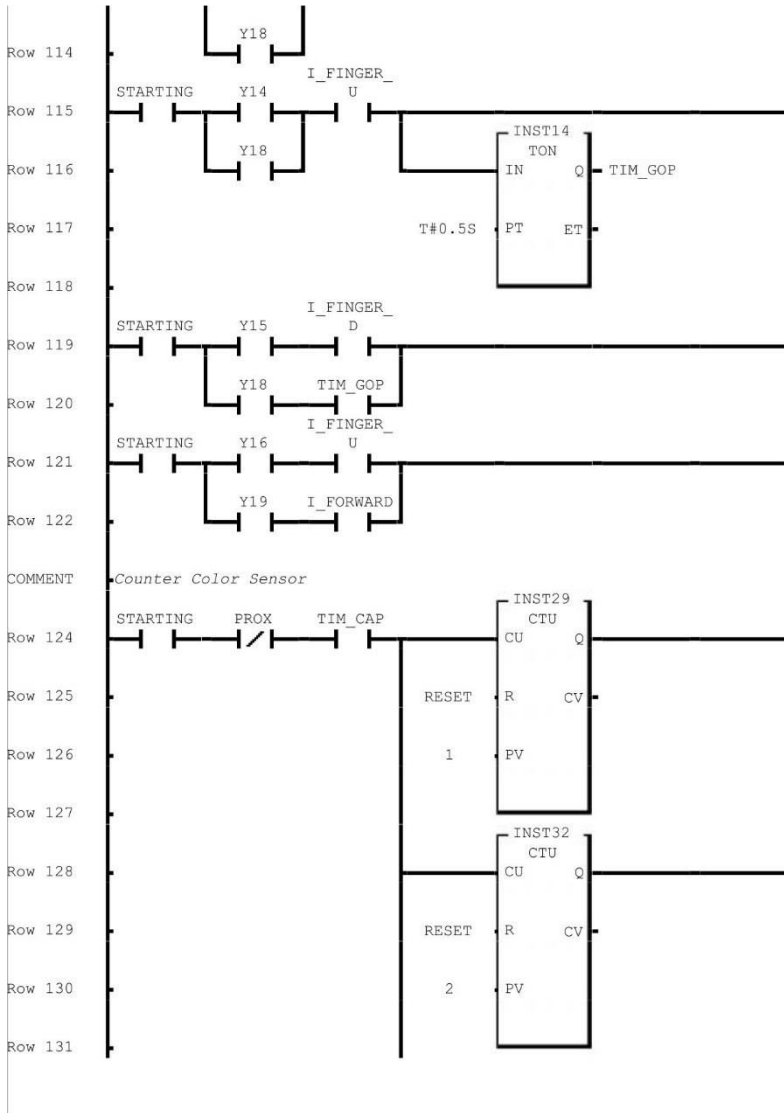
Lampiran 8



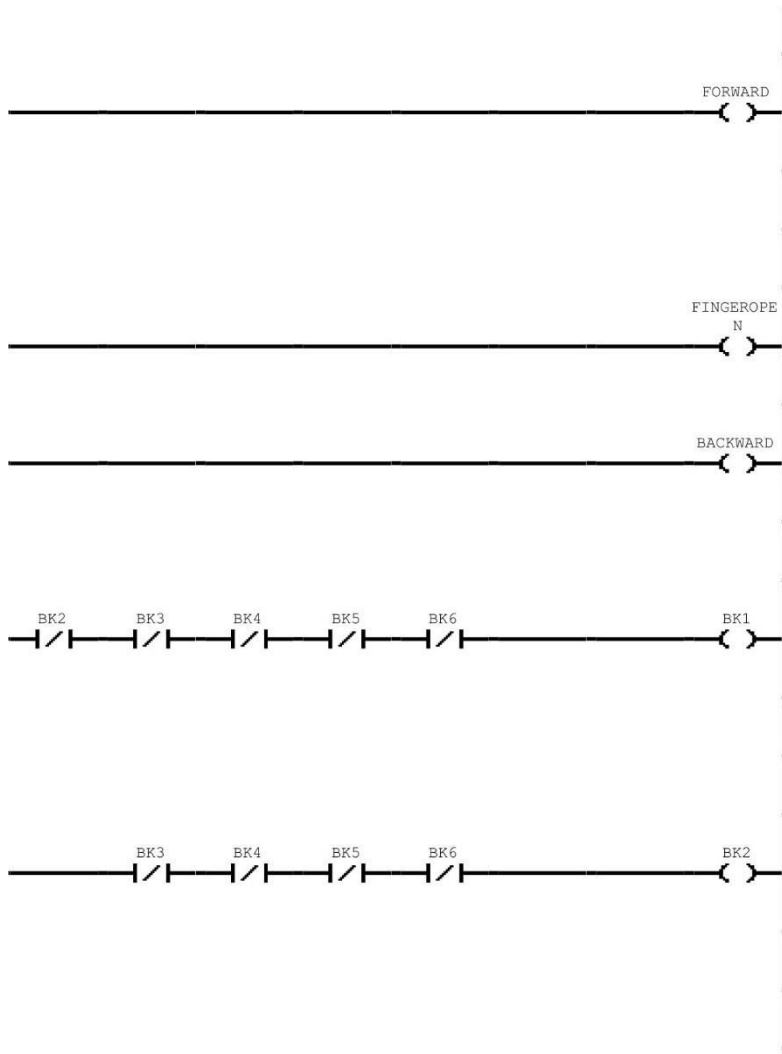
Lampiran 9



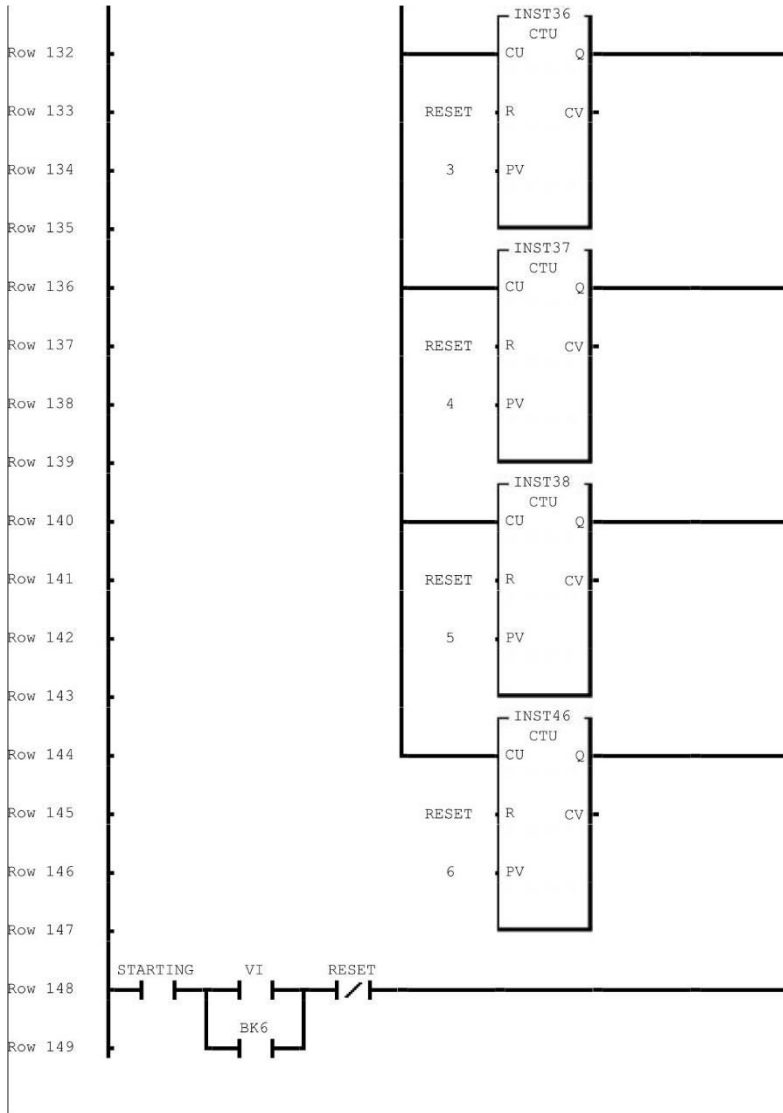
Lampiran 10



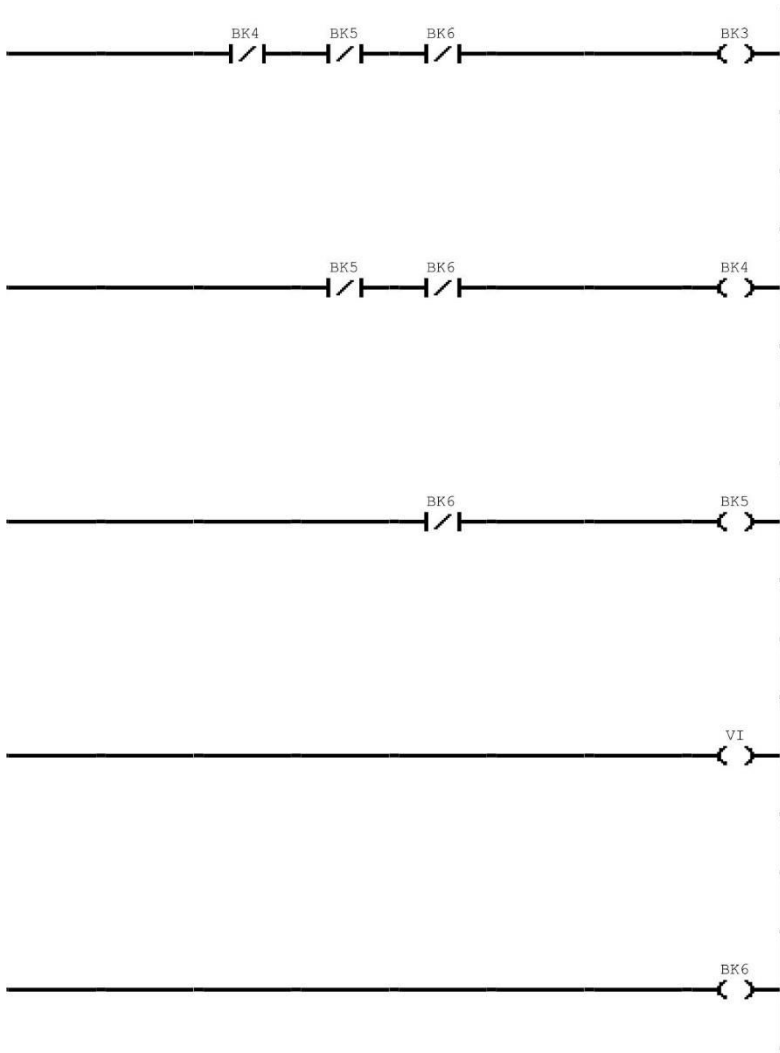
Lampiran 11



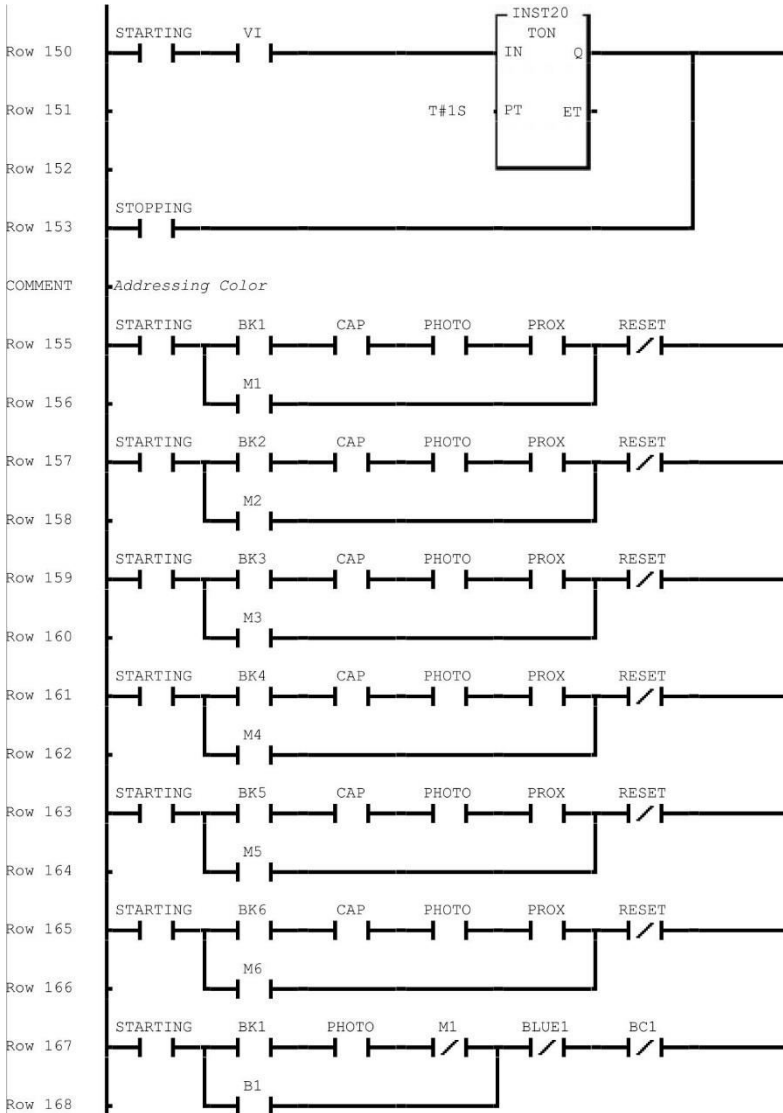
Lampiran 12



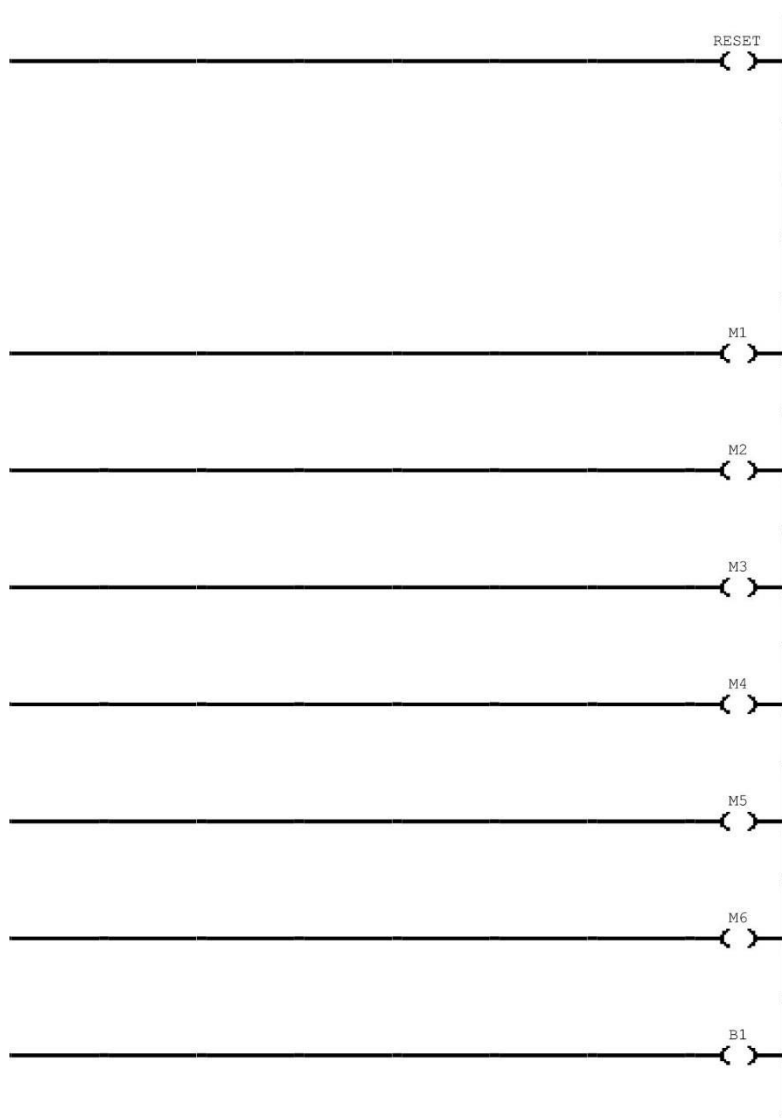
Lampiraan 13



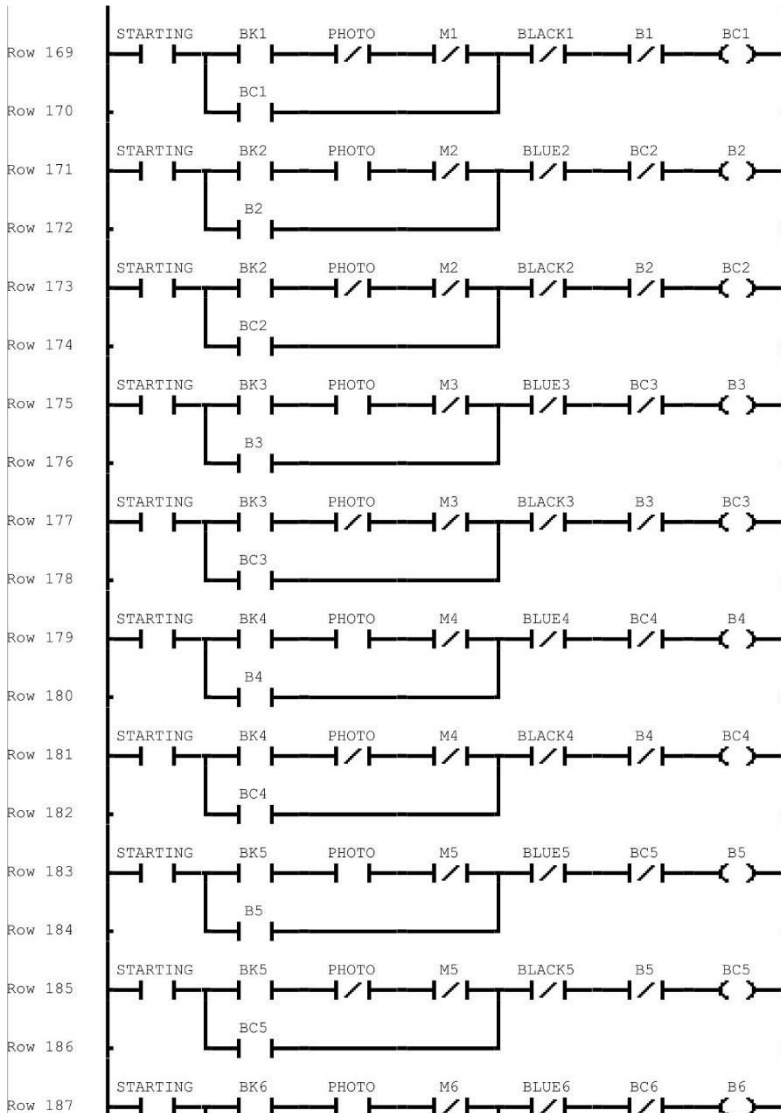
Lampiran 14



Lampiran 15

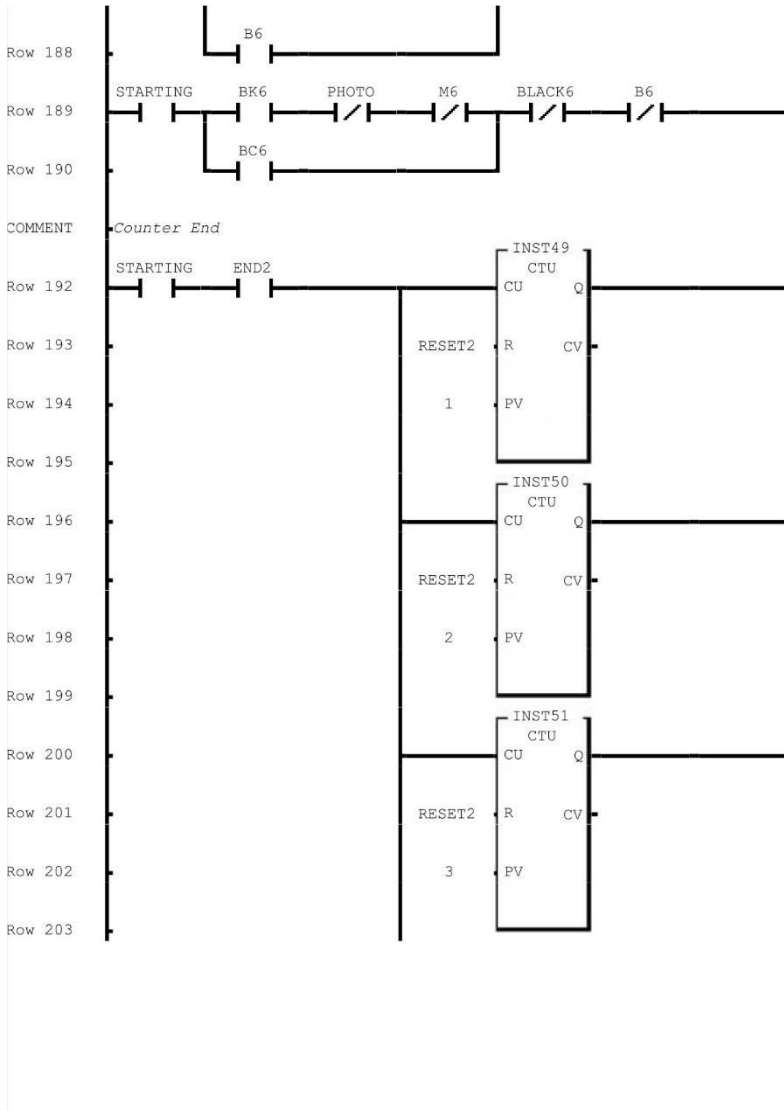


Lampiran 16

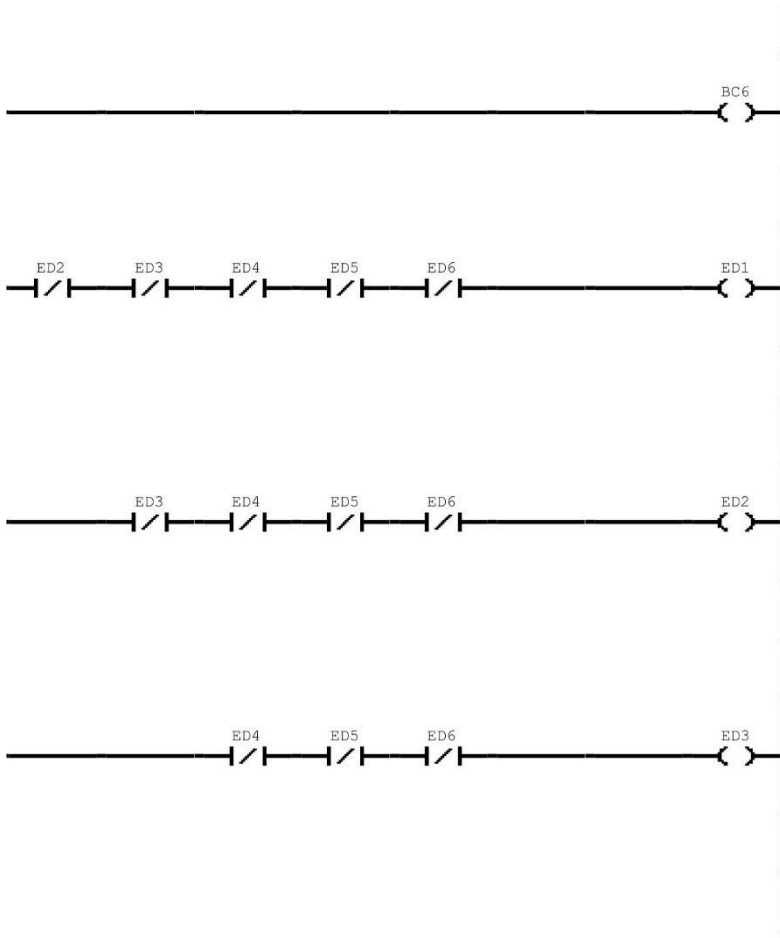


[Halaman ini sengaja dikosongkan]

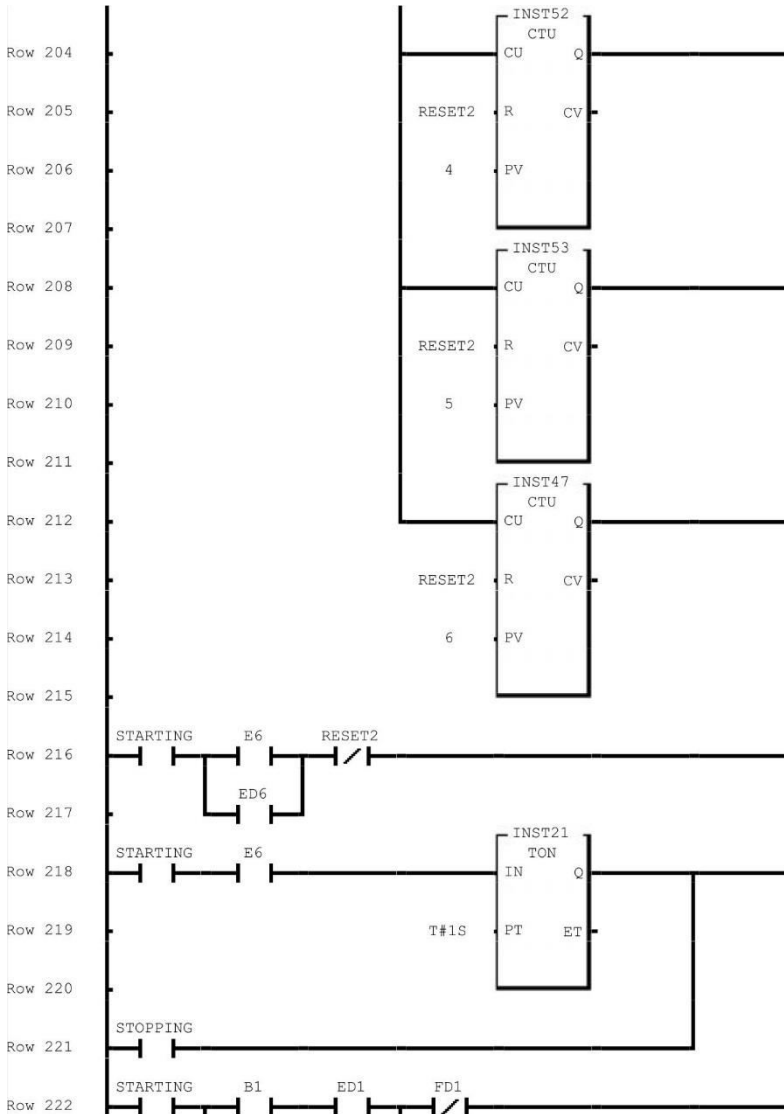
Lampiran 17



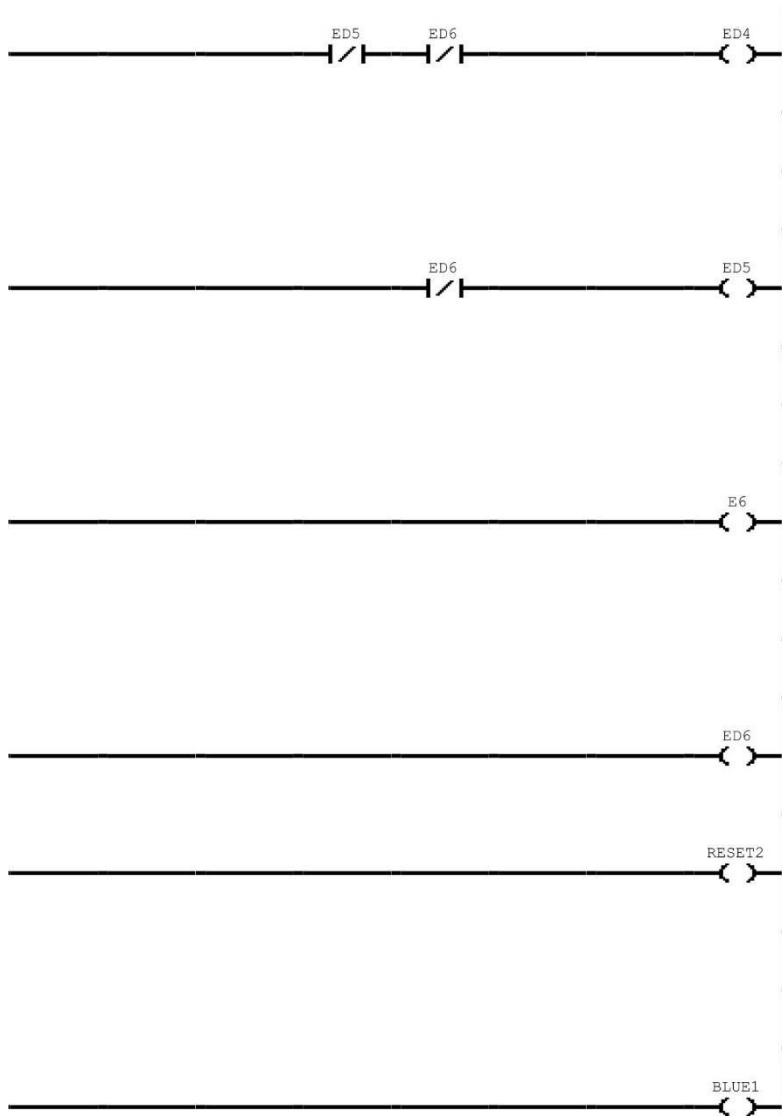
Lampiran 18



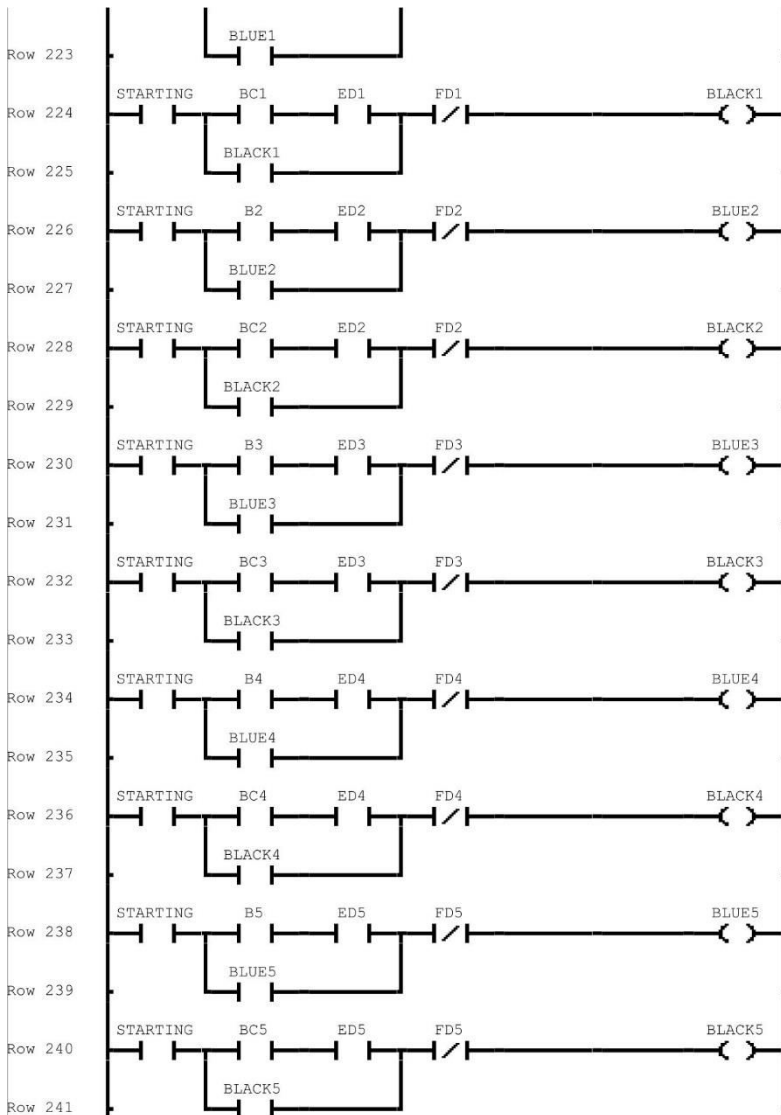
Lampiran 19



Lampiran 20

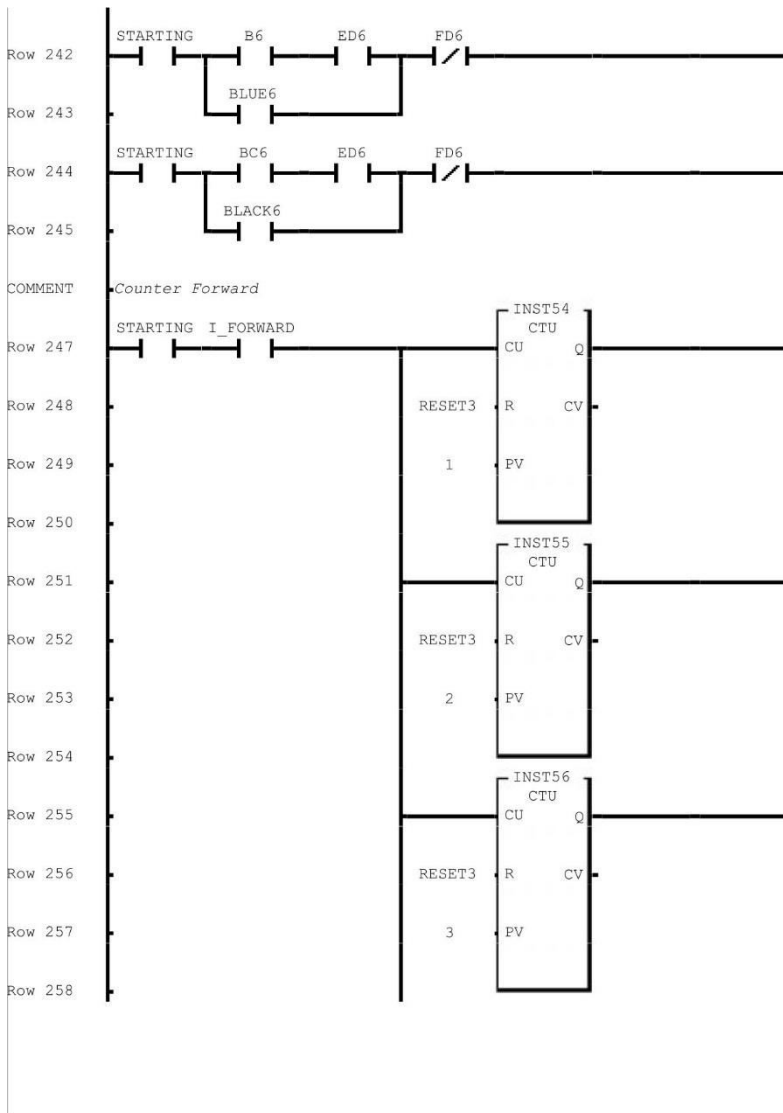


Lampiran 21

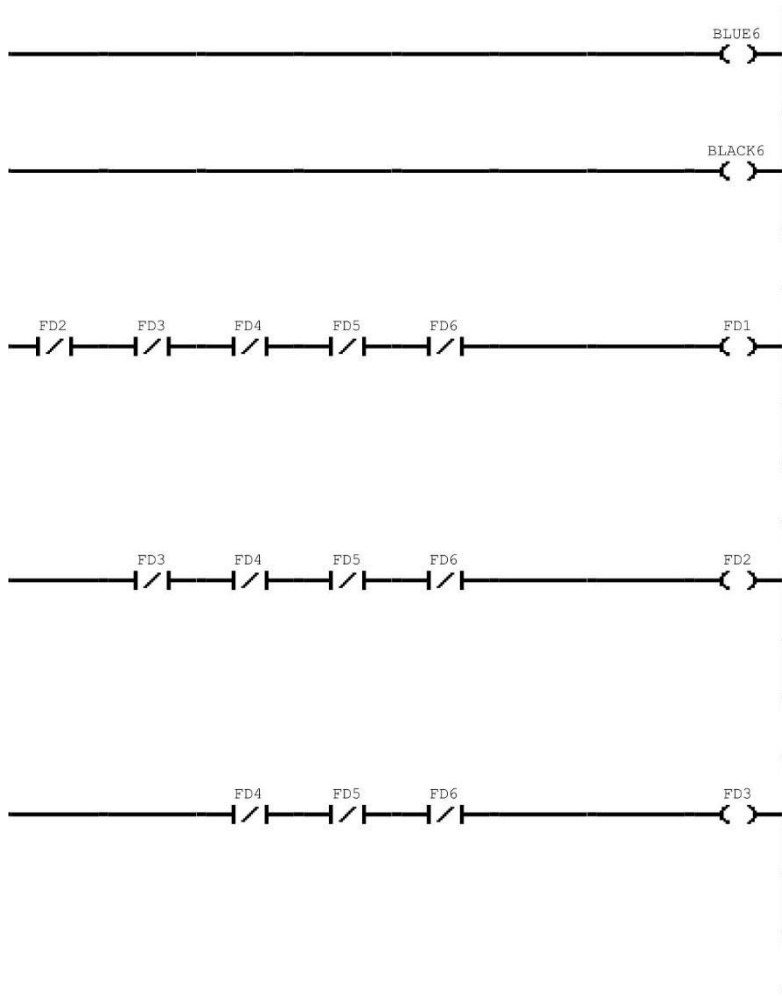


[Halaman ini sengaja dikosongkan]

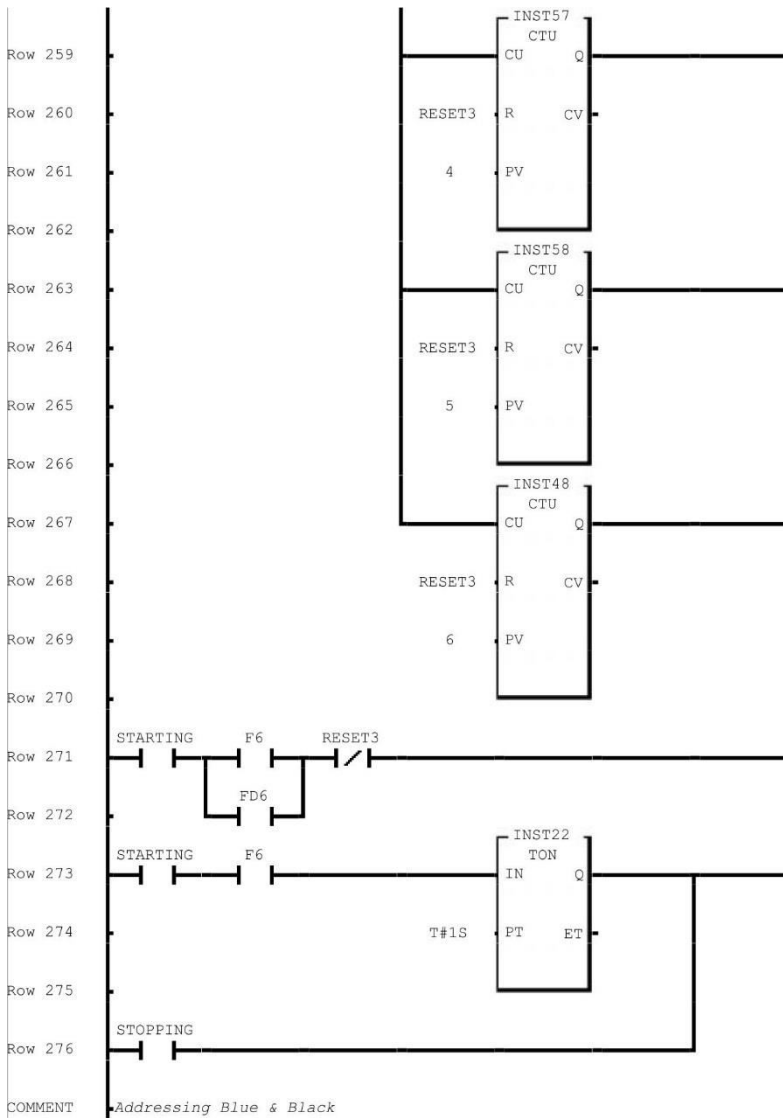
Lampiran 22



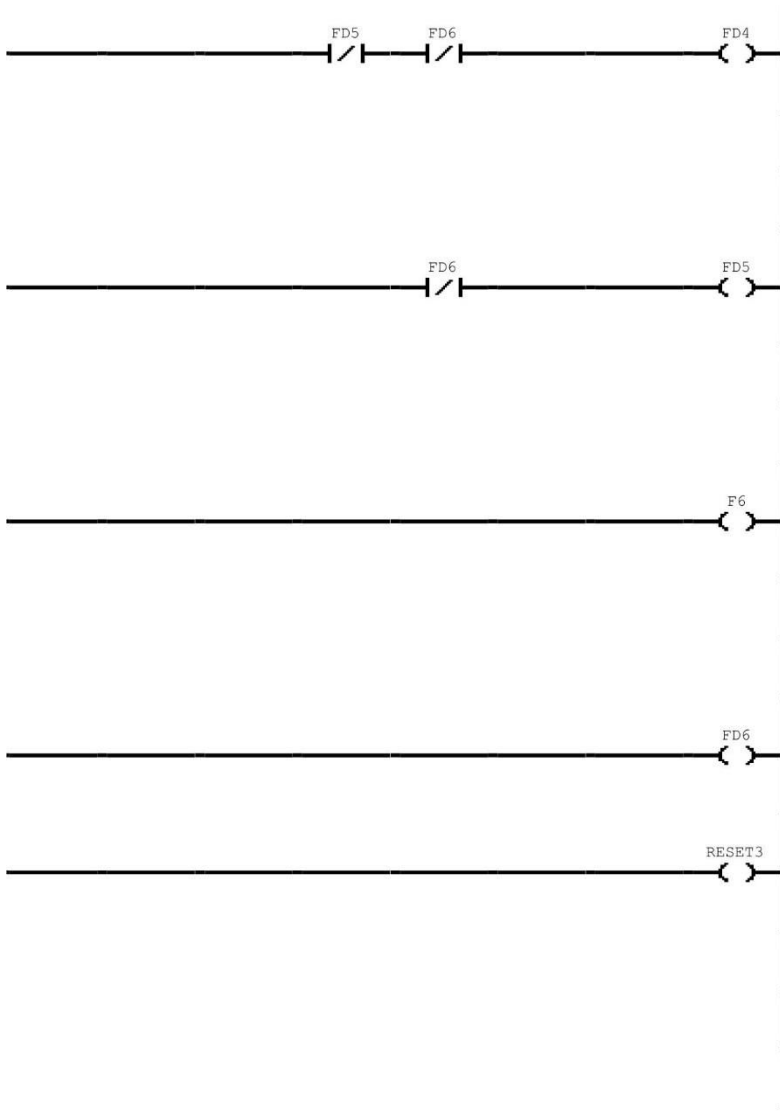
Lampiran 23



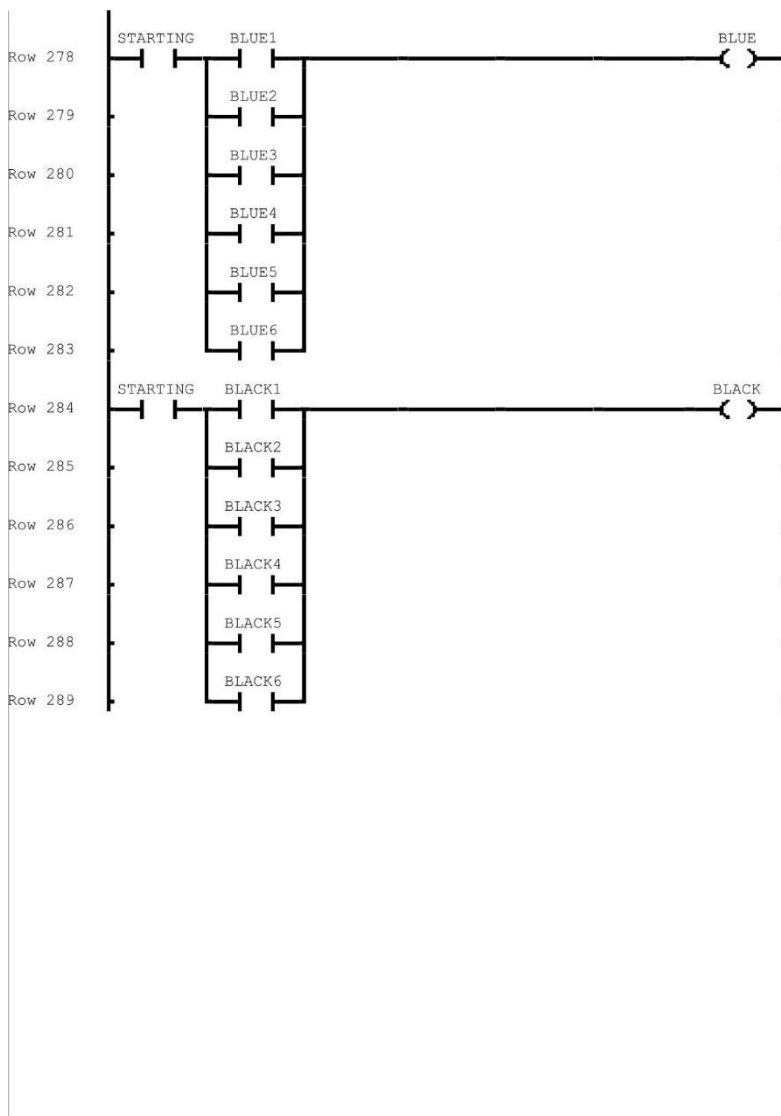
Lampiran 24



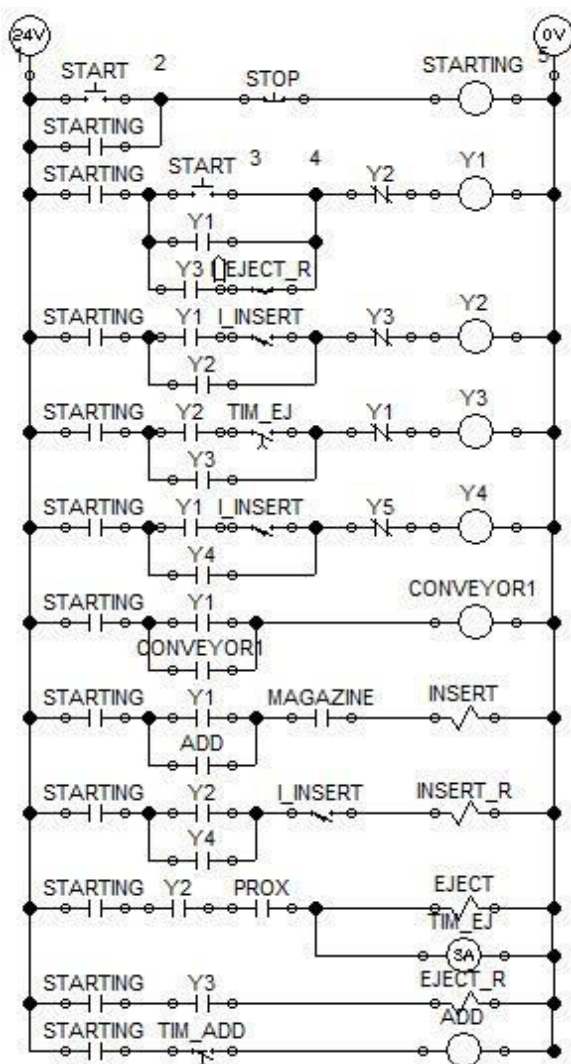
Lampiran 25



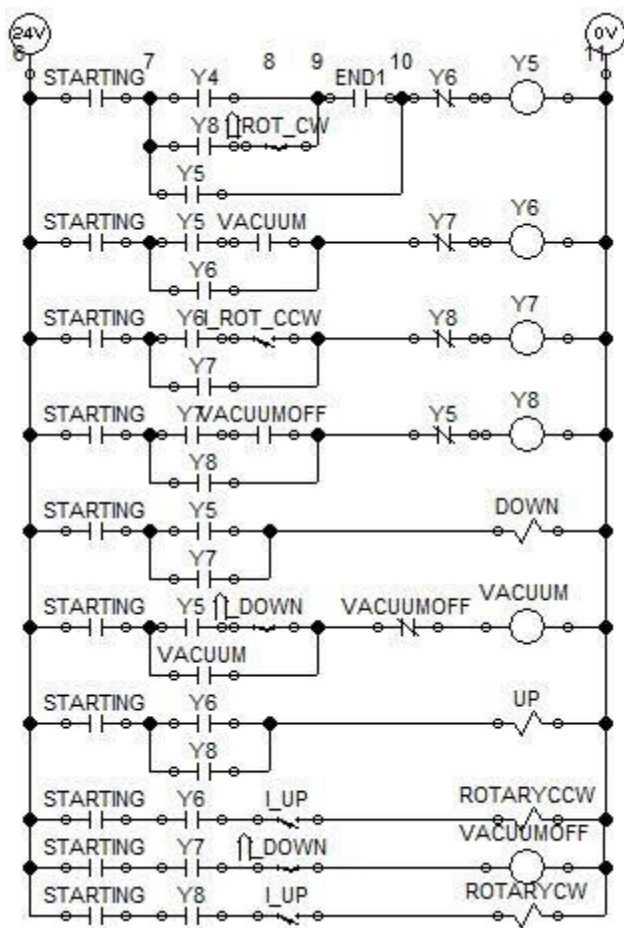
Lampiran 26



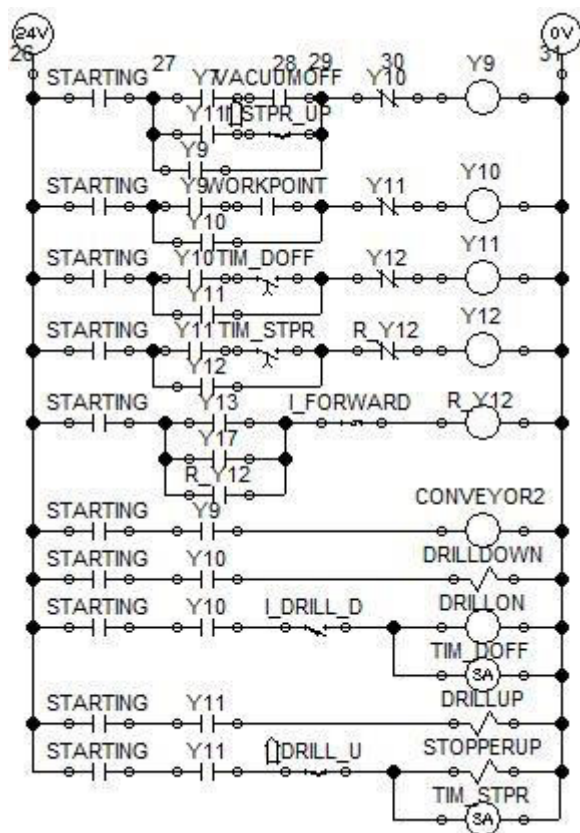
Lampiran 27



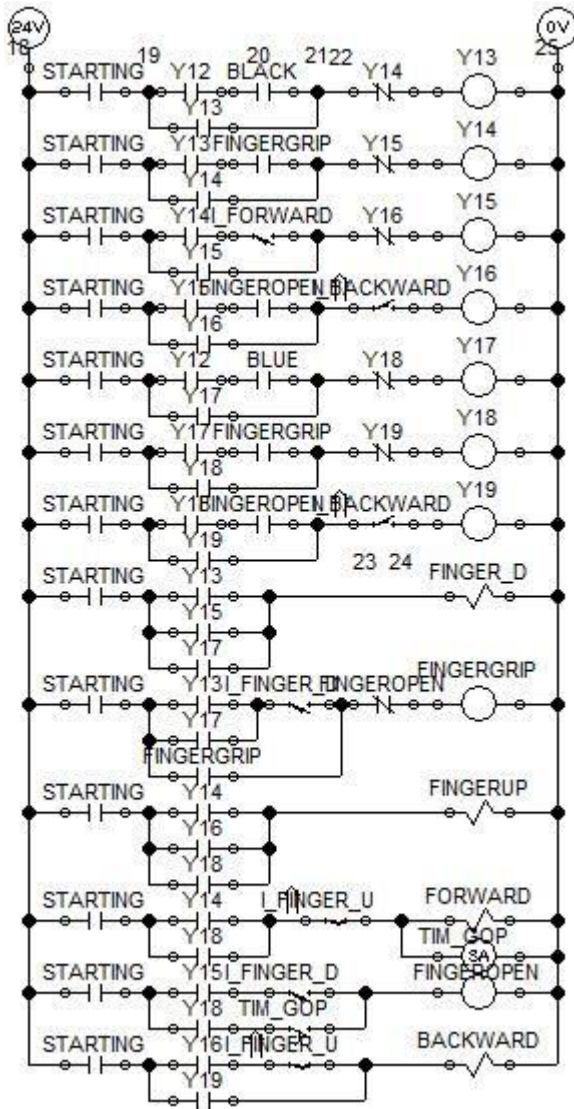
Lampiran 28



Lampiran 29

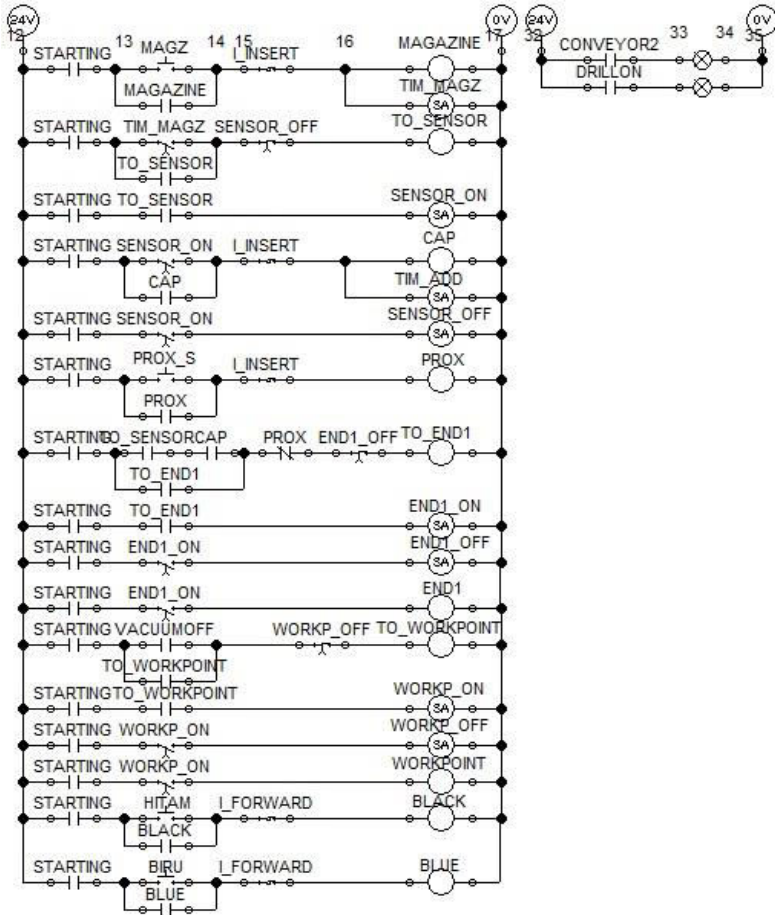


Lampiran 30



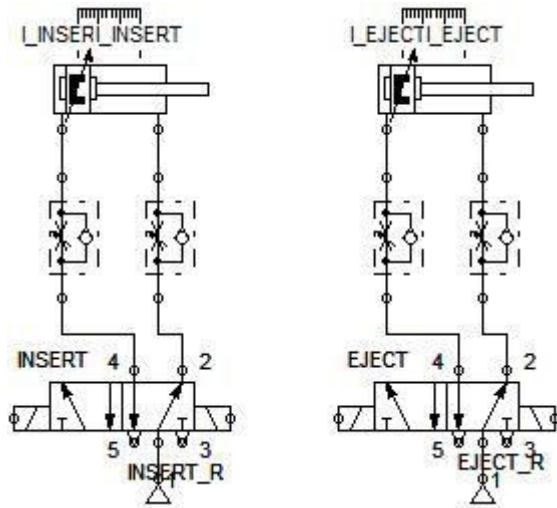
Lampiran 31

Ladder diagram otomatis pemicu gerakan

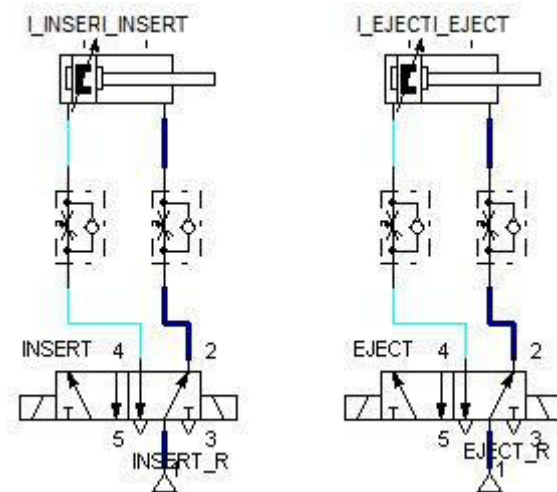


Lampiran 32

Pneumatik modul *separation*

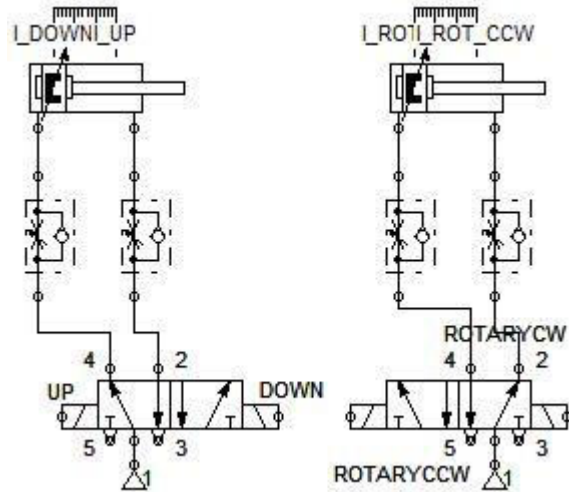


Kondisi inisial dari pneumatik modul *separation*

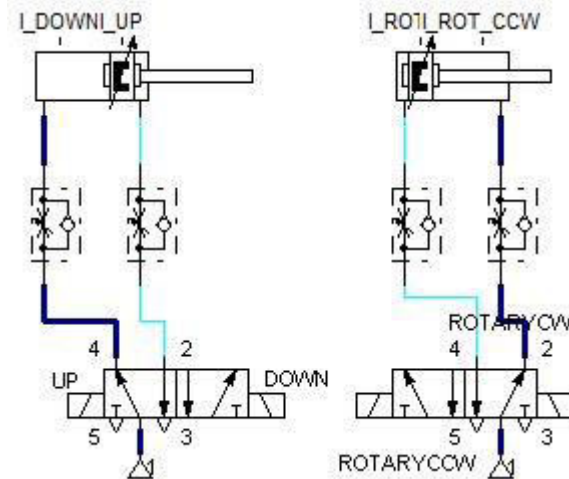


Lampiran 33

Pneumatik modul *pick & place*

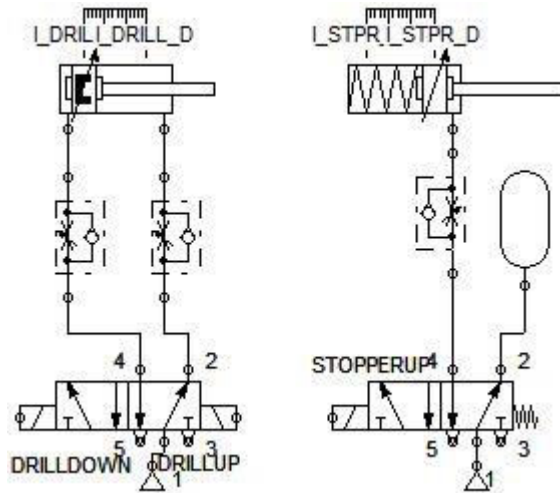


Kondisi inisial dari pneumatik modul *pick & place*

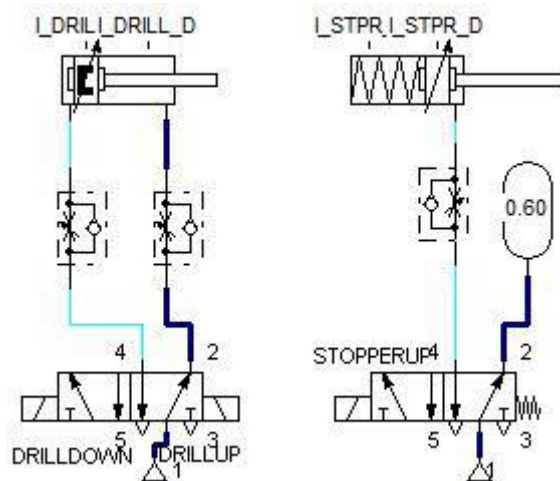


Lampiran 34

Pneumatik modul *stopper*

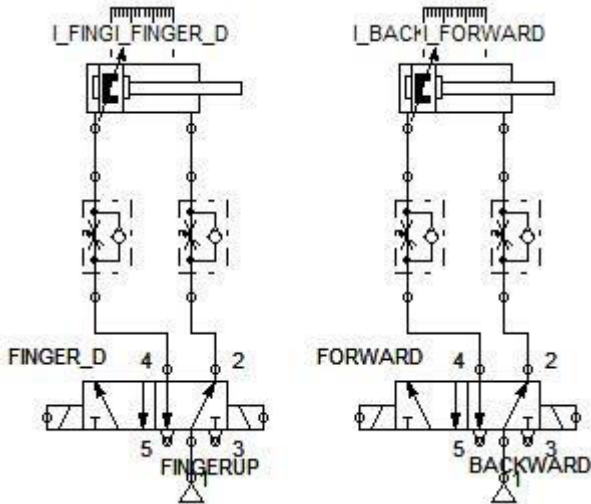


Kondisi inisial dari pneumatik modul *stopper*

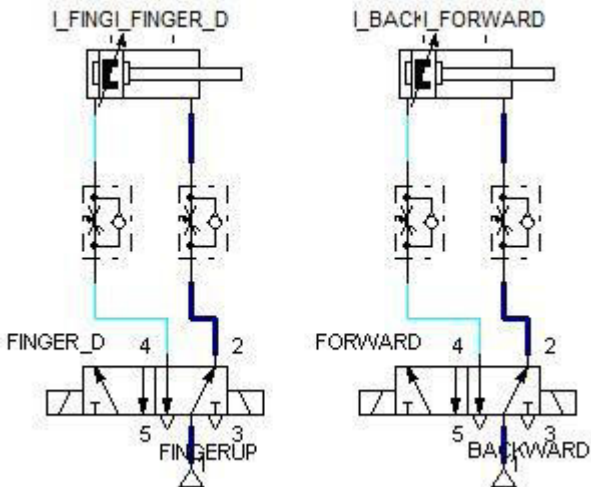


Lampiran 35

Pneumatik modul *line movement*



Kondisi inisial dari pneumatik *line movement*



[Halaman ini sengaja dikosongkan]

RIWAYAT HIDUP



Izzuddin Al Qossam, lahir di Sidoarjo, Jawa Timur pada tanggal 6 Desember 1996. Putra kedua dari pasangan Bapak Etnanto Yuwono dan Ibu Haneny Rudjiati. Penulis memulai pendidikan formal dari SD IT Nurul Iman Semarang, MTs Baitussalam Semarang dan MA Kafila Jakarta. Setelah lulus dari jenjang SMA/MA pada tahun 2014, penulis melanjutkan studi S1 jurusan Teknik Elektro di Institut Teknologi Sepuluh Nopember (ITS) Surabaya dengan mengambil fokus bidang studi Teknik Sistem Pengaturan. Buku tugas akhir ini merupakan salah satu karya penulis sebagai syarat kelulusan tingkat *Strata-1*.

