

Perancangan Sederhana

I. Tujuan

- Mahasiswa dapat membuat program PLC sederhana.
- Mahasiswa dapat menggunakan instruksi *timer* pada konsep PLC.
- Mahasiswa dapat menggunakan instruksi *compare* pada konsep PLC.

II. Peralatan

- *Software* Do-More Designer
- *Software* Factory I/O
- Komputer atau laptop
- Koneksi internet

III. Dasar Teori

Pada modul ini kita akan mempelajari 2 instruksi dasar yang tersisa yaitu *timer* dan *compare*. Pada modul sebelumnya kita sudah mengetahui perbedaan dan persamaan antara instruksi dasar *counter* dan *timer*, maka pada modul ini kita dapat langsung mempelajari bagaimana cara menggunakan instruksi dasar *Timer*.

Timer adalah sebuah perintah *output* yang digunakan untuk menentukan interval yang dihitung dari suatu kondisi atau keadaan. Pada *timer* memiliki beberapa *output* yang biasanya dijadikan *input* pada *ladder logic* lain seperti:

1. C(nomor alamat).Done, contohnya C1.Done. kontak ini akan aktif bila *timer* telah berjalan selama waktu *preset* yang telah ditentukan. Biasanya C(nomor alamat).Done ini digunakan sebagai *input* digital.
2. C(nomor alamat).Acc, contohnya C1.Acc. kontak ini biasanya dijadikan sebagai *input* analog.

Timer yang paling sering dipakai pada PLC Do-more berjumlah 4 yang dibagi menjadi 2 sifat yaitu hitung mundur dan maju lalu pada masing-masing sifat juga memiliki 2 tipe lain.

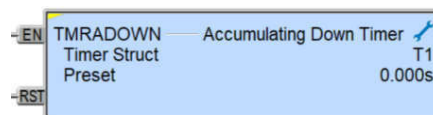
Pada sifat hitung mundur:

1. *Down Timer* (TMRDOWN)



Gambar 34 Instruksi TMRDOWN

2. *Accumulating Down Timer* (TMRADOWN)



Gambar 35 Instruksi TMRADOWN

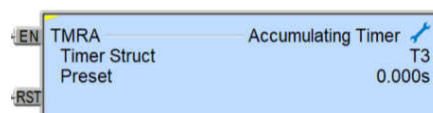
Pada Sifat hitung Maju:

3. *Up Timer* (TMR)



Gambar 36 Instruksi TMR

4. *Accumulating Up Timer* (TMRA)

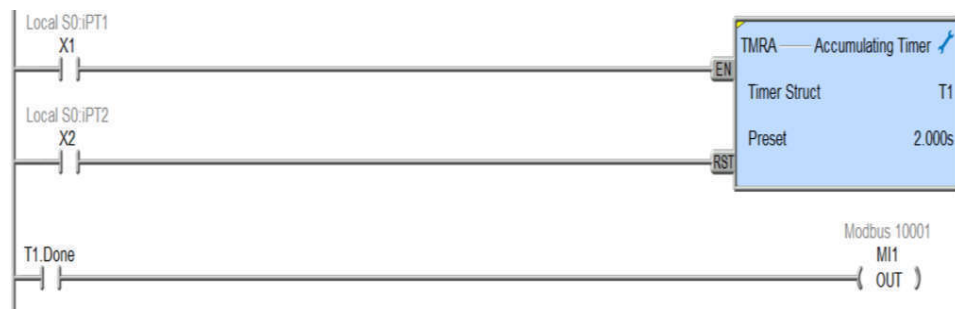


Gambar 37 Instruksi TMRA

Perbedaan 2 tipe ini adalah seperti yang terlihat gambar, pada tipe *accumulating* instruksi memiliki 2 kaki *input ladder logic* untuk reset dan EN sedangkan pada tipe lain hanya memiliki 1 kaki *input ladder logic* dengan kaki *input ladder logic* untuk EN dan reset digabung menjadi satu.

Cara kerja dari instruksi *Timer* pada contoh di bawah ini adalah, ketika EN pada *timer* (T1) mendapatkan *input* selama *preset* yang telah ditentukan (2s) maka *timer* akan mengaktifkan *contact-contact* (T1.Done)

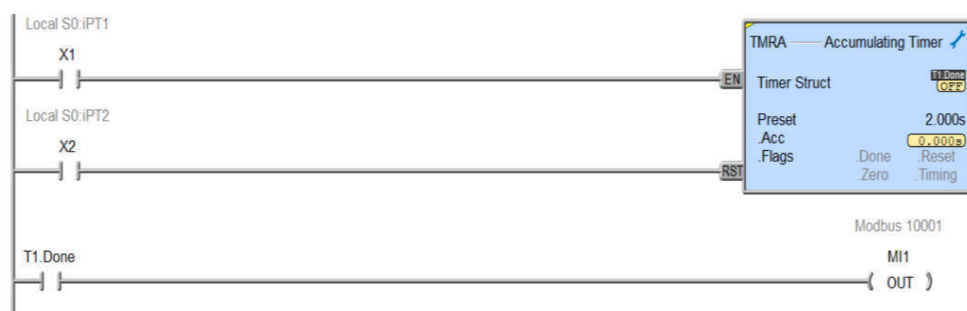
sehingga *output* (MI1) akan aktif, dan pada saat reset mendapatkan input dari MC2 maka *timer* akan mengulang kembali.



Gambar 38 contoh penggunaan instruksi *timer*

Keterangan :

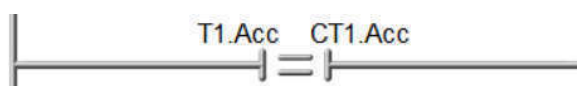
Timer akan aktif bila EN pada kondisi ON dan reset pada kondisi *OFF*. Ketika pertama kali dieksekusi *timer* akan mengukur preset dalam orde 0,1 detik.



Gambar 39 Tampilan instruksi *timer* pada saat program berjalan

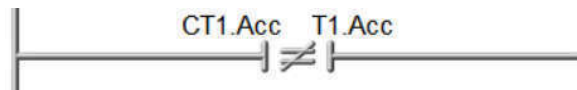
Selanjutnya kita akan mempelajari cara menggunakan instruksi *compare*. *Compare* adalah sebuah instruksi yang digunakan untuk membandingkan dua buah data. Fungsi *compare* sebenarnya memiliki fungsi yang hampir sama dengan fungsi *If Else* pada *programming language* lain yaitu ketika syarat *compare* terpenuhi maka kontak akan aktif. Pada Do-More memiliki 6 jenis instruksi *compare* yang paling sering dipakai yaitu:

1. Equal-To Relational Contact



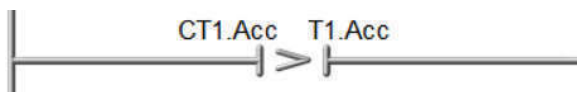
Gambar 40 Equal-To Relational Contact

2. Not-Equal-To Relational Contact



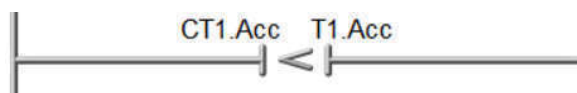
Gambar 41 Not-Equal-To Relational Contact

3. Greater-Than Relational Contact



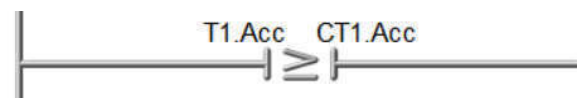
Gambar 42 Greater-Than Relational Contact

4. Less-Than Relational Contact



Gambar 43 Less-Than Relational Contact

5. Greater-Than-or-Equal-To Relational Contact



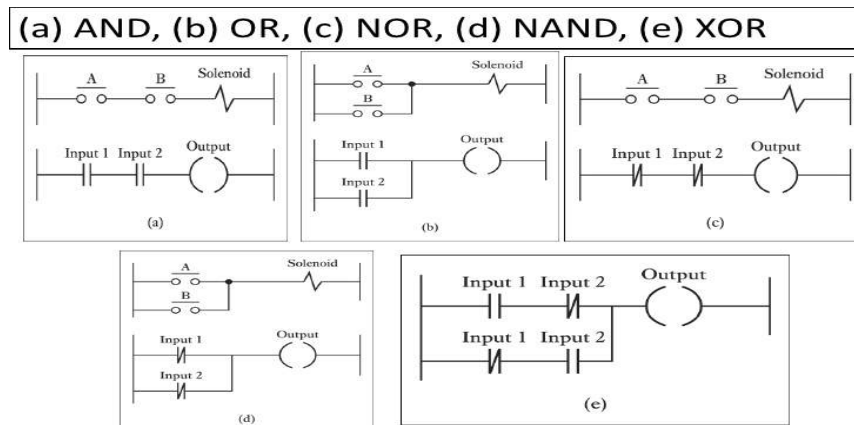
Gambar 44 Greater-Than-or-Equal-To Relational Contact

6. Less-Than-or-Equal-To Relational Contact



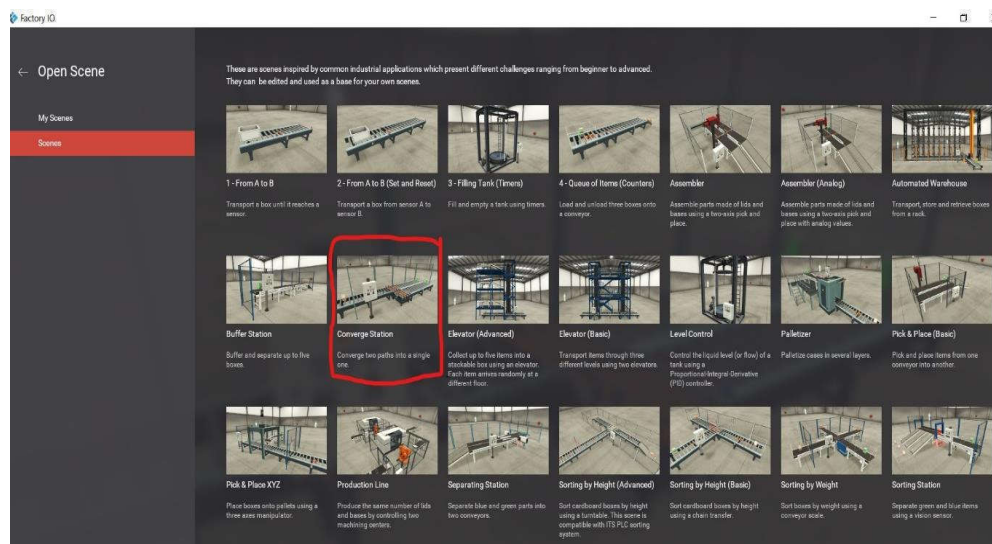
Gambar 45 Less-Than-or-Equal-To Relational Contact

Selain keenam instruksi *compare* di atas, *compare* juga dapat dilakukan pada rangkaian penempatan *ladder logic* seperti *AND*, *OR*, *NOR*, *NAND*, dan *XOR*. Rangkaian instruksi dapat dilihat pada gambar di bawah.



Gambar 46 Instruksi *Compare* pada *ladder logic*

IV. Tugas



Gambar 47 Tampilan skema *Converge Station* pada *Factory I/O*

Pilihlah skema *Converge Station* pada skema yang tersedia pada *Factory I/O*. Kemudian buatlah agar konveyor dapat berjalan saat tombol *start* ditekan dan berhenti saat tombol *stop* ditekan dengan keadaan konveyor hanya bisa bergerak ketika *switch* pada kondisi auto, *LCD counter* akan menghitung jumlah *Box* yang turun dan akan Kembali ke 0 ketika tombol reset ditekan, programlah agar barang dari kedua jalur agar dapat menuju *remove*, lalu sebagai tindak pengamanan buatlah agar seluruh kegiatan akan berhenti bila *emergency stop* ditekan. (petunjuk: pada skema terdapat 2

transfer konveyor dengan 2 sensor pada masing-masing transfer konveyor tersebut dan juga terdapat 1 sensor yang berada di akhir transfer konveyor. Gunakan lah seluruh sensor dan konveyor dengan baik dan Anda dapat menggunakan sensor yang berada di akhir transfer konveyor dengan instruksi *counter* yang telah dijelaskan sebelumnya untuk membuat LCD *counter* dapat menghitung jumlah *Box*. Gunakanlah instruksi *timer* untuk mengatur arah dan lamanya transfer konveyor bekerja dan gunakanlah instruksi *compare* untuk mengatur perpindahan transfer konveyor agar barang pada kedua transfer konveyor tidak bertabrakan)