

Pembuatan Program PLC dan Simulasi pada PLC Simulator

Jotje Rantung
Jurusan Teknik Mesin, Fakultas Teknik
Universitas Sam Ratulangi
e-mail: jrantung@yahoo.com

Abstrak

Obyek yang ditinjau dalam penulisan ini adalah proses pengisian dan penutupan Aqua 600 ml. Tujuan adalah mendapatkan model perangkat lunak PLC untuk proses pengisian dan penutupan Aqua 600 ml. Sistem terdiri dari perangkat *Programmable Logic Controller (PLC)*. Pada pelaksanaan ini penulis menggunakan *software* buatan OMRON yaitu *Zen Support Software v.3.0* yang memanfaatkan diagram *ladder* sebagai bahasa pemrograman, kemudian dirubah ke dalam kode *Mnemonic* agar dapat disimulasikan pada *software PLC simulator v.1.0* yang umum dipakai. Simulasi yang dilakukan disesuaikan dengan kondisi proses, yaitu sebagai *input* ada 2 *sensor* (*sensor* infra merah dan *sensor* tekanan), dan yang sebagai *output* (*conveyor* 1, *conveyor* 2, mesin pengisian dan mesin penutupan). Setiap *input* menghasilkan *output* yang berupa menyala lampu indikator pada program *simulator* yang sedang *running*.

Kata Kunci: PLC, Diagram Ladder, PLC Simulator

1. Pendahuluan

Teknik kontrol telah digunakan untuk waktu yang lama. Pada masa lampau tubuh manusia merupakan metode utama untuk mengontrol sistem. Dewasa ini tenaga listrik telah digunakan untuk kontrol dan saat ini kontrol elektrik digunakan untuk relay. Relay ini memungkinkan suatu power untuk dihidupkan (switch on) atau dimatikan (switch off) tanpa suatu switch mekanik. Hal-hal semacam ini secara umum menggunakan relay untuk memudahkan tujuan kontrol logika. Perkembangan komputer dengan biaya rendah telah membawa dampak pada revolusi paling berarti saat ini yaitu "Programable Logic Controller" (PLC). Kemajuan PLC dimulai pada tahun 1970-an dan telah menjadi pilihan paling utama untuk kontrol manufaktur.

PLC banyak digunakan pada aplikasi-aplikasi industri seperti proses pengepakan, penanganan bahan, perakitan otomatis, pengendalian temperatur pada proses pengolahan minyak, pengisian produk minuman pada botol, dan lain sebagainya.

Atas dasar kelebihan-kelebihan tersebut, maka penulis tertarik untuk menerapkan sistem pengisian produk minuman berbasis PLC Omron dan membuat model programnya serta mensimulasikan program dilakukan secara terintegrasi menggunakan program simulasi buatan Tang Tung Yan, yaitu PLC simulator v.1.0.

Untuk studi kasus proses pengisian aqua 600 ml menggunakan PLC dilakukan pada PT. Tirta Investama Manado.

2. Struktur Dasar PLC

Secara garis besar struktur dasar PLC dapat dibagi menjadi empat kelompok komponen utama yang terdiri dari:

1. *Antarmuka (interface) Input*
2. *Antarmuka (interface) Output*
3. *Unit Pemrosesan (CPU)*
4. *Unit Memori*

Dalam CPU sebuah PLC dapat diibaratkan sebagai kumpulan ribuan relay walaupun kenyataannya bukan berarti terdapat ribuan relay berskala kecil, tetapi dalam PLC berisi rangkaian elektronika digital yang berfungsi sebagai contact Normally Open (NO) dan contact Normally Close (NC) relay.

2.1 Dasar Pemrograman PLC

Secara teknis program pada memori PLC yang digunakan untuk mengendalikan peralatan ini dibuat dan dimasukkan dengan menggunakan perangkat pemrograman, yaitu miniprogramer atau consol, bahkan juga pemanfaatan komputer dengan perangkat lunak yang menyertainya.

Untuk mengoperasikan PLC, terlebih dahulu harus mendefinisikan program yang ditulis dalam bahasa yang sesuai dengan spesifikasi yang digunakan. Pembuatan program PLC dapat digunakan dengan tiga cara, yaitu:

1. Diagram *ladder*
2. Listing program (*Mnemonic*)
3. Diagram *blok*

Diagram *ladder* adalah sebuah diagram yang berbentuk jaringan saklar-saklar yang dihubungkan secara seri atau paralel dan hasilnya disimpan di dalam sebuah alat penyimpan dengan simbol tertentu. Terdapat beberapa simbol dasar diagram *ladder* yang digunakan untuk mengekspresikan tujuan dan arti dari sebuah *sirkuit* pengendali.

Seluruh *input* dan *output* diidentifikasi melalui alamat-alamatnya, dimana notasi yang dipergunakan bergantung pada pabrikan PLC yang bersangkutan. Dalam penulisan skripsi ini karena yang dipakai sebagai *software editor* adalah *Zen Support Software v.3.0* dan program *simulator*-nya adalah *PLC Simulator* yang mendukung perangkat PLC buatan *OMRON*, maka notasi pengalamatan dan kode instruksinya menggunakan standar yang dikeluarkan oleh *OMRON* sebagai pabrikan pembuat perangkat PLC tersebut, meskipun sebuah standar *IEC 1131-3* telah diajukan. Alamat-alamat ini mengidentifikasi lokasi *input* atau *output* di dalam memori PLC.

Kode *Mnemonic* yang berupa listing program merupakan kode pemrograman lainnya yang merupakan instruksi masukan dan keluaran yang dirangkai oleh sejumlah kondisi *input-output*. Diagram tangga atau *ladder* tidak dapat langsung dikirimkan ke PLC menggunakan consol pemrograman (*programming consol*). Untuk mengirimkan diagram tangga menggunakan *consol* pemrograman maka harus dilakukan *konversi* diagram tangga ke kode-kode *Mnemonic*. *Software* bawaan seperti *Syswin v.3.4* yang dipakai pada perangkat *OMRON PLC Sysmac* dapat melakukan hal ini dengan *otomatis*. Kode *Mnemonic* menyediakan informasi yang sama persis dengan diagram tangga, hanya saja dalam bentuk yang langsung bisa diketikkan ke PLC yang bersangkutan melalui *consol* pemrograman.

Instruksi masukan dan keluaran yang dirangkai oleh sejumlah kondisi *input-output* adalah rangkaian fungsi logika seperti *AND*, *OR*, *NOT* dan kombinasinya. Pada pemrograman, penggunaan kode *Mnemonic* sebagai *input* awal dikodekan sebagai '*LD*', selanjutnya kode instruksi untuk *Mnemonic* dan pengalamatan *input-output* dapat dilihat pada tabel-tabel di bawah ini.

Seluruh *input* dan *output* diidentifikasi melalui alamat-alamatnya, dimana notasi yang dipergunakan bergantung pada pabrikan PLC yang bersangkutan. Dalam penulisan skripsi ini karena yang dipakai sebagai *software editor* adalah *Zen Support Software v.3.0* dan program *simulator*-nya adalah *PLC Simulator* yang mendukung perangkat PLC buatan *OMRON*, maka notasi pengalamatan dan kode instruksinya menggunakan standar yang dikeluarkan oleh *OMRON* sebagai pabrikan pembuat perangkat PLC tersebut, meskipun sebuah standar *IEC 1131-3* telah diajukan. Alamat-alamat ini mengidentifikasi lokasi *input* atau *output* di dalam memori PLC.

Kode *Mnemonic* yang berupa listing program merupakan kode pemrograman lainnya yang merupakan instruksi masukan dan keluaran yang dirangkai oleh sejumlah kondisi *input-output*. Diagram tangga atau *ladder* tidak dapat langsung dikirimkan ke PLC menggunakan *consol* pemrograman (*programming consol*). Untuk mengirimkan diagram tangga menggunakan *consol* pemrograman maka harus dilakukan *konversi* diagram tangga ke kode-kode *Mnemonic*. *Software* bawaan seperti *Syswin v.3.4* yang dipakai pada perangkat *OMRON PLC Sysmac* dapat melakukan hal ini dengan *otomatis*. Kode *Mnemonic* menyediakan informasi yang sama persis dengan diagram tangga, hanya saja dalam bentuk yang langsung bisa diketikkan ke PLC yang bersangkutan melalui *consol* pemrograman.

Instruksi masukan dan keluaran yang dirangkai oleh sejumlah kondisi *input-output* adalah rangkaian fungsi logika seperti *AND*, *OR*, *NOT* dan kombinasinya. Pada pemrograman, penggunaan kode *Mnemonic* sebagai *input* awal dikodekan sebagai '*LD*', selanjutnya kode instruksi untuk *Mnemonic* dan pengalamatan *input-output* dapat dilihat pada tabel di bawah ini.

Tabel 1. Kode *Mnemonic* pada *OMRON-ZEN*

<i>OMRON - ZEN</i>	KETERANGAN
<i>LD</i>	Mulai sebuah anak tangga dengan sebuah anak kontak terbuka
<i>LD NOT</i>	Mulai sebuah anak tangga dengan sebuah anak kontak tertutup
<i>AND</i>	Sebuah elemen seri dengan sebuah kontak terbuka
<i>AND NOT</i>	Sebuah elemen seri dengan sebuah kontak tertutup
<i>OR</i>	Sebuah elemen paralel dengan sebuah kontak terbuka
<i>OR NOT</i>	Sebuah elemen paralel dengan sebuah kontak tertutup
<i>OUT</i>	Sebuah <i>output</i>

2.2 Komputer sebagai *Programming Console*

Dibandingkan dengan *miniprogrammer* atau *console*, dewasa ini komputer lebih banyak digunakan karena pemanfaatan *console* biasanya terbatas hanya untuk *editing* program *PLC* saja. Berkaitan dengan arsitekturnya yang bersifat *general purpose* dan sistem operasinya yang standar, umumnya *vendor-vendor PLC* menyertakan perangkat lunak komputer untuk mengimplementasikan pemasukan program *ladder*, pengeditan, dokumentasi dan program *monitoring rele time PLC*.

Jika pemrograman *PLC* dengan menggunakan *console* biasanya dengan mengetikkan baris-baris simbol program pada level rendah dengan menggunakan instruksi-instruksi *Mnemonic* seperti *LD*, *NOT*, *AND* dan lain sebagainya, maka pada komputer program *PLC* dapat dibuat langsung dengan menggunakan teknik standar pemrograman sekuensial, yaitu menggunakan diagram *ladder*. Diagram *ladder* ini langsung digambar dengan fasilitas *GUI* (*Graphical User Interface*) pada perangkat lunak yang ada.

Program yang telah dibuat selanjutnya ditransfer ke *PLC* melalui modul komunikasi yang tersedia (umumnya port serial: *COM*). Perangkat lunak komputer untuk pemrograman *PLC* ini biasanya juga dilengkapi dengan fasilitas *monitoring* dan komunikasi. Gambar memperlihatkan contoh tampilan *GUI* dari perangkat lunak *Zen Support Software v.3.0*.

2.3 *Programming Console PLC*

Penggunaan komputer sebagai *programming console* lebih mudah dan praktis untuk menyusun dan mengisi program *PLC*. Fungsi komputer atau *PC* (*Personal Computer*) secara umum sebagai salah satu perangkat kerja adalah untuk melakukan *download* program *ladder* serta untuk memonitor operasi kerja *PLC*. Atau dengan kata lain *CPU* pada komputer mengatur semua proses yang terjadi di *PLC*, dimana tiga komponen utama menyusun *CPU* yaitu:

1. *Prosesor*
2. *Memori*
3. *Power Supply*

Adapun tugas-tugas utama yang dapat dilakukan dengan memanfaatkan komputer antara lain:

1. Membuat program baru
2. Menyimpan program dan editing
3. Upload dan download program antara *PLC* dan komputer
4. Menulis komentar pada program untuk memudahkan pembacaan program
5. Fasilitas *library file* untuk pengembangan
6. Print out program dikertas untuk dokumentasi

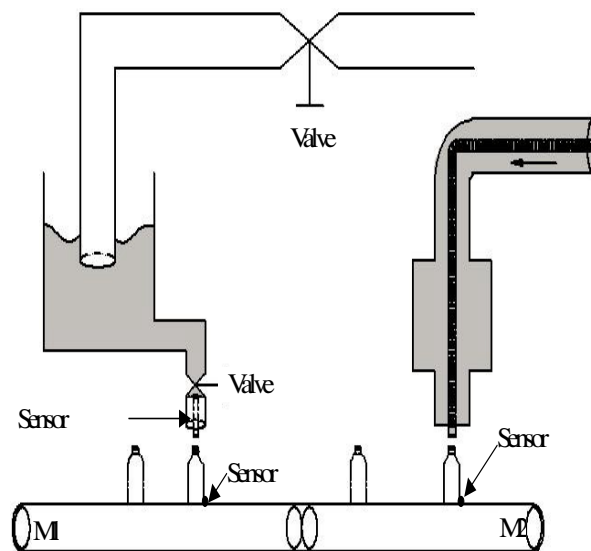
3. Siklus Pengisian

Tahapan pengisian botol dibagi menjadi lima bagian yaitu:

1. Botol kosong masuk
2. Pencucian
3. Pengisian
4. Penutupan
5. Botol penuh keluar

4. PLC Sebagai Pengendali Proses Pengisian dan Penutupan Botol

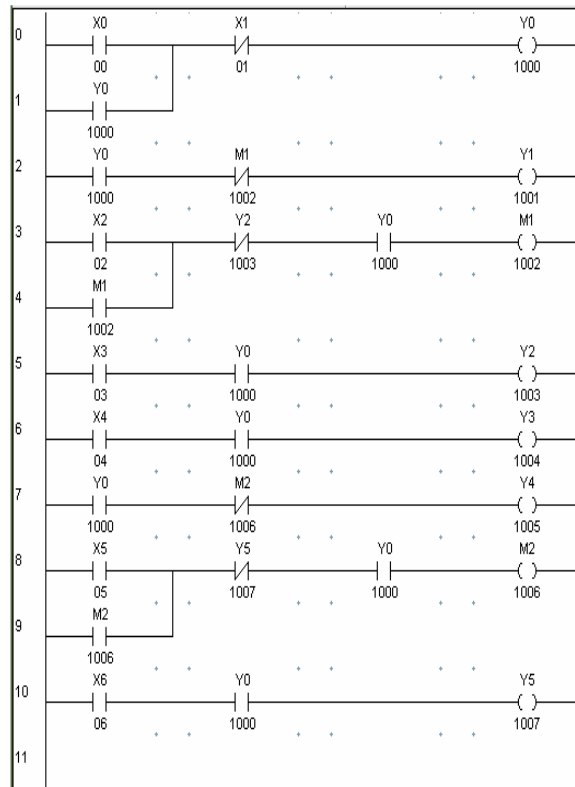
Untuk memudahkan dalam proses pengisian dan penutupan maka perlu adanya suatu sistem pengendalian yang dalam hal ini digunakan perangkat *PLC*. Adapun perangkat yang terpasang di *unit Filler PT. Tirta Investama Manado* adalah *PLC SIEMENS*. Karna keterbatasan akses terhadap *software* yang digunakan pada perangkat *PLC* yang terpasang maka dalam penulisan skripsi ini proses pengisian dan penutupan botol hanya disimulasikan dengan menggunakan *PLC Simulator v.1.0* karya *Tang Tung Yan*, dan pembuatan *Listing Program* berupa *Ladder Diagram* yang memanfaatkan *Zen Support Software v.3.0* sebagai program editornya.



Gambar 1. Proses Pengisian dan Penutupan Botol

4.1 Diagram Ladder Proses Pengisian Dan Penutupan Botol

Dari uraian *flowchart* proses pengisian dan penutupan maka dibuat diagram *ladder*-nya dengan menggunakan program *Zen Support Software v.3.0* yaitu:

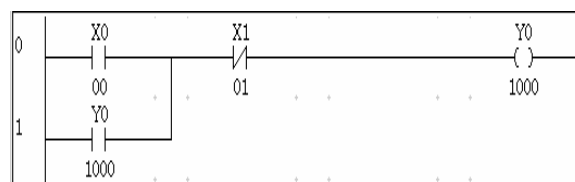


Gambar 2. Diagram *Ladder* Proses Pengisian dan Penutupan Botol

4.2 Penjelasan Diagram *Ladder*

WORK 1

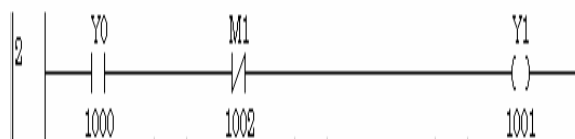
Saat *push button* di tekan (ON), maka sistem akan bekerja secara otomatis. *Conveyor* akan bekerja untuk membawa botol yang berada diatas *conveyor*.



Gambar 3. Sistem Dalam Keadaan ON dan Konveyor Bekerja

WORK 2

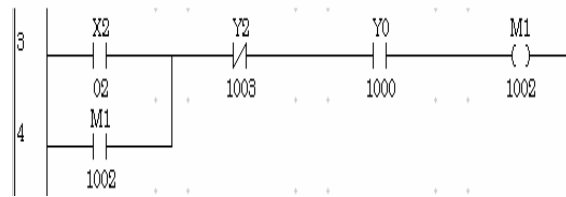
Botol yang berada diatas conveyor yang sedang berjalan akan terdeteksi oleh sensor infra merah, hal ini menyebabkan conveyor berhenti bekerja.



Gambar 4. *Sensor* ON, Motor OFF

WORK 3

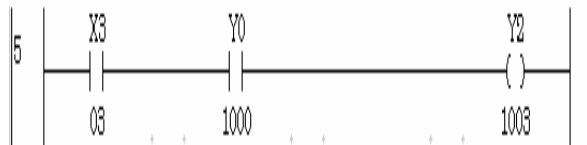
Pada saat *sensor* infra merah mendeteksi botol kosong, maka motor bekerja (M1) dan terjadi proses pengisian air kedalam botol.



Gambar 5. *Sensor* ON, Motor ON

WORK 4

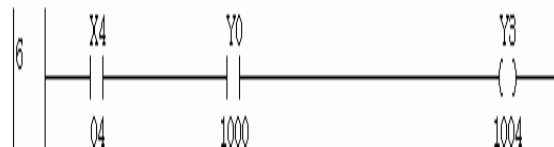
Selama proses pengisian sensor tekanan akan mendeteksi botol yang telah terisi penuh



Gambar 6. Sensor Mendeteksi Botol Penuh

WORK 5

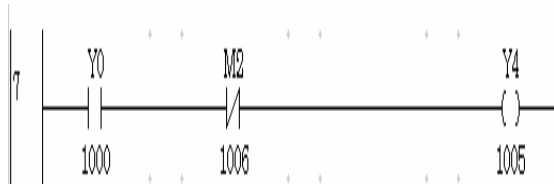
Jika sensor infra merah mendeteksi adanya botol yang tidak penuh maka alarm berbunyi (ON).



Gambar 7. Bila Ada Botol yang Tidak Teisi Penuh, Alarm ON

WORK 6

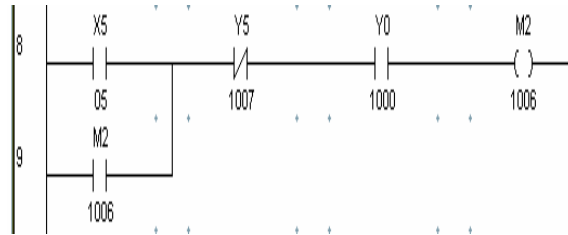
Pada saat *relay* Y0 (ON), maka *conveyor* akan bekerja dan membawa botol yang telah terisi penuh.



Gambar 8. *Relay* Y0 ON, *conveyor* ON

WORK 7

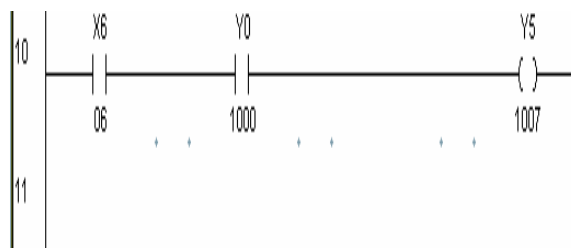
Saat sensor infra merah mendeteksi botol yang telah terisi penuh maka conveyor akan berhenti, M2 (ON) dan melakukan proses penutupan.



Gambar 9. Sensor ON, Conveyor Off, M2 ON

WORK 8

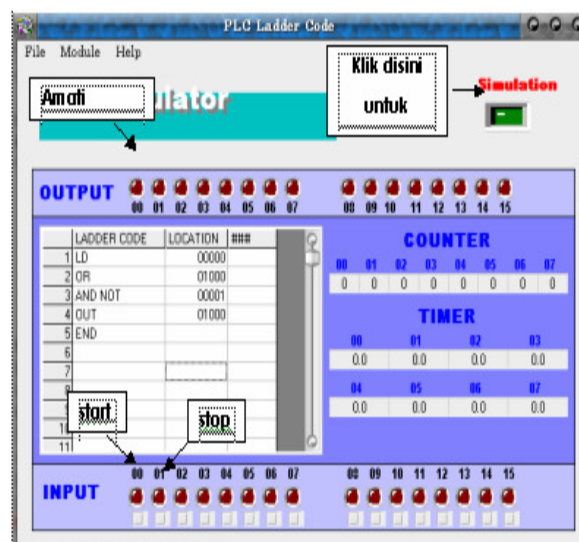
Saat sensor infra merah mendeteksi botol yang telah tertutup maka output Y5 akan kembali bekerja untuk menjalankan conveyor 2.



Gambar 10. Sensor ON, output ON

4.3 Simulasi Program dan Pemeriksaan Software

Untuk mendapatkan hasil yang diharapkan maka semua kondisi input-output harus diuji apakah berjalan sesuai yang diharapkan, bila tidak maka diadakan koreksi program kembali, selanjutnya semua hasil proses pemrograman dapat didokumentasikan secara sistematis untuk keperluan berikutnya.



Gambar 11. Simulasi Berdasarkan PLC Simulator

Untuk simulasi memanfaatkan Zen Support Software v.3.0 sebagai program editornya dan PLC Simulator v.1.0 sebagai simulatornya.

4.4 Kode Mnemonic

Tabel 2. Kode *Mnemonic* Untuk Simulasi Proses Pengisian dan Penutupan Botol

Alamat	Instruksi	Operand	Fungsi
0001	LD	00	STAR
0002	OR	1000	INTERLOCK
0003	AND NOT	01	STOP
0004	OUT	1000	OUTPUT
0005	LD	1000	SWITCH ON
0006	AND NOT	1002	SWITCH OFF
0007	OUT	1001	OUTPUT
0008	LD	02	SWITCH ON
0009	OR	1002	INTERLOCK
0010	AND NOT	1003	SWITCH OFF
0011	AND	1000	SWITCH ON
0012	OUT	1002	OUTPUT
0013	LD	03	SWITCH ON
0014	AND	1000	SWITCH ON
0015	OUT	1003	OUTPUT
0016	LD	04	SWITCH ON
0017	AND	1000	SWITCH ON
0018	OUT	1004	OUTPUT
0019	LD	1000	SWITCH ON
0020	AND NOT	1006	SWITCH OFF
0021	OUT	1005	OUTPUT
0022	LD	05	SWITCH ON
0023	OR	1006	INTERLOCK
0024	AND NOT	1007	SWITCH OFF
0025	AND	1000	SWITCH ON
0026	OUT	1006	OUTPUT
0027	LD	06	SWITCH ON
0028	AND	1000	SWITCH ON
0029	OUT	1007	OUTPUT
0030	END		AKHIR PROGRAM

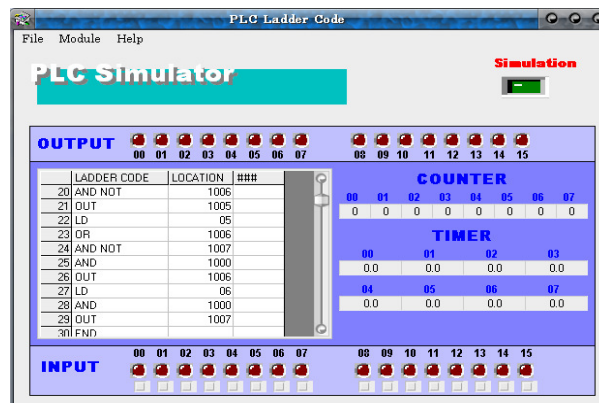
5 Kesimpulan

Model perangkat lunak *PLC* telah diperoleh berdasarkan simulasi program yang berhasil dijalankan dengan menggunakan *software* buatan *OMRON* yaitu *Zen Support Software v.3.0* yang memanfaatkan diagram *ladder* sebagai bahasa pemrograman, kemudian dirubah ke dalam kode *Mnemonic* agar dapat disimulasikan pada *software PLC simulator v.1.0* yang umum dipakai.

6. Daftar Pustaka

- Ackermann, R. 1994. *Programmable Logic Controller* (Buku Test Tingkat Dasar TP 3001-FESTO DIDACTIC). Esslingen.
- Anonimous, 1997. Omron Asia Pacific PTE.LTD.
- Anonimous, 2001. *Sarcmi Stilljet Use And Maintenance Filling and Closing Monobloc For Plastic Bottles* (No Mesin KAO610) Sasib S.P.A, Beverage Hs Division.
- Balza, A. 2007. *Pemrograman PLC Menggunakan Simulator*. Andi, Yogyakarta.
- Bolton, W. 2004. *Programmable Logic Controller (PLC)*. Erlangga.
- Bolton, W. 2006 *Mechatronics, Electronic Control Sistem In Mechanical Engineering*.
- Mawei, M. 2006. Studi Aplikasi *Programmable Logic Controller (PLC)* Sebagai Alat Pengendali Temperatur Pada Unit HE722 Di Proses Refinery PT. Bimoli Bitung. Skripsi Program S1 Teknik Mesin Unsrat, Manado.
- Putra, E. 2004. *PLC, Konsep Pemrograman dan Aplikasi*. Gava Media, Yogyakarta.
- Setiawan, I. 2006. *Programmable Logic Controller dan Teknik Perancangan Sistem Kontrol*. Andi, Yogyakarta.
- Suhendar, 2005. *Programmable Logic Controller (PLC)*. Graha Ilmu, Yogyakarta.
- Wijaya, M. 2003. *Pengenalan Dasar-Dasar PLC Disertai contoh aplikasinya*. Gava Media, Yogyakarta.

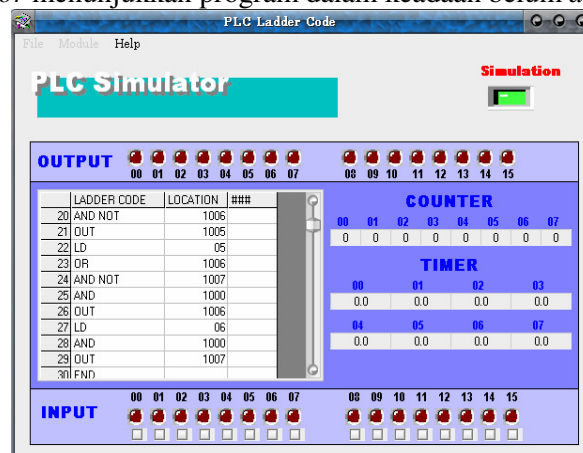
Lampiran Hasil Simulasi



Gambar L1. Tampilan *PLC Simulator* sebelum *Dieksekusi Indikator Simulasi*

Catatan:

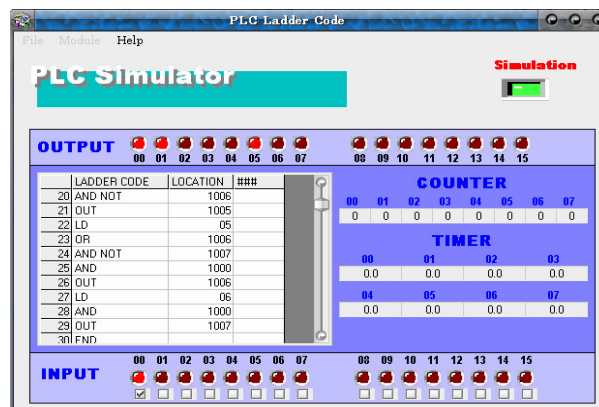
Warna hijau pada *simulator* menunjukkan program dalam keadaan belum *dieksekusi*.



Gambar L2. Tampilan *PLC Simulator* setelah *Dieksekusi Indikator Simulasi On*

Catatan:

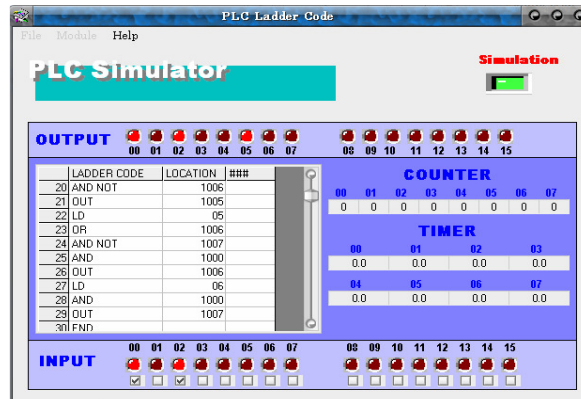
Warna kuning pada *simulator* menunjukkan program dalam keadaan *running* dan tidak ada kesalahan pada program.



Gambar L3. Kondisi pada saat *Conveyor 1 dan 2 On*

Catatan:

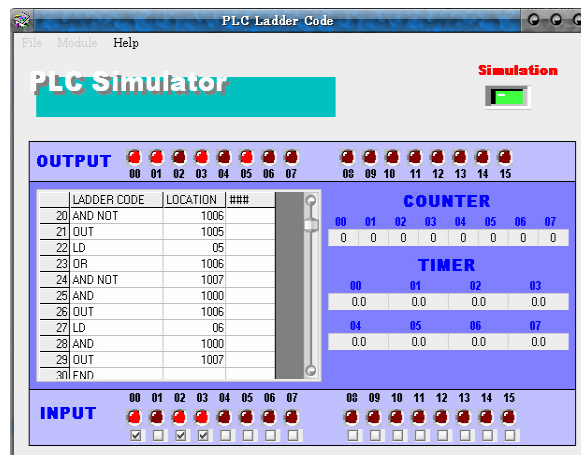
Input 00 (saklar IO) ON maka sistem bekerja secara *otomatis*, *conveyor 1 dan 2 ON* (bekerja).



Gambar L4. Sensor Mendeteksi Botol Kosong

Catatan:

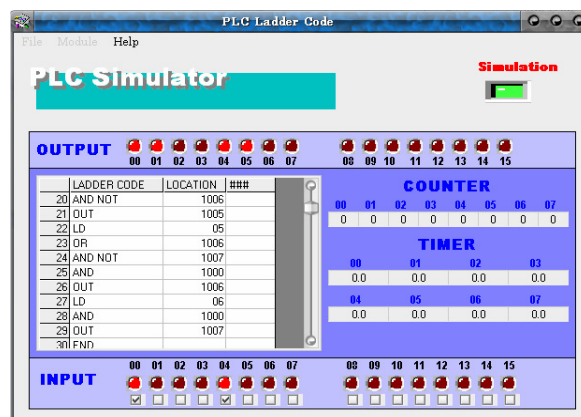
Input 02 (sensor infra merah) ON, mendeteksi botol kosong maka conveyor 1 OFF dan terjadi pengisian, conveyor 2 tetap bekerja.



Gambar L5. Relay On untuk Mematikan Mesin Pengisian

Catatan:

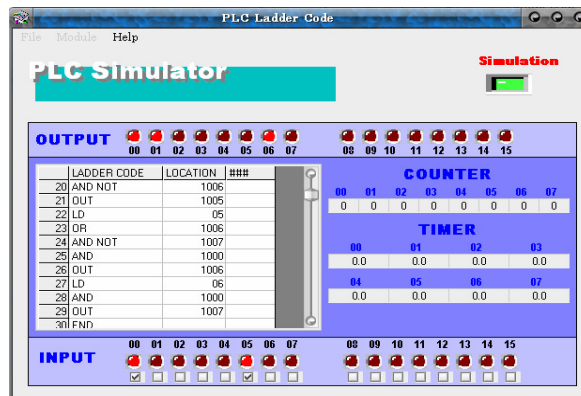
Input 03 (Relay) ON, untuk mematikan mesin pengisian dan menjalankan kembali conveyor 1.



Gambar L6. Alarm On

Catatan:

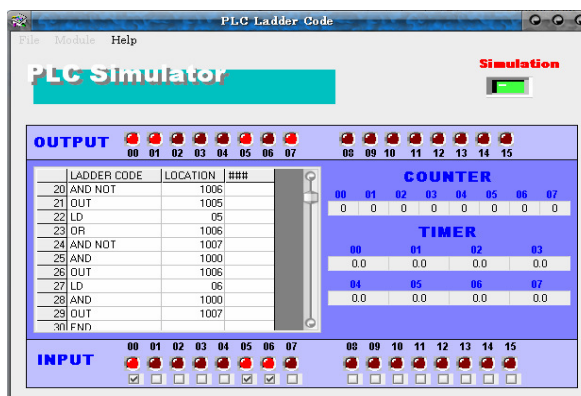
Input 04 (Alarm) ON, jika botol terdeteksi tidak penuh.



Gambar L7. Sensor Mendeteksi Botol Penuh

Catatan:

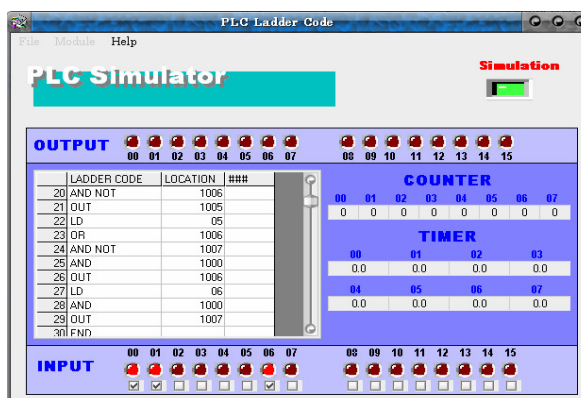
Input 05 (sensor infra merah) ON, mendeteksi botol yang penuh maka conveyor 2 OFF dan terjadi penutupan botol.



Gambar L8. Relay On untuk Mematikan Mesin Penutupan

Catatan:

Input 06 (Relay) ON, untuk membunuh mesin penutupan dan menjalankan kembali conveyor 2.



Gambar L9. Sistem akan Mati

Catatan:

Input 01 ON maka semua sistem akan mati.