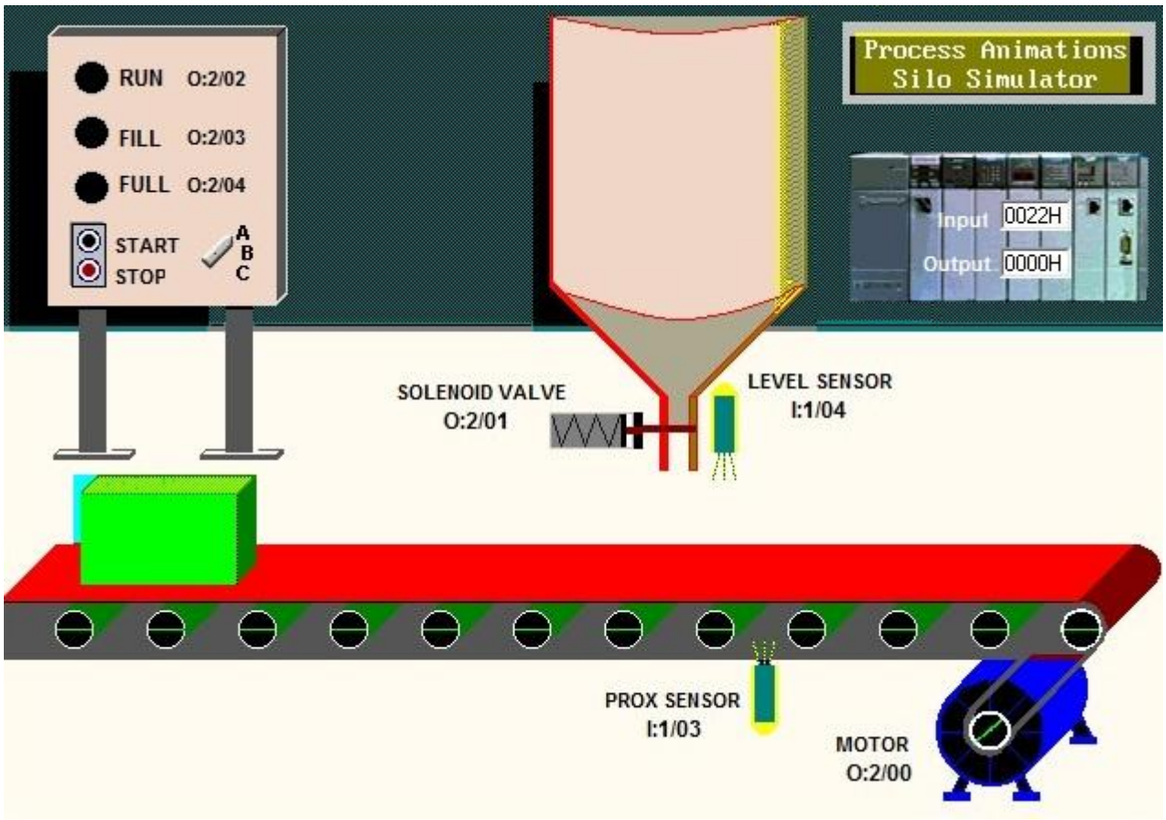
Silo Simulation



Terdapat tombol selektor :

Jika selektor terhubung ke A maka I:1/05 akan aktif. Maka silo akan berfungsi sesuai proses mode A.
Jika selektor terhubung ke B maka I:1/06 akan aktif. Maka silo akan berfungsi sesuai proses mode B.
Jika selektor terhubung ke C maka I:1/07 akan aktif. Maka silo akan berfungsi sesuai proses mode C.

- Lampu RUN O:2/02 akan menyala selama sistem beroperasi
- Lampu FILL O:2/03 akan menyala selama sistem mengisi
- Lampu FULL O:2/04 akan menyala saat box telah terisi penuh. Lampu akan tetap menyala hingga box melewati PROX SENSOR I:1/03
- Proses ini dapat dihentikan dan dijalankan kembali menggunakan tombol STOP I:1/01 dan tombol START I:1/00
- Dimanapun program dihentikan, maka saat program dijalankan akan melanjutkan ke operasi berikutnya. (jika dihentikan saat mengisi dan belum penuh, maka saat program dijalankan akan melanjutkan mengisi dahulu sampai box penuh)

Proses Mode A (mode kontinyu)

- Saat tombol START ditekan, program secara otomatis akan membawa box kebawah solenoid lalu mengisinya hingga LEVEL SENSOR I:1/04 aktif. Lalu memindahkan box dari bawah solenoid dan mendatangkan box baru. Proses ini akan bekerja terus secara otomatis

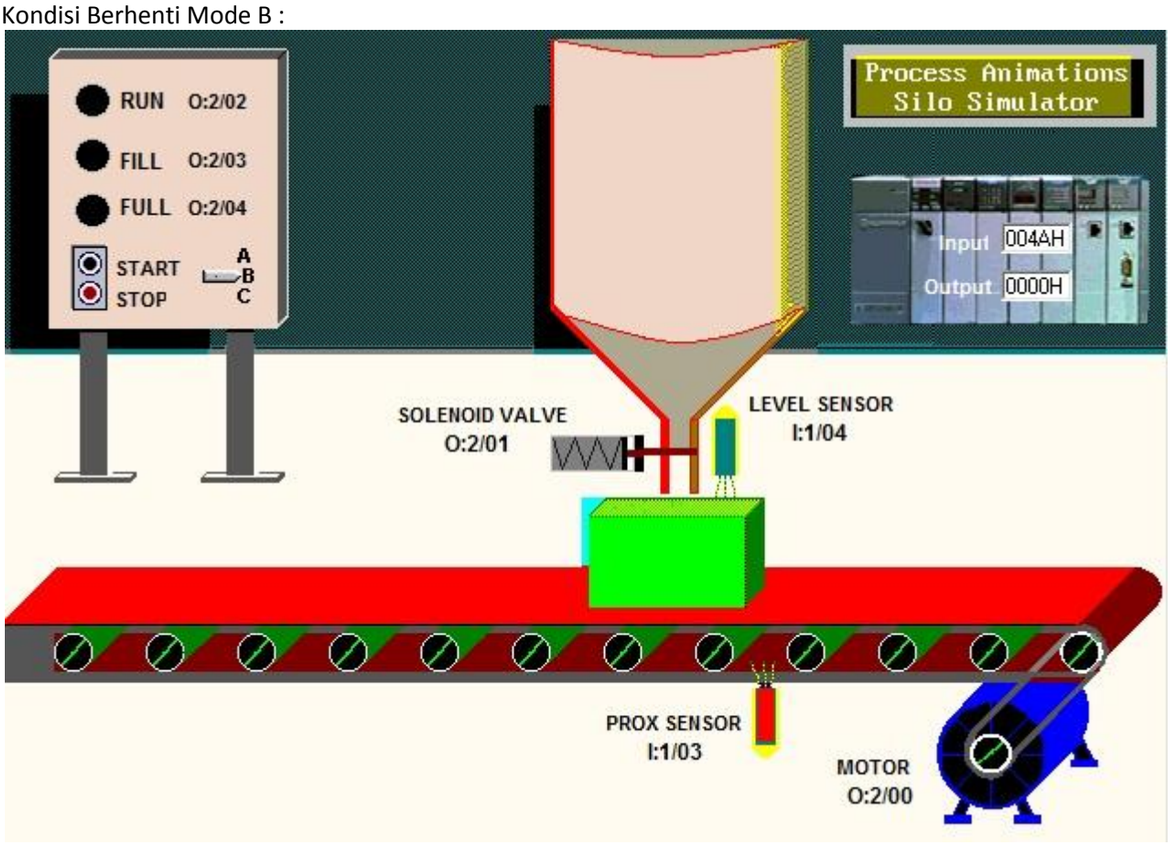
Proses Mode B (mode kontinyu dengan jumlah tertentu)

- Sama seperti Mode A, tetapi proses hanya akan bekerja otomatis untuk mengisi 5 box saja. (Gunakan counter)
- Jika sudah 5 box yang terisi, maka proses akan otomatis berhenti. Jika tombol START ditekan, maka proses akan bekerja mengisi 5 box lagi kemudian berhenti.
- Setelah 5 box, **sistem baru berhenti setelah box yang penuh dibawa kekanan keluar dari layar dan box baru dibawa kebawah solenoid tetapi belum diisi**.
- Jika misalnya setelah 3 box, tombol STOP ditekan, maka proses akan berhenti. Dan jika kemudian tombol START ditekan, maka proses akan bekerja otomatis untuk mengisi 2 box lagi kemudian berhenti.

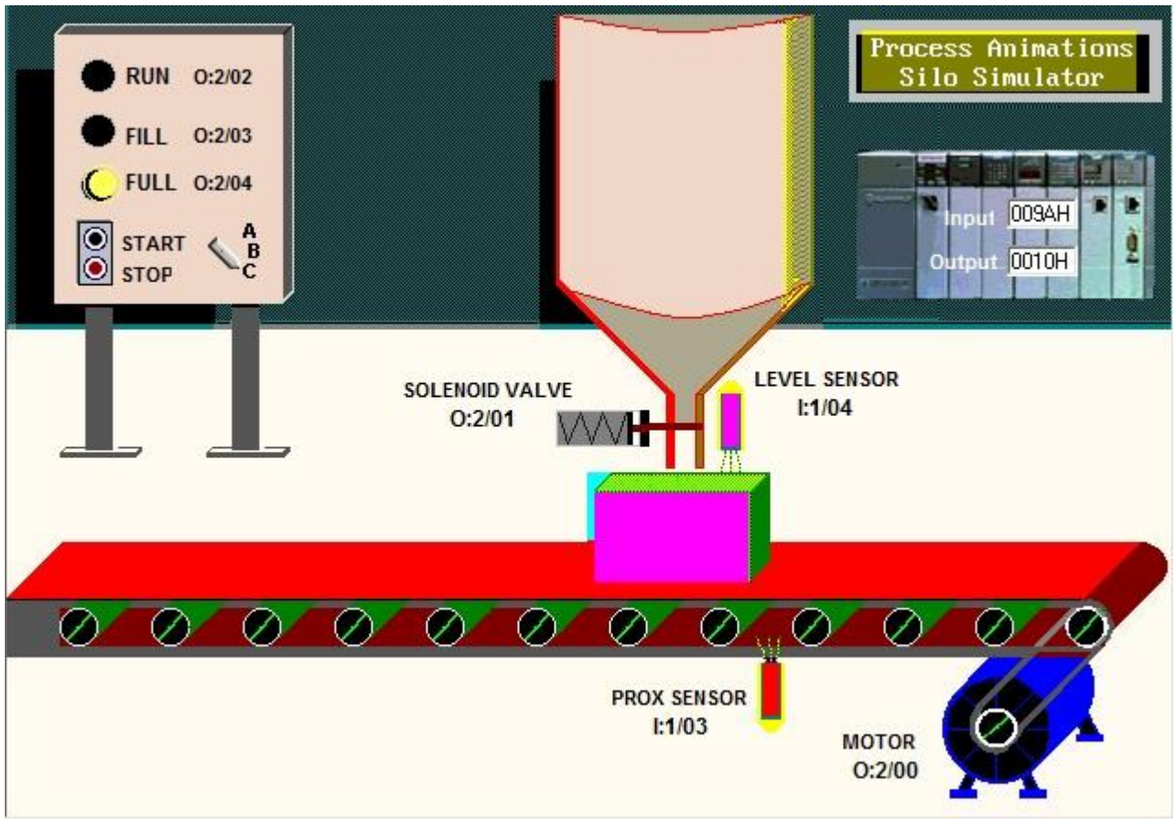
Proses Mode C (mode operasi dengan manual restart : Pemantauan Manual)

- Saat tombol START ditekan, program secara otomatis akan membawa box kebawah solenoid lalu mengisinya hingga LEVEL SENSOR I:1/04 aktif.
- Saat box telah terisi penuh, maka sistem akan berhenti bekerja. Tombol START I:1/00 harus ditekan terlebih dahulu agar conveyor beroperasi kembali. (**sistem berhenti saat box ada dibawah solenoid**)
- Jika setelah box terisi penuh lalu tombol START I:1/00 **DITEKAN TERUS**, maka box berikutnya setelah terisi **akan tetap berhenti**. (menunggu tombol START dilepas dahulu. Dan jika tombol START ditekan kembali maka barulah sistem akan kembali beroperasi)

Kondisi Berhenti Mode B :



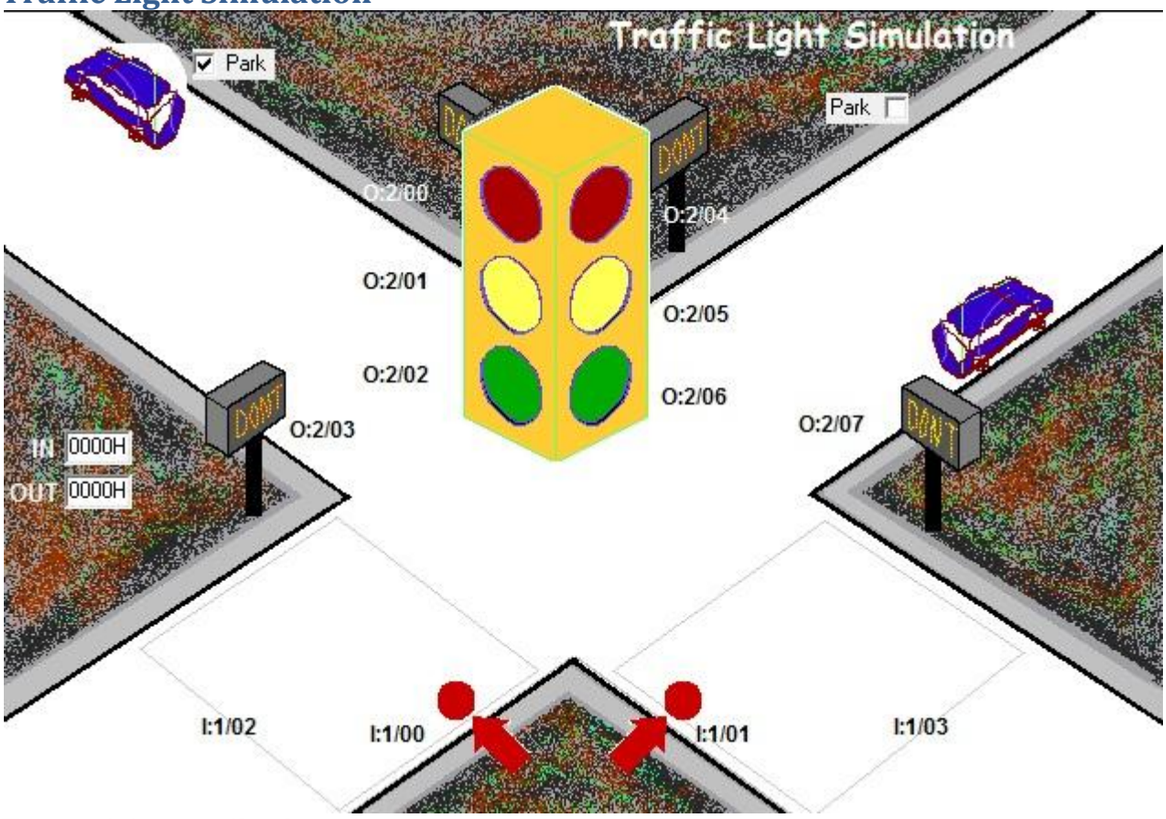
Kondisi Berhenti Mode C :



TEST POINT :

1. Dimanapun program dihentikan, maka saat program dijalankan akan melanjutkan ke operasi berikutnya. (jika dihentikan saat mengisi dan belum penuh, maka saat program dijalankan akan melanjutkan mengisi dahulu sampai box penuh)
2. Pada mode B setelah START ditekan, maka jumlah box yang diisi adalah 5 buah
3. Pada mode B jika misalnya setelah 3 box, tombol STOP ditekan, maka proses akan berhenti. Dan jika kemudian tombol START ditekan, maka proses akan bekerja otomatis untuk mengisi 2 box lagi kemudian berhenti.
4. Pada mode B sistem baru berhenti setelah box yang penuh dibawa kekanan keluar dari layar dan box baru dibawa kebawah solenoid tetapi belum diisi.
5. Pada mode C sistem berhenti saat box ada dibawah solenoid
6. Jika setelah box terisi penuh lalu tombol START I:1/00 **DITEKAN TERUS**, maka box berikutnya setelah terisi **akan tetap berhenti**. (menunggu tombol START dilepas dahulu. Dan jika tombol START ditekan kembali maka barulah sistem akan kembali beroperasi)

Traffic Light Simulation



Tahap 1 (mode fix timer)

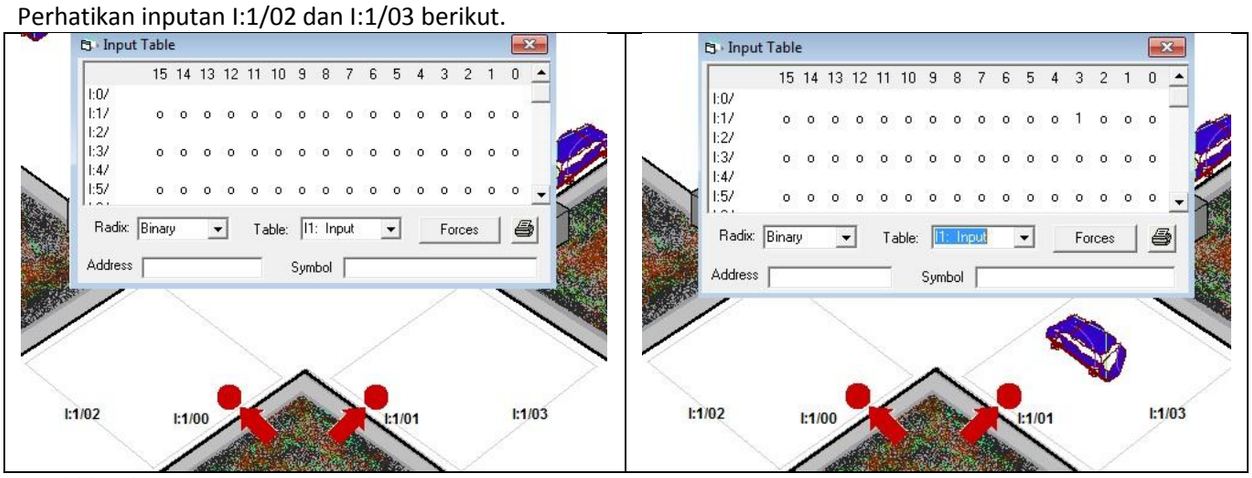
Lampu akan menyala sesuai urutan waktu sebagai berikut :

Merah (O:2/00)			Hijau (O:2/02)	Kuning (O:2/01)	Merah (O:2/00)
Hijau (O:2/06)	**Kuning (O:2/05)**	**Merah (O:2/04)**			
8 dt	4 dt	1 dt	8 dt	4 dt	1 dt

- Saat O:2/06 menyala, maka lampu jalan O:2/03 akan menyala (bertulisan WALK)
- Saat O:2/05 menyala, maka lampu jalan O:2/03 akan menyala berkedip (DON'T – WALK)
- Saat O:2/04 menyala, maka lampu jalan O:2/03 akan mati (bertulisan DON'T)
- Prinsip yang sama juga berlaku untuk lampu jalan O:2/07

Tahap 2 (mode semi inteleget timer)

Perhatikan inputan I:1/02 dan I:1/03 berikut.



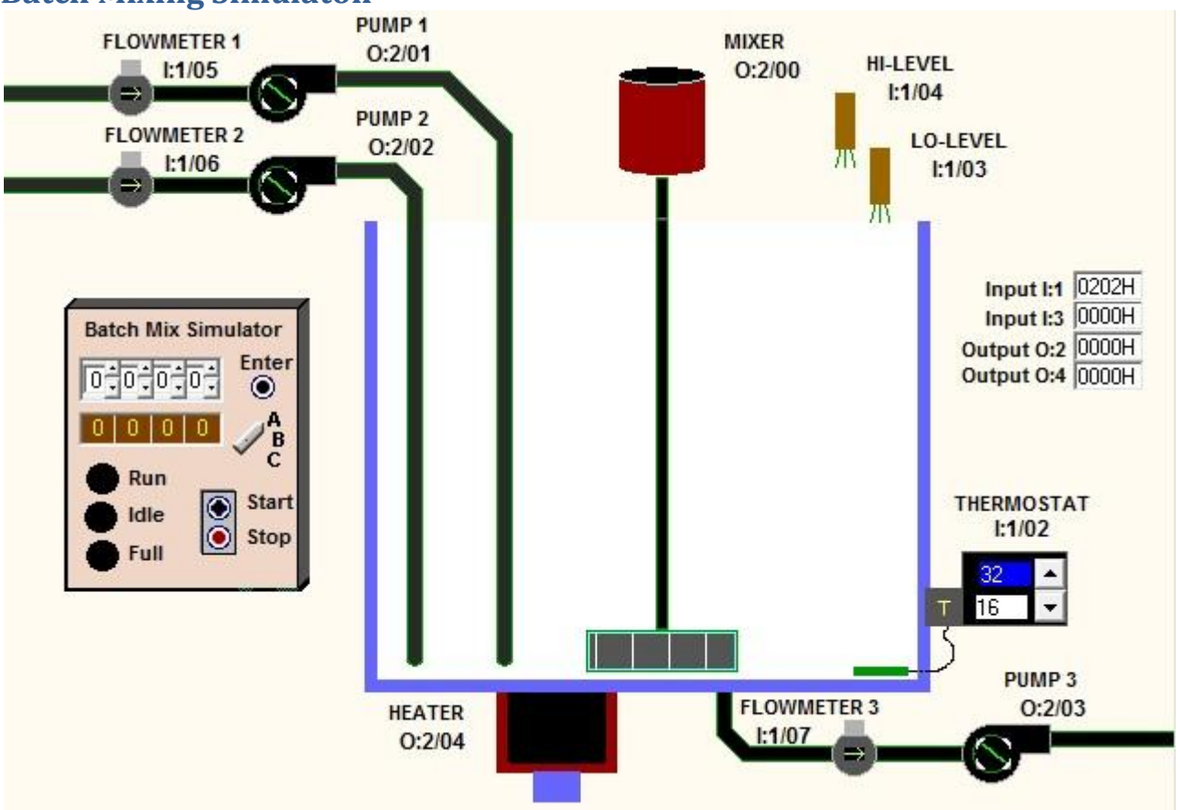
Modifikasikan mode fix timer sehingga :

- Lampu Hijau (O:2/06) baru akan berubah menjadi Kuning (O:2/05) jika setelah 8 detik terdeteksi ada mobil di I:1/02
- Walaupun tidak ada mobil di I:1/02, Lampu Hijau (O:2/06) akan tetap berfungsi seperti fix timer jika ada yang menekan tombol penyeberangan I:1/01 :
  - Jika ditekan sebelum Lampu Hijau menyala 8 detik maka Lampu Hijau tetap menyala dahulu hingga 8 detik
  - Jika ditekan setelah Lampu Hijau menyala 8 detik, maka Lampu Hijau segera berubah menjadi Kuning
- Prinsip yang sama juga berlaku untuk Lampu Hijau (O:2/02)
- Proses ini perlu disimulasikan dengan memarkirkan salah satu mobil kemudian diaktifkan kembali.

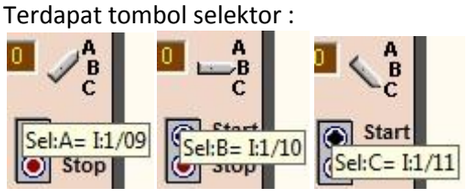
TEST POINT :

1. Pewaktuan pada mode fixed timer benar
2. Saat tidak ada mobil di I:1/03, maka O:2/02 akan menyala terus
3. Jika sebelum Lampu Hijau O:2/02 menyala 8 detik ada mobl di I:1/03 maka pada detik ke 8 lampu berubah menjadi kuning.
4. Jika setelah Lampu Hijau O:2/02 menyala 8 detik kemudian datang mobl di I:1/03 maka pada detik tersebut lampu berubah menjadi kuning.
5. Jika sebelum Lampu Hijau O:2/02 menyala 8 detik ada yang menekan tombol penyeberangan I:1/00 maka Lampu Hijau akan tetap menyala dahulu hingga 8 detik
6. Jika setelah Lampu Hijau O:2/02 menyala 8 detik ada yang menekan tombol penyeberangan I:1/00 maka Lampu Hijau akan segera berubah menjadi Kuning
7. Hal yang sama berlaku untuk Lampu Hijau O:2/06

Batch Mixing Simulaton



Terdapat tombol selektor :



Jika selektor terhubung ke A maka I:1/09 aktif dan akan memfungsikan proses mode A.
Jika selektor terhubung ke B maka I:1/10 aktif
Jika selektor terhubung ke C maka I:1/11 aktif

Proses Mode A (Otomatis untuk menghasilkan satu kali pencampuran)

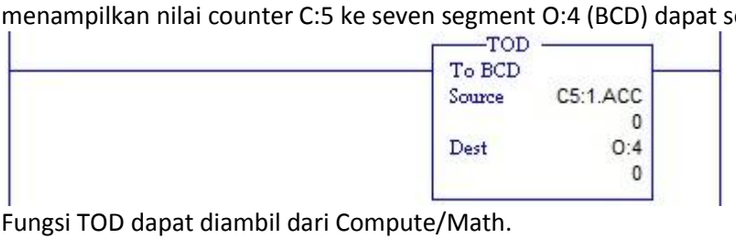
- Proses akan berjalan kontinyu dan dapat dihentikan serta di*start* kembali menggunakan tombol START dan STOP.
- Tangki pertama-tama diisi oleh PUMP 1. PUMP P1 dikendalikan oleh *counter* hingga ketinggian air dalam tangki sekitar 90%. Setelah tercapai, PUMP P1 berhenti dan PUMP P2 aktif sehingga ketinggian air mengaktifkan sensor HI-LEVEL I:1/04.
- Setelah proses pengisian selesai, lampu FULL akan menyala, HEATER O:2/04 dan MIXER O:2/00 akan aktif untuk memulai proses pemanasan.
- THERMOSTAT I:1/02 digunakan untuk mengetahui apakah temperatur air dalam tangki sudah mencapai suhu yang diinginkan. Setelah tercapai, maka HEATER akan mati tetapi MIXER tetap menyala.
- MIXER tetap beroperasi 4 detik setelah suhu air campuran yang diinginkan telah tercapai. Dan setelah mixer berhenti, PUMP P3 akan mengosongkan tangki hingga sensor LO-LEVEL I:1/03 tidak mendeteksi ketinggian air. (Hingga saat ini disebut satu pencampuran telah selesai)
- Setelah **tangki kosong, proses akan berhenti**
- Lampu RUN akan menyala selama sistem beroperasi
- Lampu FULL akan menyala selama tangki penuh dan PUMP P3 belum aktif.
- Lampu IDLE akan menyala jika sistem berhenti beroperasi (tombol STOP ditekan)
- Dimanapun program dihentikan, maka saat program dijalankan akan melanjutkan ke operasi berikutnya. (Jika dihentikan saat mengisi dan belum penuh, maka saat program dijalankan akan melanjutkan mengisi dahulu sampai tangki penuh.
  Jika dihentikan saat mengosongkan dan belum kosong, maka saat program dijalankan akan melanjutkan mengosongkan dahulu sampai tangki kosong,
  Jika dihentikan saat memanaskan dan belum sampai suhu yang diinginkan, maka saat program dijalankan akan melanjutkan memanaskan dahulu sampai suhu yang diinginkan.)

Jika Selektor di B dan C (proses akan mencampur dengan perbandingan PUMP 1 dan PUMP 2 tertentu dan menghasilkan jumlah tertentu)
Selektor di B (mengatur jumlah campuran yang dihasilkan)

- Sama seperti Mode A, tetapi proses akan bekerja otomatis untuk menghasilkan pencampuran sejumlah tertentu. Jumlah pencampuran yang dihasilkan dapat ditentukan oleh operator dengan cara mengetikkan angka pada I:3 (BCD) seperti berikut kemudian menekan ENTER I:1/08

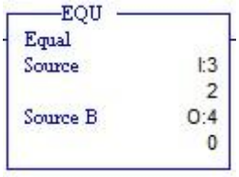
- Gunakan *counter* untuk menghitung banyaknya campuran yang telah diproses.
- Jumlah campuran yang dihitung oleh *counter* ditampilkan ke seven segment O:4 (BCD)
- Setelah jumlah campuran [O:4 (BCD)] sama dengan jumlah yang diinginkan [I:3 (BCD)], maka proses akan berhenti dan nilai O:4 (BCD) akan tetap menampilkan jumlah campuran yang sudah dihasilkan. Jika proses di START kembali, barulah nilai O:4 (BCD) akan reset kembali ke 0000

Jika counter C:5 digunakan untuk menghitung jumlah campuran yang sudah dihasilkan, maka cara untuk menampilkan nilai counter C:5 ke seven segment O:4 (BCD) dapat sebagai berikut.



Fungsi TOD dapat diambil dari Compute/Math.

Untuk membandingkan apakah nilai I:3 (BCD) sudah sama dengan O:4 (BCD) dapat sebagai berikut.



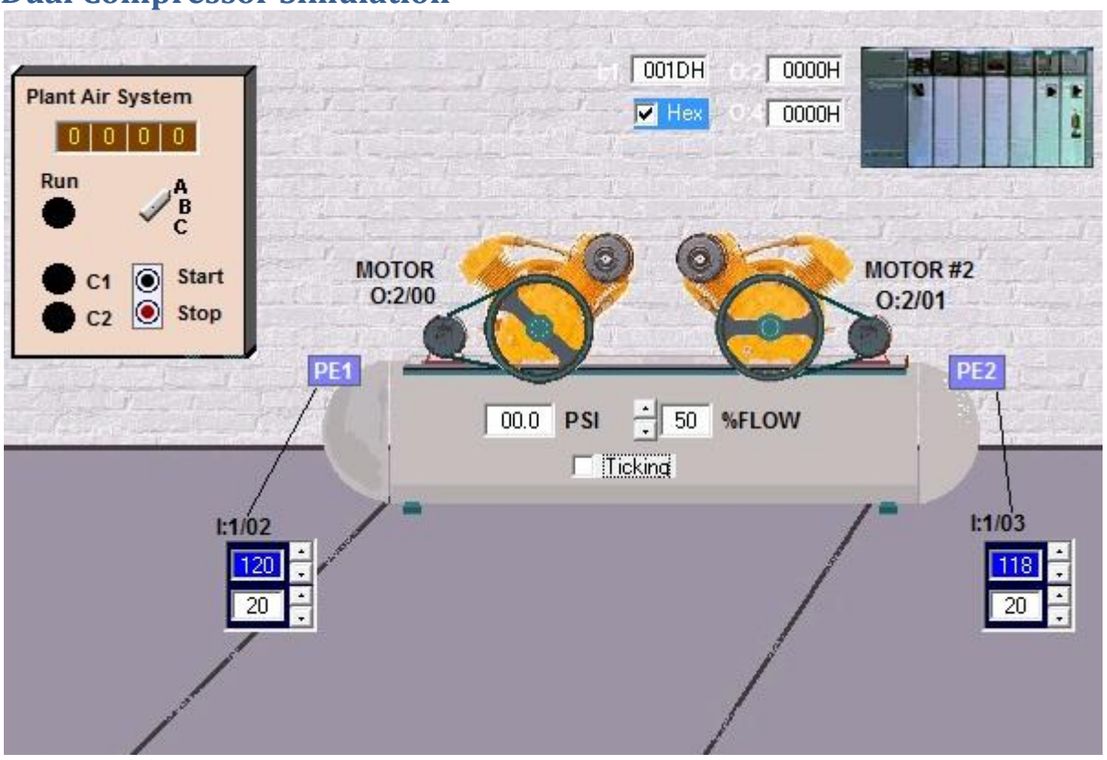
Selektor di C (mengatur perbandingan PUMP 1 dan PUMP 2 dalam persen)

- Sama seperti Mode A. Tetapi jika pada Mode A perbandingan campuran P1 dan P2 adalah 90% : 10%, maka pada Mode C, perbandingan campuran dapat ditentukan oleh operator dengan cara mengetikkan angka pada I:3 (BCD) [dalam persen tentunya], kemudian menekan ENTER.
- Selama PUMP P1 hidup, maka jumlah persentase aliran dari PUMP P1 ditampilkan ke seven segment O:4 (BCD). Tampilan O:4 (BCD) akan reset ke 0000 saat pengosongan tangki mulai berlangsung.
- Untuk kasus ini Anda tentu perlu menggunakan fungsi-fungsi Compute/Math seperti Multiplication

TEST POINT :

1. Pada Mode A, PUMP 1, PUMP 2, HEATER, MIXER dan PUMP 3 bekerja sesuai yang diharapkan.
2. Dimanapun program dihentikan, maka saat program dijalankan akan melanjutkan ke operasi berikutnya. (Jika dihentikan saat mengisi dan belum penuh, maka saat program dijalankan akan melanjutkan mengisi dahulu sampai tangki penuh.
   Jika dihentikan saat mengosongkan dan belum kosong, maka saat program dijalankan akan melanjutkan mengosongkan dahulu sampai tangki kosong,
   Jika dihentikan saat memanaskan dan belum sampai suhu yang diinginkan, maka saat program dijalankan akan melanjutkan memanaskan dahulu sampai suhu yang diinginkan.)
3. Pada mode B, jumlah pencampuran yang dihasilkan dapat ditentukan oleh operator dengan cara mengetikkan angka pada I:3 (BCD) seperti berikut kemudian menekan ENTER I:1/08
4. Jumlah campuran yang dihitung oleh *counter* ditampilkan ke seven segment O:4 (BCD). Setelah jumlah campuran yang dihasilkan sesuai dengan yang diharapkan, O:4 (BCD) akan tetap menampilkan jumlah campuran yang sudah dihasilkan. Jika proses di START kembali, barulah nilai O:4 (BCD) akan reset kembali ke 0000.
5. Pada mode C, perbandingan campuran dapat ditentukan oleh operator dengan cara mengetikkan angka pada I:3 (BCD) [dalam persen tentunya], kemudian menekan ENTER.
6. Selama PUMP P1 hidup, maka jumlah persentase aliran dari PUMP P1 ditampilkan ke seven segment O:4 (BCD). Tampilan O:4 (BCD) akan reset ke 0000 saat pengosongan tangki mulai berlangsung.

Dual Compressor Simulation



Terdapat selektor

Jika selektor terhubung ke A maka I:1/04 akan aktif.
Jika selektor terhubung ke B maka I:1/05 akan aktif.
Jika selektor terhubung ke C maka I:1/06 akan aktif.

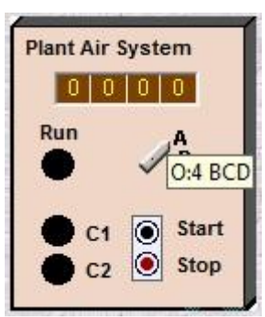
- Lampu RUN akan menyala selama sistem beroperasi
- Proses ini dapat dihentikan dan dijalankan kembali menggunakan tombol STOP dan tombol START

Proses yang dilakukan :

- Set batas atas I:1/02 menjadi 42 dan deltanya 20
- Set batas atas I:1/03 menjadi 40 dan deltanya 20
- Jika rentang tekanan pada compressor berkisar antara 22 hingga 42 maka MOTOR 1 O:2/00 dan MOTOR 2 O:2/01 akan bekerja bergantian :
  Motor 1 akan bekerja terlebih dahulu. Kemudian setelah tekanan mencapai 42 maka motor akan mati. Tekanan lalu akan turun. Setelah tekanan mencapai nilai 22 maka motor 2 akan hidup.
  Saat motor 2 hidup, maka tekanan compressor akan naik kembali. Setelah mencapai 42 maka motor 2 mati. Setelah tekanan turun kembali menjadi 22, maka Motor 1 yang kembali dihidupkan. Dst.
- Jika tekanan jatuh hingga nilai 20 maka kedua motor harus dihidupkan bersama. (Berarti beban kerja compressor tidak dapat ditangani oleh hanya satu motor). [Terjadi jika flow keluaran compressor lebih dari 80]

Langkah berikut dibuat untuk mengantisipasi jika flow keluaran compressor adalah 80. (Dapat menyebabkan satu motor bekerja terus tanpa henti karena tekanan masukan sama dengan tekanan keluaran) :

- Jika suatu motor telah bekerja melebihi waktu normal untuk mencapai tekanan penuh (batas atas) pada flow keluaran 50 % atau 60 %, maka motor kedua harus dihidupkan. (Cara 1)
- Ukurlah waktu normal tekanan turun pada compressor dari maximum ke minimumnya (42 ke 22). Maka jika waktu penurunan tekanan compressor lebih cepat berarti dibutuhkan dua motor yang selanjutnya bekerja. (Cara 2)
- Gabungkan cara I dengan cara II. Kekurangan pada cara I adalah adanya waktu untuk motor harus bekerja sendiri terlebih dahulu sedangkan kelemahan dari cara II adalah tidak dapat mendeteksi jika perubahan flow terjadi setelah mencapai tekanan 22.



- Jika selektor ke A maka O:4/BCD akan menampilkan total waktu bekerja motor I
- Jika selektor ke B maka O:4/BCD akan menampilkan total waktu bekerja motor II
- Waktu ditampilkan dalam detik
- Program Anda harus dapat bekerja sedemikian rupa agar perbandingan total waktu kerja motor I dan motor II relatif sama.

TEST POINT :

1. Saat START pertama kali (tekanan dalam compressor = 0), MOTOR 1 dan MOTOR 2 akan bekerja bersama.
2. Pada flow keluaran 50%, MOTOR 1 O:2/00 dan MOTOR 2 O:2/01 akan bekerja bergantian.
3. Pada flow keluaran 90%, tekanan akan jatuh dibawah batas bawah sensor 2 dan kedua MOTOR akan bekerja bersama

Pada flow keluaran 50%, setelah batas atas tercapai dan MOTOR mati :

4. Jika flow keluaran dijadikan 80% saat tekanan turun, maka saat MOTOR mulai bekerja kembali kedua MOTOR akan langsung bekerja

Pada flow keluaran 50% saat pengisian tekanan dan hanya 1 MOTOR yang bekerja :

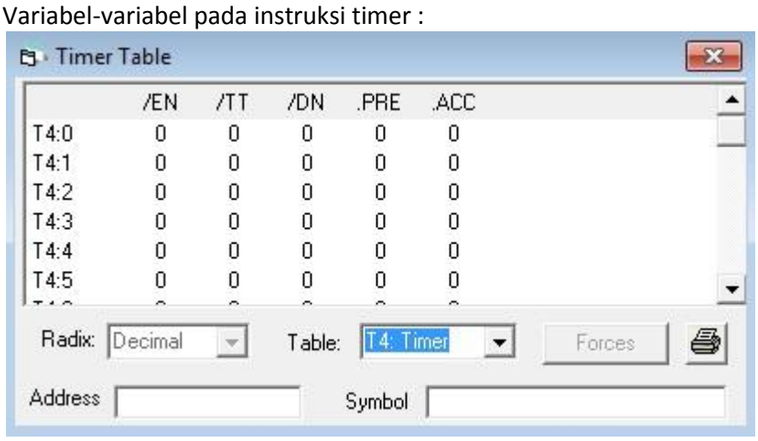
5. Jika flow keluaran dijadikan 80% saat tekanan diisi, maka setelah beberapa lama kedua MOTOR harus langsung bekerja

6. Waktu kerja masing-masing motor ditampilkan di O:4/BCD menggunakan selektor dan dalam satuan detik.
7. Perbandingan total waktu kerja motor I dan motor II relatif sama.

Timer/Counter Instructions

- Instruksi-instruksinya : TON TOF RTO CTU CTD RES

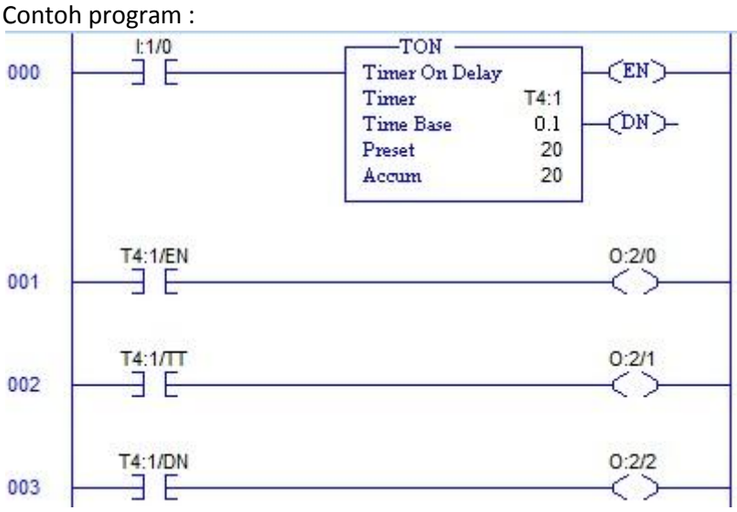
- Variabel-variabel pada instruksi timer :



  - PRE (Preset value) : nilai yang ingin dicapai oleh timer
  - ACC (Accumulated value) : nilai yang akan bertambah saat timer berjalan

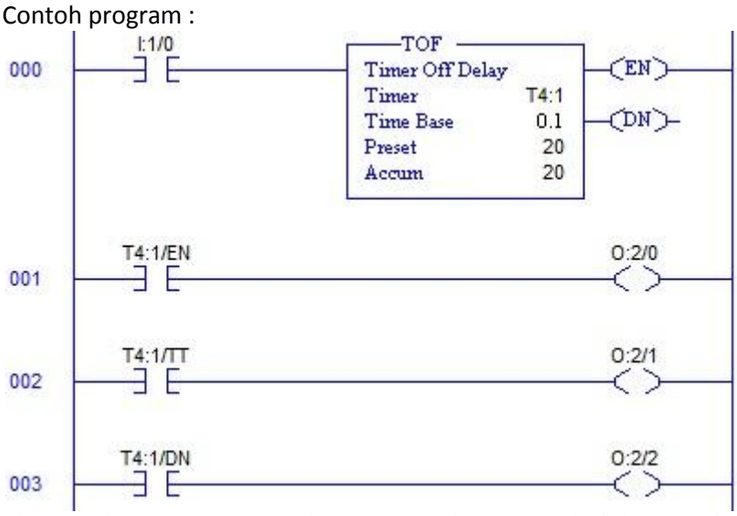
Instruksi TON (Timer ON Delay)

- Instruksi TON mulai menghitung waktu saat kondisi inputnya bernilai *true*. Selama kondisi input *true*, timer akan meningkatkan nilai ACC seiring dengan waktunya hingga mencapai nilai PRE. **Nilai ACC akan reset saat kondisi input dari timer bernilai *false***.
- EN (Enable Bit) : aktif saat timer mendapat masukan
- TT : aktif saat timer bekerja dan nilai ACC belum sama dengan nilai PRE
- DN (Done Bit) : aktif saat nilai ACC sama dengan nilai PRE
- Contoh program :



Instruksi TOF (Timer OFF Delay)

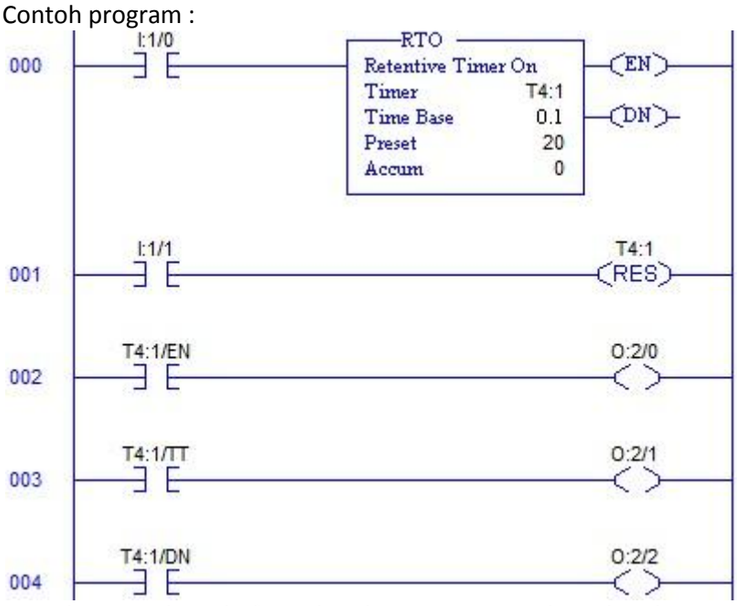
- Contoh program :



  - Jika mendapat input *true*, nilai ACC TOF akan 0, EN aktif dan DN aktif
  - Jika mendapat input *false*, nilai ACC TOF akan mulai bertambah, TT aktif dan DN aktif
  - Saat nilai ACC = PR maka EN, TT dan DN tidak aktif.

Instruksi RTO (Retentive Timer ON)

- Contoh program :



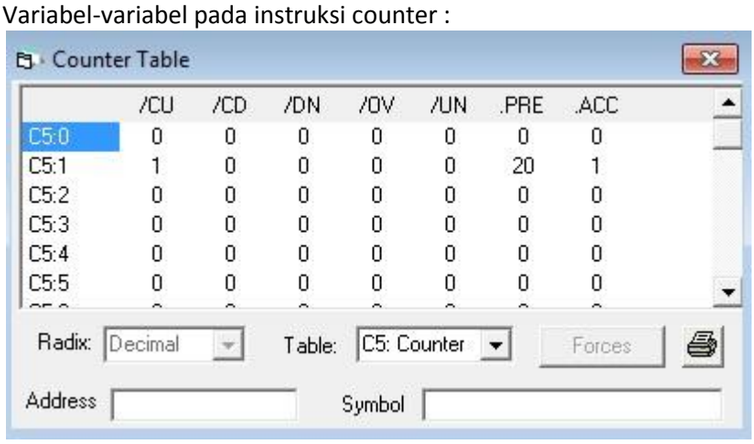
- Instruksi RTO akan aktif saat kondisi input *true*. Selama kondisi input *true*, timer akan meningkatkan nilai ACC seiring dengan waktunya hingga mencapai nilai PRE.
- **Saat kondisi input bernilai false, maka peningkatan nilai ACC akan berhenti (<u>tidak reset</u>)**. Saat input kembali bernilai *true*, maka timer akan melanjutkan peningkatan nilai ACC dari nilai sebelumnya yang telah tercapai.
- Untuk mengembalikan kembali nilai ACC menjadi 0, maka ACC Timer harus di-reset menggunakan instruksi RESET (seperti rung 001 diatas)

Instruksi RES (Reset)

- Digunakan untuk mereset nilai ACC dari timer atau counter.
- Sewaktu nilai inpu RES bernilai *truei,* maka akan mereset timer atau counter yang memiliki alamat yang sama dengan instruksi RES

Instruksi Counter

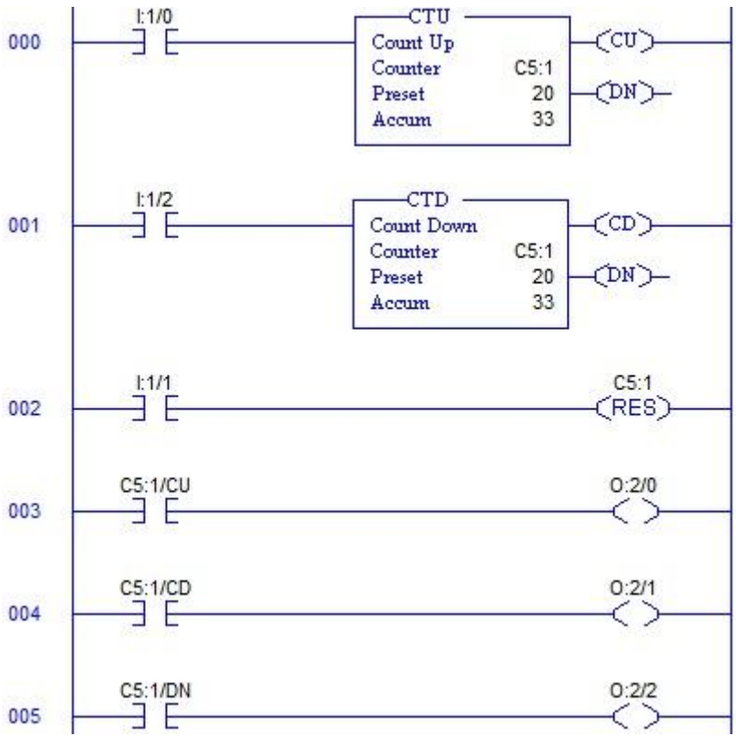
- Berfungsi untuk menghitung jumlah perubahan input.
- Variabel-variabel pada instruksi counter :



- CU : akan aktif saat nilai ACC bertambah (input CTU true)
- CD : akan aktif saat nilai ACC berkurang (input CTD true)
- DN : akan aktif saat nilai ACC sama dengan atau lebih dari nilai PRE
- ACC dapat menghitung melebihi nilai PRE

Instruksi CTU (Count Up) dan CTD (Count Down)

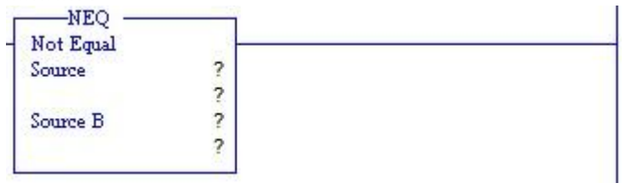
- Pada instruksi CTU, setiap terjadi perubahan *false-true* pada masukan, ACC dari counter akan bertambah satu
- Pada instruksi CTD, setiap terjadi perubahan *false-true* pada masukan
- Contoh program :



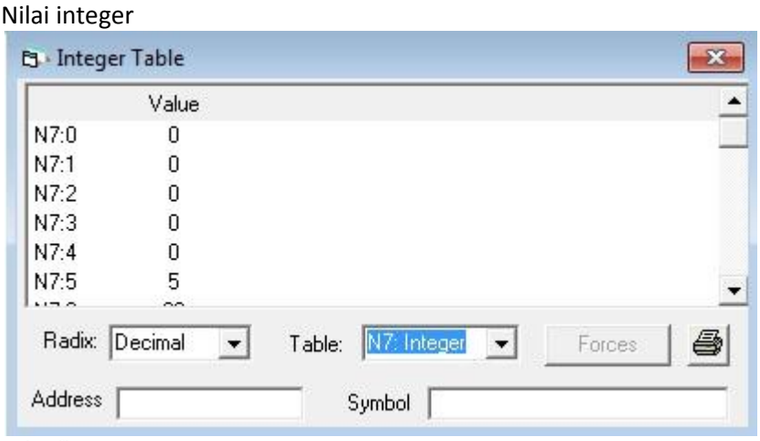
Compare Instructions

- Instruksi-instruksinya : LIM MEQ EQU NEQ LES GRT LEQ GEQ

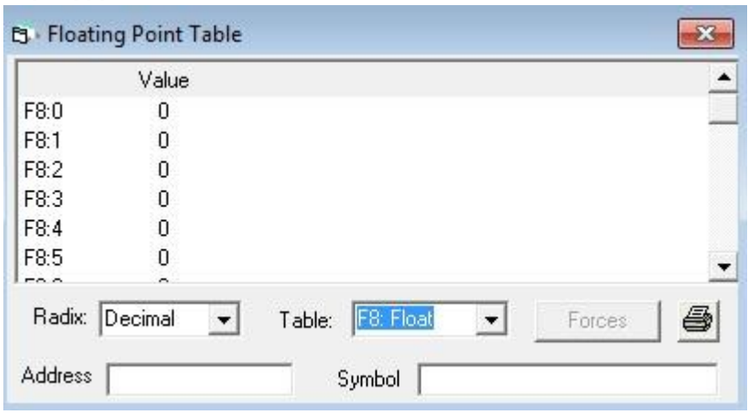
- Contoh :



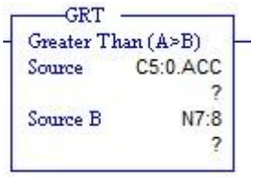
  - Nilai Source dapat berupa :
    - ACC dari counter atau timer
    - Nilai konstanta diketik langsung
    - Nilai integer



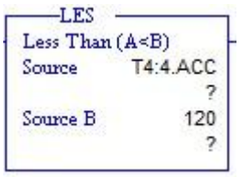
    - Nilai float



- Contoh :



Membandingkan apakah nilai ACC dari counter 0 (C5:0) lebih dari nilai N7:8



Membandingkan apakan nilai ACC dari timer 4 (T4:4) kurang dari 120

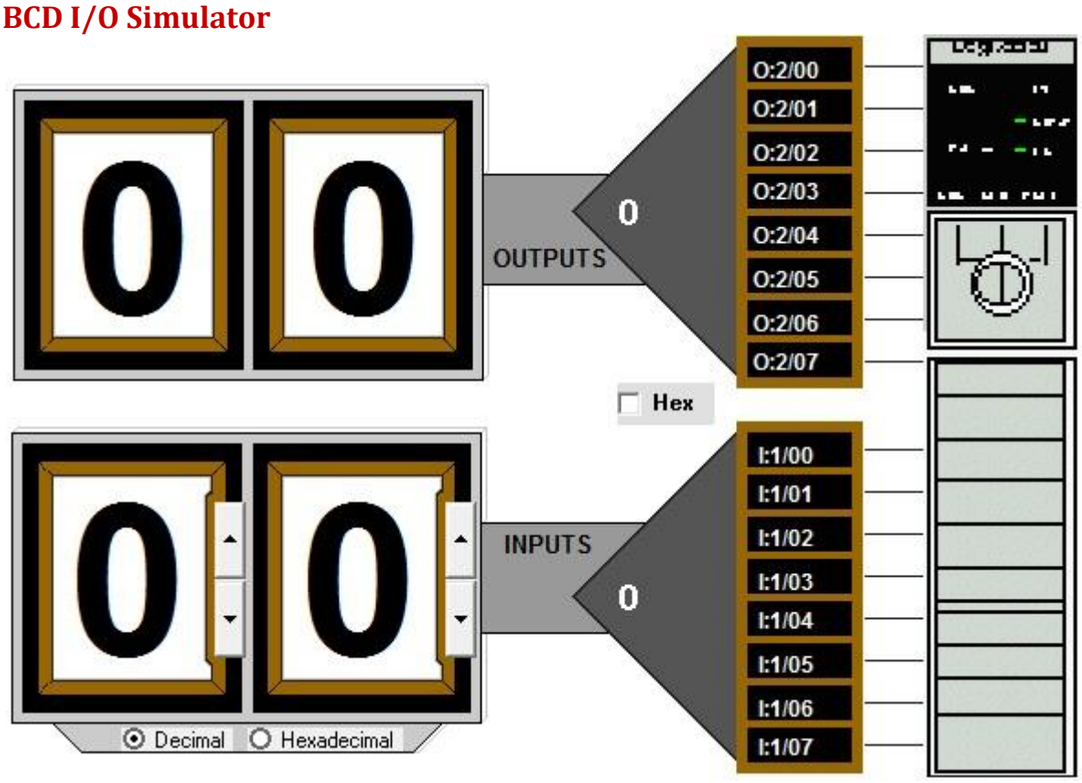
Compute Instructions

- Instruksi-instruksinya : 

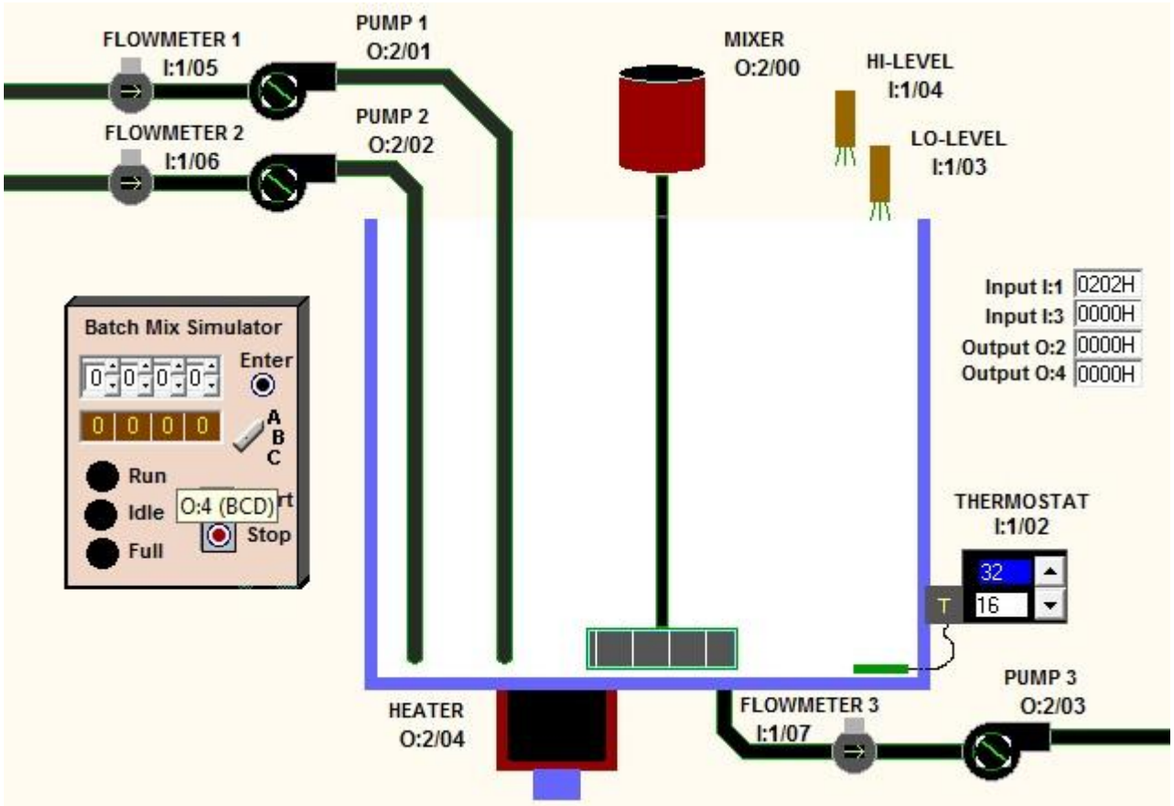
  - ADD : Addition
  - SUB : Subtraction
  - MUL : Multiplication
  - DIV : Division
  - SQR : Square Root
  - NEG : Negate
  - TOD : Convert to BCD
  - FRD : Convert from BCD

7 Segment Numeric Display

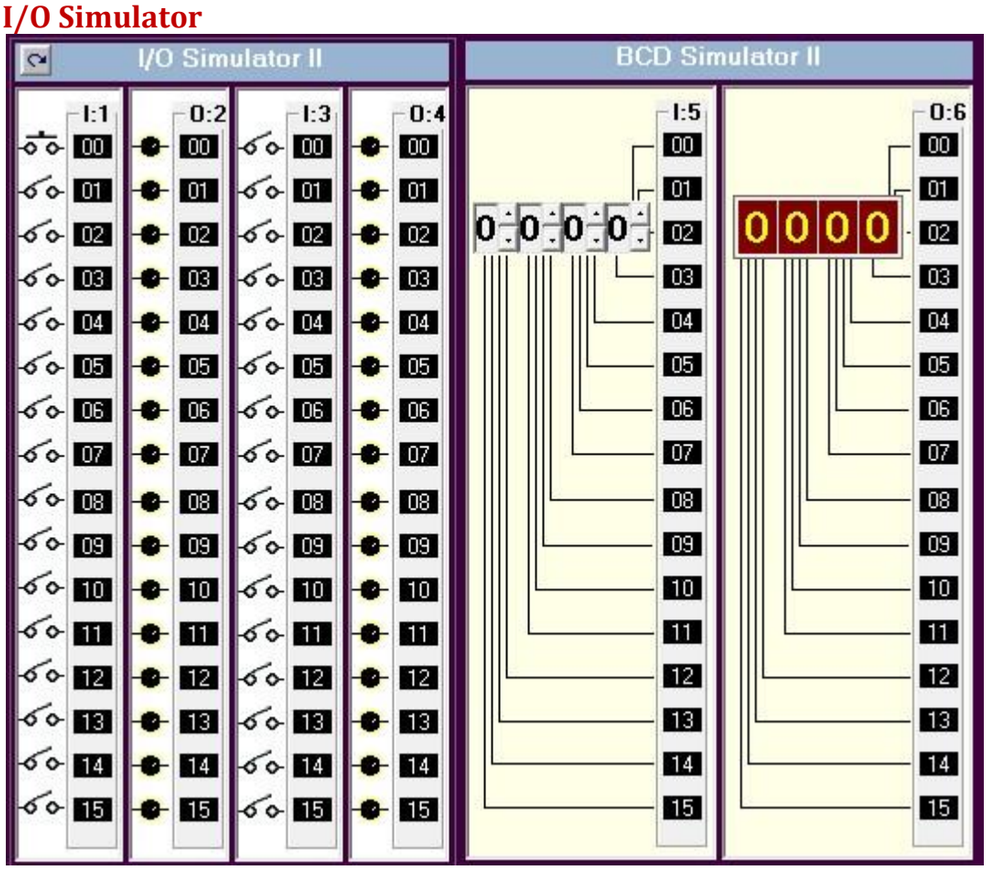
BCD I/O Simulator



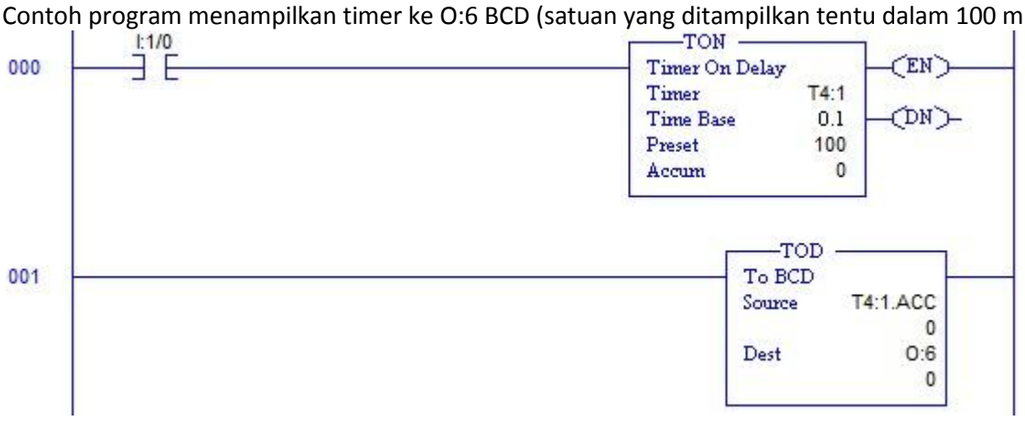
Batch Simulator



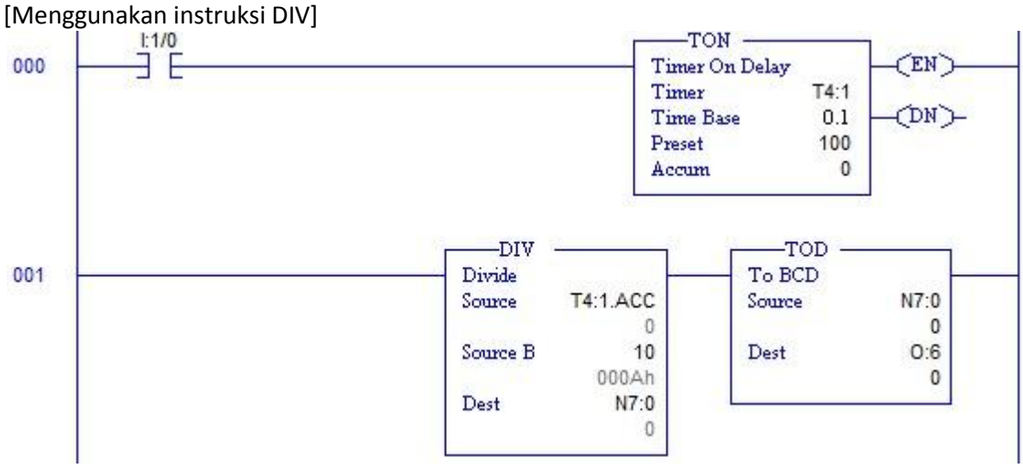
I/O Simulator



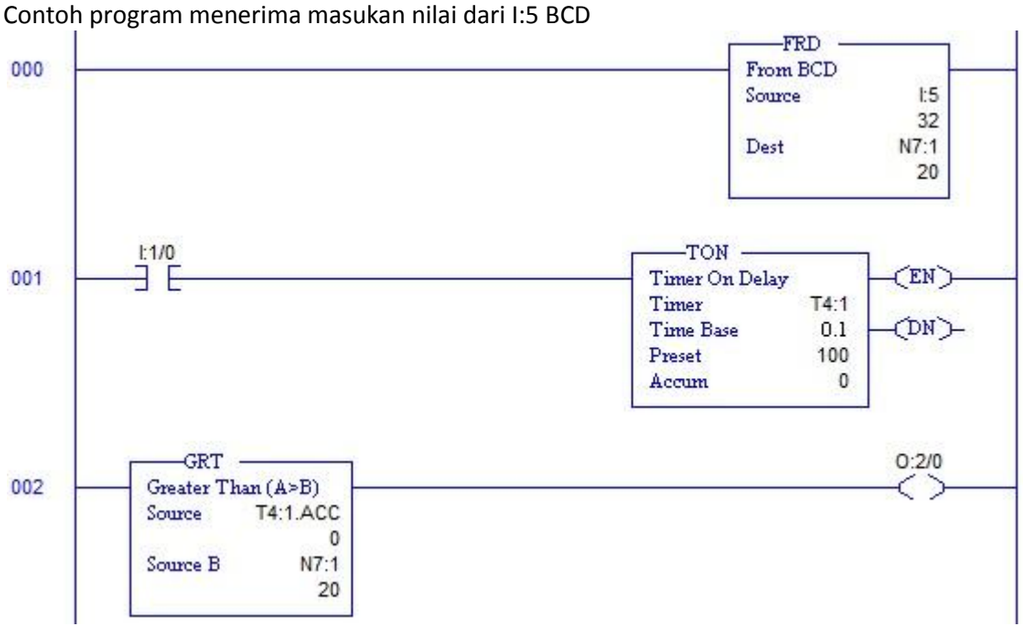
Contoh program menampilkan timer ke O:6 BCD (satuan yang ditampilkan tentu dalam 100 mikro detik)



Contoh program menampilkan timer ke O:6 BCD dalam satuan detik
[Menggunakan instruksi DIV]



Contoh program menerima masukan nilai dari I:5 BCD



Nilai timer T4:1 akan dibandingkan dengan nilai pada I:5 BCD. Jika lebih besar, maka akan menyalakan lampu O:2/0