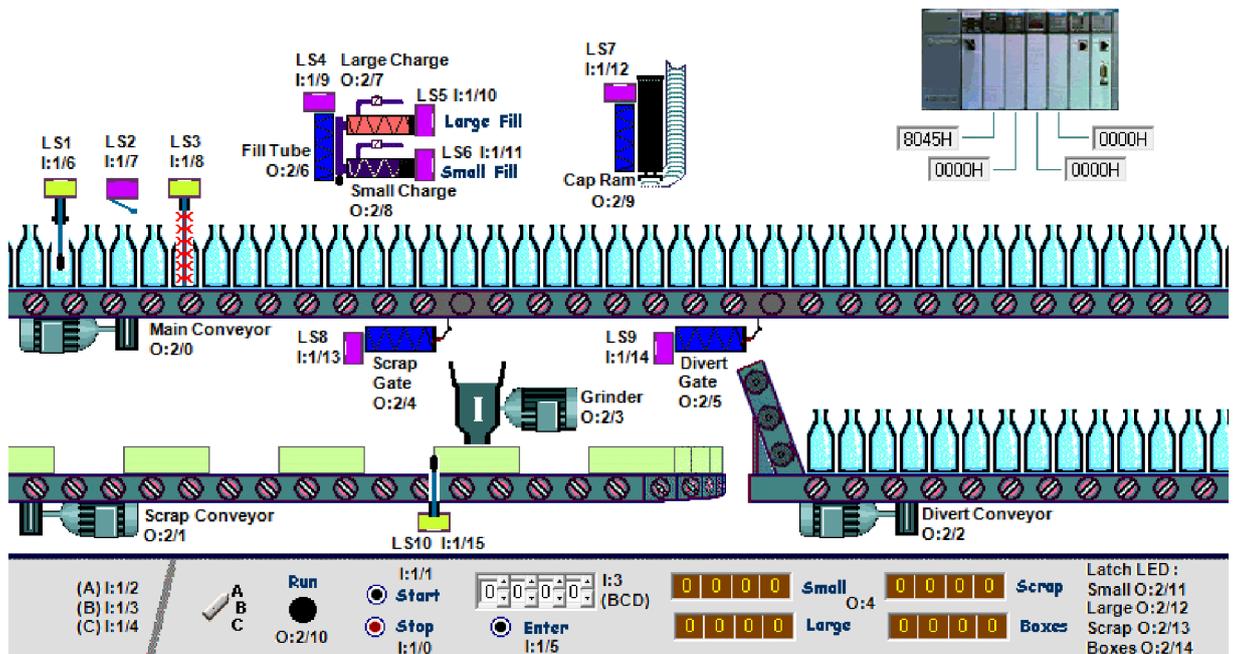


DASAR-DASAR SIMULASI PENGISIAN BOTOL

I.1 Latihan

Dari menu “Simulations” pilihlah “Bottle Line Simulator”.



I.1.1 Latihan 1 : Pengontrolan main conveyor

Buatlah program agar operator dapat menjalankan dan menghentikan *Main Conveyor* (O:2/00) menggunakan saklar START (I:1/01) dan STOP (I:1/00).

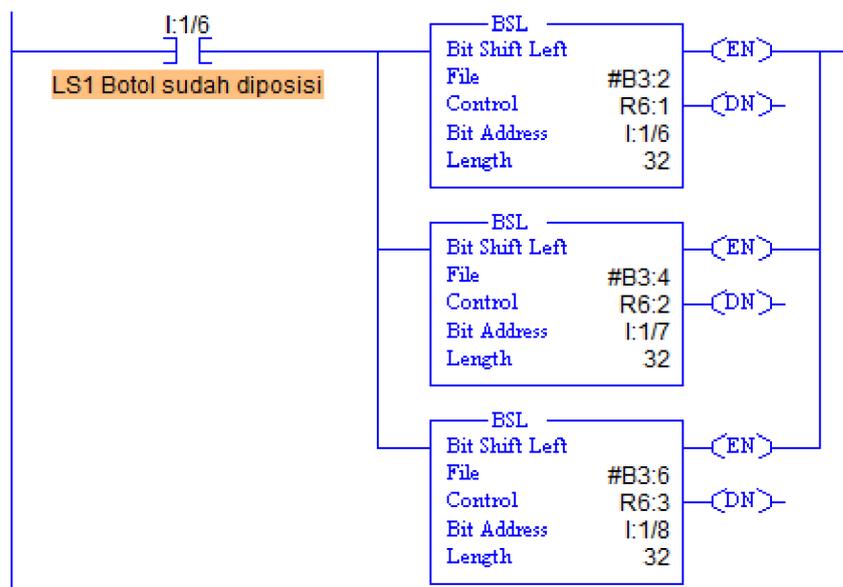
- Main Conveyor akan berjalan saat tombol START telah ditekan dan akan tetap terus berjalan meskipun tombol OPEN dilepas.
- Main Conveyor baru akan berhenti saat tombol STOP ditekan.
- Semenjak tombol START ditekan (sistem/main conveyor bekerja), maka lampu RUN akan menyala. Lampu RUN baru akan mati jika tombol STOP ditekan (sistem/main conveyor berhenti).

I.1.2 Latihan 2 : Mendeteksi botol

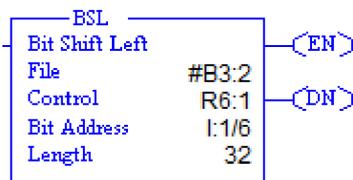
Terdapat tiga jenis sensor yang digunakan pada sistem ini yaitu

- Sensor untuk mendeteksi keberadaan botol (*Exist Sensor* LS1 I:1/16)
- Sensor untuk mendeteksi ukuran botol (*Large Sensor* LS2 I:1/17)
LS2 akan bernilai 1 jika ukuran botol besar dan akan bernilai 0 jika ukuran botol kecil.
- Sensor untuk mendeteksi apakah botol utuh atau rusak (*Broken Sensor* LS3 I:1/18)
LS3 akan bernilai 0 jika botol utuh dan akan bernilai 1 jika botol rusak.

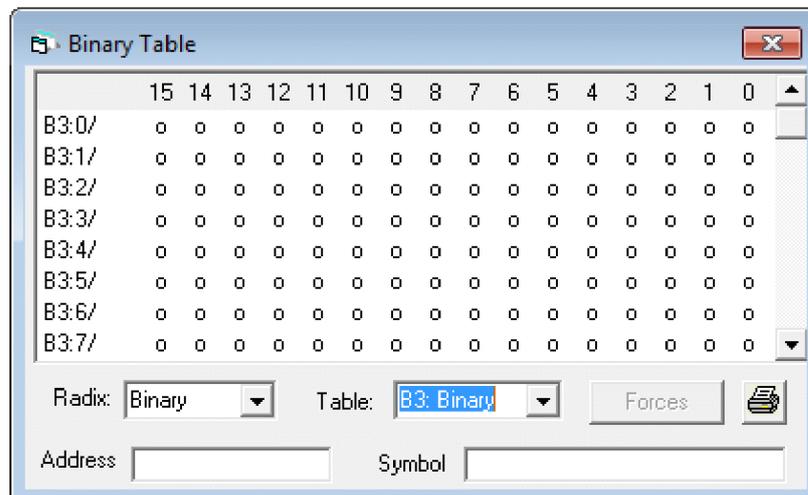
Pada simulasi ini Anda akan berlatih menggunakan instruksi BSL untuk memproses masukan dari ketiga sensor diatas. Pertama-tama, dibawah program yang telah Anda buat pada Latihan 1, tambahkanlah rung berikut ini.



Instruksi BSL sebagai berikut



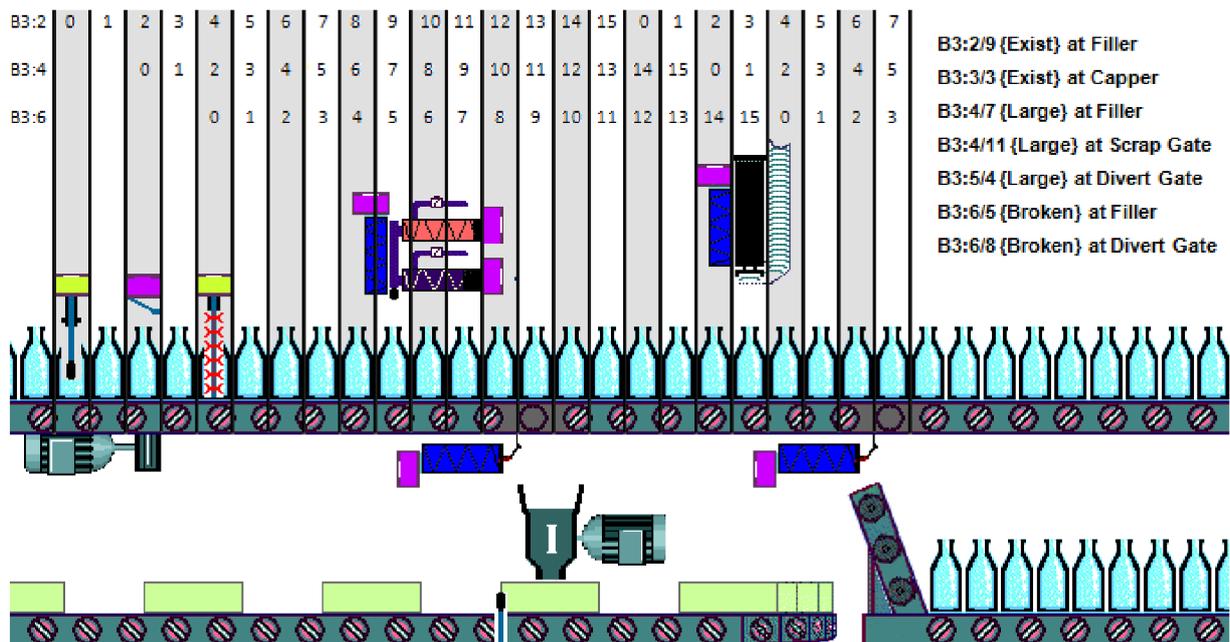
mengartikan Anda akan menggunakan alamat B3:2 sepanjang 32 bit (berarti dari B3:2/00 – B3:2/15 dan dilanjutkan dengan B3:3/00 – B3:3/15) dengan masukan I:1/6 (sensor LS1). Setiap kali nilai masukan BSL bernilai true, maka nilai I:1/6 akan dimasukkan ke bit B3:2/00. Nilai bit B3:2/00 sebelumnya akan dipindahkan ke B3:2/01. Nilai bit B3:2/01 sebelumnya juga akan dipindahkan ke B3:2/02. Dan seterusnya.



Maka berdasarkan program diatas Anda seharusnya memahami hal-hal berikut.

- Bit B3:2 dan B3:3 menerima masukan dari LS1 (sensor keberadaan botol / *exist sensor*). Maka bit B3:2 dan B3:3 akan menunjukkan apakah disuatu posisi sudah ada keberadaan botol atau tidak.
- Bit B3:4 dan B3:5 menerima masukan dari LS2 (sensor ukuran botol / *large sensor*). Maka bit B3:4 dan B3:5 akan menunjukkan apakah disuatu posisi terdapat botol yang berukuran besar atau kecil.
- Bit B3:6 dan B3:7 menerima masukan dari LS3 (sensor botol utuh atau rusak / *broken sensor*). Maka bit B3:6 dan B3:7 akan menunjukkan apakah disuatu posisi terdapat botol yang utuh atau rusak.

Sehingga seharusnya Anda dapat memahami bahwa bit-bit B3:2 – B3:7 dapat menunjukkan posisi-posisi botol seperti ditunjukkan pada gambar berikut.



Anda perlu memahami dan memberi perhatian pada bit-bit berikut ini.

Bit B3:2/9 menunjukkan apakah sudah terdapat botol dibawah *Filler* (pengisi).

Bit B3:3/3 menunjukkan apakah sudah terdapat botol dibawah *Capper* (pemasang tutup botol).

Bit B3:4/7 menunjukkan apakah dibawah *Filler* terdapat botol besar atau kecil.

Bit B3:4/11 menunjukkan apakah botol diatas *Scrap Gate* (gerbang untuk menurunkan botol rusak) adalah botol berukuran besar atau kecil.

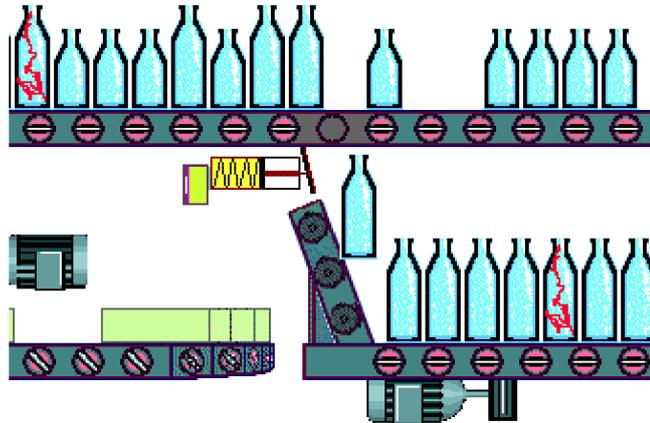
Bit B3:5/4 menunjukkan apakah botol diatas *Divert Gate* (gerbang untuk menurunkan botol besar) adalah botol berukuran besar atau kecil.

Bit B3:6/5 menunjukkan apakah dibawah *Filler* terdapat botol utuh atau rusak.

Bit B3:6/8 menunjukkan apakah botol diatas *Divert Gater* adalah botol yang utuh atau rusak.

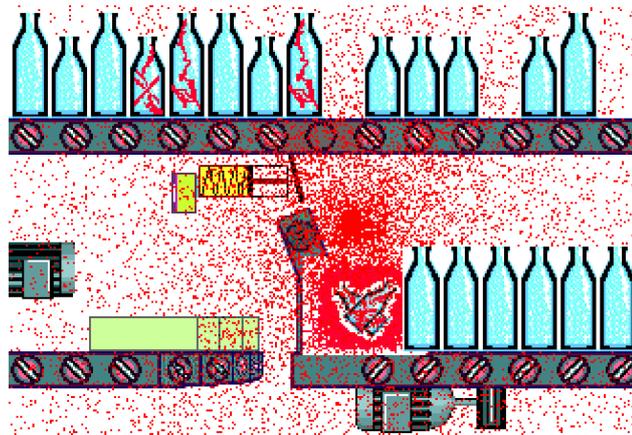
I.1.3 Latihan 3 : Memindahkan Botol Besar

Tambahkan program Anda sehingga semua botol besar akan berpindah ke konveyor bawah yang ada pada posisi sebelah kanan simulasi.



Tentu Anda harus menggunakan bit yang tepat diantara bit-bit B3:4 atau B3:5 karena diantara bit-bit B3:4 dan B3:5 terdapat informasi mengenai apakah pada suatu posisi terdapat botol besar atau botol kecil.

Hati-hati, karena botol bisa pecah jika ada kesalahan dalam pemrograman (seperti jika Divert Conveyor tidak berjalan saat botol baru datang). Maka Anda harus memastikan bahwa botol tidak rusak dalam proses ini.



Saat pengujian, gunakan “Reset Simulation” dan “Clear All Data Files (Data Table)” dalam Menu “Simulations” untuk me-*Restart* program Anda.

I.1.4 Memahami pengoperasian motor conveyor botol rusak

Pada bagian ini kita akan menggunakan sensor dan aktuator berikut ini :

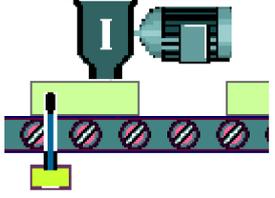
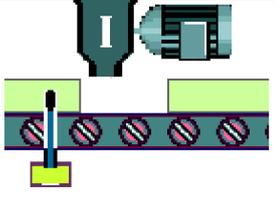
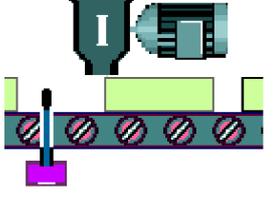
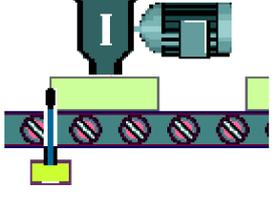
- Scrap conveyor (motor conveyor botol rusak O:2/1)
- LS10 (Sensor pedeteksi ada box I:1/15)
- Tombol ENTER (I:1/5)

Pahami bahwa sensor LS10 akan high jika terkena oleh box dan akan low jika tidak terkena box.

Buatlah program agar berfungsi sebagai berikut.

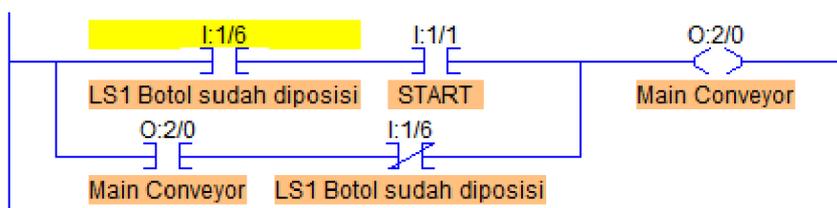
- Jika tombol ENTER ditekan, maka Scrap conveyor akan mulai menyala hingga datang box yang baru.
- Jika kecepatan simulasi terlalu cepat, anda bisa memelankannya dengan mengatur tombol Scans.

Ilustrasi prinsip yang diinginkan adalah sebagai berikut.

	<p>Saat tombol ENTER ditekan, Scrap Conveyor mulai berjalan. Scrap Conveyor berjalan maka box akan mulai melalui LS10. Karena LS10 masih mendeteksi box maka LS10 bernilai TRUE (berwarna hijau)</p>
	<p>Selama box melewati LS10, LS10 tetap bernilai TRUE (berwarna hijau)</p>
	<p>Setelah box selesai melewati LS10, maka LS10 bernilai FALSE (berwarna ungu)</p>
	<p>Jika setelah dari bernilai FALSE kemudian LS10 mendeteksi box lainnya (kembali bernilai TRUE), maka barulah Scrap Conveyor akan kembali mati.</p>

I.1.5 Mengamati jumlah Botol yang dapat masuk kedalam box

Tambahkan rung berikut dibawah rung Anda pada latihan sebelumnya.



Rung diatas berfungsi agar Main Conveyor bergerak jika tombol START ditekan.

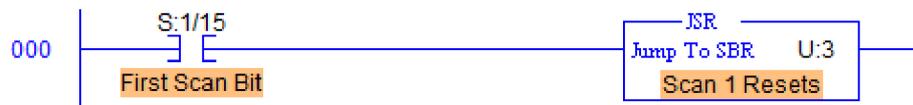
Tambahkan rung berikutnya agar jika tombol STOP ditekan, maka botol yang ada diatas posisi Scrap Gate akan dijatuhkan ke dalam box. Berarti setiap penekanan tombol STOP, Anda harus memfungsikan Scrap Gate (O:2/4) dan Grinder (O:2/3). Jika hanya Scrap Gate yang menyala dan Grinder mati, maka botol akan pecah.

Jika sudah selesai, hitunglah berapa banyak botol kecil yang dapat masuk kedalam box hingga penuh dan berapa banyak botol besar yang dapat masuk kedalam box sehingga box penuh.

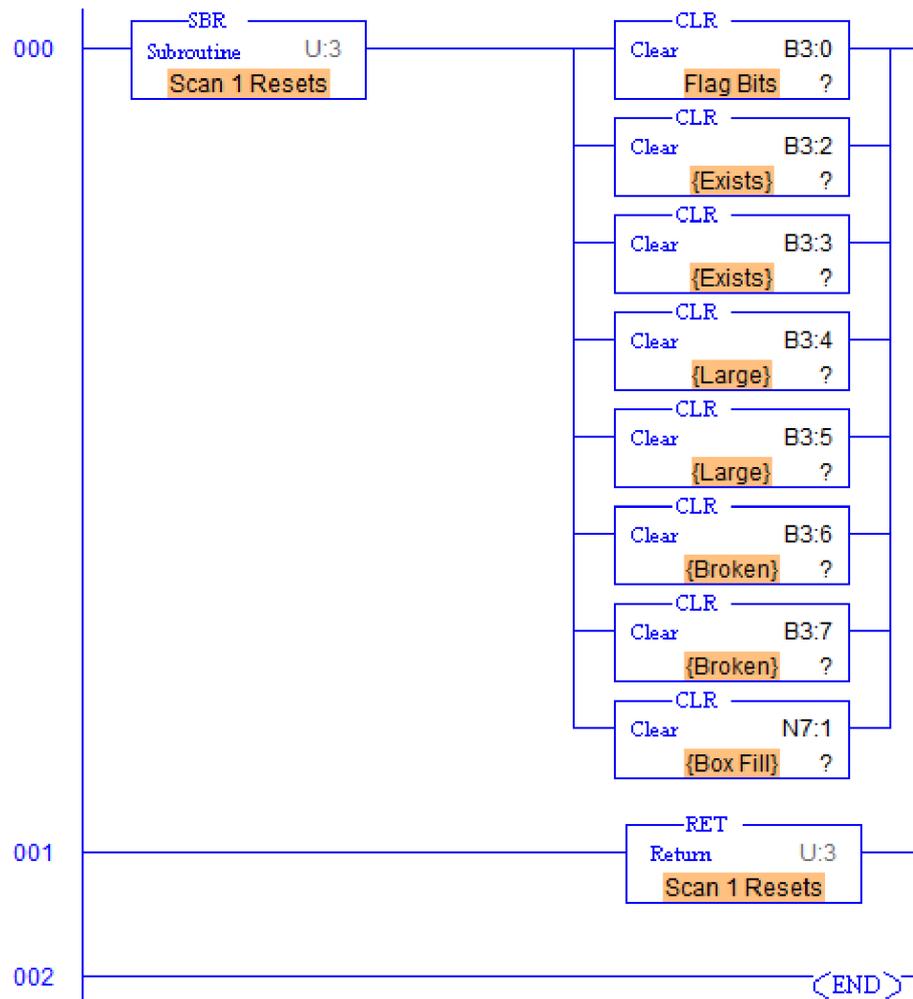
I.1.6 Latihan 4 : Pengepakan Botol Rusak

Pada latihan ini Anda diminta untuk membuang botol rusak ke pembuangan.

Tetapi sebelum melanjutkan, Anda mungkin mengalami kesulitan dalam melakukan *restarts* program karena harus melakukan “Clear All Data Files” setiap me-restart simulasi. Untuk mempermudah membersihkan variabel-variabel awal saat program pertama kali dijalankan, tambahkanlah ladder berikut ini pada awal program Anda.



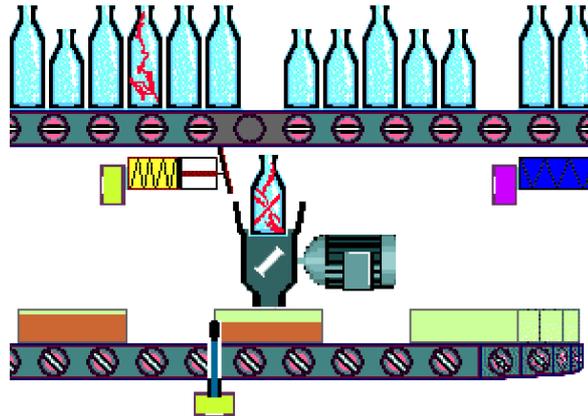
Pada alamat **SBR 3**:, buatlah program sebagai berikut.



Dengan menggunakan program diatas, maka setiap kali program dijalankan maka langkah pertama yang dilakukan adalah membersihkan seluruh memori berikut.

- B3:0 (digunakan untuk *flag bits*)
- B3:2 dan B3:3 (memori apakah disuatu posisi sudah ada botol atau belum)
- B3:4 dan B3:5 (memori apakah disuatu posisi terdapat botol berukuran besar atau kecil)
- B3:6 dan B3:7 (memori apakah disuatu posisi terdapat botol utuh atau rusak)
- N7:1 (memori mengenai jumlah botol yang sudah masuk ke kotak botol rusak)

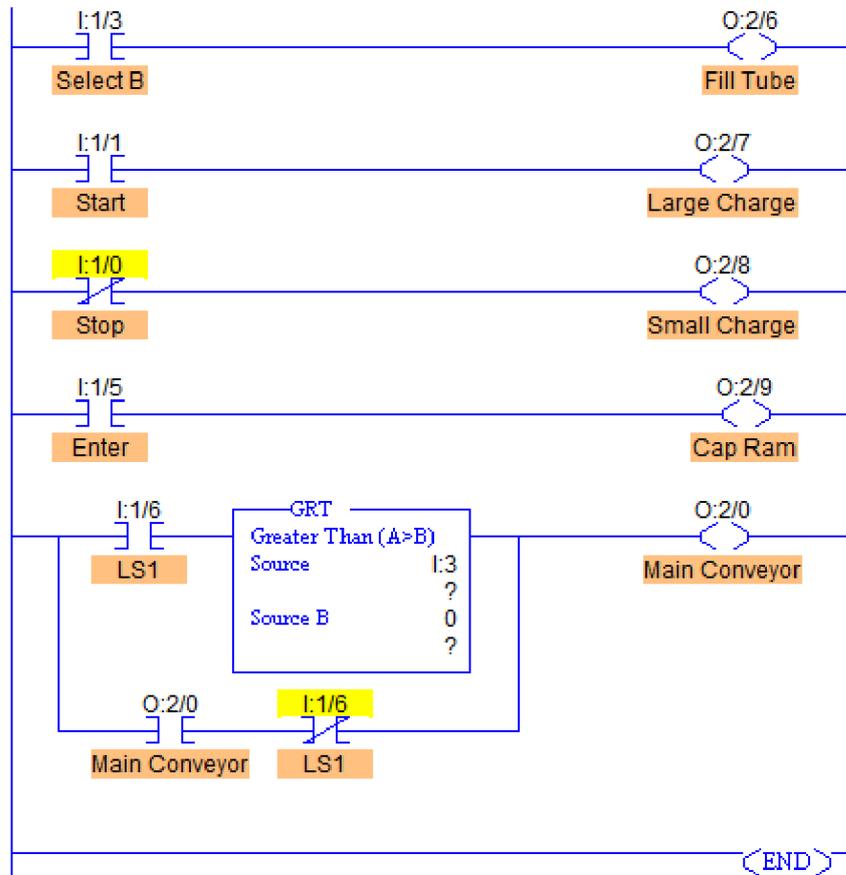
Selanjutnya, Anda harus mengeluarkan botol rusak ke bawah jalur. Pastikan dibawah *grinder* selalu terdapat kotak kardus yang masih memiliki ruang kosong untuk menampung kepingan botol.



Karena biaya untuk menyediakan kotak kardus perlu dihemat, maka Anda perlu memastikan bahwa tiap kotak kardus dapat diisi hingga kapasitas maksimumnya dan tidak tumpah. Karena botol kecil hanya memerlukan $\frac{2}{3}$ volume dibandingkan botol besar, maka Anda harus menyesuaikan perbedaan ini pada program Anda.

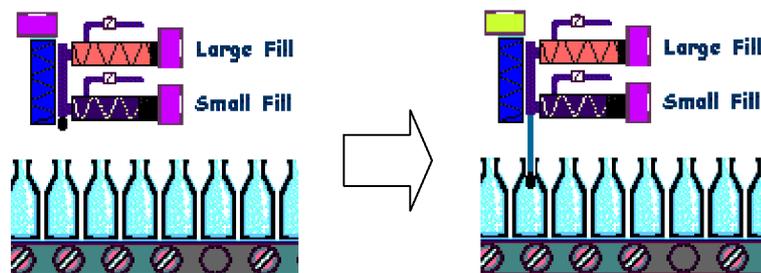
I.1.7 Memahami cara Mengisi dan Menutup Botol

Untuk memahami mekanisme pengisian dan penutupan botol pada simulasi ini, cobalah membuat ladder diagram sebagai berikut.



Maka proses untuk pengisian botol adalah sebagai berikut.

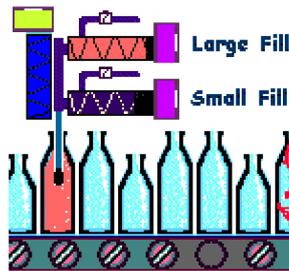
Pertama, **Fill Tube** harus diturunkan dahulu.



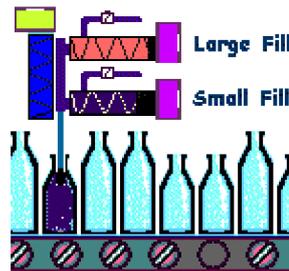
Caranya pindahkan selektor dari menunjuk ke A menjadi menunjuk ke B.



Kedua, jika botol berukuran besar, maka tekan tombol **Start** untuk mengisinya (akan terisi oleh cairan berwarna merah).



Jika botol berukuran kecil, maka tekan tombol Stop untuk mengisinya (akan terisi oleh cairan berwarna biru).



Untuk menaikkan kembali **Fill Tube**, maka pindahkan lagi selektor dari menunjuk ke B menjadi menunjuk ke A.

Hati-hati terhadap keadaan berikut :

- Jika botol yang sudah penuh tetap diisi, maka cairan akan tumpah
- Jika botol kecil diisi oleh cairan berwarna merah, maka cairan akan tumpah
- Jika botol besar diisi oleh cairan berwarna biru, maka hanya akan terisi sebagian
- Jika motor conveyor dijalankan sedangkan **Fill Tube** belum dinaikkan, maka botol akan pecah

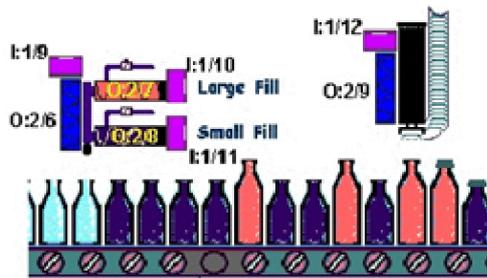
Untuk menjalankan conveyor maka naikan nilai angka dari BCD seperti berikut. Dan untuk memmatikannya, jadikan 0 nilai dari BCD.

	
BCD tidak bernilai 0, conveyor akan berjalan	BCD bernilai 0, conveyor akan mati

Penutupan botol dilakukan dengan menekan tombol Enter yang ada dibawah BCD.

I.1.8 Latihan 5 : Mengisi dan Menutup Botol

Pada operasi ini Anda diminta membuat program agar sistem secara otomatis akan menghentikan main conveyor setiap kali ada botol utuh yang berada dibawah **Fill Tube**, kemudian menurunkan **Fill Tube** O:2/6 untuk menyiapkan proses pengisian botol. Selanjutnya program harus mendeteksi ukuran botol yang ada pada posisi **Fill Tube**. Jika botol berukuran besar, maka O:2/7 yang diaktifkan, sedangkan jika botol berukuran kecil, maka O:2/6 yang diaktifkan. Untuk itu diperlukan data-data boolean pada bit *array* untuk menentukan aksi yang benar.



Pemasangan tutup botol dilakukan pada setiap botol yang datang pada bagian pemasangan botol. Pemasangan ini dilakukan dengan menjalankan O:2/9. Yang perlu diperhatikan adalah pemasangan botol hanya boleh dilakukan jika ada botol yang benar datang. Karena jika ada kesalahan maka akan membuang tutup botol dan juga akan merusak conveyor.