



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - TE 145561

**RANCANG BANGUN MEKANIK DAN PEMROGRAMAN
PLC PADA *AUTOMATION SORTING LINE SYSTEM***

Vrenky Meidianto
NRP. 1031150000044

Pembimbing
Eko Setijadi, S.T., M.T., Ph.D.
Imam Arifin, S.T., M.T.
Mohammad Hafid, S.Pd.

DEPARTEMEN TEKNIK ELEKTRO OTOMASI
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember
Surabaya 2018



ITS
Institut
Teknologi
Sepuluh Nopember

FINAL PROJECT - TE 145561

**MECHANICAL DESIGN AND PLC PROGRAMMING ON
AUTOMATION SORTING LINE SYSTEM**

Vrenky Meidianto
NRP. 1031150000044

Supervisors
Eko Setijadi, S.T., M.T., Ph.D.
Imam Arifin, S.T., M.T.
Mohammad Hafid, S.Pd.

DEPARTMENT OF ELECTRICAL AND AUTOMATION ENGINEERING
Faculty of Vocations
Institut Teknologi Sepuluh Nopember
Surabaya 2018

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul:

“RANCANG BANGUN MEKANIK DAN PEMROGRAMAN PLC PADA *AUTOMATION SORTING LINE SYSTEM*”

adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya saya 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, 10 Juli 2018



Vrenky Mejdianto
NRP 1031150000044

RANGKAI BANGUN MEKANIK DAN PEMROGRAMAN PLC PADA AUTOMATION SORTING LINE SYSTEM



TUGAS AKHIR

Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Ahli Madya Teknik

Pada
Departemen Teknik Elektro Otomasi

Fakultas Vokasi

Institut Teknologi Sepuluh Nopember



Menyebut:



Pembimbing 1

Pembimbing 2

Pembimbing 3



Setiadi, S.T., M.T., Ph.D.

Imam Arifin, S.T., M.T.

Mohammad Hafid, S.Pd.

NIP. 19721001200312110

NIP. 197302222002121001



SURABAYA
JUNI, 2018



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

RANCANG BANGUN MEKANIK DAN PEMROGRAMAN PLC PADA AUTOMATION SORTING LINE SYSTEM



TUGAS AKHIR

Disusun oleh:

Vrenky Meidianto

1031150000044

Mengajar:

FT Industri: Robotics Automation

Pembimbing Lapangan



Mohammad Hafid, S.Pd.

HRD



Puspita Kinasih Santya Putri, S.Psi.

RANCANG BANGUN MEKANIK DAN PEMROGRAMAN PLC PADA *AUTOMATION SORTING LINE SYSTEM*

Vrenky Meidianto
1031150000044

Pembimbing I : Eko Setijadi, S.T., M.T., Ph.D.
Pembimbing II : Imam Arifin, S.T., M.T.
Pembimbing III : Mohammad Hafid, S.Pd.

ABSTRAK

Penerapan teknologi pada industri mulai berkembang dengan pesat. Salah satunya dalam proses pemilihan barang atau produk tertentu untuk dipasarkan maupun disimpan. Untuk mendapatkan hasil yang maksimal dan tingkat ketelitian yang tinggi diperlukan sebuah sistem beteknologi canggih atau dapat disebut dengan sistem otomatis untuk memilah produk sesuai yang diinginkan agar hasil produksi meningkat dan mengurangi kesalahan pemilhan. Pada tugas akhir ini dibuat *Automation Sorting Line System* menggunakan pendeteksi *barcode*. Dalam proses pemilihan produk akan dipilah berdasarkan *barcode* yang ada pada setiap produk. Dalam prosesnya sistem berteknologi ini akan membaca *barcode* pada produk setelah itu akan memilah sesuai dengan kode tersebut. Sistem ini mengurangi kekeliruan yang diakibatkan pemilhan secara manual oleh manusia. Dari hasil implementasi alat bekerja sesuai dengan apa yang dikehendaki yaitu memilah produk menggunakan *barcode* dan langsung menempatkan pada tempat yang sesuai dengan kode tersebut.

Kata kunci: *Automation Sorting Line System, PLC Omron NX1P2, pemrograman PLC*

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

MECHANICAL DESIGN AND PLC PROGRAMMING ON AUTOMATION SORTING LINE SYSTEM

Vrenky Meidianto
1031150000044

Supervisor I : Eko Setijadi, S.T., M.T., Ph.D.
Supervisor II : Imam Arifin, S.T., M.T.
Supervisor III : Mohammad Hafid, S.Pd.

ABSTRACT

Application of technology in the industry began to grow rapidly. One of them in the process of selecting certain goods or products for marketed and stored. To obtain maximum results and high level of accuracy required a technologically advanced system or can be called with an automated system to sort the desired product according to increase production yield and reduce errors election. In this final project created Automation Sorting Line System using barcode detection. In product selection process will be sorted according to barcode that exist in each product. In the process this technological system will read the barcode on the product after it will sort out according to the code. This system reduces errors caused by manual sorting by humans. From the results of the implementation of the tool works in accordance with what is desired is to sort the product using a barcode and immediately put in place according to the code.

Keywords: Automation Sorting Line System, PLC Omron NX1P2, pemrogramming PLC

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

KATA PENGANTAR

Dengan menyebut nama Allah SWT yang Maha Pengasih lagi Maha Panyayang, Kami panjatkan puja dan puji syukur atas kehadirat-Nya, yang telah melimpahkan rahmat, hidayah, dan inayah-Nya kepada kami, sehingga kami dapat menyelesaikan buku Tugas Akhir yang berjudul “ Rancang Bangun Mekanik dan Pemrograman PLC pada *Automation Sorting Line System*”.

Buku Tugas Akhir telah kami susun dengan maksimal dan mendapatkan bantuan dari berbagai pihak sehingga dapat memperlancar pembuatan buku ini. Untuk itu kami menyampaikan banyak terima kasih kepada semua pihak yang telah berkontribusi dalam pembuatan makalah ini.

Terlepas dari semua itu, Kami menyadari sepenuhnya bahwa masih ada kekurangan baik dari segi susunan kalimat maupun isinya. Oleh karena itu dengan tangan terbuka kami menerima segala saran dan kritik dari pembaca agar kami dapat memperbaiki buku Tugas Akhir ini.

Akhir kata kami berharap semoga buku ini dapat memberikan manfaat maupun inspirasi terhadap pembaca.

Surabaya, 10 Juli 2018

Vrenky Meidianto
NRP. 1031150000044

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

DAFTAR ISI

LEMBAR PENGESAHAN	Error! Bookmark not defined.
LEMBAR PENGESAHAN PERUSAHAAN.....	Error! Bookmark not defined.
ABSTRAK.....	v
<i>ABSTRACT</i>	vii
KATA PENGANTAR	ix
DAFTAR ISI	xi
DAFTAR GAMBAR	xiii
DAFTAR TABEL	xv
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang.....	1
1.2. Rumusan Masalah.....	4
1.3. Batasan Masalah	4
1.4. Tujuan.....	4
1.5. Metodologi Penelitian.....	4
1.6. Sistematika Laporan	5
BAB II AUTOMATION SORTING LINE SYSTEM.....	7
2.1. Cara Kerja <i>Automation Sorting Line System</i>	7
2.2. <i>Belt Conveyor</i>	8
2.3. Motor Penggerak	8
2.4. <i>Programmable Logic Controller (PLC)</i>	9
2.5. PLC NX1P2 - 9024DT	12
2.6. Pemrograman PLC.....	12
2.7. Sysmac Studio	18
2.8. Dataman60 Cognex	19
2.9. Sensor <i>Infrared</i>	19
2.10. Sensor <i>Limit Switch</i>	20
BAB III PERENCANAAN DAN PEMBUATAN ALAT.....	21
3.1. Perancangan dan Pembuatan Mekanik	22
3.1.1. Mekanik <i>Main Line</i>	23
3.1.2. Mekanik Aktuator	24
3.1.3. Mekanik <i>Line Picking</i>	29

3.1.4. Mekanik pelindung Sistem.....	30
3.1.5. Mekanik Penyangga Motor DC dan Penyangga Sensor Cognex DM60.....	31
3.2. Perancangan Pemrograman PLC	35
3.2.1. Memahami Kebutuhan Kontrol Dari Sistem.....	37
3.2.2. Membuat <i>Flow Chart</i> umum.....	39
3.2.3. Mendaftarkan <i>Input / Output</i> Sistem.....	41
3.2.4. Menerjemahkan <i>Flow Chart</i> ke Dalam Program PLC	42
3.2.5. Simulasi Program PLC.....	48
3.2.6. Menghubungkan <i>Input / Output Device</i> Dengan PLC	48
3.2.7. Jalankan Sistem / Plan Dengan PLC	48
BAB IV PENGUJIAN DAN ANALISA	49
4.1. Cara Kerja Alat.....	49
4.2. Pengujian Mekanik dan Pemrograman PLC pada <i>Automation Sorting Line System</i>	49
4.3. Pengujian Main Line	50
4.4. Pengujian <i>Line Picking</i>	51
4.5. Pengujian Mekanik Aktuator	53
4.6. Pengujian Mekanik Pelindung	54
4.7. Pengujian Mekanik Penyangga DM60 dan Motor DC	55
4.8. Analisa Program <i>Sorting Line</i>	56
4.9. <i>Sorting Line</i>	57
4.10. <i>Input</i>	58
4.11. <i>Line Picking</i>	59
4.12. <i>Function Block Barcode</i>	61
4.13. <i>Function Block Box Sorting</i>	62
4.14. <i>Function Block Tracking Management</i>	64
4.15. <i>Function Block Box Tracking</i>	66
DAFTAR PUSTAKA	71
LAMPIRAN	73

DAFTAR GAMBAR

Gambar 2.1	<i>Automation Sorting Line System</i>	7
Gambar 2.2	<i>Belt Conveyor</i>	8
Gambar 2.3	Motor DC.....	9
Gambar 2.4	Diagram Konseptual Aplikasi PLC.....	9
Gambar 2.5	Blok diagram PLC	10
Gambar 2.6	Blok diagram CPU dan PLC	11
Gambar 2.7	Ilustrasi <i>Scanning</i>	11
Gambar 2.8	PLC NX1P2 902DT	12
Gambar 2.9	Simbol-simbol dasar ladder diagram.....	13
Gambar 2.10	Rangkaian <i>Latch</i>	14
Gambar 2.11	<i>Timer Function</i>	15
Gambar 2.12	<i>Counter Function</i>	16
Gambar 2.13	<i>Compare Function</i>	16
Gambar 2.14	Tampilan sysmac studio.....	19
Gambar 2.15	Dataman60 Cognex.....	19
Gambar 2.16	Sensor <i>Infrared</i>	20
Gambar 2.17	Sensor <i>Limit Switch</i>	20
Gambar 3.1	Pembagian Tugas	21
Gambar 3.2	<i>Block Diagram</i> Mekanik	23
Gambar 3.3	Desain Mekanik <i>Main Line</i> , ukuran dalam mm.	23
Gambar 3.4	Mekanik <i>Main Line</i>	24
Gambar 3.5	Desain Tiang Penyangga Tempat Servo	25
Gambar 3.6	Desain Tempat Servo	26
Gambar 3.7	Desain Penyangga Servo.....	27
Gambar 3.8	Hasil Mekanik Penyangga Servo	27
Gambar 3.9	Desain Pendorong	28
Gambar 3.10	Hasil Mekanik Pendorong.....	28
Gambar 3.11	Desain Mekanik <i>Line Picking</i>	29
Gambar 3.12	Hasil Mekanik <i>Line Picking</i>	29
Gambar 3.13	Desain Mekanik Pelindung <i>Main Line</i>	30
Gambar 3.14	Mekanik Pelindung <i>Main Line</i>	31
Gambar 3.15	Desain Tempat Sensor DM60	32
Gambar 3.16	DesainTiang Penyangga Sensor DM60.....	32
Gambar 3.17	Desain Mekanik Penyangga Sensor DM60.....	33
Gambar 3.18	Mekanik Penyangga Sensor DM60.....	33
Gambar 3.19	Tiang Penyangga Motor DC	34

Gambar 3. 20	Penyangga Motor DC.....	35
Gambar 3. 21	<i>Block Diagram</i>	36
Gambar 3. 22	Bentuk Tombol pada HMI	37
Gambar 3. 23	Tombol dan Indikator pada Panel	38
Gambar 3. 24	<i>Flow Chart</i> Sistem	40
Gambar 3.25	Pengelompokan Pemrograman Secara Umum	43
Gambar 3.26	<i>Function Block</i>	43
Gambar 3.27	<i>Function Block Box Sorting</i>	44
Gambar 3.28	<i>Function Block Tracking Manajemen</i>	45
Gambar 3.29	<i>Function Block Box Tracking</i>	47
Gambar 3.30	<i>Function Block Blink</i>	47
Gambar 4.1	<i>Automation Sorting Line</i>	49
Gambar 4.2	Pengujian tanpa beban	50
Gambar 4.3	Pengujian menggunakan beban 3 <i>box</i>	51
Gambar 4.4	Pengujian Sensor <i>Limit Switch</i> pada <i>Line Picking</i>	52
Gambar 4.5	Mekanik <i>Line Picking</i>	52
Gambar 4.6	Mekanik Aktuator	53
Gambar 4.7	Mekanik Pendorong <i>Box</i>	54
Gambar 4.8	Mekanik Pelindung	55
Gambar 4.9	Penyangga Sensor DM60	55
Gambar 4.10	Penyangga Motor DC.....	56
Gambar 4.11	Tampilan Pemrograman	57
Gambar 4.12	Pemrograman <i>Input</i>	58
Gambar 4.13	<i>Timing Chart Input</i>	59
Gambar 4.14	<i>Pemrograman Line Picking</i>	60
Gambar 4.15	<i>Timing Chart Line Picking</i>	60
Gambar 4.16	Barcode pada HMI	61
Gambar 4.17	<i>Timing Chart Barcode</i>	62
Gambar 4.18	<i>Function Block Sorting Management</i>	63
Gambar 4.19	Hasil Uji Coba.....	63
Gambar 4.20	<i>Timing Chart Box Sorting</i>	64
Gambar 4.21	<i>Function Block Tracking Management</i>	65
Gambar 4.22	<i>Timing Chart Tracking Management</i>	65
Gambar 4.23	<i>Function Block Box Trackin</i>	66
Gambar 4.24	<i>Timing Chart Box Tracking</i>	67

DAFTAR TABEL

Tabel 3. 1 Daftar Input pada sistem	41
Tabel 4. 1 Pengujian <i>Main Line</i>	51

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

BAB I

PENDAHULUAN

1.1. Latar Belakang

Pelaksanaan magang selama 6 bulan bertempat pada PT Industrial Robotic Automation yang bergerak pada bidang integrator. Banyak proyek yang sudah di tangani oleh PT ini salah satunya adalah pada PT BINI (Bentoel Grup). Proyek yang ditangani pada PT BINI adalah proyek robot yaitu *sistem robot palletizer*. Sistem robot ini yaitu sistem pada *warehouse* untuk menata *box* rokok pada rak – rak penyimpanan untuk diedarkan di luar negeri ataupun dalam negeri. Di dalam sistem terdapat beberapa alat atau mesin yang saling terintegrasi seperti alat *weigher*, *sorting line*, lengan robot, *shuttle car*, *pallet dispenser*, dan *pallet wrapper*. Semua alat tekoneksi satu dengan yang lain untuk mencapai tujuan yang sama. *Weighter* adalah timbangan yang berupa *belt conveyor*, suatu alat timbang yang berjalan proses penimbangannya, hal inilah yang menjadi keunggulan dari mesin *weighter*, dengan cara kerjanya yaitu mengambil rata-rata berat selama proses perjalanannya.

Sorting line adalah alat untuk memilah *box* rokok secara otomatis sesuai dengan *barcode* ataupun sesuai dengan *patern* yang ada pada *box* rokok tersebut, setiap *box* mempunyai identitas masing-masing sesuai dengan jenisnya ada yang mempunyai *barcode* ada juga yang mempunyai identitas berupa gambar. Selanjutnya lengan robot, yaitu sama dengan namanya adalah robot lengan yang berfungsi untuk menata *box* pada *pallet* untuk selanjutnya disimpan pada rak-rak penyimpanan, lengan robot yang digunakan adalah lengan robot dari PT. ABB. Jadi PT. Industrial Robotics Automation membeli rbot dari PT. ABB selanjutnya diprogram dan diintegrasikan dengan mesin yang lain oleh PT. Insudtrial robotics Automation.

Shuttle car ialah mesin untuk menukar atau mengganti *pallet* yang sudah penuh terisi *box* dengan *pallet* kosong, *pallet* yang sudah penuh dengan 24 *box* terdiri dari 3 shaf dan tiap tiap-tiap shaf terdapat 8 *box*, setelah terisi penuh akan digantikan dengan *pallet* kosong, kemudian *pallet dispenser* yang bertugas untuk mengisi *pallet* pada *shuttle car* akan aktif, pengisian *pallet* pada *shuttle car* juga dilakukan secara otomatis melalui mesin ini. Jadi *pallet* diletakan pada *pallet dispenser* lalu mesin akan secara otomatis mengisinya jika kondisi

kosong, dan yang terakhir adalah *wrapper pallet* yaitu berfungsi untuk membungkus *box* agar tidak terjatuh saat pengiriman ataupun penyimpanan. Pembungkusan dilakukan dengan teknologi canggih, digunakannya teknologi untuk mempercepat produksi ataupun proses dalam industri untuk memenuhi kebutuhan konsumen, kinerja alat ini adalah akan membungkus *box* mulai dari bawah sampai atas, untuk memastikan agar *box* tidak terjatuh saat diletakan pada rak-rak oleh *forclift*.

Setelah 3 bulan terlewati berlanjut ke 3 bulan terakhir yaitu untuk persiapan tugas akhir yang salah satu proyeknya pada *sistem robot palletizer* yaitu proyek untuk memprogram sensor pendeteksi *barcode* dan *patern* tetapi proyek ini didiskusikan dengan *supervisor electric* terlebih dahulu sebelum diputuskan untuk dijadikan topik tugas akhir *supervisor* mengatakan terlalu mudah untuk di kerjakan 3 orang dan di jadikan topik tugas akhir mahasiswa. Karena pemrograman sensor cukup dilakukan dengan satu orang saja tidak perlu dilakukan oleh tiga orang, jadi hal tersebut tidak memenuhi kriteria untuk dijadikan topik tugas akhir yang berkelompok. Jadi kami dan pembimbing berunding untuk memilih satu alat yang ada pada proyek tersebut untuk dijadikan topik tugas akhir kami bertiga. Kami harus memilih diantara beberapa mesin yang ada pada sistem *robot palletizer* mulai dari lengan robot yang dalam akhir ini menjadi sorotan khusus pada dunia industri, tetapi lengan robot terlalu sulit untuk dikerjakan selama 3 bulan kedepan yang dibuat mulai dari awal. Akhirnya pilihan tertuju pada mesin pemilahan *box* rokok.

Pada hal ini alat yang paling penting dalam sistem tersebut yaitu *sorting line*, alat untuk memilah *box* rokok sesuai dengan jenisnya berdasarkan kode ataupun gambar, dengan adanya kode ataupun gambar pada *box* tersebut dapat menjadikan identitasnya. *Sorting line* dibutuhkan karena dengan adanya mesin pemilahan secara otomatis ini dapat membantu mempercepat proses pemilihan dibandingkan dengan tenaga manusia yang akan menurun tingkat kecepatan dan ketepatan seiring dengan lamanya waktu bekerja yang dijalankan. Pemilihan *sorting line* ini dipengaruhi oleh penerapan teknologi pada industri yang semakin berkembang pesat. Semakin meningkatnya kebutuhan produsen, maka industri dituntut untuk bekerja lebih cepat dan tepat untuk memenuhi kebutuhan pasar. Dengan adanya teknologi menjadikan salah satu cara untuk mengatasi permasalahan tersebut salah satunya dalam proses pemilahan barang atau produk tertentu

untuk dipasarkan maupun disimpan [1]. Pemilahan secara manual yang dilakukan oleh tenaga manusia, dapat mengakibatkan kesalahan dan kurang efektif karena lamanya waktu bekerja pada proses tersebut. Juga bergantung pada keterampilan tenaga kerja dan keakuratannya dipengaruhi oleh banyak faktor, seperti kelelahan. Untuk mengatasi kekurangan ini, peningkatan efisiensi penyortiran, meningkatkan kualitas produk dan menghemat energi [2]. Untuk mendapatkan hasil maksimal dan tingkat ketelitian yang tinggi diperlukan sebuah sistem berteknologi canggih atau dapat disebut sistem otomatis untuk memilah produk sesuai yang diinginkan agar hasil produksi meningkat dan mengurangi kesalahan pada proses pemilahan.

Pada Tugas Akhir ini satu topik di kerjakan tiga orang mahasiswa dengan bagian bagian tertentu yaitu yang pertama adalah mekanik dan pemrograman, kedua yaitu *electric* dan *wiring*, ketiga adalah komunikasi HMI. Dengan pembagian – pembagian tersebut diharapkan dapat belajar untuk mengelola sebuah proyek di industri. Pembagian akan ditentukan sesuai minat yang diinginkan, dengan saya sendiri memilih bagian mekanik dan pemrograman PLC dengan judul rancang bangun mekanik dan pemrograman PLC pada *automation sorting line system*. Dalam judul tersebut meliputi desain mekanik dan pemrograman PLC yang menerima data *barcode* dari sensor DM60 dari cognex Dalam proses pemilihan produk akan dipilah berdasarkan kode yang ada pada setiap produk. Sistem berteknologi ini akan membaca *barcode* pada produk setelah itu akan memilah sesuai dengan kode tersebut. Dengan menggunakan sistem ini dapat mengurangi kekeliruan yang diakibatkan oleh pemilahan secara manual oleh manusia karena industri selalu menuntut ketelitian dan kecepatan yang maksimal agar produksi meningkat untuk mencukupi kebutuhan produsen.

Dari hasil implementasi alat bekerja sesuai dengan apa yang dikehendaki yaitu memilah produk menggunakan barcode dan secara langsung akan membandingkan data *barcode* dari sensor DM60 dengan data dari HMI selanjutnya akan menempatkan pada tempat yang sesuai dengan kode tersebut. Sesuai dengan desain mekanik yang sudah dibuat miring untuk menurunkan *box* pada posisi yang sudah ditempatkan berjalan sesuai dengan yang diharapkan yaitu *box* turun dengan baik.

1.2. Rumusan Masalah

Proses pemilahan yang dilakukan dengan bantuan tenaga manusia memiliki beberapa kekurangan, yaitu semakin lama proses berlangsung, tingkat kecepatan untuk melakukan pemilahan juga berkurang. Hal tersebut dapat terjadi karena manusia memiliki rasa lelah yang berpengaruh pada tingkat ketelitian semakin berkurang karena manusia memiliki rasa jenuh untuk melakukan suatu pekerjaan yang sama secara terus menerus. Kelemahan tersebut yang akhirnya membuat pabrik harus siap dengan kerugian atau berkurangnya produksi yang di dapat.

1.3. Batasan Masalah

Dalam pembuatan Tugas Akhir ini, penulis membuat batasan dalam pengerjaan yaitu membangun *automation sorting line system* dengan 1 *main line* menggunakan *conveyor belt* dan 3 *line picking* secara umum tanpa ada perhitungan secara khusus. Benda yang dideteksi adalah *box* dengan ukuran 8cm x 8cm x 8cm yang peletakan awalnya harus dekat dengan pembatas agar *box* optimal dalam proses pemilahan. Proses pemilahan dapat dilakukan secara bersamaan sampai dengan 5 *box* dan menggunakan 3 bahasa pemrograman PLC yaitu *ladder diagram*, *function block diagram* dan *structure text*.

1.4. Tujuan

Pembuatan alat ini bertujuan untuk membuat *prototype* dari alat *sorting line system* agar dapat di jadikan pembelajaran bagi peserta magang untuk mengenali cara menyelesaikan proyek dan mengetahui yang dibutuhkan di dunia industri untuk kedepannya. Dengan adanya tujuan ini bermaksud agar selesai magang juga siap untuk terjun didunia industri yang keras dan profesional.

1.5. Metodologi Penelitian

Dalam pelaksanaan tugas akhir ini, ada beberapa tahapan kegiatan yang dapat diuraikan untuk mencapai tujuan penyelesaian buku tugas akhir, yang pertama adalah studi pustaka dan survei data awal. Pada tahap ini akan mempelajari tentang buku ataupun jurnal-jurnal yang dapat menunjang teori maupun prinsip kerja pada *automation sorting line system* dan dasar desain mekanik serta

Software (pemrograman). Kemudian yang kedua mengidentifikasi masalah yang akan di bahas, dalam kegiatan ini juga dilakukan analisa kondisi lingkungan berupa wawancara guna mengetahui masalah yang dialami saat menggunakan alat yang dikendalikan secara konvensional dan efektifitas alat ini jika di realisasikan.

Selanjutnya perancangan desain mekanik. Dalam perancangan ini dilakukan desain mekanik *Automation Sorting Line System* pada *software inventor*. Mekanik ini akan di desain seefisien mungkin dengan ukuran yang lebih kecil tetapi tetap menjadikannya mesin yang tepat guna sekaligus berkualitas agar dapat menjadi referensi sistem pada proyek selanjutnya PT Industrial Robotics Automation dan menjadi bahan pembelajaran bagi kami untuk menangani sebuah proyek.

Perancangan dan pembuatan program. Langkah yang dilakukan selanjutnya adalah melakukan perancangan dan pembuatan program pada PLC Omron NX1P2-9024DT. Semuanya dibuat menggunakan tiga bahasa pemrograman PLC yaitu *Ladder diagram*, *function block* dan *structur text*, digunakannya hal tersebut bertujuan untuk memudahkan pemrograman dan pencarian masalah jika terjadi *error*.

Pada tahap selanjutnya adalah pengujian dan analisis data. Pada fase ini dilakukan pengujian di setiap komponen untuk memastikan kinerja atau hasil yang di dapatkan sesuai dengan hasil yang diharapkan, untuk proses pengujian alat ini juga menganalisis data-data yang didapat saat melakukan uji coba dan mencari besar nilai kesalahan yang terdapat pada sistem ini, kemudian yang terakhir yaitu penyusunan laporan. Setelah alat berhasil dibuat, dilakukan penyusunan laporan untuk semua tahap yang terselesaikan agar hasil yang telah di dapatkan dari semua tahapan pada alat dapat di jelaskan secara rinci dan spesifik sesuai dengan data-data yang telah di dapatkan.

1.6. Sistematika Laporan

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

Bab I Pendahuluan

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

Bab II Teori Dasar

Pada hal ini menjelaskan tentang tinjauan pustaka, dan kosep dari Desain mekanik dan manajemen pemrograman PLC pada *Automation Sorting Line System*.

Bab III Perancangan Sistem

Bab ketiga membahas desain dan perancangan desain mekanik serta manajemen pemrograman PLC pada *Automation Sorting Line System*.

Bab IV Implementasi dan Analisis Sistem

Bab ini membuat hasil implementasi serta analisis dari hasil tersebut.

Bab V Penutup

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

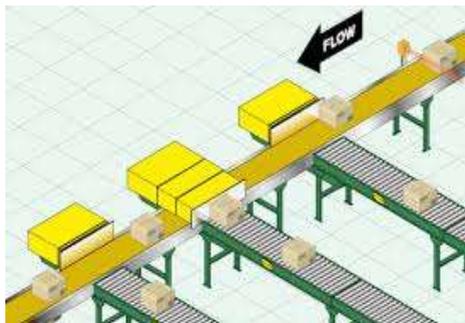
BAB II

AUTOMATION SORTING LINE SYSTEM

Pada bab ini membahas dasar-dasar teori yang dapat menunjang pembuatan alat dari segi pencangan mekanik dan pemrograman PLC yang digunakan.

2.1. Cara Kerja *Automation Sorting Line System*

Alat ini mempunyai 2 bagian jalur yaitu jalur utama (*main line*) dan jalur pemilahan (*line picking*), dengan adanya jalur pemilahan inilah proses pemilahan dilakukan sesuai dengan yang dikehendaki operator. Seperti Gambar 2.1 jalur utama adalah jalur yang pertama kali dipakai untuk mengirimkan *box* menuju proses pemilahan. Proses pemilahan dilakukan dengan pendorong agar *box* dapat masuk pada jalur pemilhan [3]. Pemilihan didasarkan pada *barcode* yang ada pada tiap-tiap *box* setelah data sudah didapat oleh sensor pembaca *barcode* selanjutnya akan di bandingkan dengan data pada HMI untuk menentukan *box* masuk pada *line picking* mana. Setelah sudah ditentukan maka *box* akan di *tracking* untuk menuju tempat pemilahan selanjutnya *box* akan di dorong oleh servo untuk masuk pada *line* yang sudah ditetapkan.



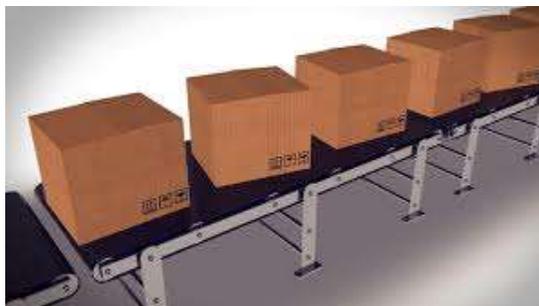
Gambar 2.1 *Automation Sorting Line System*

Sorting Line berguna untuk memilah barang (*box*) berdasarkan kode ataupun sejenisnya dan akan secara otomatis melakukan

pemisahan sesuai dengan yang diinginkan atau pengaturan kode awalnya. Demikian proses pemilihan dengan tenaga manusia dapat digantikan dengan mesin yang jauh lebih efisien dan lebih cepat.

2.2. *Belt Conveyor*

Alat tersebut terdiri dari sabuk yang tahan terhadap pengangkutan benda padat. Sabuk yang digunakan pada *belt conveyor* ini dapat dibuat dari berbagai jenis bahan. Misalnya dari karet, plastik, kulit ataupun logam yang tergantung dari jenis dan sifat bahan yang akan diangkut. *Belt conveyor* dapat digunakan untuk mengangkut material baik yang berupa “*Unit Load*” atau “*Bulk Material*” secara mendatar ataupun miring [4], [5].



Gambar 2.2 *Belt Conveyor*

Unit Load adalah benda yang biasanya dapat dihitung jumlahnya satu per satu, misalnya kotak, kantong, balok dan lain-lain. Sedangkan *Bulk Material* adalah material yang berupa butir-butir, bubuk atau serbuk misalnya pasir, semen dan lain-lain.

2.3. *Motor Penggerak*

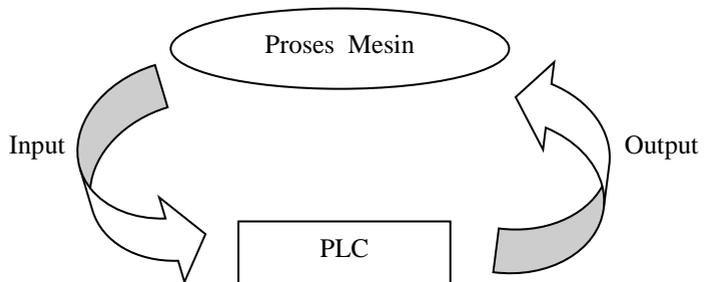
Komponen Motor DC *gearbox* digunakan sebagai penggerak karena memiliki torsi yang tinggi untuk menggerakkan suatu beban yang berat [6]. Seperti yang ditunjukkan pada Gambar 2.3 berikut adalah salah satu contoh motor *gearbox* yang digunakan sebagai penggerak pada konveyor.



Gambar 2.3 Motor DC

2.4. Programmable Logic Controller (PLC)

Pada dasarnya PLC adalah sebuah komputer yang khusus dirancang untuk mengontrol suatu proses atau mesin. Proses yang dikontrol ini dapat berupa regulasi kontinu seperti sistem-sistem servo, atau hanya melibatkan kontrol dua keadaan (*On/Off*) saja, tetapi dilakukan secara berulang-ulang seperti umum dijumpai pada sistem *conveyor*, dan lain sebagainya [7]. Gambar 2.4 berikut memperlihatkan konsep pengontrolan yang dilakukan oleh sebuah PLC.

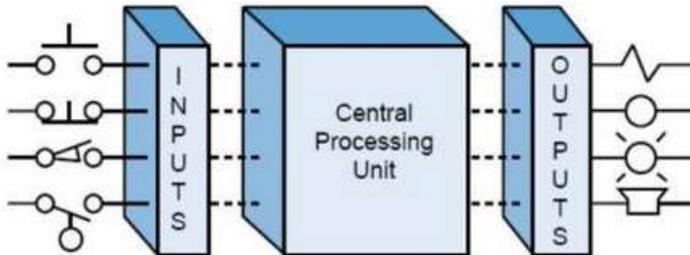


Gambar 2. 4 Diagram Konseptual Aplikasi PLC

Secara umum, PLC terdiri dari dua komponen penyusun utama yaitu *power supply*, memori, alat pemrogram, *Central Processing Unit* (CPU) dan Sistem antarmuka input/output. *Power supply* difungsikan sebagai penyedia daya bagi PLC. Memori digunakan untuk menyimpan data dalam PLC. Alat pemrogram digunakan untuk membuat atau mengedit program dalam PLC. *Unit prosesor* atau sentral unit (unit pengolahan pusat) (CPU) adalah unit yang berisi mikroprosesor yang menginterpretasikan sinyal-sinyal input dan melaksanakan tindakan-tindakan pengendalian, sesuai dengan program yang tersimpan di dalam memori, lalu mengkomunikasikan keputusan-keputusan yang diambilnya sebagai sinyal-sinyal kontrol ke antarmuka output [8].

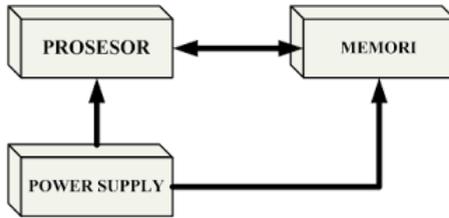
Input merupakan bagian dari PLC yang berhubungan dengan perangkat luar yang memberikan masukan pada CPU. Perangkat luar input dapat berupa tombol, *switch*, sensor atau piranti ukuran lain. *Output*, merupakan bagian PLC yang berhubungan dengan perangkat luar yang memberikan keluaran dari CPU. Perangkat luar output dapat berupa lampu, katub (*valve*), motor, dan lain-lain.

Bagian *input* dan *output* adalah antarmuka dimana prosesor menerima informasi dari dan mengonsumsi informasi kontrol ke perangkat-perangkat eksternal. Seperti pada Gambar 2.5 berikut ini.



Gambar 2.5 Blok diagram PLC

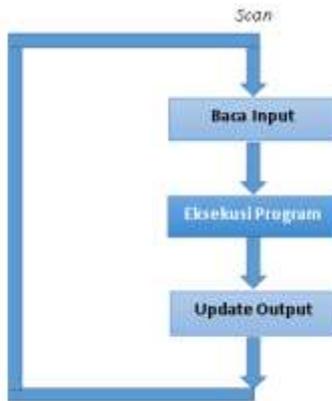
Fungsi dari CPU adalah mengatur semua proses yang terjadi di PLC. Ada tiga komponen utama penyusun CPU ini yaitu : *prosesor*, *memory*, dan *power supply*. Interaksi antara ketiga komponen ini dapat dilihat pada Gambar 2.6.



Gambar 2.6 Blok diagram CPU dan PLC

Selama prosesnya, CPU melakukan tiga operasi utama yaitu pertama membaca data masukan dari perangkat luar via modul input dapat berupa sensor dan lain lain, kedua yaitu mengeksekusi program kontrol yang tersimpan di memori PLC , program yang telah dibuat pada PLC akan dieksekusi sesuai dengan perintahnya, dan yang terakhir adalah meng-update atau memperbaharui data pada modul output. Setelah proses eksekusi program maka hasil akhir akan dikeluarkan pada modul output.

Ketiga proses tersebut dinamakan *scanning*, seperti pada Gambar 2.7 berikut ini.



Gambar 2.7 Ilustrasi *Scanning*

Tahap 1. Baca Input, PLC akan melihat status input apakah sedang dalam kondisi ON atau OFF. Tahap 2. Eksekusi Program, PLC akan mengeksekusi program yang dibuat per intruksi. Tahap 3. *Update Output*, ini tergantung input mana yang ON selama tahap 1 dan hasil eksekusi dari tahap 2.

2.5. PLC NX1P2 - 9024DT

PLC omron merupakan salah satu PLC yang sering digunakan dalam suatu *controller* pada mesin. Salah satunya adalah tipe NX1P2-9024DT yang memiliki 14 input dan 10 output yang dapat digunakan untuk menjalankan suatu mesin [9]. PLC jenis ini inputnya akan aktif ketika mendapatkan tegangan listrik 24v dikarenakan jenis PLC ini adalah NPN dan outputnya mengeluarkan tegangan 0v. Bentuk fisik PLC ditunjukkan pada Gambar 2.8 berikut.



Gambar 2.8 PLC NX1P2 902DT

2.6. Pemrograman PLC

Dapat dituliskan dengan beberapa bahasa pemrograman PLC dan komponen-komponen yang dibutuhkan dalam pemrograman yaitu:

Pemrograman tangga untuk PLC

Salah satu metode pemrograman PLC yang sangat umum digunakan adalah yang didasarkan pada penggunaan diagram-diagram tangga. Menuliskan sebuah program dengan demikian, sama halnya dengan menggambarkan sebuah rangkaian pensaklaran. Diagram-diagram tangga terdiri dari dua garis vertikal yang mempresentasikan rel-rel daya. Komponen-komponen rangkaian disambungkan sebagai

garis-garis horisontal, yaitu anak-anak tangga, diantarakedua garis vertikal ini [10].

Dalam menggambarkan sebuah diagram tangga, diterapkan konvensi-konvensi tertentu :

- Garis-garis vertikal diagram merepresentasikan rel-rel daya, di mana di antara kedua komponen-komponen rangkaian tersambung.
- Tiap-tiap anak tangga mendefinisikan sebuah operasi di dalam proses kontrol.
- Sebuah diagram tangga dibaca dari kiri ke kanan dan dari atas ke bawah. Prosedur membaca semua anak tangga program ini disebut sebagai sebuah siklus.
- Tiap-tiap anak tangga harus dimulai dengan sebuah input atau sejumlah input dan harus berakhir dengan setidaknya sebuah output.
- Input-input dan output-output seluruhnya diidentifikasi melalui alamat-alamatnya, notasi yang diergunakan bergantung pada pabrikan PLC yang bersangkutan.



Gambar 2.9 Simbol-simbol dasar ladder diagram

Gambar 2.9 memperlihatkan simbol-simbol baku yang digunakan untuk perangkat-perangkat input dan output. Input hanya direpresentasikan hanya dengan dua simbol, yaitu untuk kontak secara normal terbuka dan kontak secara normal tertutup. Output hanya direpresentasikan dengan satu simbol.

Fungsi-fungsi Logika

Banyak situasi yang mengharuskan dilakukannya tindakan-tindakan pengontrolan dilaksanakan ketika suatu kombinasi dari kondisi-kondisi tertentu terpenuhi. Ada beberapa fungsi-fungsi logika yang akan membantu pemrograman pada PLC [10] yaitu AND merupakan sebuah situasi dimana sebuah perangkat output tidak akan menyala terkecuali jika kedua saklar normal-terbuka berada dalam keadaan tertutup. Hal ini menggambarkan kombinasi logika OR (atau), dimana kedua inputnya harus 'hidup' untuk menghasilkan sebuah output. OR adalah situasi dimana sebuah perangkat output akan menyala apabila keduanya dalam kondisi awal terbuka menjadi dalam kondisi tertutup. Hal ini menggambarkan kombinasi logika OR (atau), dimana salah satu inputnya harus 'hidup' untuk menghasilkan sebuah output. NOT akan bekerja jika terdapat sebuah output ketika tidak ada input dan tidak ada output ketika terdapat sebuah input. Dapat dicontohkan dengan kontak dalam keadaan awal tertutup. Gerbang logika ini terkadang disebut sebagai pembalik (inverter).

Rangkaian Penahan (Latch)

Salah satu elemen penting pembentuk diagram *ladder* ini adalah rangkaian penahan. Hal ini muncul dari kenyataan bahwa input yang sifatnya sesaat digunakan untuk menghidupkan/mematikan output secara terus menerus [11]. Berdasarkan sifatnya, topologi rangkaian ini dapat dikatakan juga sebagai operasi Set/Reset. Sesuai dengan aplikasi yang dikontrol, jumlah input/output yang digunakan untuk operasi Set/Reset ini dapat lebih dari satu. Rangkaian penahan dapat dilihat pada Gambar 2.10 berikut ini.

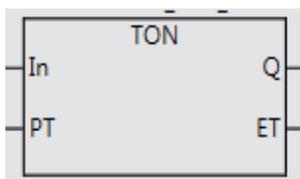


Gambar 2.10 Rangkaian *Latch*

Timer

Fungsi *Timer* adalah pemberi waktu / penundaan waktu dalam suatu proses. *Timer* berasal dari *built in clock oscillator* dalam CPU.

Terdapat beberapa bentuk *timer* yang dapat dijumpai pada PLC, yaitu *timer on-delay*, *timer off-delay* dan *timer pulse*. *Timer on -delay* akan hidup setelah satu periode waktu tunda yang telah ditetapkan. *Timer of- delay* berada dalam keadaan hidup selama periode waktu yang telah ditetapkan. Jenis *timer* lainnya yang sering dijumpai adalah *timer pulsa*. *Timer* jenis ini berubah menjadi aktif dan tidak aktif selama periode waktu yang telah ditetapkan. Durasi waktu yang ditetapkan untuk sebuah timer disebut sebagai *preset time* [11] . Fungsi ini dapat dilihat pada Gambar 2.11 berikut ini.

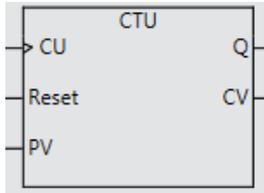


Gambar 2.11 *Timer Function*

Counter

Sebuah *counter* (piranti pencacah) seperti pada Gambar 2.12 dibawah ini memungkinkan untuk melakukannya pencacahan (atau penghitungan) terhadap sejumlah sinyal input. *Counter* ditetapkan untuk menghitung suatu nilai (atau jumlah) tertentu, dan ketika pulsa-pulsa dengan jumlah ini telah diterima, *counter* akan mengoperasikan kontak-kontaknya.

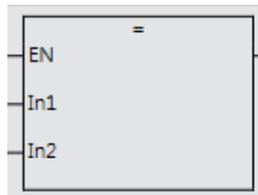
Terdapat dua tipe *counter*, yaitu *up-counter* (pencacah maju) dan *down-counter* (pencacah mundur). *Up-counter* (pencacah maju) menghitung maju dari nol hingga mencapai suatu nilai yang ditetapkan dengan kata lain, setiap kejadian akan menyebabkan nilai perhitungan bertambah satu [12], [13]. Ketika *counter* mencapai nilai yang ditetapkan, keadaan kontak-kontaknya berubah. *Down-counter* (pencacah-mundur) melakukan penghitungan mundur dari suatu nilai yang ditetapkan hingga mencapai nol, dengan kata lain, setiap kejadian akan mengurangi suatu nilai yang ditetapkan. Ketika *counter* mencapai nilai nol, keadaan kontak-kontaknya berubah.



Gambar 2.12 Counter Function

Compare

Fungsi ini berguna untuk membandingkan dua buah bilangan dan mengeset *flag-flag* spesial bit yang terkait berdasarkan hasil pembandingnya. Fungsi tersebut dapat kita lihat pada gambar 2.13 berikut ini. *Enable* adalah ntuk mengaktifkan *compare function*, In1 merupakan masukan pertama yang akan dibandingkan dengan In2 dan ketika sama akan menghasilkan output.



Gambar 2.13 Compare Function

Diagram Blok Fungsi

Salah satu format yang cukup banyak digunakan di dalam program adalah diagram blok fungsi (*function block diagram*). Diagram ini pada dasarnya adalah sama dengan yang digunakan untuk menjabarkan sistem-sistem logika. Blok fungsi adalah sebuah unit intruksi program yang apabila dieksekusi menghasilkan sebuah nilai output atau lebih

Tipe Data String

Tipe ini digunakan untuk merepresentasikan data yang berupa teks (kumpulan karakter). Dengan adanya tipe data ini dapat memuncuk teks untuk mengetahui hasil output yang dikeluarkan.

Fungsi IF

Salah satu permasalahan yang pasti akan dijumpai dalam pembuatan program adalah suatu percabangan. Percabangan yang dimaksud adalah suatu pemilihan *statement* yang akan dieksekusi dimana pemilihan tersebut didasarkan atas kondisi tertentu. Untuk mengimplementasikan suatu percabangan yaitu salah satunya dengan menggunakan struktur IF.

Statemen-statemen yang terdapat dalam sebuah blok percabangan akan dieksekusi hanya jika kondisi yang didefinisikan terpenuhi (bernilai benar). Artinya jika kondisi tidak terpenuhi (bernilai salah), maka *statemen-statemen* tersebut juga tidak ikut dieksekusi atau dengan kata lain diabaikan.

Struktur IF Satu Kondisi

Struktur ini merupakan struktur yang paling sederhana karena hanya melibatkan satu buah ekspresi yang akan diperiksa.

Bentuk umum baru struktur percabangan yang memiliki satu kondisi adalah sebagai berikut :

```
IF (kondisi) THEN
(kondisi)
END_IF
```

Struktur IF Dua Kondisi

Struktur percabangan jenis ini sedikit lebih kompleks bila dibandingkan dengan struktur yang hanya memiliki satu buah kondisi. Sebenarnya konsepnya sangat sederhana, yaitu pada struktur ini terdapat sebuah statemen khusus yang berguna untuk mengatasi kejadian apabila kondisi yang didefinisikan tersebut tidak terpenuhi (bernilai salah).

Bentuk umum daru struktur percabangan yang memiliki dua kondisi adalah sebagai berikut :

```
IF (kondisi) THEN
    (statemen_jika_kondisi_terpenuhi)
ELSE
    (statemen_jika_kondisi_tidak_terpenuhi)
END_IF
```

Struktur IF Tiga Kondisi atau Lebih

Percabangan jenis ini merupakan perluasan dari struktur yang memiliki dua kondisi diatas, yaitu dengan menyisipkan (menambahhkan) satu atau lebih kondisi didalamnya.

Bentuk umum baru struktur percabangan yang memiliki lebih dari dua kondisi adalah sebagai berikut :

```
IF (kondisi1) THEN
    (statemen_jika_kondisi_terpenuhi)
ELSIF (kondisi2) THEN
    (statemen_jika_kondisi2_terpenuhi)
ELSIF (kondisi3) THEN
    (statemen_jika_kondisi3_terpenuhi)
.....
ELSE
    (statemen_jika_kondisi_tidak_terpenuhi)
END_IF
```

2.7. Sysmac Studio

Software ini dipergunakan untuk memprogram PLC jenis omron. *Software* ini adalah software terbaru dari omron yang mempunyai keunggulan semakin interaktif kepada pengguna yang memiliki banyak fungsi didalamnya yang salah satunya adalah fungsi pengalamatan yang hanya memakai nama bukan kode-kode seperti pada PLC yang lain [14]. Tampilan dari sysmac studio ditunjukan pada Gambar 2.14 seperti berikut.



Gambar 2.14 Tampilan sysmac studio

2.8. Dataman60 Cognex

Sensor pembaca code 1D dan 2D, alat ini mendukung koneksi antarmuka RS-232, USB, dan Ethernet ditambah I / O diskrit [15]. Hal ini merupakan keunggulan dari sensor dataman60 yang digunakan dalam *automation sorting line system*. Dalam pengambilan data berupa kode alat ini memiliki 2 kondisi yaitu *good read* dan *bad read*, ketika kondisi *good read* maka kode sudah didapatkan datanya an pada kondisi *bad read* maka data kosong atau tidak dapat membaca kode. Setelah kondisi sudah terpenuhi maka data akan langsung dikirim. Dataman60 dapat dilihat pada Gambar 2.15 berikut ini .



Gambar 2.15 Dataman60 Cognex

2.9. Sensor Infrared

Menurut penelitian yang dilakukan oleh G. Benet, F. Blanes, J.E. Simo dan P. Perez pada tahun 2002 tentang Using Infrared Sensors for Distance Measurement in Mobile Robots terdapat juga sensor

inframerah yang menggunakan intensitas dari pantulan cahaya untuk mengestimasi jarak dari suatu benda, tetapi tidak dipergunakan secara umum dan hanya sejumlah kecil saja yang tercatat dalam penggunaan metode ini. Dengan menggunakan metode ini maka respon yang terjadi akan lebih cepat dan sangat membantu dalam menambah performa respon real-time dari suatu mobile robot (V. Colla,1996). Hal tersebut dapat dilihat bentuk fisiknya pada gambar 2.15 berikut.



Gambar 2.16 Sensor *Infrared*

2.10. Sensor *Limit Switch*

Seperti terlihat pada Gambar 2.16, merupakan sensor dan merupakan komponen yang sangat penting dalam mendukung terjadinya kontrol proses yang berfungsi sebagai berikut :

- Menyediakan input dari proses dan dari lingkungan eksternal.
- Mengubah informasi fisik misalnya posisi untuk sinyal listrik.
- Terkait dengan variabel fisik pada cara yang diketahui sehingga sinyal listriknya dapat digunakan untuk memonitor dan mengontrol proses.

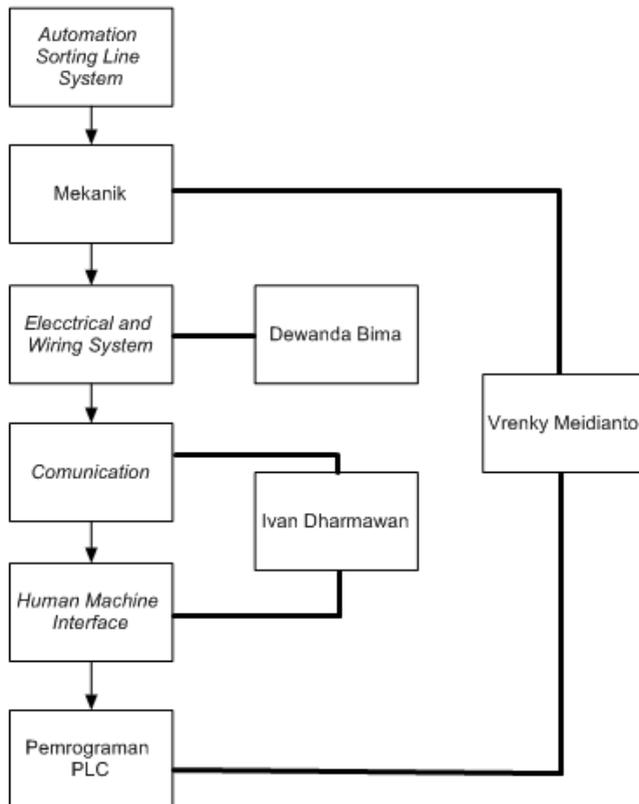


Gambar 2. 17 Sensor *Limit Switch*

BAB III

PERENCANAAN DAN PEMBUATAN ALAT

Pada bab ini akan dibahas tahapan dalam perencanaan dan pembuatan tugas akhir rancang bangun mekanik dan pemrograman PLC pada *Automation Sorting Line System*. Untuk mencapai keberhasilan maka akan dibagi tugas-tugas pengerjaan sistem ini. Seperti yang terlihat pada Gambar 3.1 berikut ini.



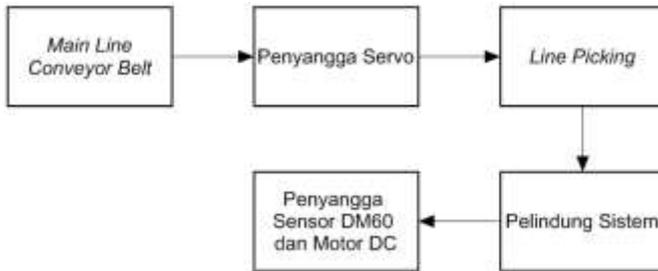
Gambar 3.1 Pembagian Tugas

Dalam membuat sebuah sistem harus memiliki perencanaan dan pembagian tugas yang matang, agar alat yang dibuat dapat selesai tepat waktu. Sistem ini dibagi menjadi beberapa bagian dan diharapkan dengan pembagian ini sistem dapat terealisasi dengan baik. Pertama adalah mekanik dan pemrogram, kedua *electrical and wiring, communication* dan *human machine interface*. Mekanik adalah awal sebuah rancangan untuk menentukan bentuk dari kerangka sistem tersebut, pada bagian ini sangat penting dari proses pembuatan sistem dikarenakan mekanik adalah bentuk fisik dari sistem tersebut. Selanjutnya adalah *electric* sekaligus *wiring*. Pada tahap ini dilakukan pengadaan komponen-komponen *electric* yang dibutuhkan dalam sistem dan dilakukannya pengkabelan untuk menyinkronkan semua komponen. Selanjutnya adalah *communication*, yaitu untuk komunikasi antar komponen seperti HMI dengan PLC ataupun PLC dengan sensor *barcode* DM60. Setelah itu adalah perancangan *human machine interface*, seperti mendesain tampilannya dll.

Setelah semua sudah dilakukan maka tahap selanjutnya adalah pemrograman, hal ini adalah tugas terakhir dari pembagian tugas pada sistem *automation sorting line system* karena pemrograman dapat berjalan setelah semua tahapan-tahapan sebelumnya sudah dilakukan, meskipun dapat dilakukan dengan simulasi tetapi keadaan dalam simulasi dengan tersambung alat secara langsung kemungkinan dapat berbeda. Hal tersebutlah yang membuat pemrogram itu tidak dapat berjalan dengan maksimal ketika belum dijalankan menggunakan *hardware*-nya. Pada pemrograman ini akan dibuat keseluruhan program mulai dari mesin aktif sampai proses pemilahan terselesaikan. Program yang dibuat meliputi penerimaan data *barcode* dari HMI dan data dari sensor pendeteksi *barcode* DM60 setelah itu dikomparasi untuk menentukan data mana yang cocok, selanjutnya pelaksanaan proses pemilahan sampai selesai.

3.1. Perancangan dan Pembuatan Mekanik

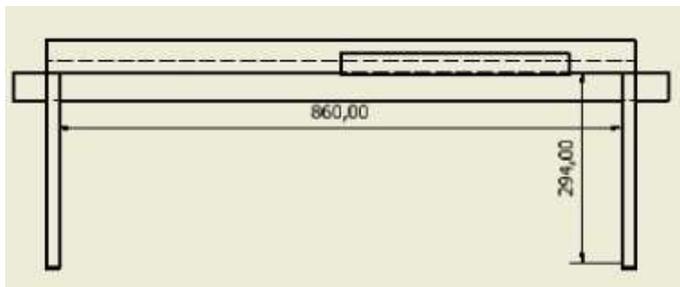
Dalam sub bab ini, akan dibahas mengenai perancangan mekanik pada *Automation Sorting Line System* menggunakan *software design* 3D inventor. Pada alat ini terdapat beberapa bagian yaitu mekanik *main line*, aktuator, *line picking*, pelindung dan penyangga. Proses pembuatan desain seperti pada Gambar 3.2 berikut ini.



Gambar 3.2 *Block Diagram Mekanik*

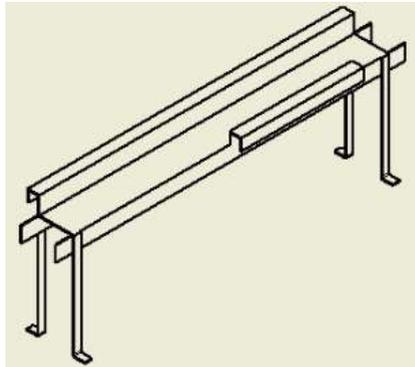
3.1.1. Mekanik *Main Line*

Mekanik *Mine Line* pada *Automation Sorting Line System* ini adalah jalur utama untuk menghantarkan *box* menuju proses pemilahan pada *line picking*. Alat ini didesain menggunakan *software* inventor. Didesain sedemikian rupa seperti yang ditunjukkan pada Gambar 3.3 dibawah ini. Desain dengan panjang lintasan 100cm dan lebar 10cm mempunyai ketinggian 30cm diukur dari permukaan tanah sampai *belt conveyor*. Pembatas atau pemhalang konveyor memiliki 2 ketinggian yaitu 5 cm dan 3 cm dikarenakan 5 cm untk menyesuaikan bok agar tetap pada posisi yang lurus dan ketinggian 3m untuk memastikan bahwa *box* tidak keluar jalur *conveyor* dan tidak menghalangi pembacaan sensor perekam *barcode* DM60 dari cognex. Seperti pada Gambar 3.3 berikut ini.



Gambar 3.3 *Desain Mekanik Main Line*, ukuran dalam mm.

Mekanik *mine line* menggunakan *conveyor belt* untuk menjaga kualitas dan mengurangi resiko terjadi cacatnya *box* saat berjalan diatas konveyor serta menggunakan kain keras sebagai *belt* penghantar. Untuk mengatasi kain melengkung diberikan plat besi diantara kedua roler untuk menyangga *belt* agar tidak melengkung ketika ada beban diatasnya digunakannya jenis ini sangat cocok untuk pemindahan barang berupa *box* atau sejenisnya. Dengan *belt conveyor box* akan terjaga keamanannya dan mengurangi terjadinya gesekan dibandingkan ketika menggunakan roller yang rentan terjadi gesekan antara roller dengan objek. Mekanik yang sudah di jelaskan di atas seperti pada Gambar 3.4 berikut ini.



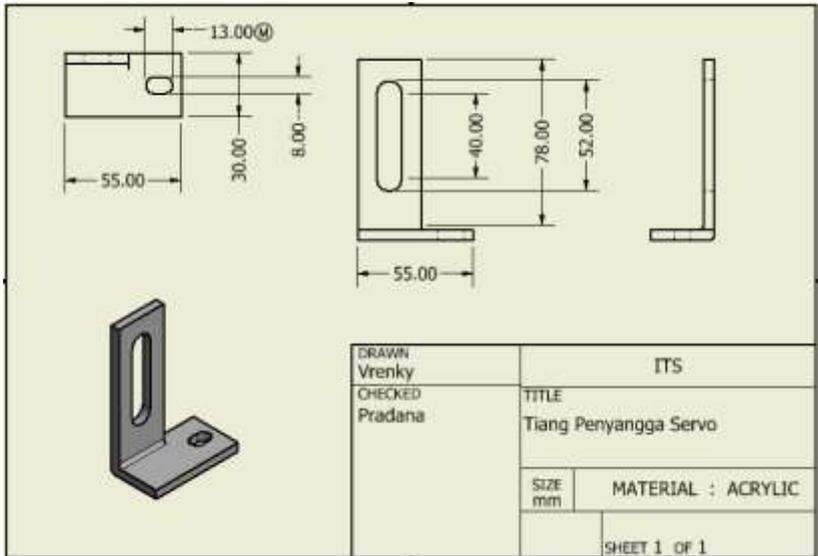
Gambar 3.4 Mekanik *Main Line*

3.1.2. Mekanik Aktuator

Pada desain mekanik aktuator ini dibagi menjadi 2 bagian yaitu desain mekanik penyangga servo dan desain mekanik pendorong *box* pada servo.

1. Mekanik penyangga servo

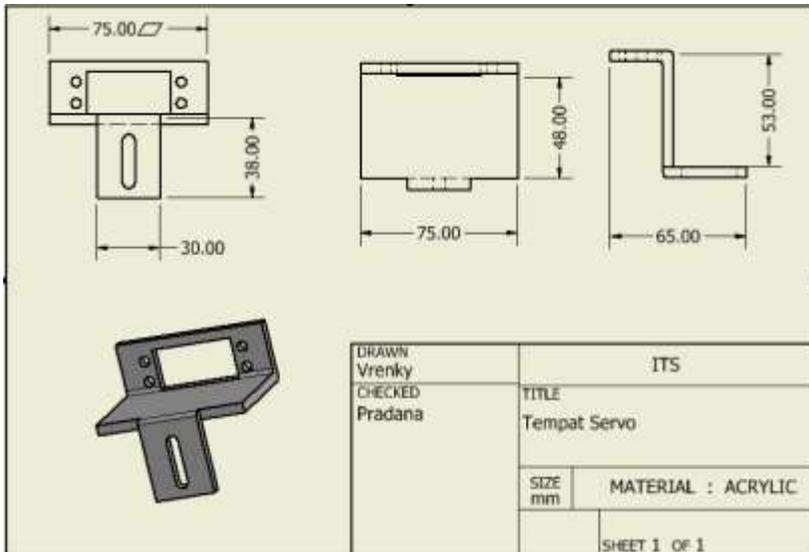
Mekanik penyangga servo ini didesain menggunakan *adjuster* agar dapat diatur ketinggian servo untuk memposisikan dorongan yang sesuai dengan kondisi *box*, sehingga memudahkan pengaturan untuk mendapatkan titik dorong yang tepat pada *box* dan mengurangi resiko miringnya benda ketika pendorong kurang tepat pada titik dorongnya. Seperti yang terlihat pada Gambar 3.5 berikut ini.



Gambar 3.5 Desain Tiang Penyangga Tempat Servo

Pada Gambar 3.5 terlihat tiang tempat servo didesain sedemikian rupa untuk dapat digunakan seperti yang diinginkan, pada mekanik ini memiliki *adjuster* digunakan untuk mengatur tinggi servo pada sistem. Hal tersebutlah yang membuat mekanik ini fleksibel dari segi pengaturan untuk menyesuaikan servo. Komponen ini dibuat dengan bahan akrilik 6mm, ketebalan tersebut diharapkan agar mekanik kuat menahan beban servo saat diam ataupun beban servo saat mendorong *box*. Tinggi tiang penyangga ini adalah 7,8 cm, lebar penahan 5,5 cm. Setelah itu dibuat tempat untuk peletakan servo sebagai aktuator pendorong *box* seperti yang terlihat pada Gambar 3.6 berikut.

Pada desain selanjutnya ini adalah tempat untuk meletakkan servo. Komponen ini berbentuk seperti tangga agar servo tepat dan kuat melekat pada tempatnya dan agar servo dapat di topang di mekaniknya tidak menggantung. Dibuat dari akrilik dengan ketebalan 6mm juga, agar mampu menahan servo.



Gambar 3.6 Desain Tempat Servo

Kedua desain diatas disatukan untuk menjadi mekanik penyangga servo sesuai dengan yang diinginkan. Untuk menyatukan kedua kompen tersebut yang berupa akrilik dengan cara di baut dan di mur untuk mengunci kedua komponen tersebut. Dan ditengah tengahnya terdapat adjuster yang dapat diatur ketinggiannya dengan cara memberi baut ukuran 6mm yang bertujuan untuk mengatur ketinggian dari mekanik penyangga servo, seperti pada gambar 3.7 berikut. Setelah dirangkai sedemikian rupa akan didapatkan mekanik penyangga yang diharapkan mampu memenuhi kebutuhan dari sistem untuk menjadi tempat aktuator yaitu servo.

Desain dibawah ini digunakan untuk membuktikan bahwa komponen-komponen yang dibuat dapat di gabungkan secara desainnya sebelum dii realisasikan dalam bentuk nyatanya. Ketika sudah didapat desain dan penggabungan dengsn baik maka selanjutnya dapat dilakukan pembuatan dengan akrilik 6mm dan sekaligus memberi baut dan mur untuk megatur ketinggian dan mengunci servo pada penyangga.



Gambar 3.7 Desain Penyangga Servo

Desain ini dibuat untuk mendukung kinerja sistem agar sistem berjalan sesuai harapan. Setelah Dengan hasil desain yang sudah di buat selanjutnya dibuat mekanik dengan akrilik dengan ketebalan 5 mm dengan hasil seperti pada Gambar 3.8 berikut.

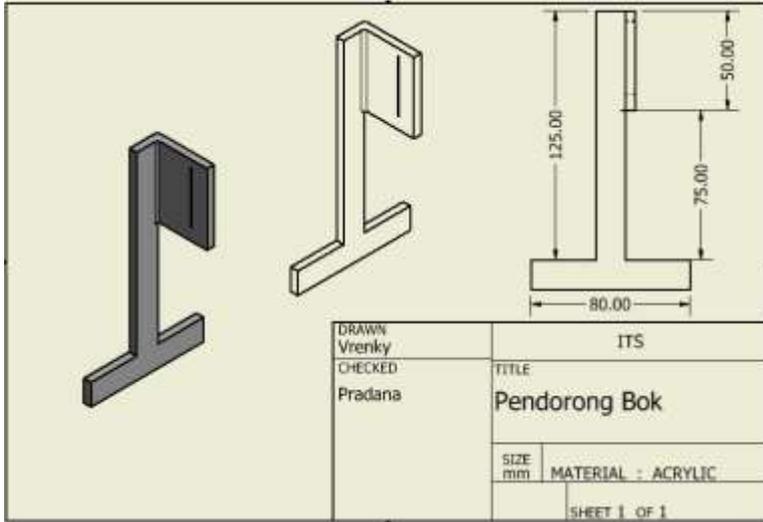


Gambar 3.8 Hasil Mekanik Penyangga Servo

2. Mekanik pendorong bok pada servo

Mekanik pendorong *box* ini di desain dengan bentuk huruf T terbalik dengan panjang pendorong adalah 8 cm dan lebar 1,5 cm. seperti Gambar 3.9. Desain yang dibuat dengan tinggi pendorong adalah 14 cm dan panjang 8 cm, dipilihnya ukuran tersebut untuk

memenuhi kebutuhan pendorong untu mendorong *box* yang ukurannya 8cm x 8cm x 8xm, agar *box* dapat terdorong dengan maksimal.



Gambar 3.9 Desain Pendorong

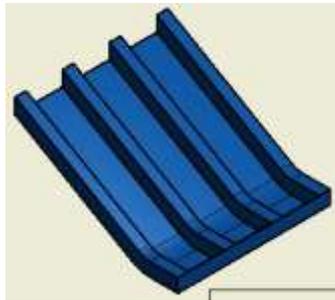
Huruf T terbalik di pilih dikarenakan dengan adanya pendorong yang panjang dapat mendorong bok dengan maksimal dan mengurangi resiko kurang tepatnya pendorong pada saat mendorong *box*. Pendorong *box* dibuat menggunakan akrilik dengan tebal 5mm. Seperti yang terlihat pada Gambar 3.10 berikut.



Gambar 3.10 Hasil Mekanik Pendorong

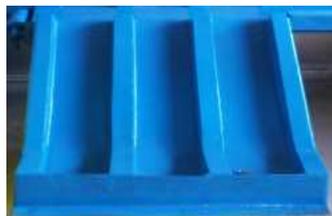
3.1.3. Mekanik *Line Picking*

Perancangan mekanik pada *line picking* ini seperti pada Gambar 3.6 alat di desain dengan 3 *line picking* yang berarti alat ini mampu menampung maksimal 3 jenis *box* saja, apabila terdapat lebih dari 3 jenis *box* yang dimasukkan maka *box* akan masuk ketempat pembuangan yang sudah di sediakan dan menampung *box* yang tidak terdeteksi *barcode*. *Line picking* dibuat miring untuk menurunkan *box* yang sudah terdorong oleh aktuator agar secara otomatis menempati tempat yang sudah disediakan. Pada bagian akhir landasan dibuat datar pengambilan *box* yang sudah masuk. Seperti pada Gambar 3.11 berikut.



Gambar 3.11 Desain Mekanik *Line Picking*

Mekanik *line picking* dibuat dengan bahan plat besi (seng) dengan ketebalan 1 mm seperti yang terlihat pada Gambar 3.12 berikut.



Gambar 3.12 Hasil Mekanik *Line Picking*

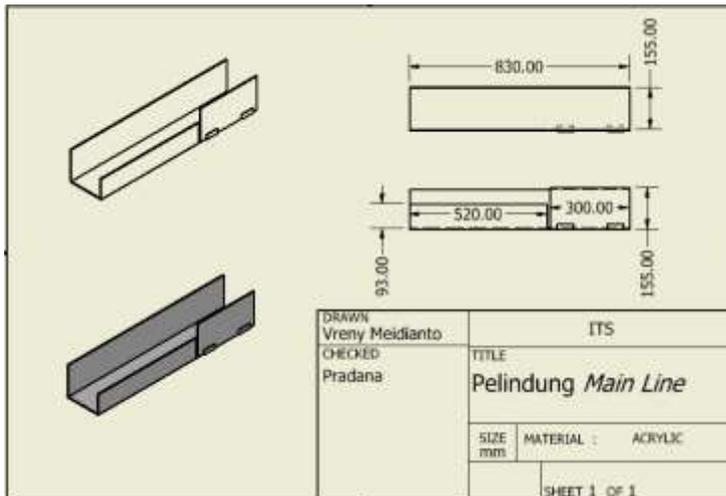
Panjang lintasan *line picking* yaitu 40 cm, dengan lebar 10 cm. Seng besi yang digunakan mempunyai tebal 1mm agar seng mudah dibentuk sesuai dengan desain yang diinginkan.

3.1.4. Mekanik pelindung Sistem

Mekanik pengaman ini didesain bertujuan untuk menjaga keamanan mesin dari segala sesuatu yang dapat mengganggu jalannya mesin. Mekanik pelindung dibagi menjadi 2 pada *automation sorting line system* yaitu.

Mekanik Pelindung *Main Line*

Bahan akrilik transparan digunakan dengan tujuan agar operator dapat memantau secara langsung proses pada mesin. Seperti ketika terjadi *box* yang tersangkut ataupun ada kejadian lainnya yang tidak diinginkan. Desain pelindung dapat dilihat pada Gambar 3.13 berikut.



Gambar 3.13 Desain Mekanik Pelindung *Main Line*

Pelindung dibuat dengan pintu samping dan dilengkapi dengan *safety door*. Ketika pintu dibuka sistem akan berhenti, apabila terjadi kegagalan sistem atau mesin bermasalah dapat diatasi dengan

membuka pintu tersebut tanpa mengurangi tingkat keamanan. Desain yang sudah dibuat direalisasikan menggunakan bahan akrilik transparan yang memiliki ketebalan 7mm seperti pada Gambar 3.14 berikut



Gambar 3.14 Mekanik Pelindung *Main Line*

Mekanik Pelindung *Line Picking*

Mekanik ini didesain dengan menjadikannya sebagai pintu pengaman untuk *line picking*, berbeda dengan mekanik pelindung pada *main line* yang mempunyai pintu samping. Hal ini dikarenakan gar lebih leluasa untuk melakukan perbaikan pada saat terjadi masalah pada mesin dan tingkat terjadinya kerusakan yang mengganggu sistem juga rendah maka dari itu pelindung didesain menjadi pintu pelindung sepenuhnya yang juga dilengkapi dengan sistem *safety door* untuk menjaga keamanan.

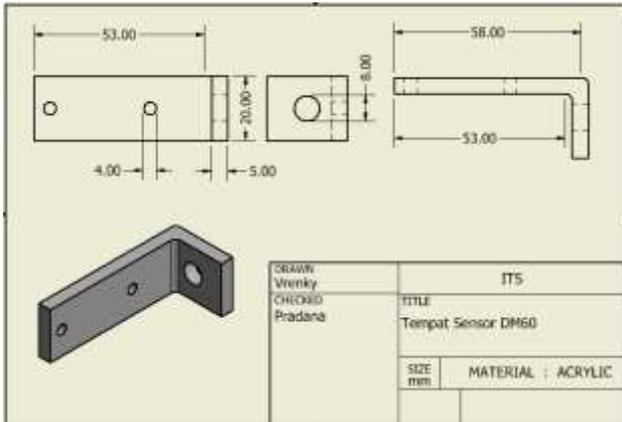
3.1.5. Mekanik Penyangga Motor DC dan Penyangga Sensor Cognex DM60

Pada sub bab ini akan menjelaskan beberapa komponen mekanik yaitu :

Penyangga Sensor Cognex

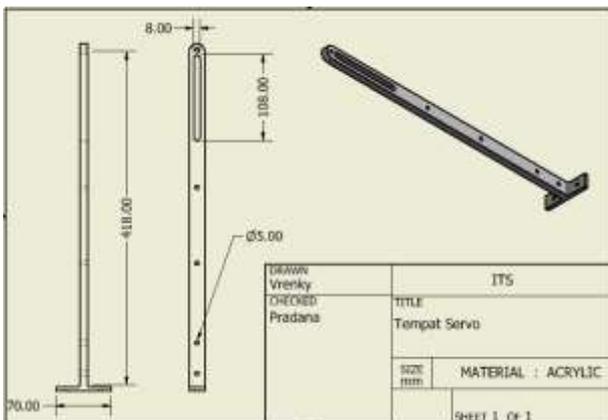
Penyangga Sensor menggunakan akrilik dengan ketebalan 10mm, dan diletakkan diluar alat konveyor dikarenakan untuk tetap menjaga kesatbilan sensor pada saat membaca kode agar tidak terjadi guncangan. Karena jika digabungkan dengan kerangka konveyor dapat memberikan getaran pada pengga sensor dan mengakibatkan proses perekaman *barcoede* menjadi tidak sempurna dan hasil tidak dapat menghasilkan , snsor *barcode* merupakan hal yang sangat penting

dalam sistem ini karena jika sensor tidak sesuai maka data tidak akan terbaca dan pensortiranpun tidak dapat ilakkan dengan sempurna.



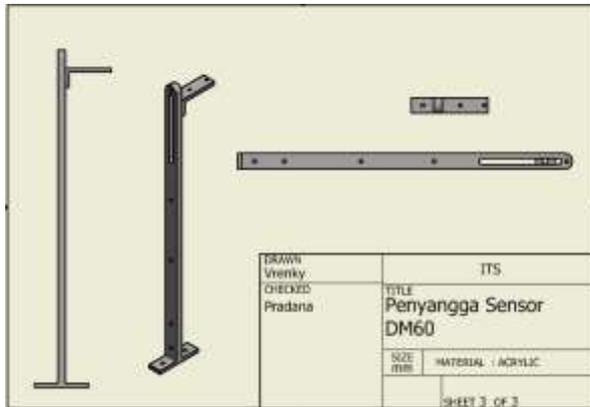
Gambar 3.15 Desain Tempat Sensor DM60

Pada Gambar 3.15 ditunjukkan desain mekanik dari tempat sensor DM60 yang dibentuk dengan sudut 90⁰ untuk memudahkan pemasangan pada tiang penyangga sensor. Seperti pada Gambar 3.16



Gambar 3. 16 DesainTiang Penyangga Sensor DM60

Mekanik diatas menggunakan *adjuster* untuk mengatur tinggi dan rendah sensor agar dapat membaca kode pada *box* dengan tepat. Pada kenyataannya *adjuster* digunakan untuk mengurangi ketidaktepatan sensor saat merekam kode pada *box*. Seperti pada Gambar 3.17



Gambar 3.17 Desain Mekanik Penyangga Sensor DM60

Dengan adanya hal tersebut sensor dapat diatur sesuai dengan kondisi agar sensor dapat merekam dengan cepat dan tepat untuk meningkatkan produksi suatu pabrik dan mengurangi kesalahan pembacaan kode yang mengakibatkan barang dianggap rusak atau tidak sempurna pada pembacaan kode oleh sensor.



Gambar 3.18 Mekanik Penyangga Sensor DM60

Gambar 3.18 diatas menunjukkan mekanik yang dibuat dari akrilik dengan ketebalan 1cm untuk digunakan sebagai tempat sensor cognex DM60.

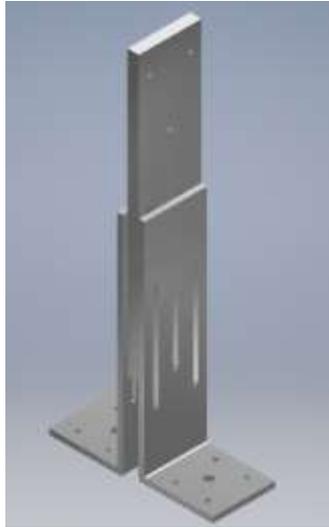
Penyangga Motor DC

Desain mekanik dibuat untuk menahan motor DC yang menjadi penggerak *conveyor belt*. Mekanik dibuat seperti pada Gambar 3.19 berikut ini. Dibutuhkan dua buah untuk menahan motor DC.



Gambar 3. 19 Tiang Penyangga Motor DC

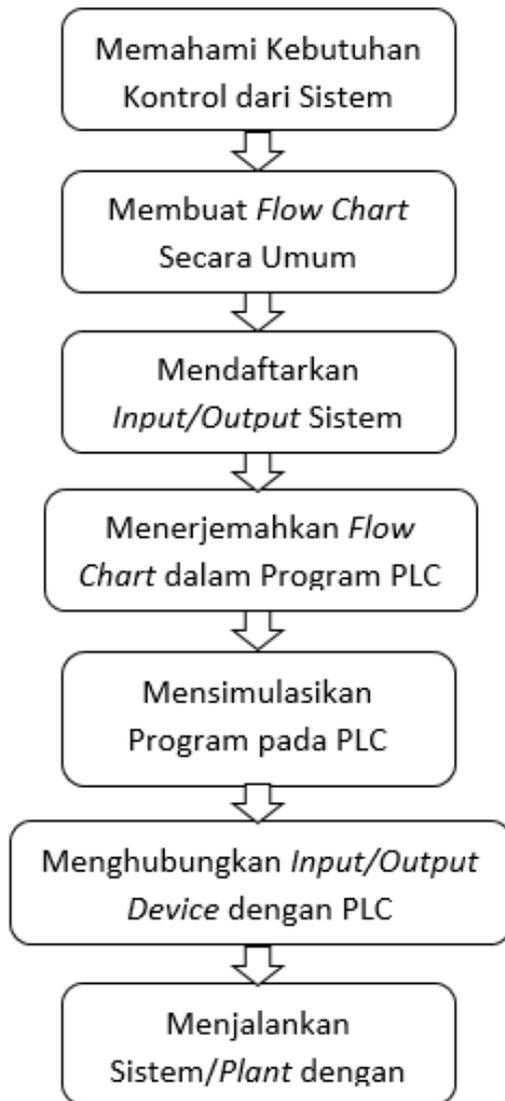
Gambar diatas digunakan 2 buah dikarenakan untuk mekanik ini didesain menyerupai huruf L maka harus menggunakan 2 buah untuk menjaga kestabilan dan kekuatan dari mekanik itu sendiri selanjutnya semua dirangkai dijadikan satu untuk menjadi desain utuh dari mekanik motor DC yang terlihat pada Gambar 3.20 berikut ini. Desain ini akan diwujudkan menggunakan bahan *acrylic*. Dengan desain yang sedemikian rupa diharapkan motor DC dapat berjalan dengan maksimal tanpa ada gangguan dari segi mekanik.



Gambar 3. 20 Penyangga Motor DC

3.2. Perancangan Pemrograman PLC

Dalam merancang program harus dilakukan langkah -langkah perancangan untuk mencapai hasil yang maksimal dan mengurangi kesalahan dalam proses pemrograman. Seperti yang terlihat pada Gambar 3.21 berikut in. Dimulai dari pemahaman kebutuhan kontrol, membuat *flow chart* setelah itu mendaftarkan *input/output*. Menerjemahkan *flowchart* kedalam bahasa pemrograman. Selanjutnya mensimulasikan program dan menjalankan alat dengan PLC. *Blok diagram* dibawah akan menuntun untuk membuat program yang seblumnya harus memenuhi tahapan-tahapan yang sudah ditentukan untuk menghindari kesalahan dan untuk mempercepat pekerjaan seorang *programmer* untuk memprogram suatau alat atau bahkan sebuah sistem yang komplek, dari tahapan inilah mampu membiasakan pola pikir yang runtun sebagai seorang *programmer* yang baik dan profesional



Gambar 3. 21 *Block Diagram*

3.2.1. Memahami Kebutuhan Kontrol Dari Sistem

Dalam pembahasan ini adalah untuk memahami kebutuhan kontrol pada *automation sorting line system*. Sistem ini ialah alat yang mempunyai 2 bagian jalur yaitu jalur utama (*main line*) dan jalur pemilahan (*line picking*). Sistem ini memiliki 1 jalur utama yang berupa *conveyor belt* dengan panjang 100cm, lebar 10cm dan 3 jalur pemilahan, jadi alat dapat memilah maksimal 3 jenis *box*.

Sistem memiliki 3 buah tombol pada HMI, 1 buah tombol pada panel dan 3 buah lampu indikator yaitu :

Tombol pada HMI

Tombol *Run*, Berfungsi untuk menyalakan / menjalankan mesin. Dengan tombol yang bersifat set / reset dapat dijadikan tombol *run* dengan baik dan aman. Tombol *Stop*, Berfungsi mematikan / menghentikan mesin untuk sementara ketika terjadi hal-hal yang tidak diinginkan misalnya *box* rusak ditengah perjalanan atau lain sebagainya. Maka tombol stop dapat dijaan untuk mengatasi permasalahan tersebut. Selanjtnya tombol *Reset* berfungsi mengembalikan keadaan mesin awal, atau membersihkan sistem agar kembali pada posisi awal. Tampilan tombol pada HMI dapat dilihat pada Gambar 3.22 berikut ini.



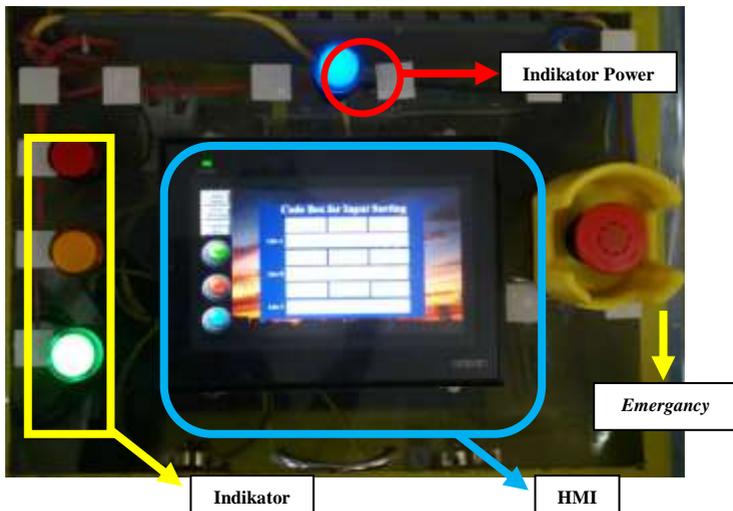
Gambar 3. 22 Bentuk Tombol pada HMI

Tombol pada panel

Tombol *Emergency*, tombol ini sama halnya dengan fungsi tombol *stop* pada hmi tetapi tombol ini terletak pada panel. Tombol ini di rangkai secara seri dengan *limit switch safety device*, pemicu dari rangkaian ini tidak hanya dengan tombol tetapi juga dengan membuka *body* pelindung juga akan memicu *emergency* hidup dan secara otomatis system akan berhenti selama tombol ataupun *safety device* tetap terpicu. Terlihat pada Gambar 3.23 dibawah ini.

Lampu indikator

1. Indikator *Run*
Befungsi sebagai informasi bahwa mesin / sistem dalam keadaan hidup / beroperasi.
2. Indikator *Standby*
Befungsi sebagai informasi bahwa mesin / sistem dalam keadaan persiapan.
3. Indikator *Stop*
Befungsi sebagai informasi bahwa mesin / sistem dalam keadaan berhenti.



Gambar 3. 23 Tombol dan Indikator pada Panel

Prinsip kerja sistem ini adalah ketika mesin sudah siap dalam artinya sudah dinyalakan, *box* dapat dimasukkan setelah itu akan menyentuh sensor *limit switch* yang berfungsi sebagai *trigger* sensor DM-60 dan motor pada *main line* berhenti untuk pelaksanaan perekaman kode oleh sensor, setelah sensor selesai perekaman data akan diterima oleh plc dan di bandingkan dari data hasil perekaman dengan data pada HMI setelah mendapatkan hasil motor akan berjalan kembali, selanjutnya *box* akan melewati sensor *limit switch* sebagai sensor *tracking*. Sensor ini akan bekerja sesuai dengan data *box* yang di dapat misalnya data *box* adalah *box A* maka harus melewati limit switch sebanyak 6 buah setelah itu servo pendorong akan mendorong bok menuju *line picking*, pada *line picking* terdapat limit swicth sebagai sensor *line picking full* dan counter dengan lampu indikator berkedip menandakan counter berjalan dan indikator menyala terus-menerus menandakan *line picking* dalam kondisi penuh.

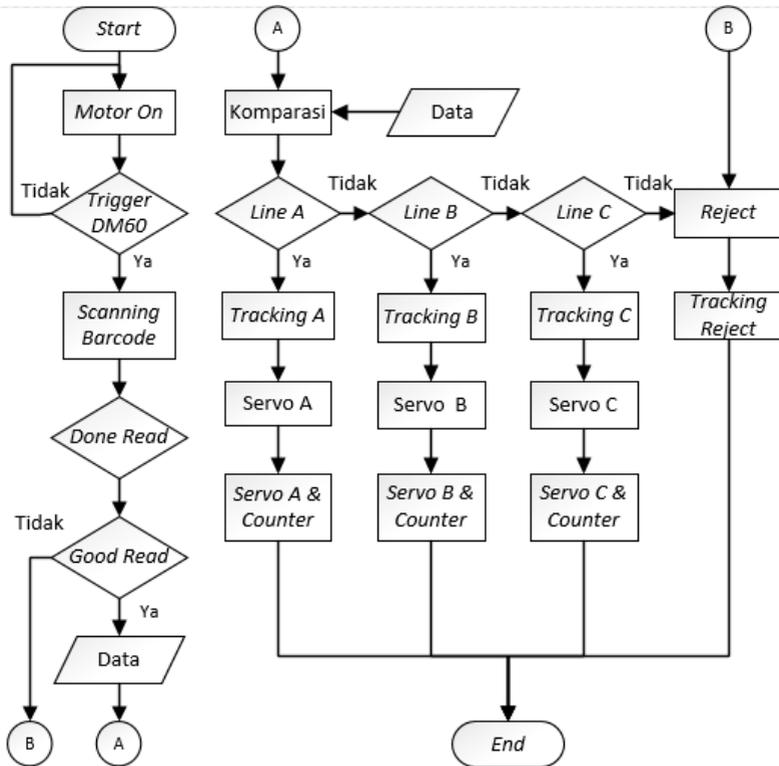
3.2.2. Membuat *Flow Chart* umum

Flow chart dibuat untuk membantu memahami alur kerja sistem yang diinginkan. Hal ini dibuat untuuk merancang sistem agar berjalan sesuai dengan yang diinginkan secara garis besarnya atau dengan garis besarnya sistem berjalan. Dibuatnya *flow chart* ini bertujuan untuk menentukan alur pemrogram yang harus dilakukan untuk sistem berjalan dengan semestinya dan juga akan berpengaruh untuk mempercepat pemrograman dikarenakan terbantu oleh garis besar bejalannya mesin ataupun sistem yang diinginkan dan mengurangi resiko terjadinya kebingungan saat memprogram.

Urutan tersebut dibuat adalah urutan mulai dari mesin dijalankan sampai selesai menjalankan semuanya. Pertama setelah mesin dijalankan yang aktif adalah motor yaitu *belt conveyor* aktif dan setelah *trigger barcode* aktif maka selanjutnya adalah proses *scanning barcode* pada eksekusi ini dilakukann sensor pendeteksi *barcode* 1D dan 2D dari cognex dan setelah selesai perekaman data maka akan mendapatkan 2 kondisi yaitu *good read* dan *bad read*, ketika *good read* maka sensor mendapatkan datanya setelah itu data akan di komparasi atau dibandingkan dengan data yang ada pada HMI untuk menentukan bahwa boox akan masuk pada *line* apa.

Setelah itu *box* akan di *tracking* untuk mengawal *box* sampai tujuan dan setelah mencapai tempat pemilhan yang dituju *box* akan

didorong oleh servo agar masuk pada *line picking* setelah itu *box* akan menekan *limit switch* dan secara otomatis yang tadi mendorong *box* akan kembali ke posisi semula dan *counter* akan berjalan menghitung *box* yang masuk pada *line picking*. *Flow chart* dapat dilihat pada Gambar 3.24 dibawah ini.



Gambar 3. 24 *Flow Chart* Sistem

Flowchart diatas dibuat untuk memudahkan pembuatan program, dikarenakan dengan adanya *flowchart* membuat program dapat mengikuti aurnya agar dengan cepat dan tepat membuat programnya.

3.2.3. Mendaftarkan *Input / Output* Sistem

Sebelum memulai pembuatan program pada PLC, perlu dibuat daftar peralatan *input* dan *output* (*input/output list*) yang digunakan pada sistem *automation sorting line system*. Hal ini penting supaya dapat mengetahui jumlah *input* dan *output* yang perlu dipenuhi oleh PLC. Hal ini sangat membantu pemrograman karena pengalamatan juga berkaitan dengan status *input* dan *output* dalam sistem.

Pertama adalah membuat *input/output list* pada Ms. Excel untuk menentukan PLC yang dipakai, agar sebuah PLC dapat bekerja maksimal sesuai spesifikasi yang dibutuhkan. Karena jika I/O nya kurang maka sistem akan kurang dalam kinerjanya dan jika I/O yang ada dalam PLC maka akan sia sia juga I/O yang tersisa tersebut, maka dari itu tentukan *input/output*nya untuk menentukan PLC yang digunakan

Input yang dibutuhkan untuk memenuhi kebutuhan sistem ini sebanyak 14 *input*, yang dimulai dari alamat 00 -13. pada tabel 3.1 ini memperlihatkan *input list* yang digunakan. *Input* di beri kode Xin untuk menandakan bahwa itu adalah *input* pada PLC agar dapat dengan mudah membedakan I/O PLC dengan memori. Tabel 3.1 berikut adalah *input* yang digunakan

Tabel 3. 1 Daftar *Input* pada sistem

No.	Address	Tag Name	Comment
1	I0.0	xIn_00_LSW	Limit switch Trigger Barcode
2	I0.1	xIn_01_LSW	Limit switch Tracking No. 1
3	I0.2	xIn_02_LSW	Limit switch Tracking No. 2
4	I0.3	xIn_03_LSW	Limit switch Tracking No. 3
5	I0.4	xIn_04_LSW	Limit switch Tracking No. 4
6	I0.5	xIn_05_LSW	Limit switch Tracking No. 5
7	I0.6	xIn_06_LSW	Limit switch Trigger Servo A
8	I0.7	xIn_07_LSW	Limit switch Trigger Servo B
9	I1.0	xIn_08_LSW	Limit switch Trigger Servo C
10	I1.1	xIn_09_LSW	Limit switch Full Box Line A
11	I1.2	xIn_10_LSW	Limit switch Full Box Line B
12	I1.3	xIn_11_LSW	Limit switch Full Box Line C
13	I1.4	xIn_12_LSW	Limit switch Full Box Reject
14	I1.5	xIn_13_SFT	Safety Device

Input yang digunakan dalam *automation sorting line system* berjumlah 14 *input* dengan begitu PLC yang digunakan adalah PLC dari Omron dengan tipe NX1P2-9024DT yang memiliki 14 *input*. Pada PLC ini semua *input* terpakai dengan maksimal untuk sistem. *Input* diberikan nama Xin untuk menandakan bahwa alamat tersebut berfungsi sebagai *input*, selanjutnya diikuti dengan bit yang digunakan pada PLC dan terakhir diberikan singkatan seperti *limit switch* disingkat menjadi LSW dan *safety device* disingkat SFT. Hal tersebut dilakukan untuk mempermudah masukan alamat sesuai dengan yang ditentukan dan juga untuk mempermudah mengetahui input apa yang dipakai pada alamat tersebut. Gambar dijelaskan pada lampiran.

Setelah semua terdaftar pada PLC maka pemrograman siap dilakukan sesuai dengan *input / output* yang diinginkan. Pada *sysmac studio* untuk memanggil *input / output* pada PLC cukup dengan menyebutkan variabel yang sudah kita tuliskan pada *input / output setting*. Pemrograman akan lebih mudah dilakukan oleh fasilitas tersebut.

3.2.4. Menerjemahkan *Flow Chart* ke Dalam Program PLC

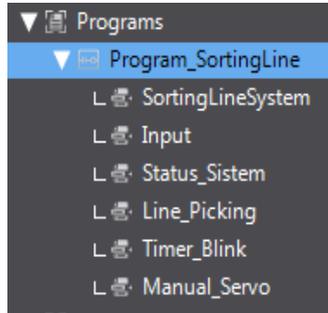
Flow chart yang sudah dibuat sebelumnya akan di terjemahkan ke dalam bahasa pemrograman PLC yaitu *Ladder diagram*, *Function block diagram* dan *Structured text*. Digunakannya 3 bahasa pemrograman ini untuk mempermudah pembuatan program dan supaya pengguna yaitu operator pabrik dapat memahami pemrograman yang sudah dibuat.

Langkah awal pembuatan program yaitu mengelompokan pemrograman sesuai dengan kebutuhan dan fungsinya. Diantaranya yaitu :

1. *Program Sorting Line*

- *Sorting Line System*
- *Input*
- *Status Sistem*
- *Line Picking*
- *Timer Blink*
- *Manual Servo*

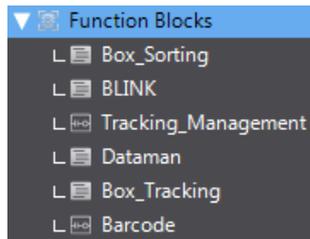
Pengelompokan pemrograman akan mempermudah memprogram dan mencari permasalahan saat terjadi *error*. Pada gambar 3.25 yaitu sub program *Sorting Line* yang memiliki beberapa *section* yaitu *Sorting Line System*, input, status sistem, line picking, dan timer blink



Gambar 3.25 Pengelompokan Pemrograman Secara Umum

Untuk meringkas program utama dibuatkan pemrograman dengan *function block* agar lebih mudah untuk memanggil program ataupun untuk menganalisa kesalahan. Terdapat beberapa *function block* untuk mendukung program utama yaitu :

- *Box Sorting*
- *Blink*
- *Tracking Management*
- *Dataman*
- *Box Sorting*
- *Barcode*

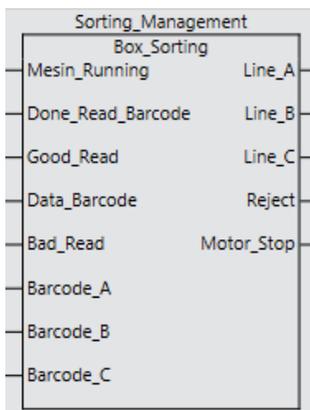


Gambar 3.26 *Function Block*

Pada gambar 3.26 *function block* digunakan untuk mendukung pemrograman. Didalam *function* digunakan 2 bahasa pemrograman yaitu *ladder diagram* dan *structure text*. Dengan cara memanggil nama variabel maka fungsi sudah siap dipakai untuk menjalankan mesin *automation sorting line system*.

Box Sorting

Function block berfungsi untuk mengkomparasi data *barcode* dari DM60 dengan data dari HMI untuk menentukan *box* masuk *line* mana sesuai dengan kode yang ada pada *box*. Pada Gambar 3.7 berikut dapat dilihat bentuk *function block box sorting management*. Ada beberapa *input* yang tersedia yaitu mesin *running* yaitu dijadikan *input* awal untuk dijadika langkah awal untuk menajalankan mesin



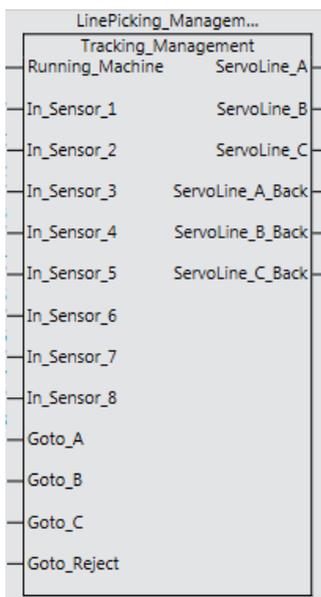
Gambar 3.27 *Function Block Box Sorting*

Tracking Management

Pada *tracking management* ini terdapat beberapa *function block* yang diibaratkan seperti orang yang dibuat untuk mengawal *box* menuju ke *line picking* yang *tracking box*, cara kerjanya yaitu dengan *tracking* sensor sesuai dengan kode yang diperoleh dari sensor DM-60. Misalnya jika kode *box* adalah A maka harus melewati 6 sensor tracking setelah itu aktuator mendorong *box* tersebut menuju *line picking* A. Disediakan 5 *function block* yang bertugas menghantarkan

box ke *line picking*, *main line* dapat menjalankan 5 *box* secara berurutan. Di dalam *function* ini memakai dua bahasa pemrograman yaitu *ladder diagram* dan *structured text*. Digunakannya dua bahasa tersebut untuk mempermudah pemrograman dan juga analisisnya, seperti halnya *contact* dan *coil* dibuat menggunakan bahasa *ladder diagram* tidak perlu menggunakan *structured text*, hal ini harus selektif untuk pemakaian bahasa pemrograman agar tidak mempersulit pemrogram dan pengguna untuk menganalisa program.

Dibawah ini Gambar 3.28 adalah bentuk dari *Function Block Tracking Management*. Dari sebuah *function* inilah terdapat berbagai macam *input* dari berbagai jenis *data type* seperti boolean, string, integer dan lain sebagainya dan juga terdapat berbagai jenis *output* dengan berbagai jenis *data type*. Karena itulah *function block* dapat meringkas suatu program yang hanya ditampilkan *input* dan *outputnya*.



Gambar 3.28 *Function Block Tracking Manajemen*

Input function block ini ada 3 jenis yaitu :

- In data

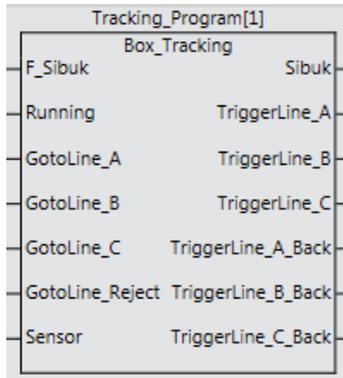
- xIn sensor (1,2,....,8)
- GoTo (A,B,C)

xIn sensor berfungsi sebagai *sensor tracking* atau sensor pelacak *box* yang berupa sensor *infrared*, yang dimaksud pelacak disini yaitu mengetahui bahwa *box* menuju ke *line picking* mana dengan menggunakan hal tersebut bok dapat dipilah sesuai dengan jenis *codenya*. *Tracking sensor* terdiri dari 5 sensor dan 3 adalah sensor tiap *line picking*, misalnya kode bok A harus melewati 6 sensor, kode bok B 7 sensor, dan kode bok C 8 sensor

Output dari *tracking management* diimplementasikan sebagai *phuser servo*, berfungsi mengaktifkan servo sebagai aktuator pendorong *box*. Seperti yang diijelaskan diatas *output* dapat berupa apapun jenis datanya, seperti yang dignakan untuk *function* ini dalah jenis data *boolean*, yaitu yang mengirim data dengan dua kondisi saja *high* dan *low*, dengan inilah *output* dapat dijadikan atau disambungkan dengan *plant* yang ada, seperti servo, led ataupun yang lainnya.

Tracking

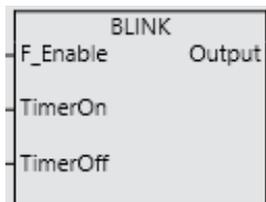
Function block tracking ini memiiki beberpa *input* yaitu diantaranya adalah FB sibuk, berfungsi untuk meberikan trigger jika *function block* sebelumnya sedang sibuk atau sedang bekerja mengawal *box* menuju *line picking*. selanjutnya *running* berfungsi sebagai input bahwa mesin sedang berjalan atau mesin dalam keadaan aktif. *GoToLine* (A, B, C, Reject), input ini berfungsi untuk mengarahkan *box* menuju *line picking* dan reject. Sensor, *output* dari *function blok* ini adalah Sibuk, Output ini berfungsi untuk memberikan trigger pada *function block* selanjutnya bahwa *function block* ini sedang sibuk menjalankan *tracking*. *Trigger Line* (A,B,C), berfungsi untuk mengaktifkan servo pada *line picking* untuk mendorong *box* menuju tempat pemilahan dan mengembalikan servo kedalam keadaan semula. *Function block* yang dimaksud adalah seperti yang terlihat pada Gambar 3.29 berikut ini.



Gambar 3.29 *Function Block Box Tracking*

Blink

Di dalam *function block* ini terdapat program untuk pewaktu nyala dan pewaktu mati yang digunakan untuk lampu berkedip seperti lampu indikator *error* dll. Di dalam *function block* ini dibuat dengan menggunakan bahasa pemrograman *structured text*, dikarenakan untuk memudahkan pemrograman pembuatan timer. Seperti pada Gambar 3.30 berikuut ini.



Gambar 3.30 *Function Block Blink*

Ada beberapa macam *input* dalam *function block* ini yaitu

- ***F Enable***
F Enable berfungsi mengaktifkan *function block blink*, bisa dikatakan ini adalah kontak untuk menjalankan *function block blink*.
- ***Timer On***

Timer On berfungsi memberikan waktu menyala / hidup, waktu dapat diatur sesuai kebutuhan

- **Timer Off**

Timer Off berfungsi memberikan waktu mati / tidak hidup, waktu dapat diatur sesuai kebutuhan

Pada *function block* *Blink* hanya memiliki 1 *output* yaitu diberi nama *output*. *Output* ini berfungsi untuk memberikan sinyal keluaran yaitu logika 1 dan 0, pada logika 1 berarti *timer on* sedang aktif dan keadaan logika 0 *timer off* sedang aktif.

Semua variabel yang digunakan lintas grup program harus di daftarkan pada global variabel dengan terdaftaranya hal tersebut variabel dapat diakses disegala program, karena variabel sudah menjadi variabel global atau variabel luas.

3.2.5. Simulasi Program PLC

Diperlukannya simulasi oleh seorang pemrogram dikarenakan *sysmac studio* memiliki fasilitas simulasi dan memiliki aliran arus listrik yang dapat dilihat oleh pemrogram dan juga pada tahap ini juga menguji *input / output* pada PLC apakah sudah terhubung dengan program atau tidak sebelum melakukan penyambungan *input/output device* dengan PLC, dengan mensimulasikan program kita dapat mengetahui apakah program yang sudah dibuat sesuai dengan yang diharapkan atau belum

3.2.6. Menghubungkan Input / Output Device Dengan PLC

Hal dilakukan pada bagian *wiring system*, karena untuk membuat suatu sistem berjalan membutuhkan kolaborasi dengan berbagai devisi untuk mencapai hasil yang maksimal. Tetapi penghubungan *input / output* ini dalam pengawasan pemrogram dikarenakan jika salah akan berpengaruh dengan program yang sudah dibuat. Pemrogram hanya memastikan bahwa semua tersambung sesuai alamat yang sudah di buat dalam *input / output setting* dalam PLC.

3.2.7. Jalankan Sistem / Plan Dengan PLC

Pelaksanaan untuk pengoperasian sistem / *plant* dengan PLC akan dibahas pada bab selanjutnya.

BAB IV

PENGUJIAN DAN ANALISA

4.1. Cara Kerja Alat

Automation sorting line system bekerja dengan menerima data barcode dimana data tersebut diperoleh dari sensor cognex DM60, setelah tu data akan dibandingkan dengan data *barcode* pada HMI. Dengan pembandingan inilah data untuk penyortiran dapat dilakukan dan alat ini memiliki 1 jalur utama (*main line*) yang berupa *belt conveyor* dan 3 jalur pemilahan (*line picking*) yang dibuat miring dengan sudut 30°. Jalur utama yang berupa *belt conveyor* untuk mentransfer barang menuju tempat pemilahan, *conveyor* dijalankan menggunakan motor DC, *belt* yang digunakan adalah kain keras. *Belt* dapat diatur tingkat kekencangannya karena kerangka untuk salah satu roller di buat tempat *adjuster* yaitu agar *belt* dapat diatur kekencangannya melalui roller tersebut. Proses pemilahan akan dilakukan menggunakan aktuator servo dan secara otomatis *box* akan terdorong ke tempat pemilahan. Gambar 4.1 memperlihatkan dari keseluruhan sistem.



Gambar 4.1 *Automation Sorting Line*

4.2. Pengujian Mekanik dan Pemrograman PLC pada *Automation Sorting Line System*

Pada sub bab ini akan membahas seluruh pengujian mekanik yang sudah dibuat. Pengujian alat ini dilakukan agar dapat menyimpulkan apakah alat sudah sesuai dengan yang diinginkan atau ada evaluasi yang akan dilakukan untuk penelitian kedepan. Mekanik merupakan komponen paling dasar sebelum melakukan hal lainnya

seperti pemasangan komponen *electric* dan wiring. Maka dari itu mekanik harus diuji terlebih dahulu untuk memastikan bahwa tahap selanjutnya dapat dilakukan agar tidak terjadi kesalahan yang mendalam yang disebabkan oleh komponen dasar yang kurang mendukung.

4.3. Pengujian Main Line

Diadakan pengujian untuk membuktikan apakah desain yang telah dibuat sesuai dengan yang diinginkan. Pengujian ini menjalankan *conveyor main line* tanpa beban dan dengan menggunakan beban apakah sama dapat berjalan sesuai yang diharapkan atau tidak. Pengujian ini dilakukan seperti pada Gambar 4.2.



Gambar 4.2 Pengujian tanpa beban

Dengan dilakukannya pengujian tanpa beban *belt conveyor* mengalami kendala yaitu semakin lama *conveyor* berjalan *belt* semakin miring dan akhirnya terlalu mepet dengan *body conveyor*. Dengan kendala seperti itu maka diatasi dengan cara mengencangkan *belt* pada satu sisinya agar tidak bergesekan dengan *body*. Selanjutnya dilakukan pengujian ulang dengan menggunakan beban sebanyak 3 *box* seperti yang terlihat pada Gambar 4.3 berikut ini.

Uji coba ini akan memperlihatkan 3 *box* berjalan bersamaan untuk mengetahui bahwa ketika pemilahan dengan beban 3 buah *box* secara bersamaan apakah *conveyor* dapat berjalan atau masih terdapat kendala.



Gambar 4.3 Pengujian menggunakan beban 3 *box*

Setelah dilakukan pengujian dengan 3 beban *box*, konveyor mengalami beberapa kali tersendat dikarenakan *belt slip* pada *roller*. Dengan adanya kendala tersebut maka cara untuk mengatasi hal tersebut *belt* conveyor dikencangkan untuk menghindari *slip* pada *roller*. Pengujian dilakukan menggunakan 3 beban dikarenakan proses pemilihan pada *line picking* terdapat 3 jalur pemilahan. Dapat disimpulkan setelah *belt* conveyor dikencangkan adalah conveyor pada *main line* dapat berjalan dengan baik untuk menghantarkan *box* pada proses pemilahan

Tabel 4. 1 Pengujian *Main Line*

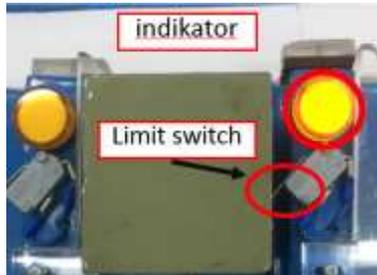
No.	Beban <i>Box</i>	Hasil
1	Tanpa <i>box</i>	Konveyor lancar
2	1 <i>box</i>	<i>Box</i> lancar
3	2 <i>box</i>	<i>Box</i> lancar
4	3 <i>box</i>	<i>Box</i> lancar

Dari tabel pengujian 4.1 dapat disimpulkan bahwa mekanik *main line* berjalan dengan baik tanpa beban dan menggunakan 3 beban sekaligus.

4.4. Pengujian *Line Picking*

Dilakukannya pengujian untuk membuktikan apakah mekanik yang telah dibuat sesuai dengan kenyataan yang ada atau tidak.

Pengujian meliputi penurunan *box* pada *line picking*, setelah *box* terdorong oleh servo *box* akan mengenai *limits switch* pada *line picking* yang berfungsi sebagai *counter* dan sensor *line full* seperti yang ditunjukkan pada Gambar 4.4 dibawah ini.



Gambar 4.4 Pengujian Sensor *Limit Switch* pada *Line Picking*

Pada pengujian diatas pada *line* dua *box* tersangkut dikarenakan sensor *body* penahan yang terbuat dari *acrylic* terlalu menjorok kedalam yang mengakibatkan *box* tertahan oeh *body* dan *sensor* dengan danya kendala tersebut maka dilakukan pengikisan *acrylic* agar *box* tidak tersangkut kembali. Hal tersebut berhasil dengan presentasi uji coba 10 kali berhasil semua tanpa ada *box* yang tersangkut kembali.

Pengujian selanjutnya ini dilakukan seperti pada Gambar 4.5 berikut yaitu menguji apakah *box* dapat turun dengan baik setelah terdorong oleh servo atau belum.



Gambar 4.5 Mekanik *Line Picking*

Dari hasil pengujian pada *line picking box* dapat turun dengan baik sesuai yang diharapkan pada desain mekanik *line picking* yang otomatis *box* turun setelah terdorong oleh servo dan *limit switch* berfungsi sesuai dengan fungsinya yaitu sebagai *counter* dan sensor *line full*. *Limit switch* sebagai counter jika tertahan kurang dari 1 detik dan ketika *limit switch* tertekan selama lebih dari 2 detik maka berfungsi sebagai pendeteksi *line picking full* maka *buzzer* akan berbunyi untuk memberi tanda bahwa *line picking* ada yang penuh dan harus diambil *box*nya agar dapat memilah kembali pada *line* tersebut.

4.5. Pengujian Mekanik Aktuator

Mekanik aktuator yaitu berfungsi untuk penyangga atau tempat servo dan mekanik pendorong *box*. Pengujian ini dilakukan untuk melihat hasil desain mekanik yang sudah dibuat, dari pengujian penyangga aktuator dapat dijalankan dengan aman dikarenakan dapat diatur ketinggian dan kekuatannya juga terbilang aman dengan adanya *adjuster*, aktuator dapat diatur ketinggiannya untuk menyesuaikan dengan keadaan yang ada. Pengujian dilakukan seperti pada Gambar 4.6 berikut ini.



Gambar 4.6 Mekanik Aktuator

Pengujian untuk pendorong sendiri dilakukan dengan cara menggerakkan servo dan meletakkan *box* didepannya, untuk hasil yang diperoleh adalah bok terdorong oleh pendorong yang sudah didesain

dengan huruf T terbalik agar dapat mendorong bok dengan baik. Pengujian dilakukan seperti Gambar 4.7

Hal tersebut yaitu *box* dapat terdorong tanpa ada miring saat proses pendorongan. Maka dari itu dengan desain tersebut cocok digunakan untuk pendorong dengan *box* tetap berjalan tanpa berhenti ketika melakukan pendorongan.



Gambar 4.7 Mekanik Pendorong *Box*

4.6. Pengujian Mekanik Pelindung

Pada pelindung ini dilakukan pengujian untuk memastikan apakah pelindung sudah terpasang sesuai yang diinginkan dan kekuatan pelindung apakah dapat diandalkan atau perlu di perbaiki lagi. Pengujian dilakukan seperti Gambar 4.8. pelindung dibuat sesuai dengan desain tetapi hasilnya tidak sesuai yaitu dimensi pelindung terlalu lebar dari desain yang sudah dibuat. Oleh karena itu pelindung di pasang dengan menambahkan besi persegi untuk memberikan isi pada ruang yang berlebih.

Pengujian dilakukan dengan menggoyangkan pelindung untuk membuktikan bahwa pelindung sudah kuat untuk melindungi alat tersebut. Uji buka tutup pintu pelindung untuk memastikan apakah pintu yang terintegrasi dengan *safety device* berfungsi sempurna atau belum. Dengan terujinya pelindung ini dapat membuat sistem menjadi aman dibandingkan dengan tana pelindung. Hal tersebut untuk mengurangi resiko kecelakaan kerja yang dipengaruhi pada faktor-faktor yang salah satunya adalah kelelahan atau jatuh yang tinggi.



Gambar 4.8 Mekanik Pelindung

Dari hasil pengujian tersebut didapatkan kekuatan dan pintu pelindung aman untuk diterapkan pada *automation sorting line system*. dinyatakan aman ditinjau dari kekuatan untuk melindungi gangguan dari luar sistem atau lebih menuju ke keselamatan kerja operator.

4.7. Pengujian Mekanik Penyangga DM60 dan Motor DC

Penyangga DM60 sebagai sensor pendeteksi *barcode* dan motor DC sebagai penggerak konveyor ini diuji agar mengetahui apakah sudah sesuai atau tidak dengan yang diinginkan pada *automation sorting line system*. Pengujian dilakukan pada mekanik penyangga sensor DM60 yaitu dengancara merakit mekanik dan memfokuskan titik tembak *barcode* pada *box* seperti Gambar 4.9 berikut.



Gambar 4.9 Penyangga Sensor DM60

Dari hasil perakitan mekanik dapat berfungsi sesuai dengan yang diinginkan, dengan memiliki *adjuster* pada alat ini membuat titik fokus dapat diatur ketinggiannya.

Pengujian penyangga motor DC dilakukan dengan cara memasang motor dc lalu digerakan . Cara itu dilakukan untuk pengujian apakah mekanik penyangga mampu menahan getaran dari motor dc atau tidak. Hasil yang diperoleh mekanik mumpuni untuk menyangga motor dc. Seperti yang ditunjukkan pada Gambar 4.10 berikut ini.



Gambar 4.10 Penyangga Motor DC

Dari pengujian diatas dinyatakan bahwa mekanik motor DC dapat berfungsi dengan baik dengan adanya *adjuster* yang terdapat dalam mekanik ini mengatasi permasalahan kurang akuratnya posisi sensor dan motor yang disebabkan tidak presisi dengan objek, adanya *adjuster* ini permasalahan tersebut dapat diatasi.

4.8. Analisa Program Sorting Line

Pada Sub Program ini terdapat beberapa section program yang dipakai untuk menjalankan alat *sorting line* yaitu *sorting line system*, *input*, status sistem, *line picking*, *timer blink*. Secara lebih lengkapnya akan dibahas pada sub bab berikutnya.

4.9. Sorting Line

Sorting Line adalah tempat dimana seluruh program diintegrasikan agar sebuah sistem berjalan sesuai dengan yang diharapkan. Didalamnya terdapat 3 jenis bahasa pemrograman yaitu *Ladder Diagram*, *Function Block Diagram*, dan *Structure Text*. Seperti yang terlihat pada gambar 4.11 di bawah ini.



Gambar 4.11 Tampilan Pemrograman

Terdapat beberapa *function block* didalamnya yaitu :

- *DM_Connect*
Function Block ini berisi skuensial komunikasi dari sensor DM60
- *Barcode*
Function block ini berisi untuk mengkomparasi data *barcode* dari HMI dan sensor DM60
- *Box_Sorting*
Function Block ini berisi mengenai proses sorting
- *Box_Tracking*

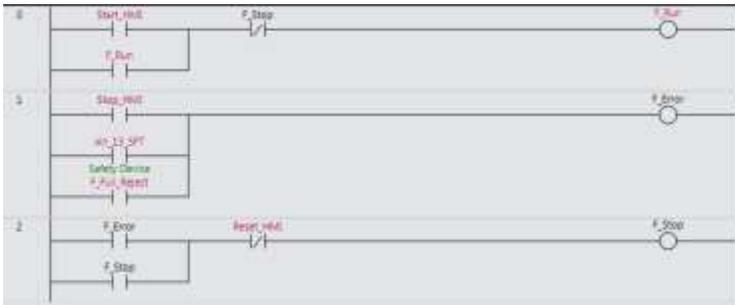
Function Block ini berisi mengenai proses *tracking box* menuju ke *line picking*.

Fucntion yang pertama kali dieksekusi dalam program setelah mesin berjalan adalah *Box_Sorting* karena input pertama kali masuk akan menuju ke limit switch trigger barcode dalam hal ini memiliki alamat *xin_00* setelah itu menunggu data dari function block tersebut, setelah data diterima akan diproses kedalam *Box_sorting* dan akan dicocokkan dengan data pada HMI kemudian kecocokan akan dikirim ke function block *box_tracking*, fungsi function block ini adalah mengantar bok menuju *line picking* yang akan dipilah menurut kode yang sudah di dapat dari hasil penyamaan dari function block *box sorting*.

Ladder diagram menjadi Bahasa pemrograman yang utama dikarenakan Bahasa tersebut sudah umum digunakan didalam pemrograman PLC dan juga mudah dipahami dalam pengaplikasiannya. Setelah itu terdapat *function block diagram*, digunakannya bahasa pemrograman tersebut ditujukan untuk mempermudah pemrograman dan juga memperingkas pemrograman.

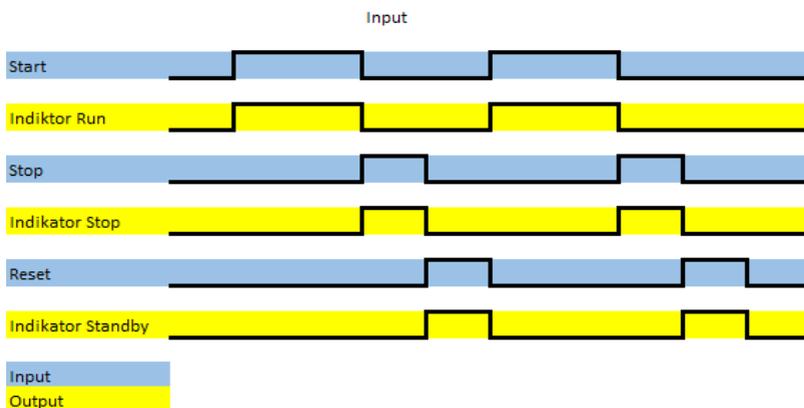
4.10. Input

Pemrograman ini dibuat untuk menjalankan sistem. Program yang digunakan adalah pemrograman dengan *ladder diagram* seperti yang ditunjukkan pada Gambar 4.12 berikut, digunakannya bahasa tersebut dikarenakan lebih mudah untuk mengatifikkan contact dan coi pada sistem.



Gambar 4.12 Pemrograman *Input*

Analisa pemrogram dilakukan dengan menggunakan *timing chart* untuk mengetahui bagaimana program bekerja dan bagaimana pengeksekusian *sequensial*. Hal ini ditunjukkan pada Gambar 4.13 berikut ini. Dengan adanya *timing chat* memudahkan progeammer dan pembaca menganalisa sebuah program.

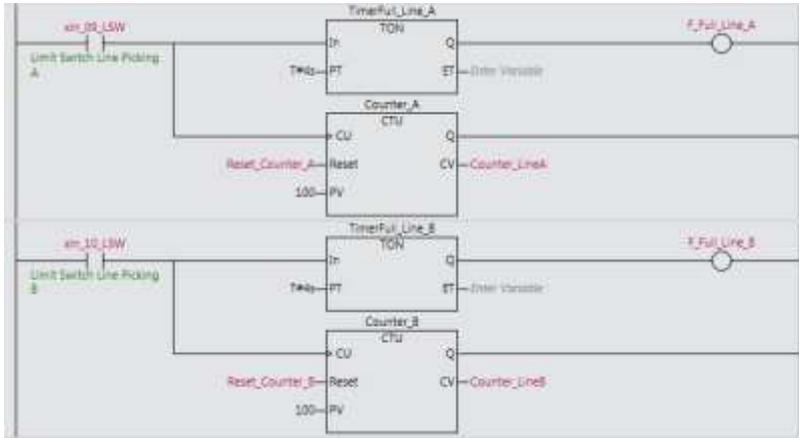


Gambar 4.13 *Timing Chart Input*

Pada HMI terdapat 3 tombol yaitu *start*, *stop*, dan *reset* seperti pada namanya seperti *start* yaitu berfungsi untuk mengaktifkan mesin dan secara otomatis indikator *run*, *stop* berfungsi untuk mengheentikan sistem apabila terjadi *error* maupun kendala yang lain yang akan diberi isyarat dengan lampu indikator stop warna merah menyala dan *reset* adalah untuk mengembalikan ke kondisi normal ataupun awal.

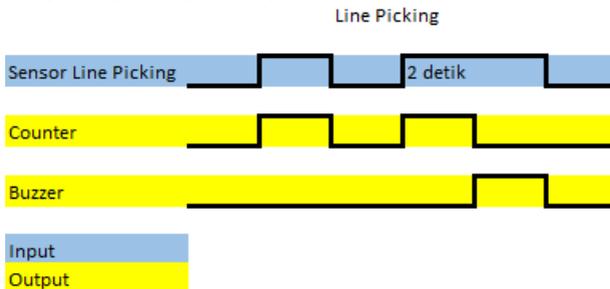
4.11. *Line Picking*

Pada pemrograman ini berfungsi untuk menunjukkan status *line picking* seperti jumlah *box* yang masuk pada *line picking* dan untuk menunjukkan apakah *line* penuh atau tidak yang diatur menggunakan timer selama 2 detik setelah limit switch tertekan selama waktu yang ditentukan maka inikator buzzer akan menyala menandakan bahwa *line picking* dalam kondisi penuh. Pemrograman dapat dilihat pada Gamabar 4.14 berikut ini.



Gambar 4.14 Pemrograman Line Picking

Analisa dilakukan dengan *timing chart* sama halnya yang dilakukan untuk analisa program yang lain untuk mengetahui kinerja program tersebut. Sensor *limit switch* merupakan sensor untuk penghitung jumlah *box* yang masuk dan sebagai sensor untuk mengetahui bahwa *line* sudah penuh atau belum. Squensial pada program *line picking* ditunjukkan pada Gambar 4.15 berikut.



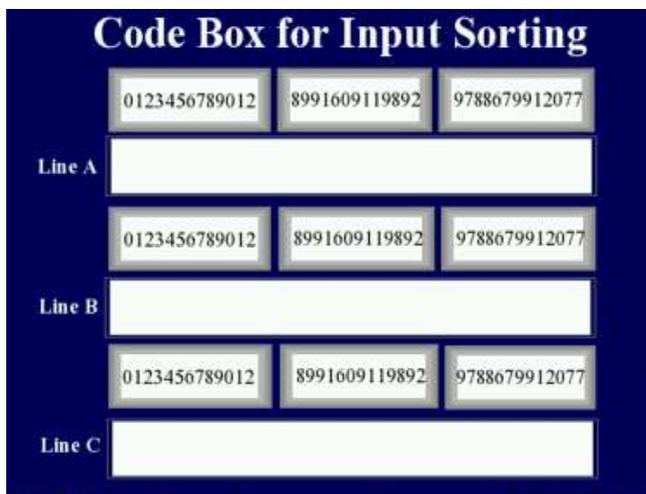
Gambar 4.15 Timing Chart Line Picking

Ketika sensor tertekan selama 2 detik maka *line* penuh, secara otomatis buzzer akan menyala selama sensor masih aktif dan *buzzer*

akan mati ketika sensor sudah tidak aktif. Jika sensor tertekan kurang dari 2 detik maka sensor berfungsi sebagai *counter* untuk menghitung jumlah *box* yang masuk pada *line picking*.

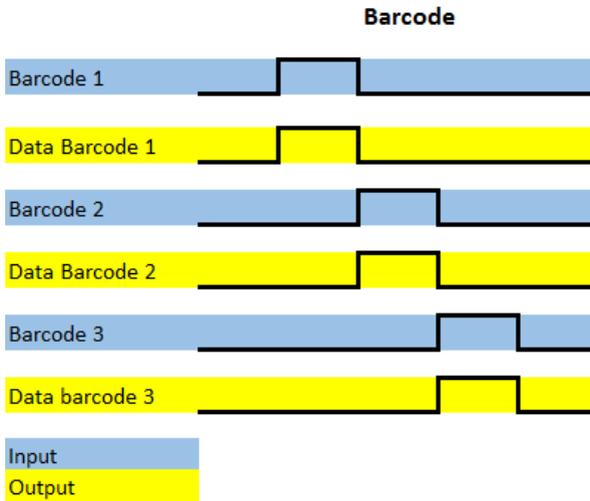
4.12. *Function Block Barcode*

Function blok ini merupakan fungsi untuk memanggil data barcode yang sudah dipilih pada panel HMI untuk ditampilkan pada layar. Data yang sudah dipilih akan disimpan pada memori PLC dan dibandingkan dengan data dari sensor DM60. Program *barcode* dapat dilihat pada Gambar 4.16 dibawah ini. Dengan adanya tombol ini yang menyediakan tiga jenis *code* memudahkan operator memilih *code* yang diinginkan sesuai dengan *line picking* yang diinginkan tanpa harus menulis dari awal, karena banyak resiko dalam penulisan yang disebabkan kurang telitinya operator dan faktor kelelahan ataupun kejenuhan operator. Selanjutnya mengakibatkan data tidak dapat dibandingkan dengan hasil data dari sensor DM60, membuat *box* tidak akan tersortir dan *box* akan menuju *line* tempat pembuangan atau sering disebut *reject*.



Gambar 4.16 Barcode pada HMI

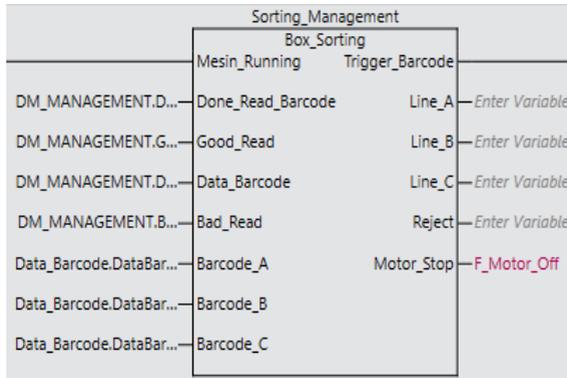
Timing chart ini difungsikan untuk mengetahui cara kerja program *barcode* yang ada pada pane HMI. Setelah tombol pada HMI ditekan maka akan secara otomatis data akan terpilih sesuai dengan yang ditekan pada HMI tersebut. Analisa program ditunjukkan pada Gambar 4.17 berikut ini.



Gambar 4.17 *Timing Chart Barcode*

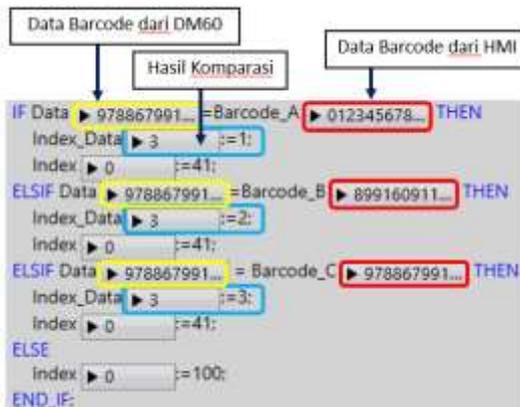
4.13. Function Block *Box Sorting*

Function blok ini berfungsi untuk menerima data dari *function blok barcode* dan menerima data dari sensor DM60 setelah itu dikomparasi untuk menentukan penyortiran *box* menuju ke *line picking*. Bentuk dari *function blok* tersebut seperti yang diperlihatkan pada Gambar 4.18. didalamnya memiliki banyak *input/output* untuk menjalankan program pada sistem *automation sorting line system*



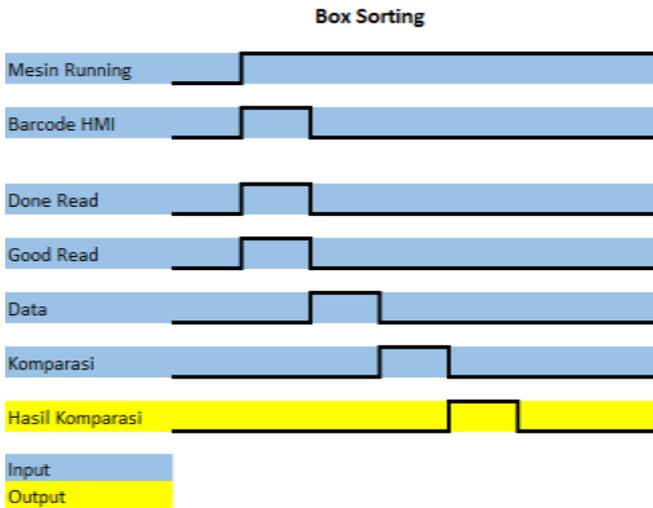
Gambar 4.18 Function Block Sorting Management

Data yang di komparasi adalah berbentuk *string* 13 digit setelah di komparasi akan menghasilkan data indeks yaitu berfungsi untuk penanda bahwa data hasil komparasi disimpan pada index data tersebut dan itu bertipe *integer index* 1 berarti *box* akan masuk pada *line* A dan seterusnya. Hal tersebut sama dengan yang ditunjukkan pada Gambar 4.19 berikut ini



Gambar 4.19 Hasil Uji Coba

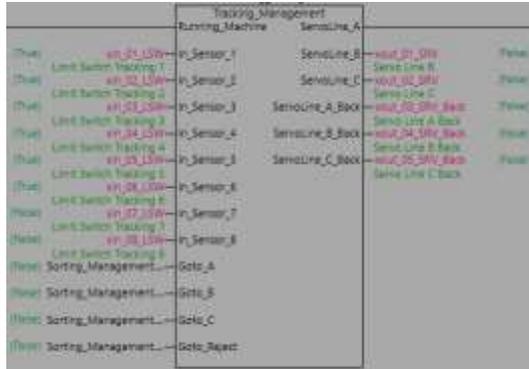
Analisa program digunakan untuk mengetahui bahwa program berjalan dengan baik seperti pada Gambar 4.20 berikut ini yaitu sama dengan sebelumnya yaitu menggunakan *timing chart* untuk menganalisisnya. *done read* didapat dari sensor DM60 yang sudah selesai pembacaan *barcodenya* setelah itu akan menghasilkan keputusan dari hasil kerja sensor tersebut yaitu *goog read* dan *bad read* ketika *good read* maka sensor DM60 akan mengirim data hasil *scanning* tetapi jika *bad read* maka sensor tidak akan menghasilkan data atau dapat dikatakan kosong. Setelah itu proses kompaasi untuk mengerahui *box* tersebut masuk *line* A, B, atau C.



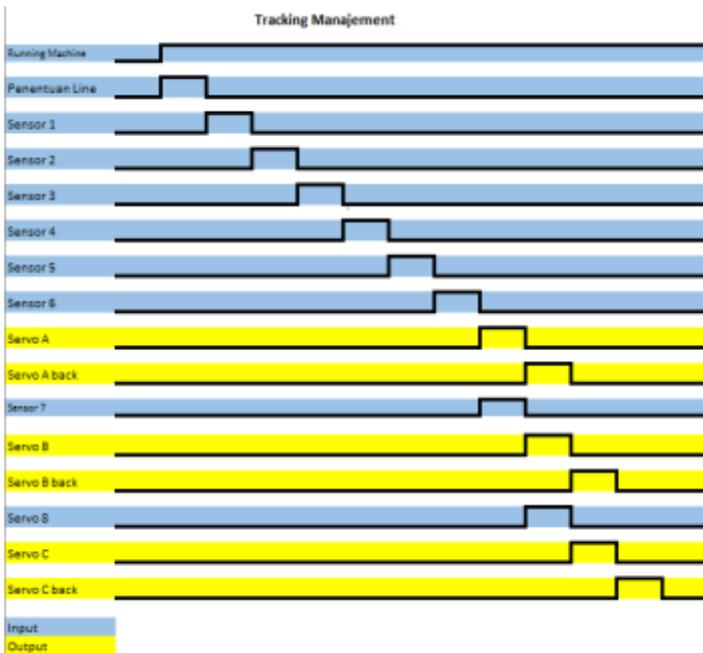
Gambar 4.20 *Timing Chart Box Sorting*

4.14. Function Block Tracking Management

Function block tracking management berfungsi untuk menerima data dari sensor *tracking* untuk melacak bok sudah sampai dimana dan untuk memastikan bahwa *box* sudah berjalan untuk proses pemilahan dan akan mengeluarkan berupa data biiner (bool) yang akan mentrigger servo untuk mendorong *box* menuju tempat pemilahan. *Funnction block tracking management* dapat dilihat seperti pada Gambar 4.21 dibawah ini.



Gambar 4.21 *Function Block Tracking Management*

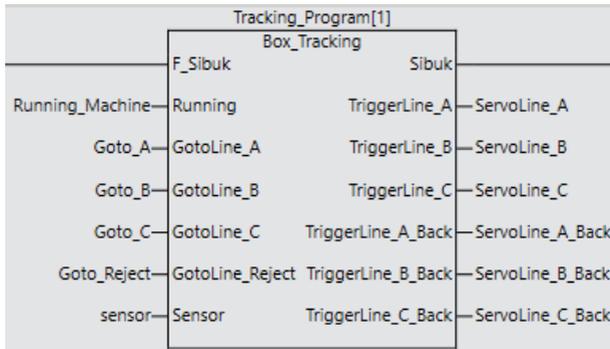


Gambar 4.22 *Timing Chart Tracking Management*

Analisa menggunakan *timing chart* untuk mengetahui bahwa program berjalan sesuai yang kita inginkan seperti yang ditunjukkan pada Gambar 4.22 di atas. Dengan demikian suatu program alur eksekusinya dapat dilihat melalui *timing chart*.

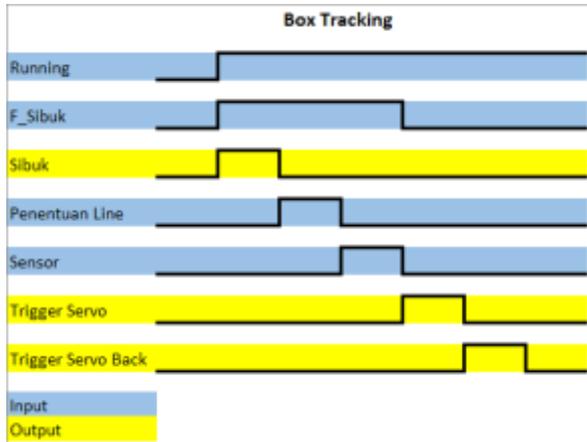
4.15. Function Block *Box Tracking*

Function block box tracking berfungsi untuk mengolah data dari *tracking management* yaitu dari sensor *tracking limit switch*, setelah data masuk *box* menuju ke *line A,B* atau *C* sensor *tracking* bekerja dan selanjutnya akan mentrigger servo dan dikeluarkan dari *function block tracking management*. Jadi *box tracking* ini ada dialam *tracking management* program yang dibuat dapat dilihat pada Gambar 4.23 dibawah ini.



Gambar 4.23 *Function Block Box Trackin*

Untuk melakukan analisa program dilakukan dengan *timing chart* untuk mengatahui *sqquential* dari program tersebut agar mudah untuk menganalisa apakah program suddah berjalan sesuai yang kita inginkan atau belum. *Timing chart* dapat dilihat pada Gambar 4.24 dibawah ini.



Gambar 4.24 *Timing Chart Box Tracking*

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

BAB V

PENUTUP

Rancang bangun mekanik dan pemrogram PLC yang telah dibuat dengan melewati berbagai uji coba dan mengalami berbagai kendala dan berbagai kondisi uji yang bertujuan untuk menemukan atau menghasilkan sebuah alat yang dapat bekerja semaksimal mungkin dan dapat diaplikasikan didunia nyatanya. Dengan adanya tujuan seperti ituah maka dengan uji coba dapat diketahui kemampuan alat sampai dimana dan jika masih ada kekurangan dengan melalui hal tersebut dapat diperbaiki lagi untuk di uji coba lagi ampai meneumakn hasil yang terbaik.

Dari berbagai uji coba yang dilakukan alat ini mempunyai kekurangan dan kelebihan antara lain dari kelebihan dan kekurangannya dalah 1. Kelebihan dari alat ini ialah *line picking* dibuat dengan model miring yang memiliki sudut 30^0 dan secara otomatis *box* akan turun pada tempat yang sudah di siapkan seteah terdorong oleh aktuator servonya, kedua adalahh sistem ini dilengkapi dengan *body* pelindung yang kokoh dan dapat diartikan tingkat keamanan dari sistem ini sudah baik diakrenakan *body* juga dlengkapi dengan *safety device* yang berfungsi apabila *body* terbuka maka sistem akan berhenti sejenak sampai pintu ditutup dan seteah itu di aktifkan kembali.kelemahannya adalah sensor *infrared* kurang akurat dikarenakan dari 10 percobaan terdapat 2 kali sensor tidak mendeteksi adanya *box*. Kelemahan selanjutnya adalah servo yang dipakai kurang maksimal seperti kurang akuratnya servo berputar untuk mendorong *box* yang mengakibatkan *box* tersangkut *body*.

Dapat disimpulkan dari kelebihan dan kelemahannya alat ini layak pakai untuk penyortiran barang dan dapat menyortir lebih tepat daripada dengan tenaga manusia.Untuk pengembangan lebih lanjut mungkin jauh lebih baik apabila rancang bangun dan pemrograman di padukan dengan konsep yang lebih matang dan dengan tingkat error yang lebih sedikit. Yang dapat dilakukan dengan komponenkomponen yang lebih baik ;agi.

Dengan demikian sistem *automation sorting line sistem* berjalan dengan maksimal dan sesuai dengan yang diharapkan tanpa banyak mengatur ataupun memodifikasi untuk penyesuaian.

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

DAFTAR PUSTAKA

- [1] S. P. Dabade, "Research in Engineering Automatic Sorting Machine Using Conveyor Belt," no. June 2015, 2016.
- [2] R. Pourdarbani, H. R. Ghassemzadeh, H. Seyedarabi, F. Z. Nahandi, and M. Moghaddam Vahed, "No Title," *Study an Autom. sorting Syst. Date Fruits*.
- [3] S. V Kulkarni, S. R. Bhosale, P. P. Bandewar, and P. G. B. Firame, "Automatic Box Sorting Machine," vol. 4, no. 04, pp. 57–58, 2016.
- [4] A. M. Zainuri, *Mesin Pemindah Bahan*. Yogyakarta: ANDI, 2008.
- [5] S. Wang, K. Hu, and D. Li, "Analysis and experimental research on air gap characteristics of permanent magnet low-resistance belt conveyor," *IET Sci. Meas. Technol.*, vol. 12, no. 4, pp. 472–478, 2018.
- [6] C. Oktora and R. I. O. Susanto, "PROTOTYPE PEMILAH BOK BERDASARKAN UKURAN FINAL PROJECT SORTING PROTOTYPE BASED ON BOX SIZE FACULTY OF SCIENCE AND TECHNOLOGY," 2017.
- [7] Hustanto and Thomas, *PLC (Programmable Logic Control) FP Sigma*. Yogyakarta: ANDI, 2007.
- [8] W. Handy, *Programmable Logic Controller Teori, Pemrograman dan Aplikasinya dalam Otomasi Sistem*. Yogyakarta: Graha Ilmu, 2009.
- [9] O. Boards, S. U. Units, P. Output, and S. Communications, "Machine Automation Controller NX1P2-."
- [10] Suhendar, *Programmable Logic Controller*. Yogyakarta: Graha Ilmu.
- [11] I. Setiawan, *Programmable Logic Control (PLC) dan Teknik Perancangan Sistem Kontrol*. Yogyakarta: ANDI, 2006.
- [12] I. Setiawan, "Programmable Logic Controller dan Teknik Perancangan Sistem Kontrol," pp. 1–14, 2006.
- [13] W. Bolton, *Programmable Logic Controllers*. 2009.
- [14] W. Resources and O. F. South, "Version 1," 2009.
- [15] R. Manual, "DataMan 60," 2017.

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

LAMPIRAN

I/O Map

Built-in I/O	Built-in I/O Settings					
Input Bit 00	Input Bit 00	R	BOOL	in_00_LSW	Limit Switch Trigger Barcode	
Input Bit 01	Input Bit 01	R	BOOL	in_01_LSW	Limit Switch Tracking 1	
Input Bit 02	Input Bit 02	R	BOOL	in_02_LSW	Limit Switch Tracking 2	
Input Bit 03	Input Bit 03	R	BOOL	in_03_LSW	Limit Switch Tracking 3	
Input Bit 04	Input Bit 04	R	BOOL	in_04_LSW	Limit Switch Tracking 4	
Input Bit 05	Input Bit 05	R	AGC0	in_05_LSW	Limit Switch Tracking 5	
Input Bit 06	Input Bit 06	R	BOOL	in_06_LSW	Limit Switch Tracking 6	
Input Bit 07	Input Bit 07	R	BOOL	in_07_LSW	Limit Switch Tracking 7	
Input Bit 08	Input Bit 08	R	BOOL	in_08_LSW	Limit Switch Tracking 8	
Input Bit 09	Input Bit 09	R	BOOL	in_09_LSW	Limit Switch Line Picking A	
Input Bit 10	Input Bit 10	R	BOOL	in_10_LSW	Limit Switch Line Picking B	
Input Bit 11	Input Bit 11	R	AGC0	in_11_LSW	Limit Switch Line Picking C	
Input Bit 12	Input Bit 12	R	BOOL	in_12_LSW	Limit Switch Reject	
Input Bit 13	Input Bit 13	R	BOOL	in_13_SFT	Safety Device	
Output Bit 00	Output Bit 00	RW	BOOL	out_00_SRV	Servo Line A	
Output Bit 01	Output Bit 01	RW	BOOL	out_01_SRV	Servo Line B	
Output Bit 02	Output Bit 02	RW	BOOL	out_02_SRV	Servo Line C	
Output Bit 03	Output Bit 03	RW	BOOL	out_03_SRV_Back	Servo Line A Back	
Output Bit 04	Output Bit 04	RW	BOOL	out_04_SRV_Back	Servo Line B Back	
Output Bit 05	Output Bit 05	RW	BOOL	out_05_SRV_Back	Servo Line C Back	
Output Bit 06	Output Bit 06	RW	BOOL	out_06_LED	Indikator Run	
Output Bit 07	Output Bit 07	RW	BOOL	out_07_LED	Indikator Stop	
Output Bit 08	Output Bit 08	RW	BOOL	out_08_MTH	Motor DC	
Output Bit 09	Output Bit 09	RW	BOOL	out_09_BDR		

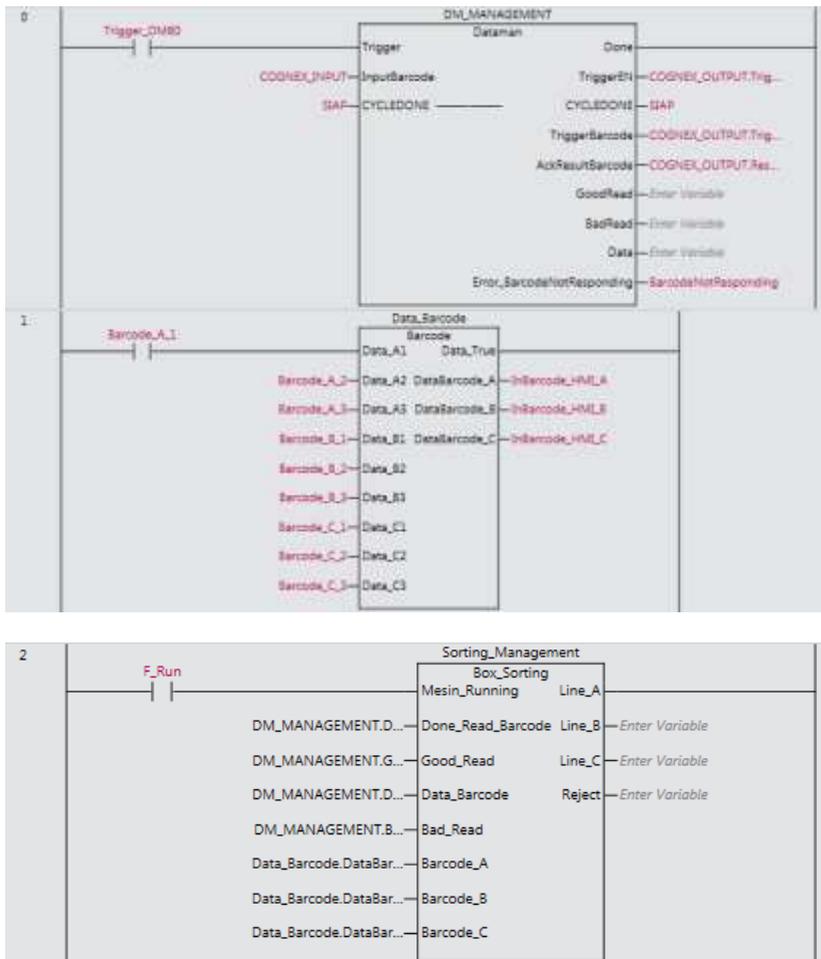
Global Variabel

in_00_LSW	BOOL	BuiltInIOV...	<input type="checkbox"/>	<input type="checkbox"/>	Do not publish	Limit Switch Tr...
in_01_LSW	BOOL	BuiltInIOV...	<input type="checkbox"/>	<input type="checkbox"/>	Do not publish	Limit Switch Tr...
in_02_LSW	BOOL	BuiltInIOV...	<input type="checkbox"/>	<input type="checkbox"/>	Do not publish	Limit Switch Tr...
in_03_LSW	BOOL	BuiltInIOV...	<input type="checkbox"/>	<input type="checkbox"/>	Do not publish	Limit Switch Tr...
in_04_LSW	BOOL	BuiltInIOV...	<input type="checkbox"/>	<input type="checkbox"/>	Do not publish	Limit Switch Tr...
in_05_LSW	BOOL	BuiltInIOV...	<input type="checkbox"/>	<input type="checkbox"/>	Do not publish	Limit Switch Tr...
in_06_LSW	BOOL	BuiltInIOV...	<input type="checkbox"/>	<input type="checkbox"/>	Do not publish	Limit Switch Tr...
in_07_LSW	BOOL	BuiltInIOV...	<input type="checkbox"/>	<input type="checkbox"/>	Do not publish	Limit Switch Tr...
in_08_LSW	BOOL	BuiltInIOV...	<input type="checkbox"/>	<input type="checkbox"/>	Do not publish	Limit Switch Tr...
in_09_LSW	BOOL	BuiltInIOV...	<input type="checkbox"/>	<input type="checkbox"/>	Do not publish	Limit Switch Li...
in_10_LSW	BOOL	BuiltInIOV...	<input type="checkbox"/>	<input type="checkbox"/>	Do not publish	Limit Switch Li...
in_11_LSW	BOOL	BuiltInIOV...	<input type="checkbox"/>	<input type="checkbox"/>	Do not publish	Limit Switch Li...
in_12_LSW	BOOL	BuiltInIOV...	<input type="checkbox"/>	<input type="checkbox"/>	Do not publish	Limit Switch R...
in_13_SFT	BOOL	BuiltInIOV...	<input type="checkbox"/>	<input type="checkbox"/>	Do not publish	Safety Device
out_00_SRV	BOOL	BuiltInIOV...	<input type="checkbox"/>	<input type="checkbox"/>	Do not publish	Servo Line A
out_01_SRV	BOOL	BuiltInIOV...	<input type="checkbox"/>	<input type="checkbox"/>	Do not publish	Servo Line B
out_02_SRV	BOOL	BuiltInIOV...	<input type="checkbox"/>	<input type="checkbox"/>	Do not publish	Servo Line C
out_03_SRV_Back	BOOL	BuiltInIOV...	<input type="checkbox"/>	<input type="checkbox"/>	Do not publish	Servo Line A B...
out_04_SRV_Back	BOOL	BuiltInIOV...	<input type="checkbox"/>	<input type="checkbox"/>	Do not publish	Servo Line B B...
out_05_SRV_Back	BOOL	BuiltInIOV...	<input type="checkbox"/>	<input type="checkbox"/>	Do not publish	Servo Line C B...
out_06_LED	BOOL	BuiltInIOV...	<input type="checkbox"/>	<input type="checkbox"/>	Do not publish	Indikator Run
out_07_LED	BOOL	BuiltInIOV...	<input type="checkbox"/>	<input type="checkbox"/>	Do not publish	Indikator Stop
out_08_MTH	BOOL	BuiltInIOV...	<input type="checkbox"/>	<input type="checkbox"/>	Do not publish	Motor DC
out_09_BDR	BOOL	BuiltInIOV...	<input type="checkbox"/>	<input type="checkbox"/>	Do not publish	

Start_HMI	BOOL		%W0.00	<input type="checkbox"/>	<input type="checkbox"/>	Do not publish
Stop_HMI	BOOL		%W0.01	<input type="checkbox"/>	<input type="checkbox"/>	Do not publish
Reset_HMI	BOOL		%W0.02	<input type="checkbox"/>	<input type="checkbox"/>	Do not publish
Barcode_A_1	BOOL		%W7.00	<input type="checkbox"/>	<input type="checkbox"/>	Do not publish
Barcode_A_2	BOOL		%W7.01	<input type="checkbox"/>	<input type="checkbox"/>	Do not publish
Barcode_A_3	BOOL		%W7.02	<input type="checkbox"/>	<input type="checkbox"/>	Do not publish
Barcode_B_1	BOOL		%W7.03	<input type="checkbox"/>	<input type="checkbox"/>	Do not publish
Barcode_B_2	BOOL		%W7.04	<input type="checkbox"/>	<input type="checkbox"/>	Do not publish
Barcode_B_3	BOOL		%W7.05	<input type="checkbox"/>	<input type="checkbox"/>	Do not publish
Barcode_C_1	BOOL		%W7.06	<input type="checkbox"/>	<input type="checkbox"/>	Do not publish
Barcode_C_2	BOOL		%W7.07	<input type="checkbox"/>	<input type="checkbox"/>	Do not publish
Barcode_C_3	BOOL		%W7.08	<input type="checkbox"/>	<input type="checkbox"/>	Do not publish
Reset_Counter_A	BOOL		%W7.09	<input type="checkbox"/>	<input type="checkbox"/>	Do not publish
Reset_Counter_B	BOOL		%W7.10	<input type="checkbox"/>	<input type="checkbox"/>	Do not publish
Reset_Counter_C	BOOL		%W7.11	<input type="checkbox"/>	<input type="checkbox"/>	Do not publish
InBarcode_HMI_A	STRING[15]		%D0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Do not publish
InBarcode_HMI_B	STRING[15]		%D8	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Do not publish
InBarcode_HMI_C	STRING[15]		%D16	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Do not publish
Counter_LineA	INT		%D32	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Do not publish
Counter_LineB	INT		%D35	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Do not publish
Counter_LineC	INT		%D38	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Do not publish
Counter_Reject	INT		%D41	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Do not publish
F_Motor_Off	BOOL		%W5.04	<input type="checkbox"/>	<input type="checkbox"/>	Do not publish
F_Full_Line_A	BOOL		%W5.00	<input type="checkbox"/>	<input type="checkbox"/>	Do not publish

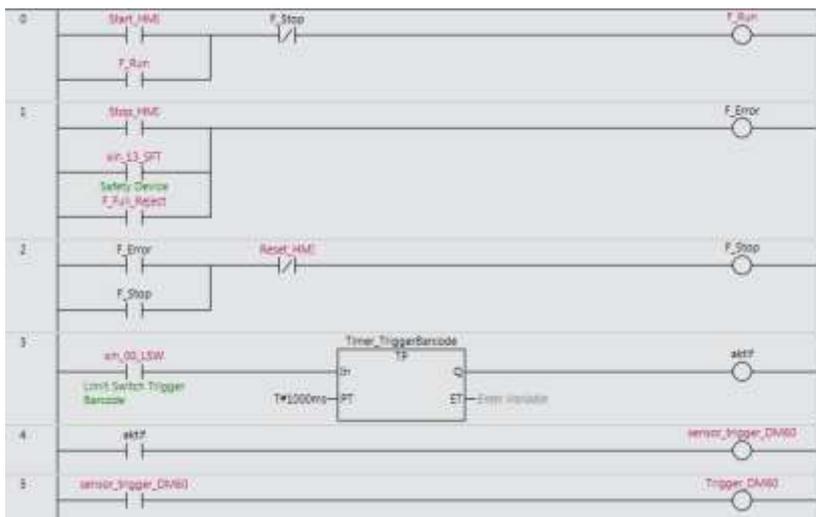
DataC	STRING[15]			<input type="checkbox"/>	<input type="checkbox"/>	Do not publish
DataB	STRING[15]			<input type="checkbox"/>	<input type="checkbox"/>	Do not publish
DataA	STRING[15]			<input type="checkbox"/>	<input type="checkbox"/>	Do not publish
Counter_Reject	INT		%D41	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Do not publish
Counter_LineC	INT		%D38	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Do not publish
Counter_LineB	INT		%D35	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Do not publish
Counter_LineA	INT		%D32	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Do not publish
COGNEX_OUTPUT	COGNEX_OUT			<input type="checkbox"/>	<input type="checkbox"/>	Output
COGNEX_INPUT	COGNEX_IN			<input type="checkbox"/>	<input type="checkbox"/>	Input
BarcodeNotRespo...	BOOL			<input type="checkbox"/>	<input type="checkbox"/>	Do not publish
Barcode_C_3	BOOL		%W7.08	<input type="checkbox"/>	<input type="checkbox"/>	Do not publish
Barcode_C_2	BOOL		%W7.07	<input type="checkbox"/>	<input type="checkbox"/>	Do not publish
Barcode_C_1	BOOL		%W7.06	<input type="checkbox"/>	<input type="checkbox"/>	Do not publish
Barcode_B_3	BOOL		%W7.05	<input type="checkbox"/>	<input type="checkbox"/>	Do not publish
Barcode_B_2	BOOL		%W7.04	<input type="checkbox"/>	<input type="checkbox"/>	Do not publish
Barcode_B_1	BOOL		%W7.03	<input type="checkbox"/>	<input type="checkbox"/>	Do not publish
Barcode_A_3	BOOL		%W7.02	<input type="checkbox"/>	<input type="checkbox"/>	Do not publish
Barcode_A_2	BOOL		%W7.01	<input type="checkbox"/>	<input type="checkbox"/>	Do not publish
Barcode_A_1	BOOL		%W7.00	<input type="checkbox"/>	<input type="checkbox"/>	Do not publish

Program *Sorting Line System*

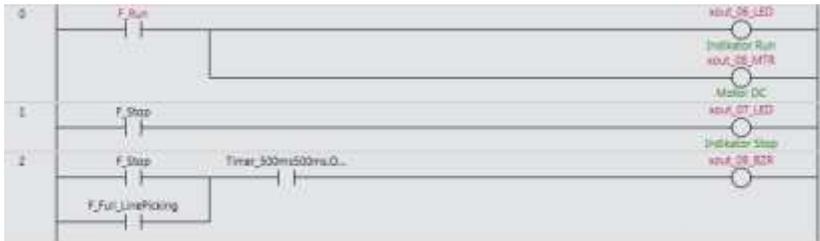




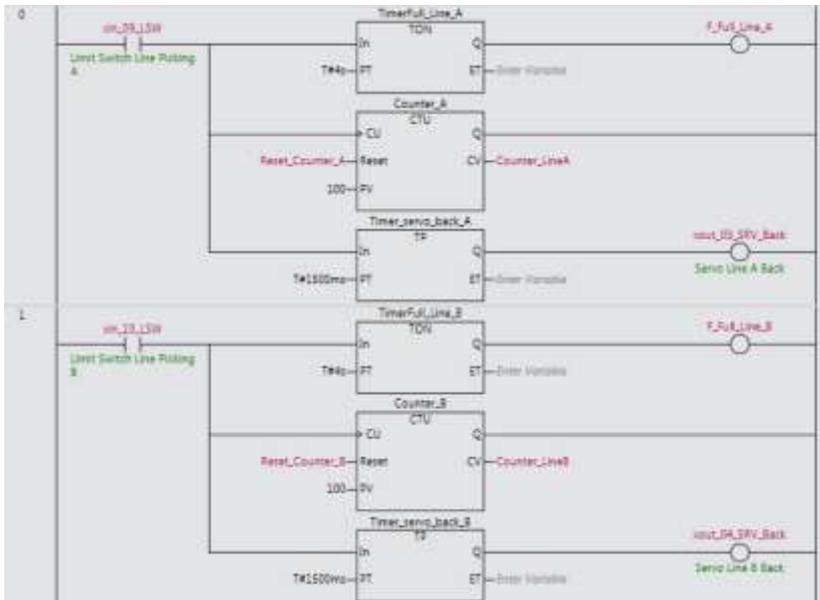
Program input

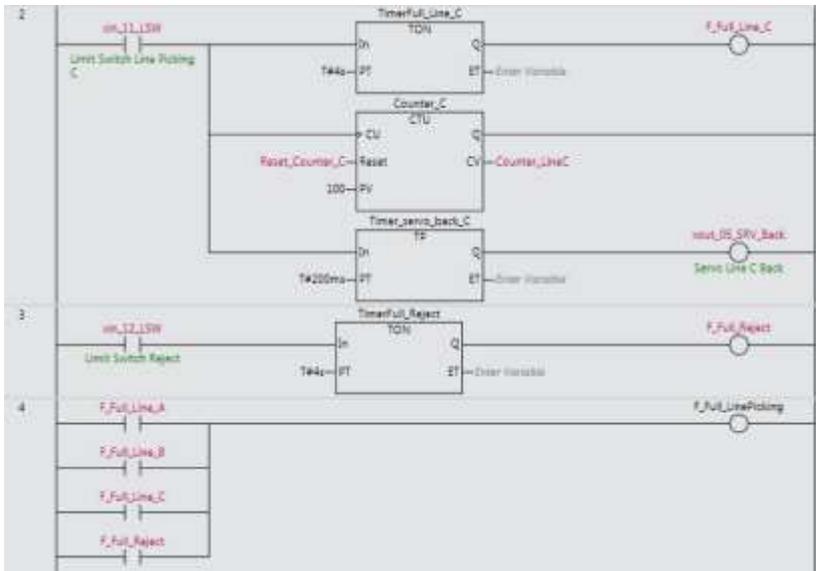


Program Status *Input*

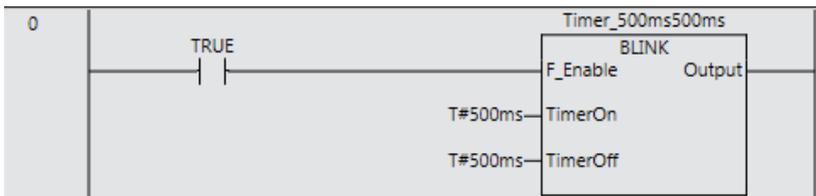


Program Line Picking

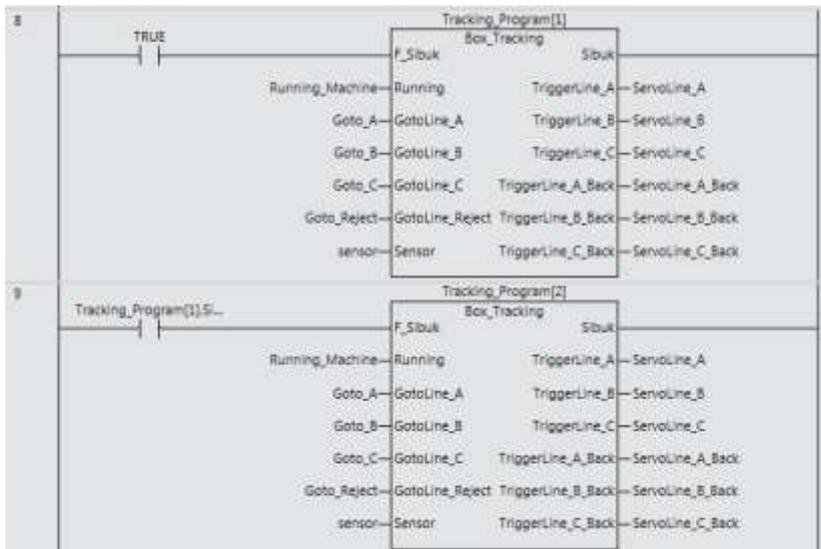


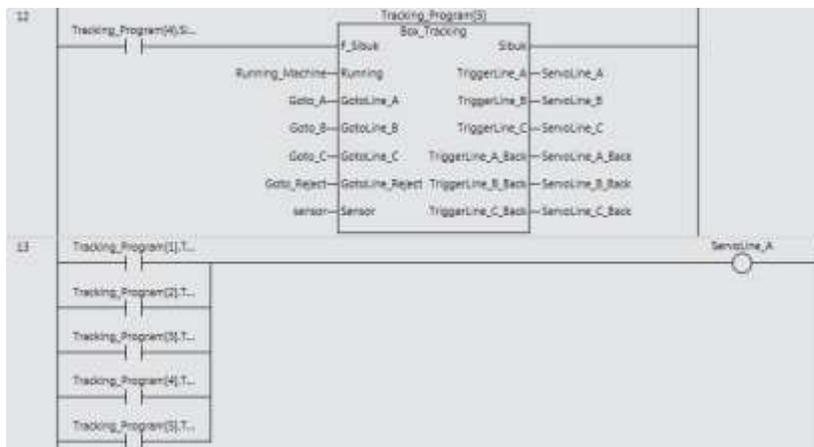
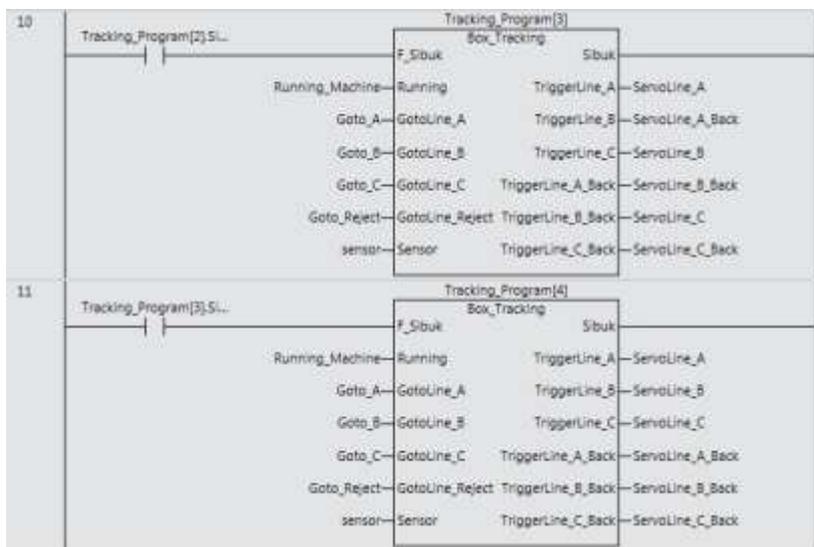


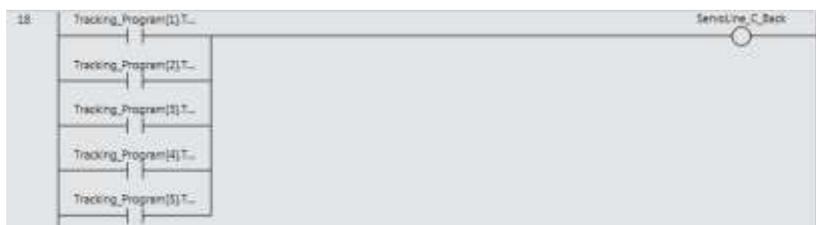
Program *Timer Blink*



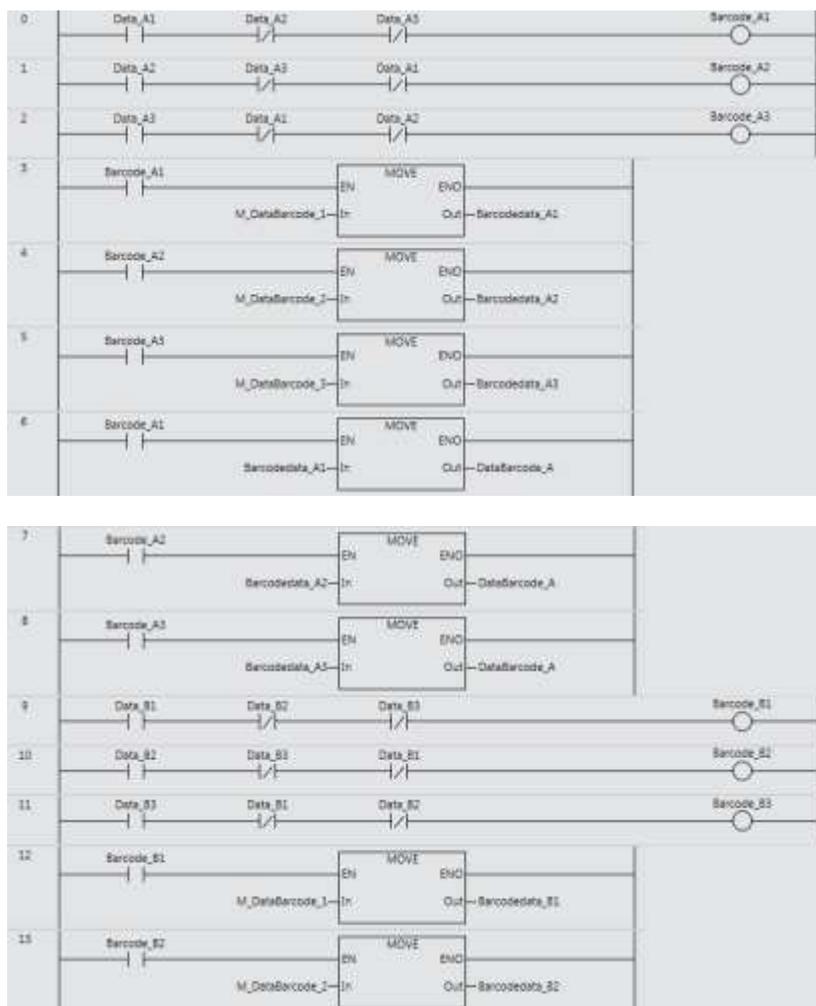
Program Tracking Management

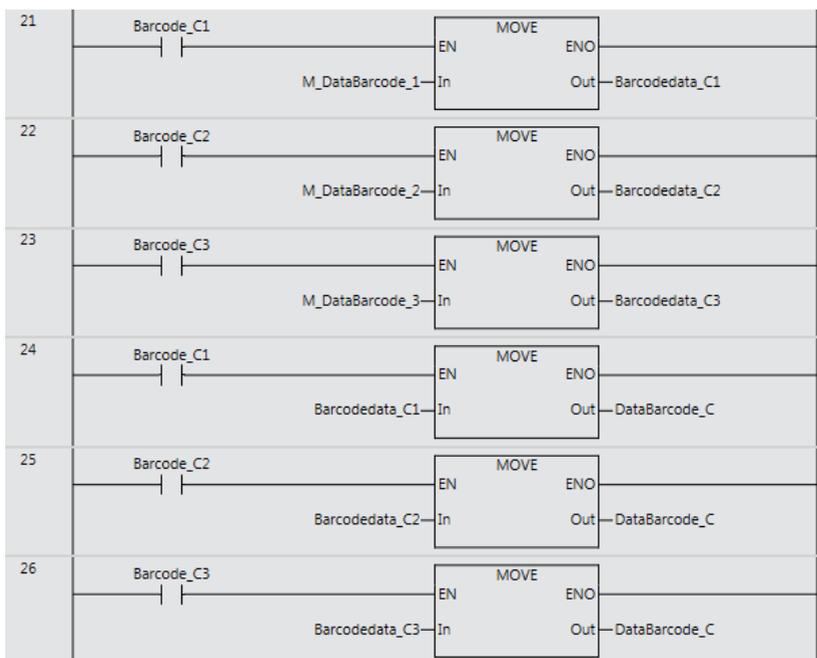
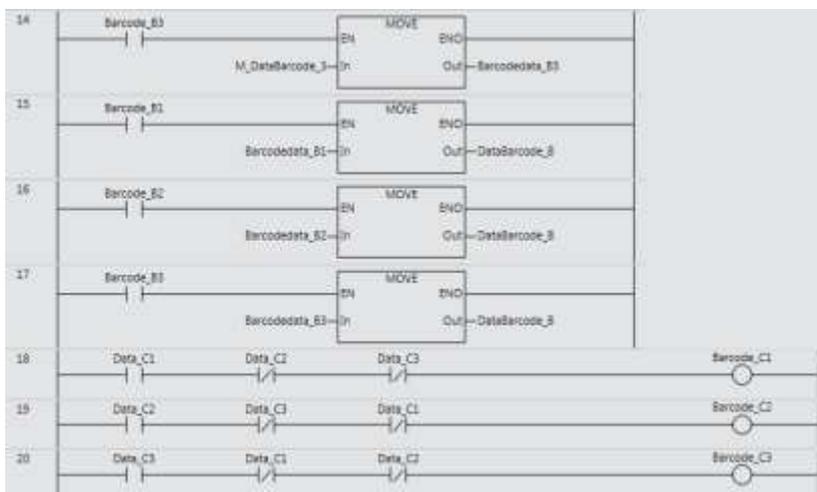






Program Barcode





Program Manual Servo



Program *Box Sorting*

```

IF Mesin_Running THEN
  CASE Index OF
    0:
      IF sensor_trigger_DM60 THEN
        Index:=10;
      END_IF;
    10:
      IF sensor_trigger_DM60 THEN
        IF Done_Read_Barcode THEN
          Index:=30;
        END_IF;
      END_IF;
  
```

```

    END_IF;
20:
    IF NOT Done_Read_Barcode THEN
        Index:=20;
    END_IF;
30:
    IF Good_Read THEN
        Data:=Data_Barcode;
        Index:=40;
    ELSIF Bad_Read THEN
        Index:=100;
    END_IF;
40:
    IF Data=Barcode_A THEN
        Index_Data:=1;
        Index:=41;
    ELSIF Data=Barcode_B THEN
        Index_Data:=2;
        Index:=41;
    ELSIF Data = Barcode_C THEN
        Index_Data:=3;
        Index:=41;
    ELSE
        Index:=100;
    END_IF;
41:
    CASE Index_Data OF
        1:
            Line_A:=TRUE;
            Index:=200;
        2:
            Line_B:=TRUE;
            DataTimer(IN:=TRUE,PT:=T#500ms);
            IF DataTimer.Q THEN
                DataTimer(IN:=FALSE);
                Index:=200;
            END_IF;
        3:
            Line_C:=TRUE;

```

```

        DataTimer(IN:=TRUE,PT:=T#500ms);
        IF DataTimer.Q THEN
            DataTimer(IN:=FALSE);
            Index:=200;
        END_IF;
    END_CASE;
100:
    Reject:=TRUE;
    Index:=200;
200:
    Line_A:=FALSE;
    Line_B:=FALSE;
    Line_C:=FALSE;
    Reject:=FALSE;
    Index:=0;
END_CASE;
ELSE
    Index:=0;
    DataTimer(IN:=FALSE);
    Line_A:=FALSE;
    Line_B:=FALSE;
    Line_C:=FALSE;
    Reject:=FALSE;
END_IF;

```

Program Box Tracking

```

IF Running THEN
    CASE Index OF
        0:
            IF F_Sibuk THEN
                IF GotoLine_A THEN
                    Index:=100;
                ELSIF GotoLine_B THEN
                    Index:=200;
                ELSIF GotoLine_C THEN
                    Index:=300;
                ELSIF GotoLine_Reject THEN
                    Index:=400;
                END_IF;
            END_IF;
        END_CASE;
    END_IF;

```

```

    END_IF;
100:
    IF NOT GotoLine_A THEN
        DataTimer(IN:=TRUE,PT:=T#250ms);
        IF DataTimer.Q THEN
            Sibuk:=TRUE;
            DataTimer(IN:=FALSE);
            Index:=101;
        END_IF;
    END_IF;
101:
    IF Sensor[1] THEN
        Index:=102;
    END_IF;
102:
    IF Sensor[2] THEN
        Index:=103;
    END_IF;
103:
    IF Sensor[3] THEN
        Index:=104;
    END_IF;
104:
    IF Sensor[4] THEN
        Index:=105;
    END_IF;
105:
    IF Sensor[5] THEN
        Index:=106;
    END_IF;
106:
    IF Sensor[6] THEN
        Index:=150;
    END_IF;
150:
    DataTimer(IN:=TRUE,PT:=T#700ms);
    IF DataTimer.Q THEN
        TriggerLine_A:=TRUE;
        DataTimer(IN:=FALSE);
    
```

```

        Index:=151;
    END_IF;
151:
    DataTimer(IN:=TRUE,PT:=T#200ms);
    IF DataTimer.Q THEN
        TriggerLine_A:=FALSE;
        DataTimer(IN:=FALSE);
        Index:=160;
    END_IF;
160:
    Sibuk:=FALSE;
    Index:=0;
200:
    IF NOT GotoLine_B THEN
        DataTimer(IN:=TRUE,PT:=T#200ms);
        IF DataTimer.Q THEN
            Sibuk:=TRUE;
            DataTimer(IN:=FALSE);
            Index:=201;
        END_IF;
    END_IF;
201:
    IF Sensor[1] THEN
        Index:=202;
    END_IF;
202:
    IF Sensor[2] THEN
        Index:=203;
    END_IF;
203:
    IF Sensor[3] THEN
        Index:=204;
    END_IF;
204:
    IF Sensor[4] THEN
        Index:=205;
    END_IF;
205:
    IF Sensor[5] THEN

```

```

        Index:=206;
    END_IF;
206:
    IF Sensor[6] THEN
        Index:=207;
    END_IF;
207:
    IF Sensor[7] THEN
        Index:=250;
    END_IF;
250:
    DataTimer(IN:=TRUE,PT:=T#550ms);
    IF DataTimer.Q THEN
        TriggerLine_B:=TRUE;
        DataTimer(IN:=FALSE);
        Index:=251;
    END_IF;
251:
    DataTimer(IN:=TRUE,PT:=T#300ms);
    IF DataTimer.Q THEN
        TriggerLine_B:=FALSE;
        DataTimer(IN:=FALSE);
        Index:=260;
    END_IF;
260:
    Sibuk:=FALSE;
    Index:=0;
300:
    IF NOT GotoLine_C THEN
        DataTimer(IN:=TRUE,PT:=T#1000ms);
        IF DataTimer.Q THEN
            Sibuk:=TRUE;
            DataTimer(IN:=FALSE);
            Index:=301;
        END_IF;
    END_IF;
301:
    IF Sensor[1] THEN
        Index:=302;

```

```

    END_IF;
302:
    IF Sensor[2] THEN
        Index:=303;
    END_IF;
303:
    IF Sensor[3] THEN
        Index:=304;
    END_IF;
304:
    IF Sensor[4] THEN
        Index:=305;
    END_IF;
305:
    IF Sensor[5] THEN
        Index:=306;
    END_IF;
306:
    IF Sensor[6] THEN
        Index:=307;
    END_IF;
307:
    IF Sensor[7] THEN
        Index:=308;
    END_IF;
308:
    IF Sensor[8] THEN
        Index:=350;
    END_IF;
350:
    DataTimer(IN:=TRUE,PT:=T#1s);
    IF DataTimer.Q THEN
        TriggerLine_C:=TRUE;
        DataTimer(IN:=FALSE);
        Index:=351;
    END_IF;
351:
    DataTimer(IN:=TRUE,PT:=T#5s);
    IF DataTimer.Q THEN

```

```

        TriggerLine_C:=FALSE;
        DataTimer(IN:=FALSE);
        Index:=360;
    END_IF;
360:
    Sibuk:=FALSE;
    Index:=0;
400:
    IF NOT GotoLine_Reject THEN
        DataTimer(IN:=TRUE,PT:=T#1000ms);
        IF DataTimer.Q THEN
            Sibuk:=TRUE;
            DataTimer(IN:=FALSE);
            Index:=401;
        END_IF;
    END_IF;
401:
    IF Sensor[1] THEN
        Index:=402;
    END_IF;
402:
    IF Sensor[2] THEN
        Index:=403;
    END_IF;
403:
    IF Sensor[3] THEN
        Index:=404;
    END_IF;
404:
    IF Sensor[4] THEN
        Index:=405;
    END_IF;
405:
    IF Sensor[5] THEN
        Index:=406;
    END_IF;
406:
    IF Sensor[6] THEN
        Index:=407;
    
```

```

    END_IF;
407:
    IF Sensor[7] THEN
        Index:=408;
    END_IF;
408:
    IF Sensor[8] THEN
        Index:=460;
    END_IF;
460:
    Sibuk:=FALSE;
    Index:=0;
    END_CASE;
END_IF;

```

Program Blink

```

IF F_Enable THEN
    CASE SeqIndex OF
        0:
            DataTimer(IN:=TRUE,PT:=TimerOn);
            IF DataTimer.Q THEN
                DataTimer(IN:=FALSE);
                Output:=TRUE;
                SeqIndex:=10;
            END_IF;
        10:
            DataTimer(IN:=TRUE,PT:=TimerOff);
            IF DataTimer.Q THEN
                DataTimer(IN:=FALSE);
                Output:=FALSE;
                SeqIndex:=0;
            END_IF;
    END_CASE;
ELSE
    SeqIndex:=0;
    DataTimer(IN:=FALSE);
    Output:=FALSE;
END_IF;

```

RIWAYAT HIDUP PENULIS

Nama Lengkap : Vrenky Meidianto
Nama Panggilan : Vrenky
Jenis Kelamin : Laki-laki
Tempat, Tanggal Lahir : Surabaya, 15 Mei 1996
Alamat :Ds. Jati 002/002 Kec.
Tarokan Kab. Kediri
No. Telepon : +62857080007508
E-mail : vrenky15@gmail.com
Agama : Islam



Latar Belakang Pendidikan

Tingkat	Nama Institusi	Jurusan	Tahun
SD	SDN Jati	-	2002-2008
SMP	SMPN 1 Grogol	-	2008-2011
SMA	SMAN 1Grogol	IPA	2011-2014
Perguruan Tinggi	Institut Teknologi Sepuluh Nopember, Surabaya	Departemen Teknik Elektro Otomasi	2015-Sekarang

Pengalaman Kepanitiaian

Organisasi	Nama Kegiatan	Posisi	Tahun
HIMA D3TEKRO FTI-ITS	Industrial Automation And Robotic Competition (IARC)	SC Line Tracer Mikro	2016
Laboratorium Sistem Komputer dan Otomasi	Siemens Scitech Competition (SSC)	OC Acara	2016