

# SoMachine Basic Manuals

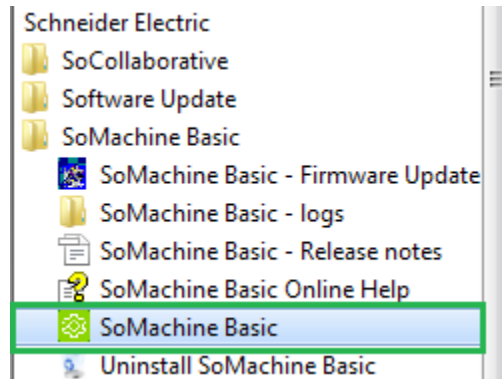
## Daftar isi

Membuat Project Baru.....	3
Membuka Project .....	5
System Settings.....	11
Programing Language.....	13
Ladder.....	13
Instructions List .....	13
Addressing Format.....	14
Address to object.....	15
Simulator.....	15
Symbols.....	18
Export Symbol ke CSV File.....	19
Import Symbols.....	21
Comment.....	23
Program Structure.....	24
Task.....	25
User-defined function.....	26
User-defined function block.....	29
Programming Rungs.....	32
Acces Bits pada Memory Word.....	33
Animations Table.....	34
Search dan Replace.....	36
Cross Reference.....	37
Memory Consumption.....	38
Operation Block.....	39
Instructions.....	42
Contacts dan Coils.....	42
Exclusive OR Operator.....	45
Rising dan Falling.....	47
Timers.....	49
Counters.....	50
Comparison.....	52
Jump.....	53
Conditional Elements.....	54
Loop Elements.....	56
Drum Assistant.....	58
Shift Bit Register.....	61
Read Var.....	63
Write Var.....	66
Write Read Var.....	68
Communication.....	70
Ethernet Port .....	70
Serial Port.....	72
Modbus TCP.....	73
Modbus Serial.....	74
Mengatur Jam RTC .....	75
Simulasi SoMachine Basic ke Ecostruxure SCADA Expert.....	76

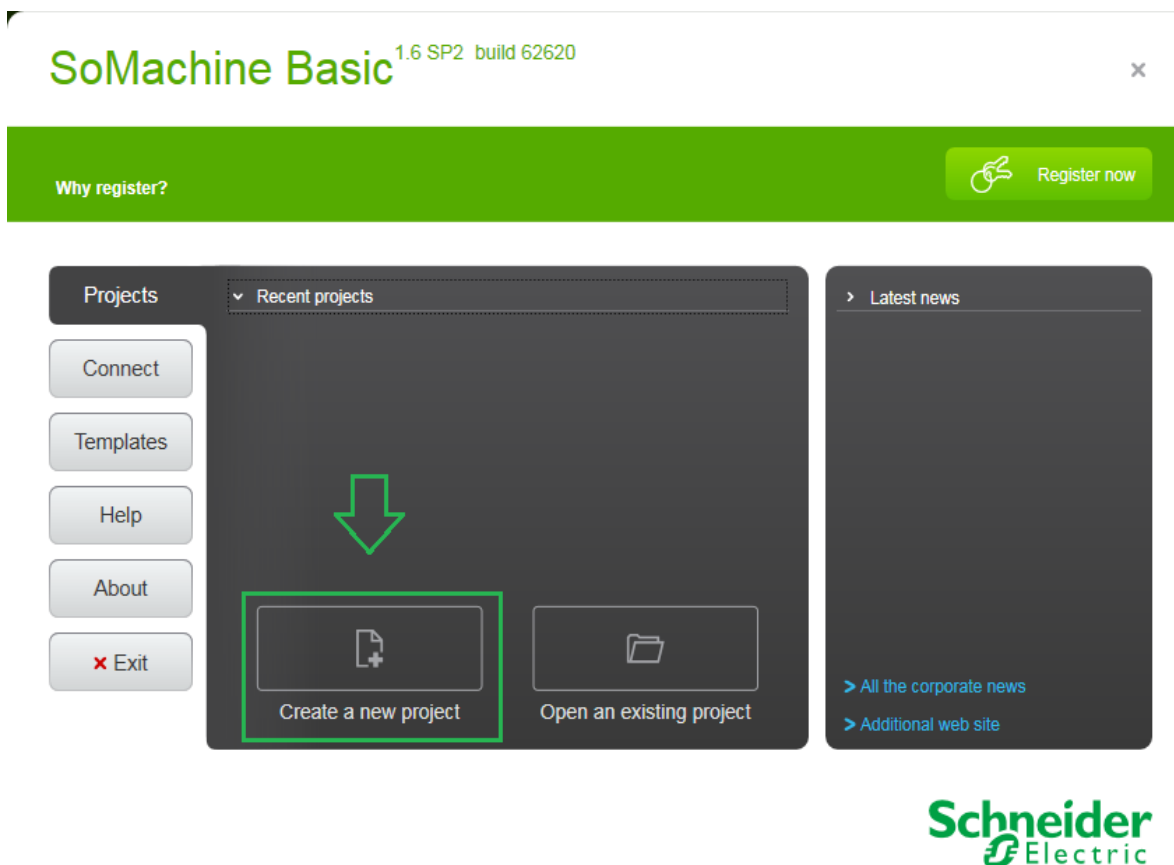
## Membuat Project Baru

Untuk membuat project baru menggunakan software SoMachine Basic langkah – langkahnya adalah sebagai berikut :

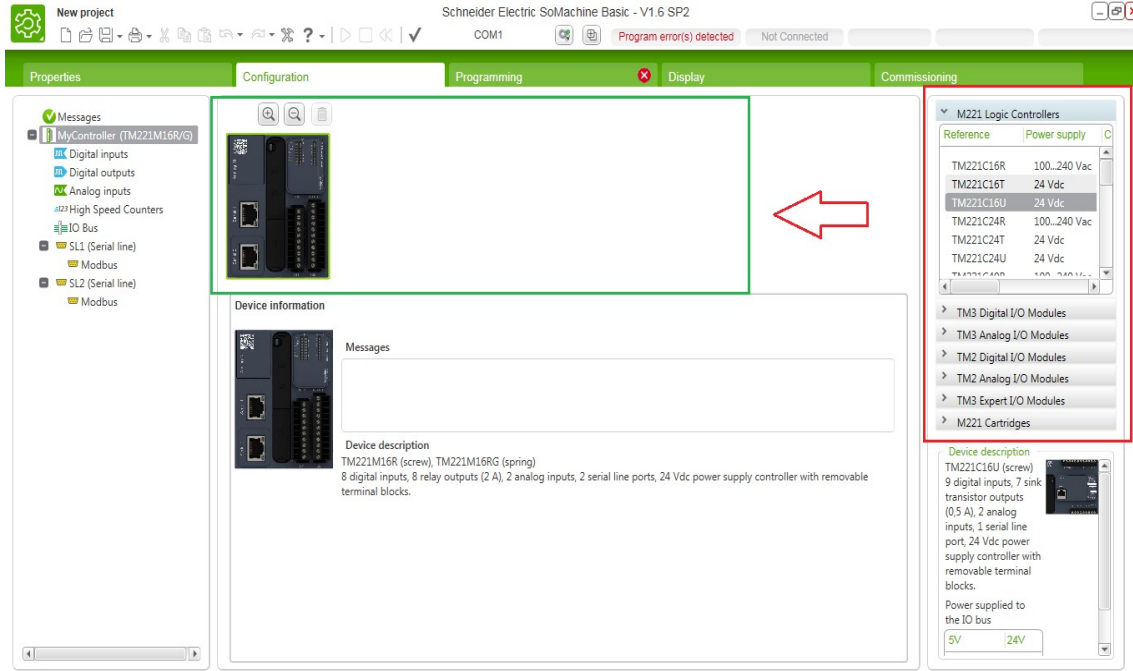
1. Double Klik pada ikon SoMachine Basic untuk mulai membuka software.



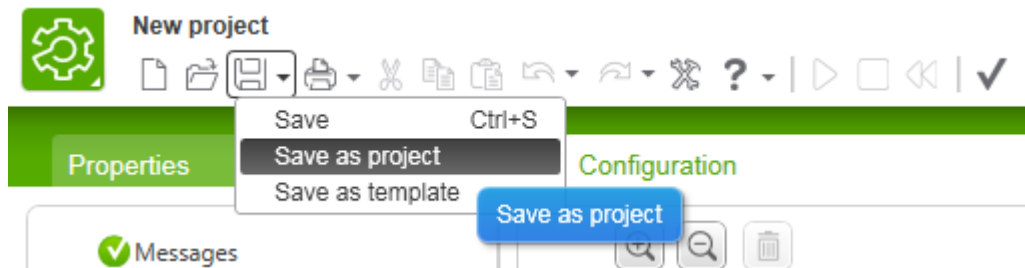
2. Pilih Create New Project untuk membuat project baru.



3. Pilih Type PLC, expand I/O module yang mau digunakan, drag object untuk menambahkannya ke hardware configuration.



4. Klik Save ikon pada pojok kiri atas untuk menyimpan project yang telah dibuat.



## Membuka Project

Langkah-langkah untuk membuka project yang pernah dibuat adalah sebagai berikut :

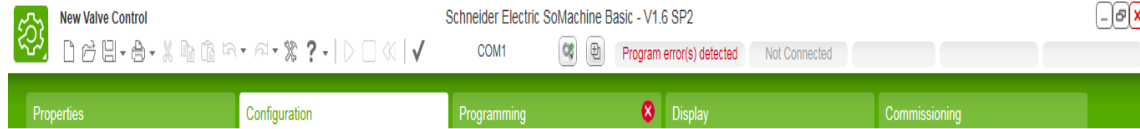
1. Klik ikon Open Existing Project yang ada di pojok kiri atas atau klik SoMachine Basic ikon untuk membuka halaman awal.



2. Klik Open Existing Project atau lihat list Recent Projects.

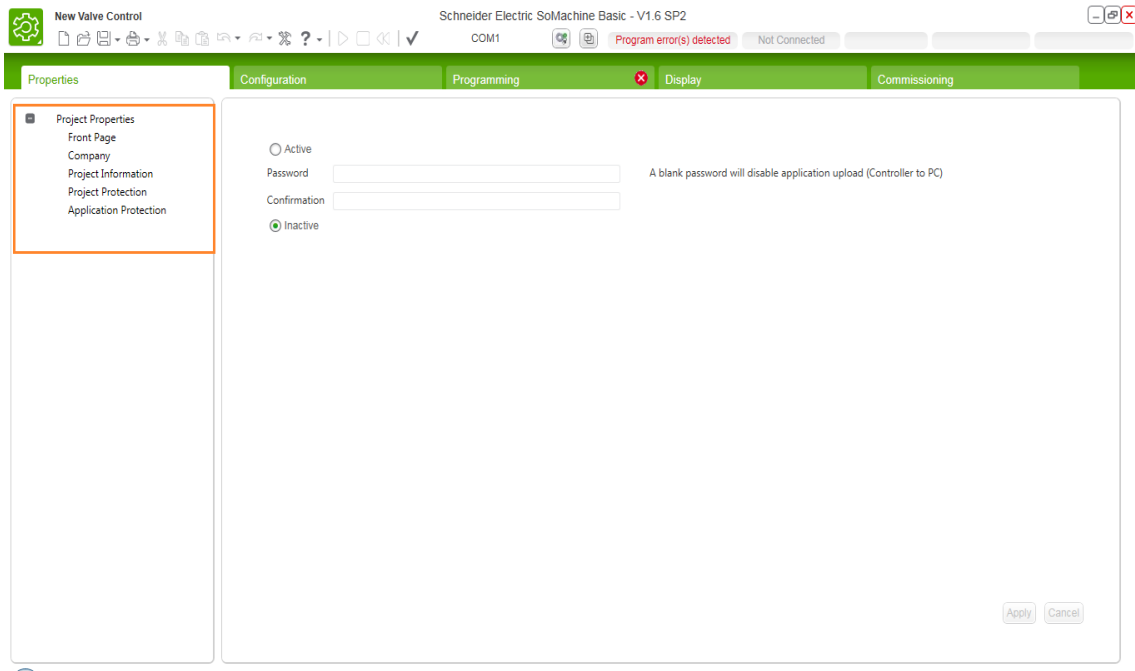


## Menu



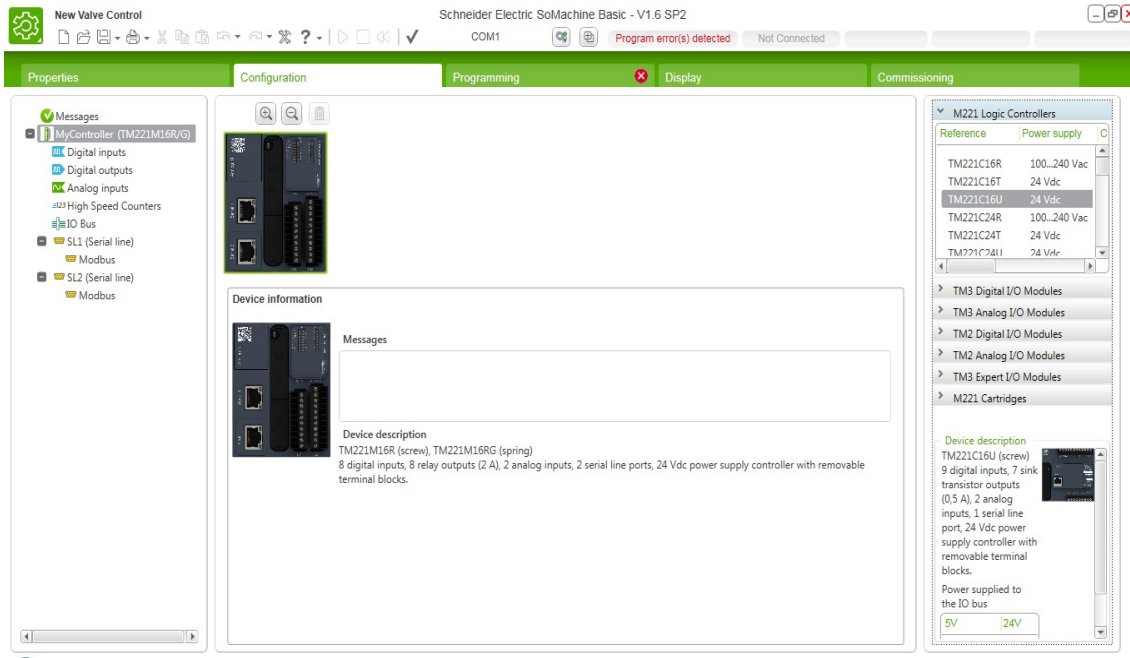
Pada Software SoMachine Basic terdapat beberapa menu utama, menu tersebut antara lain :

- *Properties* : berisi mengenai informasi tentang project yang dibuat, serta protection yang dapat dipakai untuk melindungi program diutak – atik orang lain.



- Front Page : digunakan untuk menambahkan informasi tentang pembuat project.
- Company : digunakan untuk menambahkan informasi perihal siapa owner dari project yang dibuat.
- Project Information : berisi informasi umum mengenai project yang dibuat.
- Project Protection : untuk membatasi siapa yang boleh membuka serta memodifikasi project yang telah dibuat.
- Application Protection : untuk memberikan protection saat project mau diupload ( dari PLC ke PC/laptop ).

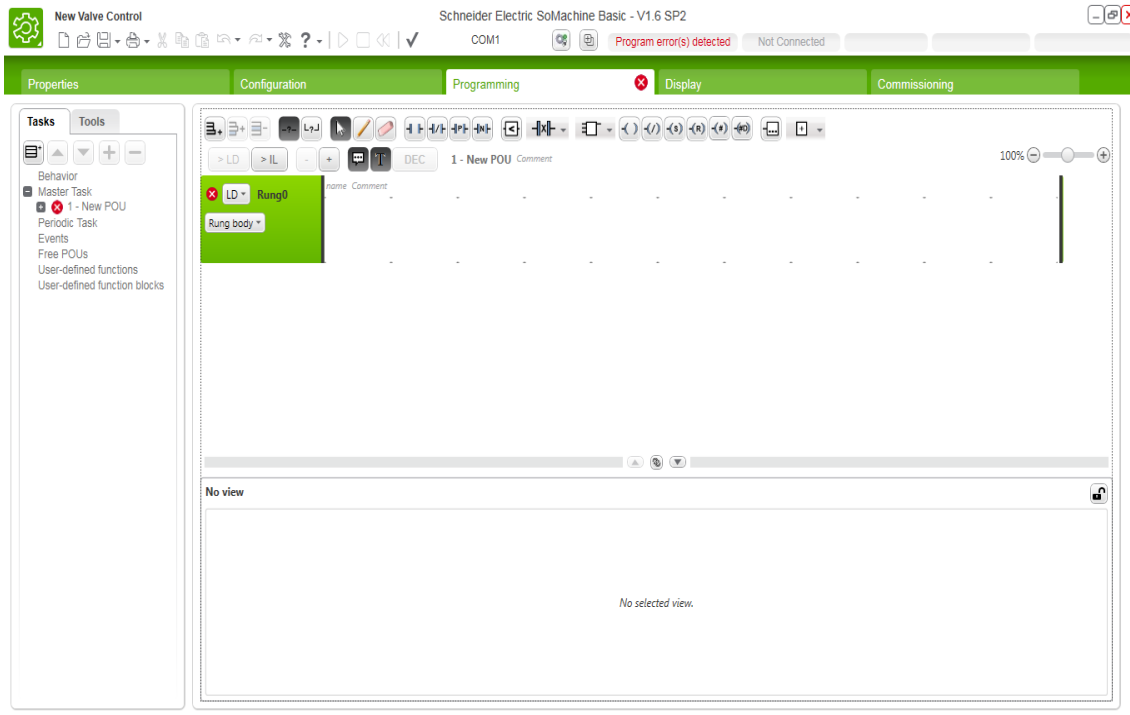
- Configuration : digunakan untuk mengkonfigurasi hardware PLC yang akan digunakan pada project. Termasuk juga memilih additional I/O module , pengaturan parameters pada hardware.



Untuk mengubah tipe PLC yang digunakan pada project maka dapat memilih tipenya terlebih dahulu di sidebar sebelah kanan, lalu drag and drop ke PLC yang digunakan sekarang. Cara yang sama juga dapat digunakan untuk menambahkan additional module (digital/analog).

Di sidebar sebelah kiri digunakan jika ingin mengatur mengenai Digital, Analog, High Speed Counters serta pengaturan komunikasi yang ingin digunakan.

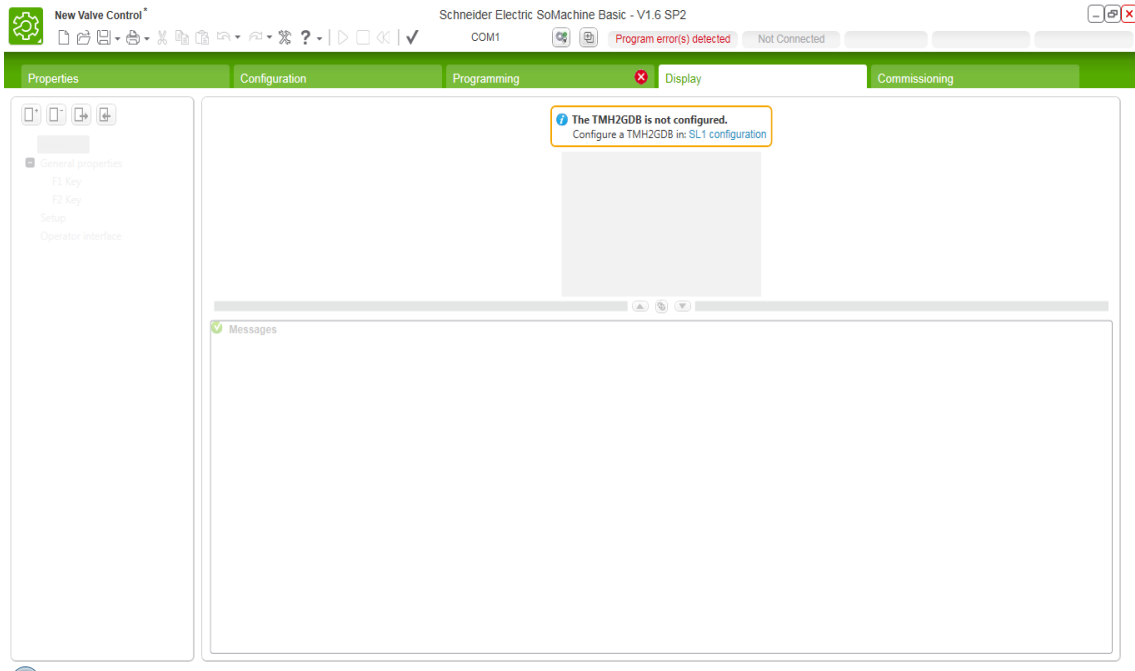
- Programming : digunakan untuk menulis, mengedit program pada project. Program dapat ditulis menggunakan bahasa Ladder atau Instruction List.



Selain digunakan untuk mengedit, menulis program pada programming tab ini juga dapat digunakan untuk memonitor/ debugging program yang telah dibuat. Debugging dapat dilakukan dengan menghubungkan PC ke PLC atau melalui simulation mode. Jika melalui simulation mode maka ada beberapa instruction yang tidak bisa dimonitor.



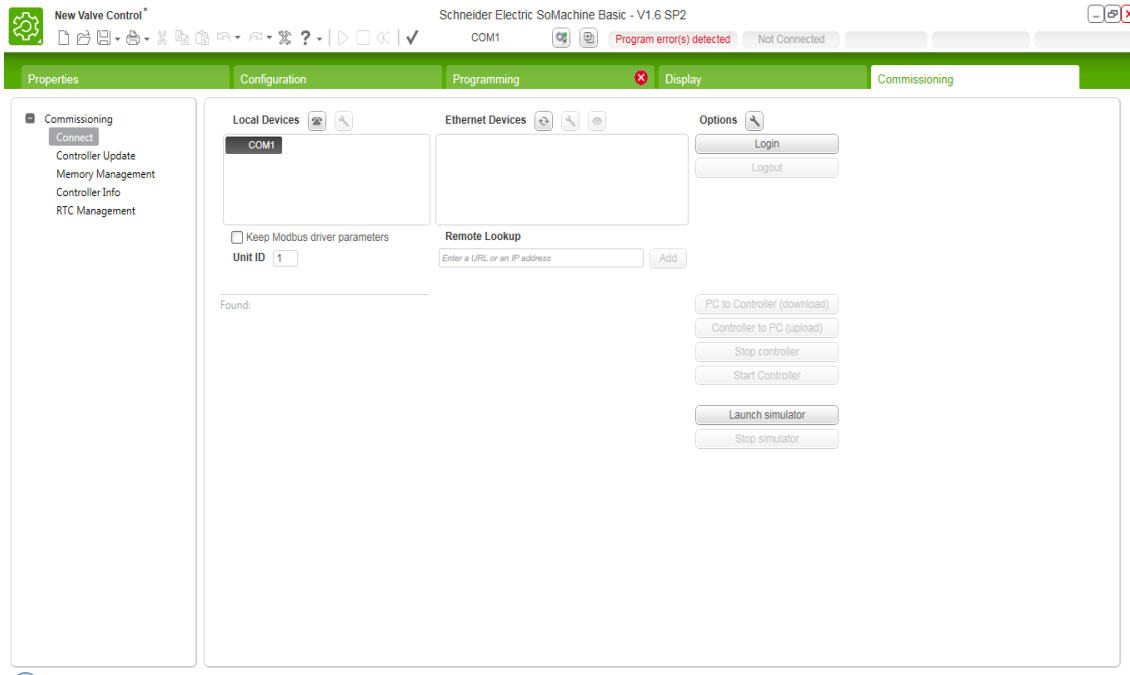
- Display : digunakan untuk mengatur dan memprogram display yang terhubung dengan PLC, display tersebut yaitu TMH2GDB.



Tampilan display TMH2GDB yang dapat diprogram melalui software SoMachine Basic.



- Commissioning : digunakan untuk mendownload serta mengupload project pada hardware PLC jika program telah selesai dibuat.



Koneksi yang digunakan untuk mentransfer program dapat melalui mode Serial, Ethernet, atau USB. Selain untuk mentransfer program, menu ini juga bisa digunakan untuk mengupdate firmware plc, mengatur jam RTC pada PLC.

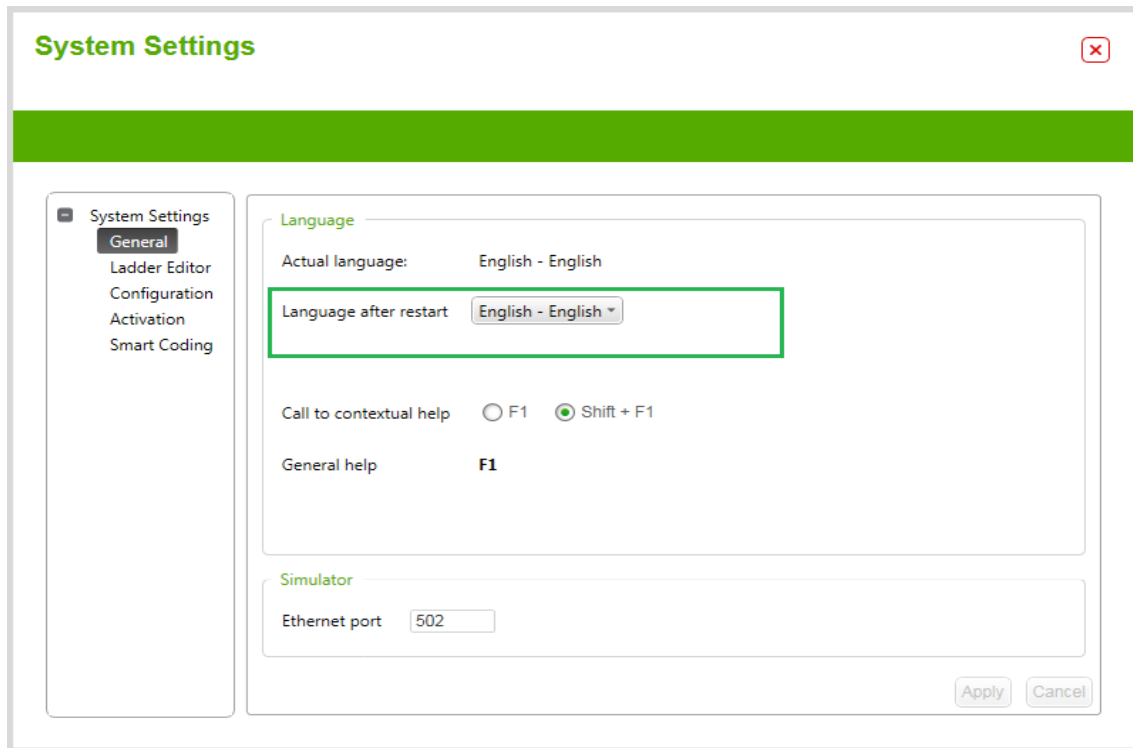
## System Settings

System Setting digunakan untuk mengatur hal - hal lebih lanjut mengenai software yang digunakan. Langkah-langkah untuk masuk ke menu system setting adalah sebagai berikut :

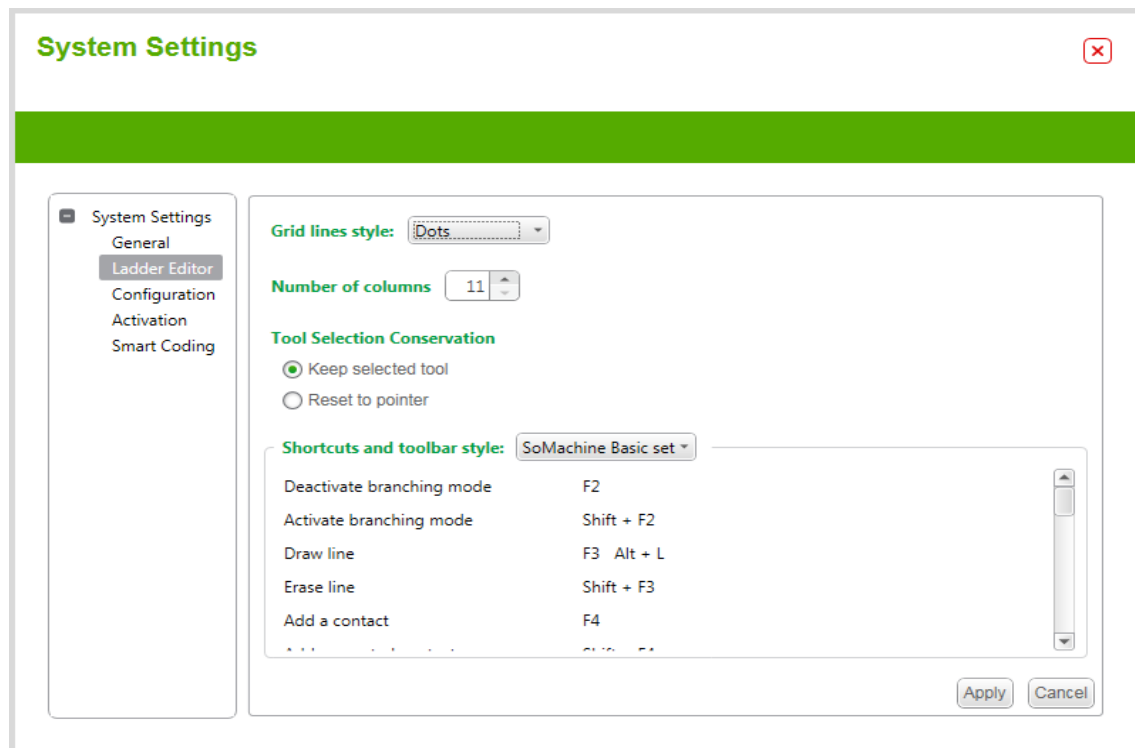
1. Klik ikon Setting di pojok kiri atas.



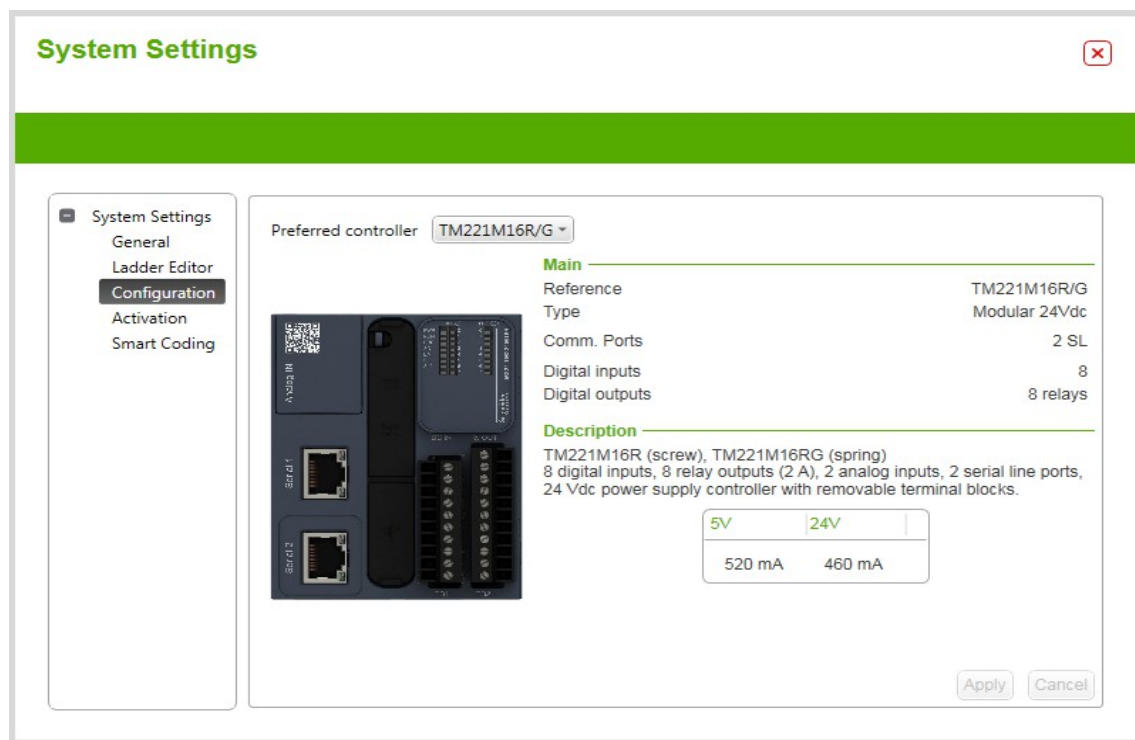
2. Klik pada bagian General untuk mengatur bahasa yang digunakan pada software. Bahasa akan aktif setelah software di-restart.



3. Klik pada Ladder Editor untuk mengatur jumlah column, shortcuts yang digunakan.



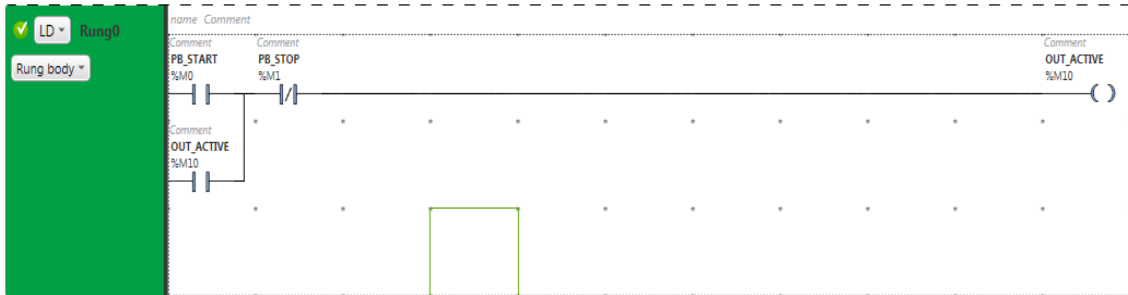
4. Klik Configuration untuk mengatur default PLC yang akan digunakan saat pertama kali membuat project.



## Programming Language

Bahasa pemrograman yang biasa digunakan untuk memprogram PLC ada beberapa macam. Pada SoMachine Basic hanya ada 2 macam bahasa yang dapat dipilih yaitu :

- **Ladder** : merupakan bahasa yang paling umum dipakai, tampilanya mirip dengan diagram listrik, mudah, jelas dilihat saat proses debugging/monitoring program.



Ladder terdiri dari Rung dan Line, bagian kiri dari sebuah rung disebut input sedangkan bagian kanan pada rung disebut Output.

- **Instructions List** : merupakan bahasa berupa list – list instruksi yang saling terhubung, bahasa ini mirip dengan bahasa mnemonic console / computer machine code.



Instructions List menggunakan metode store, load dan accumulator. Keunggulan bahasa ini yaitu lebih ringkas dan cepat saat proses scanning program.

## Addressing Format

Addressing Format merupakan cara penulisan alamat memori pada saat memprogram. PLC M221 Controller menggunakan format pengalamatan sebagai berikut :

`%<Type>[<Identifier>]<Location>`

contoh : `%MW10`

M = Type

W = Identifier

10 = Location

Type yang dapat digunakan terbagi menjadi :

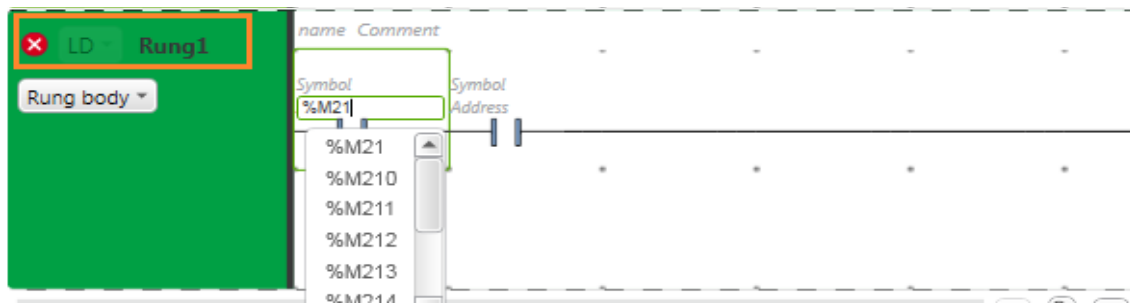
- **I** : merupakan alamat inputan fisik dari PLC baik itu yang terpasang dengan CPU atau additional module(expand io).Contoh : `%I0.0`
- **K** : merupakan alamat memori konstan. Nilai dari type memory ini adalah fixed artinya tidak dapat dirubah oleh program.biasanya nilai ini disetting sebelum sebuah program didownload ke PLC.Ccontoh : `%KW20`
- **M** : merupakan alamat internal memori yang nilainya dapat dirubah menggunakan program.Ccontoh : `%MW100`
- **Q** : merupakan alamat output fisik dari PLC baik itu yang terpasang dengan CPU atua additional module(expand io).Contoh : `%Q2.0`
- **S** : merupakan alamat memori system yang digunakan untuk mendeteksi, memonitor keadaan CPU PLC dan fitur lainnya tentang system.Ccontoh : `%SW100`

Identifier termbagi menjadi :

- **(none)** : merupakan identifier yang berupa nilai single bit, nilai valuenya 0 atau 1.Ccontoh : `%Q3.1`
- **W** : merupakan identifier yang berisi nilai Word , rentang nilainya yaitu 0 ~ 65535.Ccontoh : `%QW0.0`
- **D** : merupakan identifier yang berisi nilai Double Word , rentang nilainya yaitu 0 ~ 4294967295.Ccontoh : `%MD200`
- **F** : merupakan identifier yang berisi nilai Floating Point, rentang nilainya yaitu 0 ~ 65535.Ccontoh : `%MF30`

## Address to object

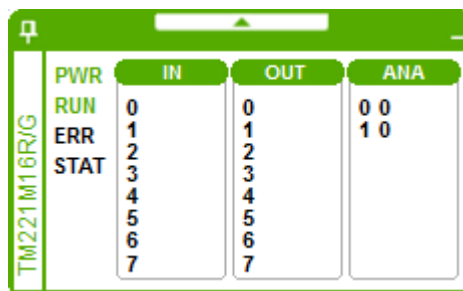
Saat membuat program, object yang dibuat harus dipasangkan dengan I/O Address atau Memory Address. Jika tidak maka program akan error sehingga tidak akan dapat dicompile atau didownload.



Untuk menambahkan address pada object maka caranya dengan double klik pada tulisan Address pada object yang akan ditambahkan.

## Simulator

Simulator digunakan untuk mensimulasikan program yang telah dibuat tanpa menggunakan hardware PLC. Dengan menggunakan simulator ini maka proses pengecekan suatu program menjadi lebih cepat dan mudah. Walaupun ada beberapa instruction yang tidak dapat disimulasikan seperti PID, akan tetapi untuk instruction dasar bisa tercover.



	PWR	IN	OUT	ANA
0	0	0	0	0 0
1	1	1	1	1 0
2	1	1	1	1 0
3	1	1	1	1 0
4	1	1	1	1 0
5	1	1	1	1 0
6	1	1	1	1 0
7	1	1	1	1 0

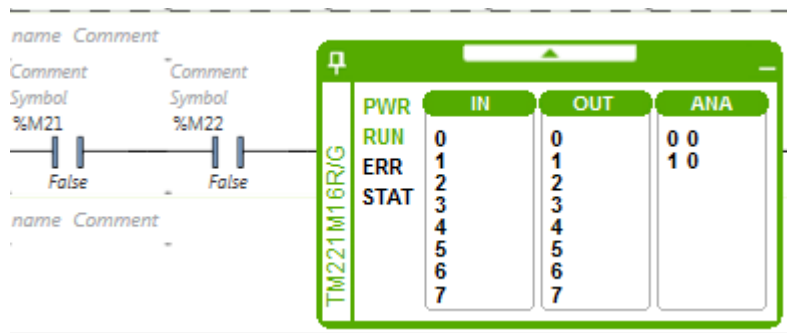
Simulator ini dapat digunakan untuk merubah nilai dari digital input/output serta analog input/output. Dibagian kiri simulator window terdapat status dari PLC Simulatornya (Run / Err). Simulator Window ini dapat bertambah racknya sesuai dengan hardware configuration yang diatur.

Untuk menjalankan simulator maka langkah-langkahnya adalah sebagai berikut :

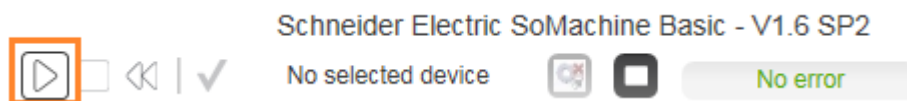
1. Klik ikon Launch Simulator atau CTRL + B untuk menjalankan simulator.



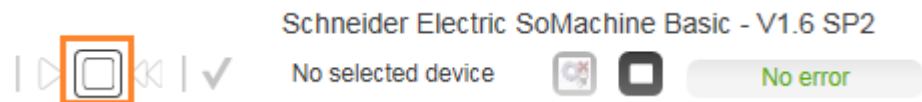
2. Tunggu sampai Simulator Window muncul. Simulator telah berjalan tetapi PLC Simulatornya masih stop, ditandai dengan tulisan Run yang berkedip – kedip.



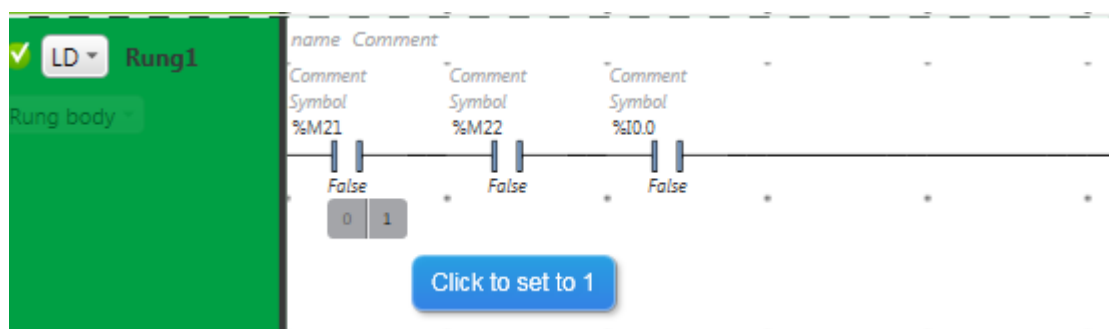
3. Untuk Starting PLC Simulator maka klik ikon Start Controller atau CTRL + M.



4. Untuk Stop PLC Simulator maka klik ikon Stop Controller atau CTRL + L

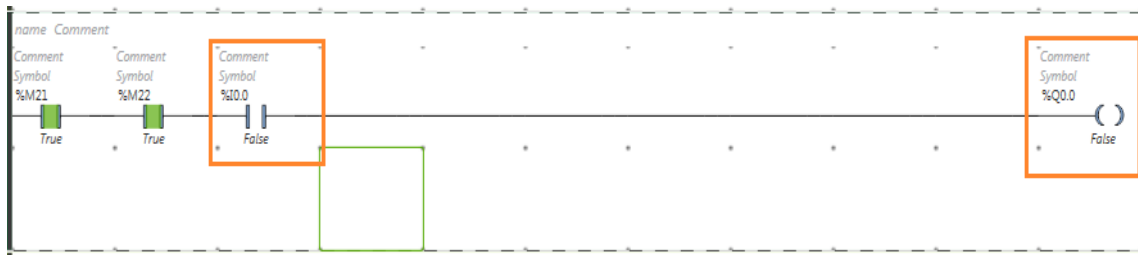


5. Untuk mengubah nilai address single bit maka arahkan cursor ke samping object nanti akan muncul nilai 0 dan 1 , pilih salah satu untuk mengubah.





6. Untuk program dengan address single bit I/O maka ada 2 cara untuk merubah nilainya saat simulasi, dapat dengan mengarahkan cursor ke object (Force Mode F0, F1) atau menggunakan simulator window dengan klik pada angka di kolom in atau out.



7. Untuk mengubah nilai pada address analog maka double klik di angka analog yang akan dirubah, kemudian akan muncul dialog Set Analog Window. Ubah nilai dari analog yang dipilih, pengubahan dapat dilakukan pula menggunakan slider.

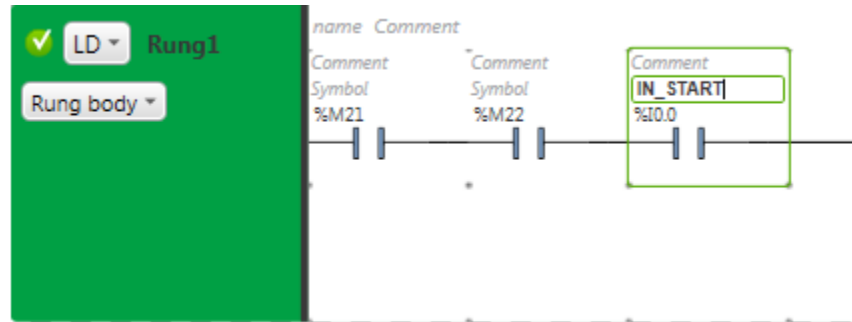
Inputs	Change Value
%IW0.0	126
%IW0.1	0

PWR	IN	OUT	ANA
0	0	0	126
1	1	1	0
2	2	2	
3	3	3	
4	4	4	
5	5	5	
6	6	6	
7	7	7	

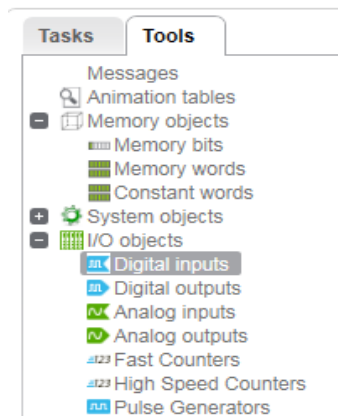
## Symbols

Symbol digunakan untuk memberikan nama pada setiap object yang telah dibuat pada saat memprogram. Symbol memudahkan identifikasi object pada saat debugging atau pengecekan program karena symbol akan lebih mudah dilihat.

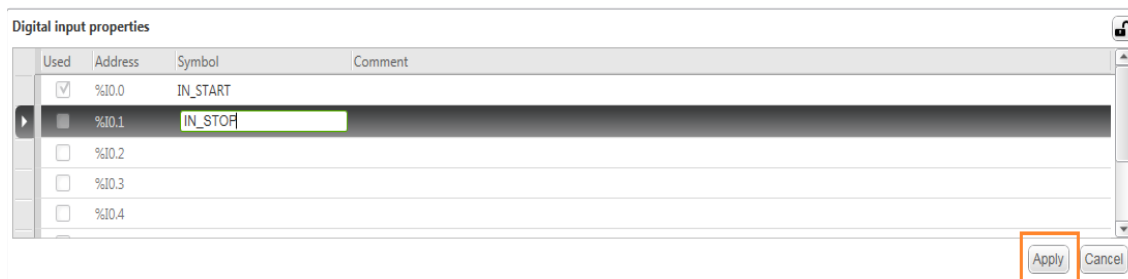
Untuk menambahkan symbol pada object, double klik pada tulisan symbols pada object lalu isikan nama sesuai keinginan. Pada saat menulis symbol tidak boleh menggunakan space/spasi, spasi dapat diganti menggunakan underscore.



Jika ingin melakukan perubahan/penambahan symbol secara cepat, terstruktur maka dapat masuk ke menu Tools dan pilih address mana yang akan dirubah / ditambahkan symbol.



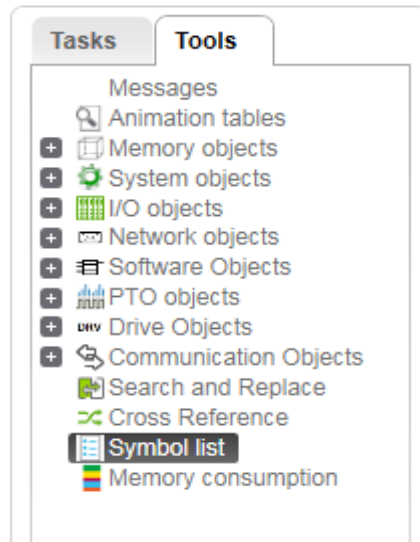
Klik Apply untuk menyimpan perubahan pada symbol.



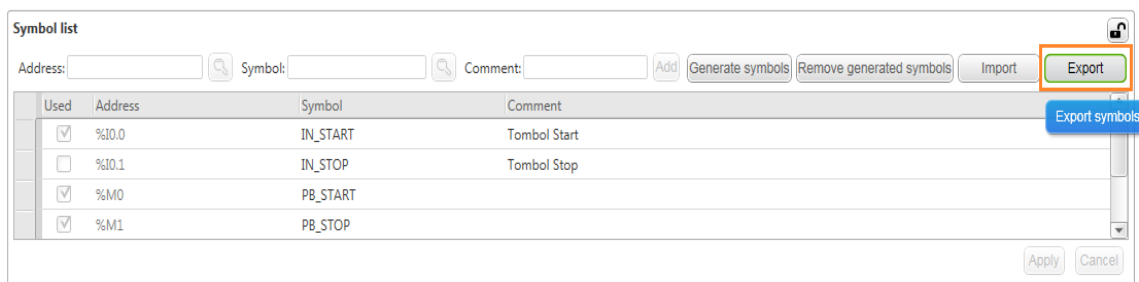
## Export Symbol ke CSV File

Untuk mengexport symbol serta comment yang telah dibuat ke dalam file CSV langkah-langkahnya adalah sebagai berikut :

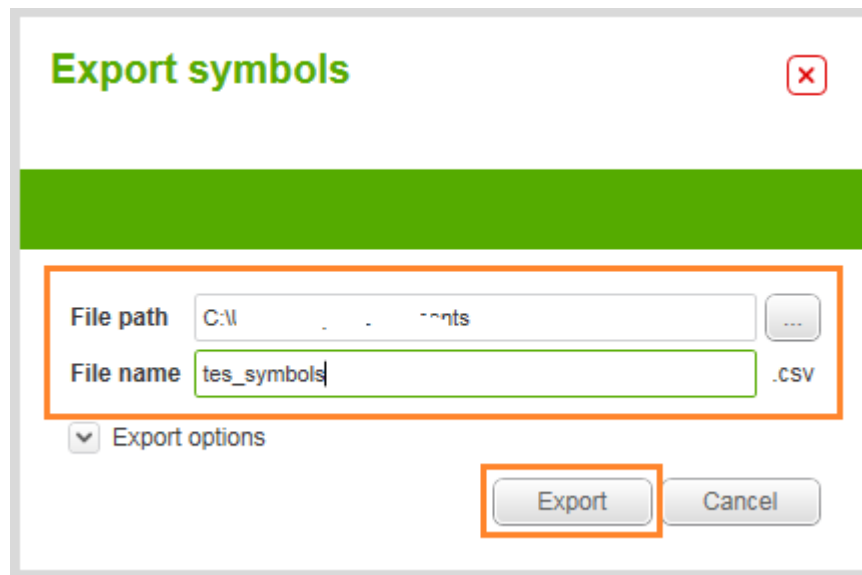
1. Buka Tools lalu pilih Symbol List.



2. Pilih Export button pada window.



- Isikan nama file serta alamat folder untuk menyimpan hasil export.



- Buka folder tempat menyimpan file, kemudian cari file bernama tes\_symbols.csv



- Buka file untuk melihat hasil exportnya.

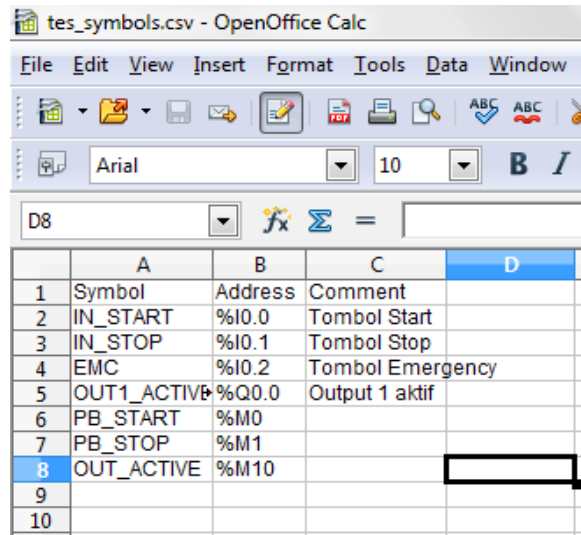
The screenshot shows the OpenOffice Calc spreadsheet titled "tes\_symbols.csv - OpenOffice Calc". The spreadsheet has a menu bar (File, Edit, View, Insert, Format, Tools, Data, Window) and a toolbar. The active cell is A1, which contains the text "Symbol". The spreadsheet data is as follows:

	A	B	C	D
1	Symbol	Address	Comment	
2	IN_START	%I0.0	Tombol Start	
3	IN_STOP	%I0.1	Tombol Stop	
4	PB_START	%M0		
5	PB_STOP	%M1		
6	OUT_ACTIVE	%M10		
7				
8				

## Import Symbols

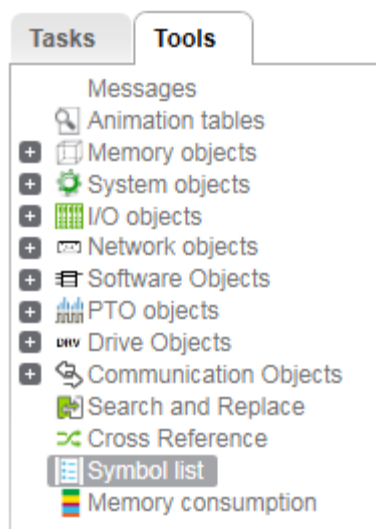
Untuk import symbol dan comment dari file CSV ke Somachine Basic langkah-langkahnya adalah sebagai berikut.

1. Siapkan file yang akan di import. Pastikan bahwa penulisan address dan symbol tidak salah.

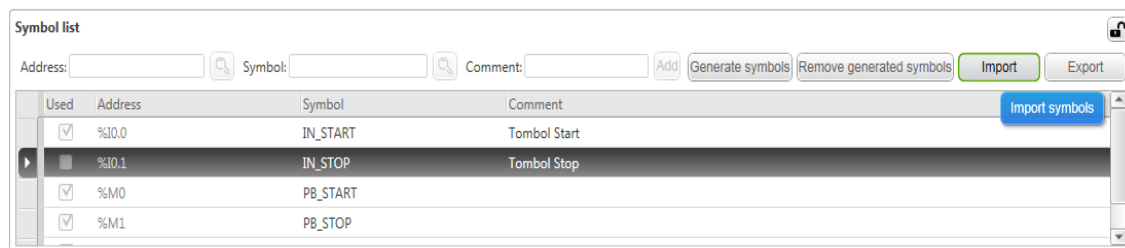


	A	B	C	D
1	Symbol	Address	Comment	
2	IN_START	%I0.0	Tombol Start	
3	IN_STOP	%I0.1	Tombol Stop	
4	EMC	%I0.2	Tombol Emergency	
5	OUT1_ACTIVE	%Q0.0	Output 1 aktif	
6	PB_START	%M0		
7	PB_STOP	%M1		
8	OUT_ACTIVE	%M10		
9				
10				

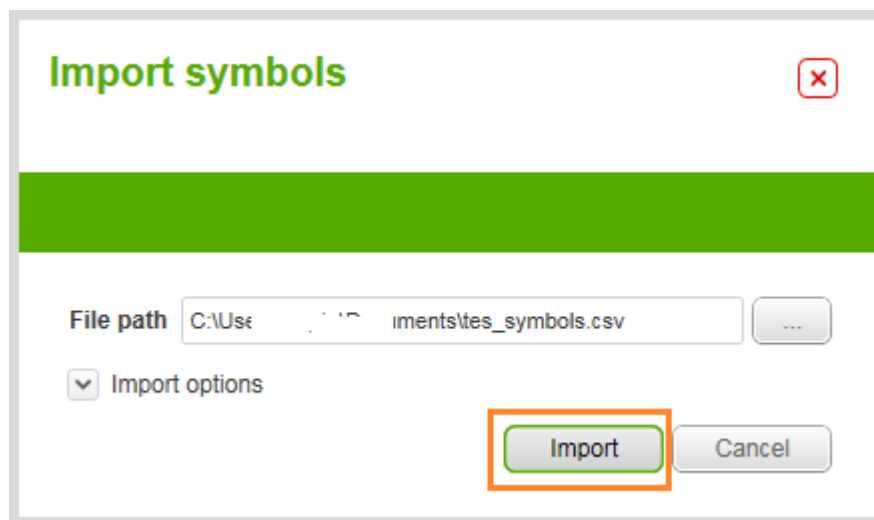
2. Masuk ke Tools, kemudian pilih symbol list.



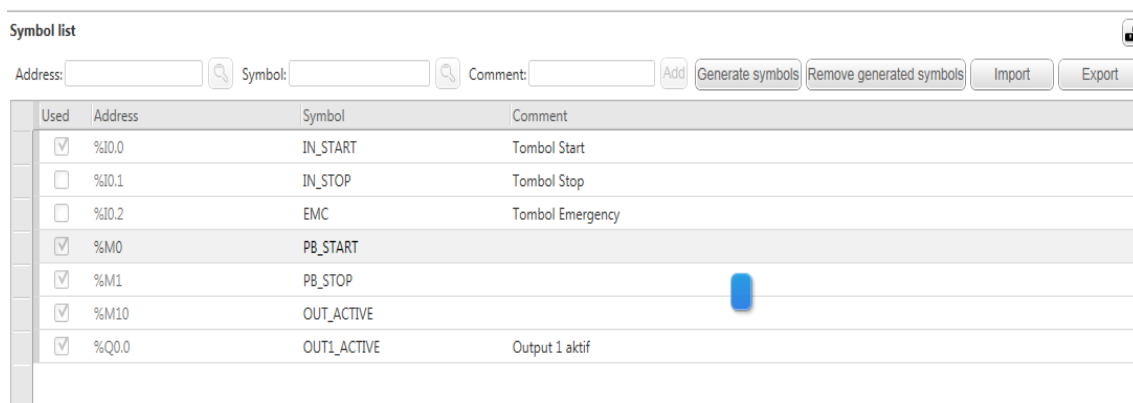
3. Pilih Import button pada Window.



4. Pilih file yang akan diimport ke SoMachine Basic. Pilih Import.



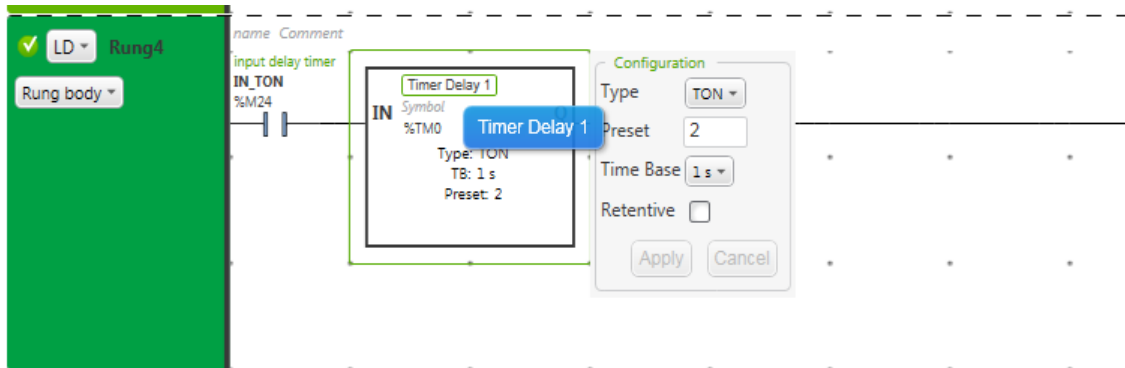
5. Symbol berhasil ditambahkan ke program.



## Comment

Comment digunakan untuk menambahkan komentar pada setiap object / rung yang dibuat. Komentar dapat ditulis menggunakan spasi dan bisa diisi dengan karakter yang lebih banyak.

Untuk menambahkan komentar pada object caranya yaitu dengan double klik comment line pada setiap object, kemudian isikan komentar sesuai keinginan.



Comment line yang ditambahkan pada object akan masuk ke table memory yang terhubung, sehingga dapat diexport atau diimport seperti halnya pada symbols.

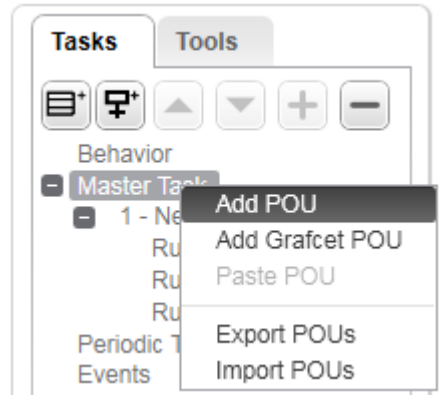
Used	Address	Symbol	Comment
<input checked="" type="checkbox"/>	%M21		
<input checked="" type="checkbox"/>	%M22		
<input checked="" type="checkbox"/>	%M23		
<input checked="" type="checkbox"/>	%M24	IN_TON	input delay timer
<input checked="" type="checkbox"/>	%M25		

## Program Structure

Struktur program pada Somachine Basic menggunakan POU(Program Operation Unit) untuk mengorganisir kode program menjadi beberapa section/bagian. Tujuannya adalah agar memudahkan dalam mencari kode tertentu pada program, serta agar lebih rapi dan terstruktur.

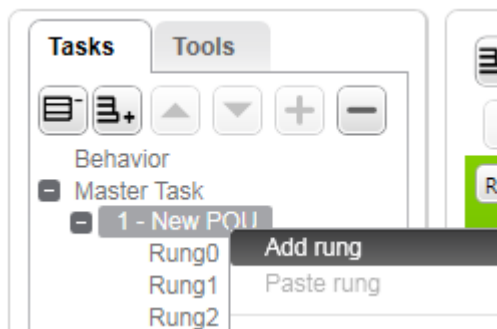
- **Add POU**

Untuk menambahkan section/POU baru, klik kanan pada Master Task , lalu pilih Add POU.



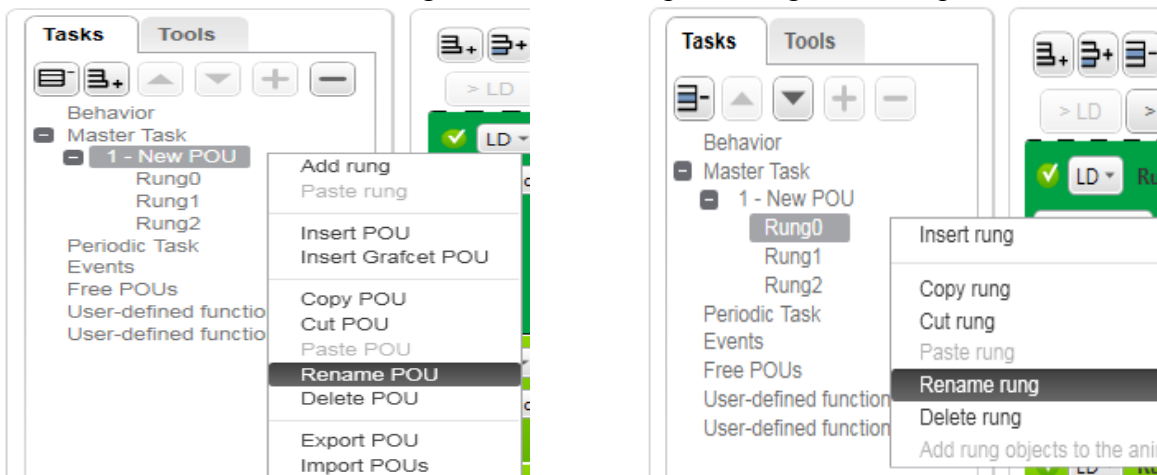
- **Add Rung**

Untuk menambahkan Rung baru, klik kanan pada POU, lalu pilih Add Rung.



- **Rename POU dan Rung**

Untuk Rename POU atau Rung maka klik kanan pada Rung/POU lalu pilih Rename.





## Task

Pada Somachine Basic terdapat task yang dapat digunakan untuk meletakkan program dan mengatur kapan program tersebut di-Scan oleh CPU PLC. Task tersebut yaitu :

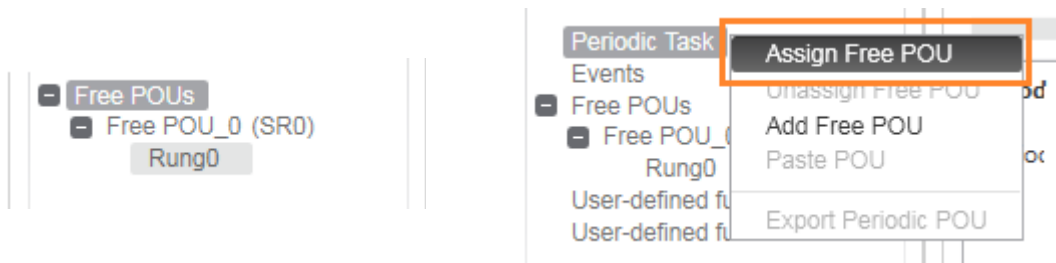
- **Master Task** : Master Task merupakan main task yang mana program di dalamnya akan selalu discanning secara continuous/ terus – menerus. Pada Master Task ada 2 mode yang dapat digunakan yaitu: Normal Mode(Scan selanjutnya akan dijalankan apabila scan yang sekarang telah selesai),Periodic Mode(Scan selanjutnya akan dijalankan apabila waktu scan yang sekarang telah tercapai, setiap scan memiliki waktu yang sama).
- **Periodic Task** : merupakan task yang akan discan pada waktu - waktu tertentu. Periodic task dapat ditentukan waktu scan programnya (1..255ms) default 255 ms.
- **Event Task** : merupakan task yang ditrigger oleh physical input atau HSC(high speed counter) block.Task ini terhubung dengan embedded input (input plc).Event task merupakan subroutine yang sangat singkat.

## Task Priorities

Task Type	Priority
Master	Lowest
Periodic	Lebih tinggi dari Master Task, tapi dibawah Event task
Event	Highest

## Free POU

Free POU merupakan POU yang tidak terhubung dengan Task atau Event manapun.Free POU juga dapat disebut library yang dapat diubah – ubah tanpa mempengaruhi main program.Free POU dapat dihubungkan ke periodical task.

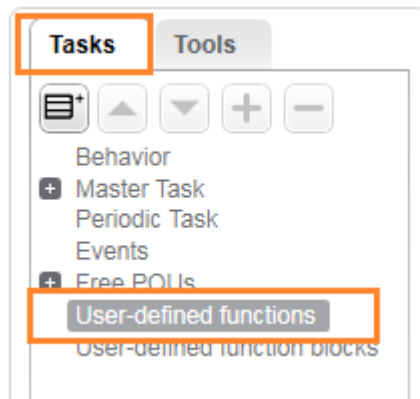


## User-defined function

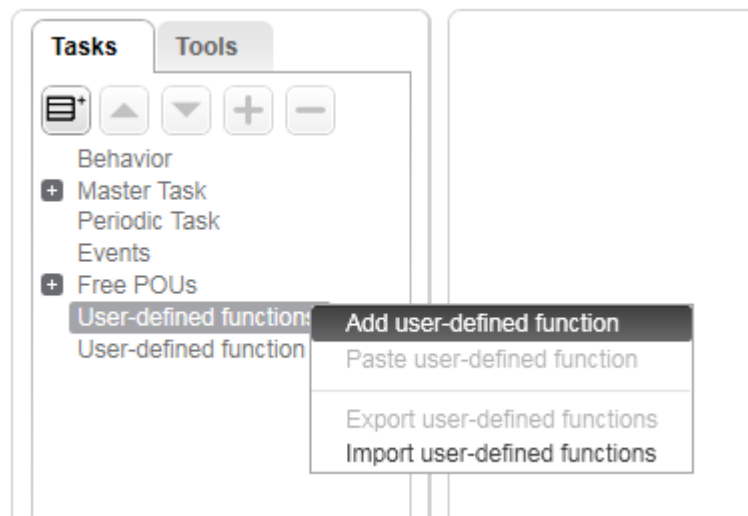
User-defined function merupakan fitur yang digunakan untuk membuat function sesuai keinginan. Function yang dibuat boleh menggunakan satu atau lebih input parameter, local variabel, dan return value. User-defined function dapat dipanggil di Master task, Periodical task, atau Free Pou's.

Langkah-langkah membuat User-defined function di Somachine Basic adalah sebagai berikut :

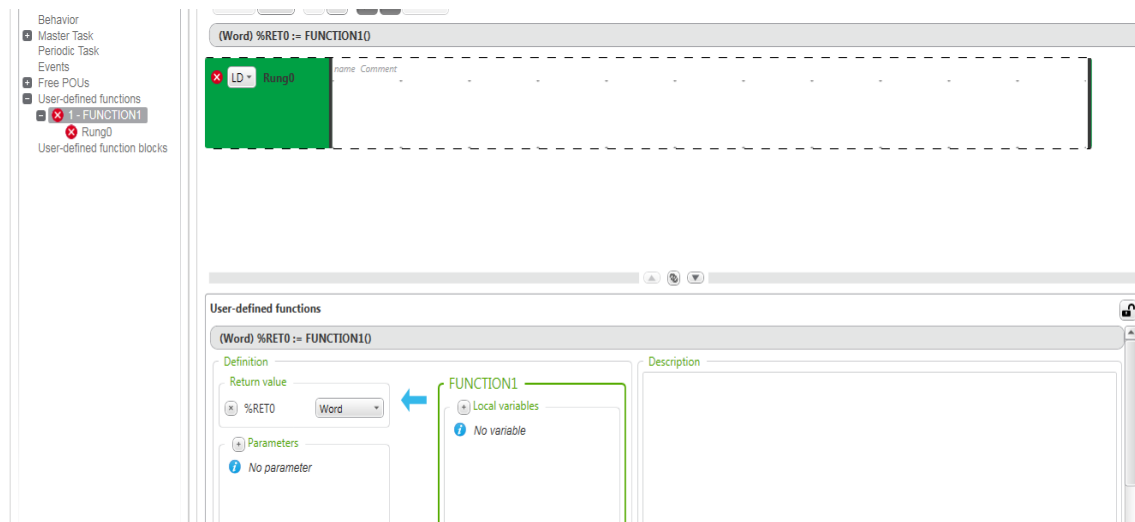
1. Masuk ke Tab Programming, lalu pilih Task.



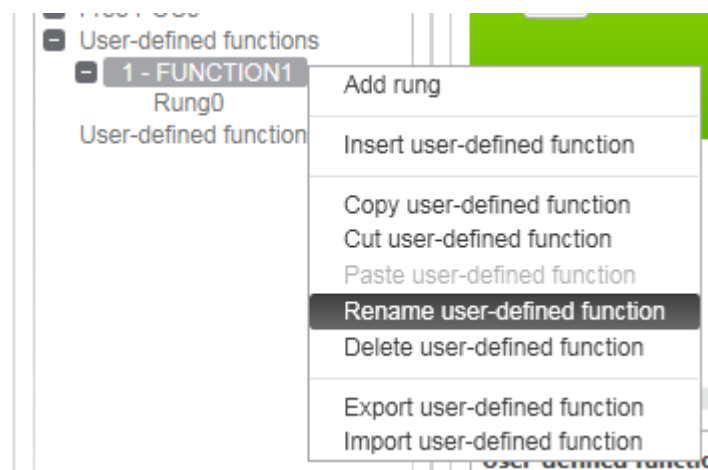
2. Untuk menambahkan User-defined function baru maka klik kanan, dan pilih add user-defined function.



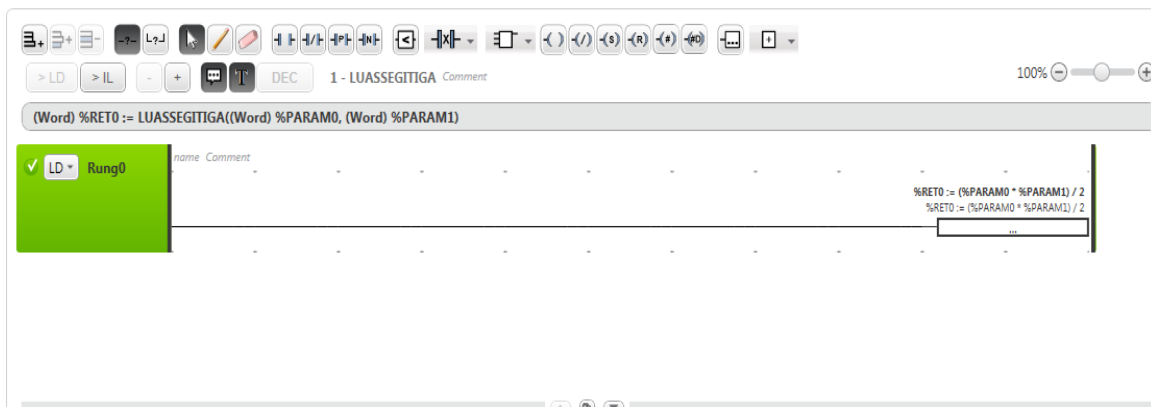
3. Tambahkan program untuk function yang telah dibuat.



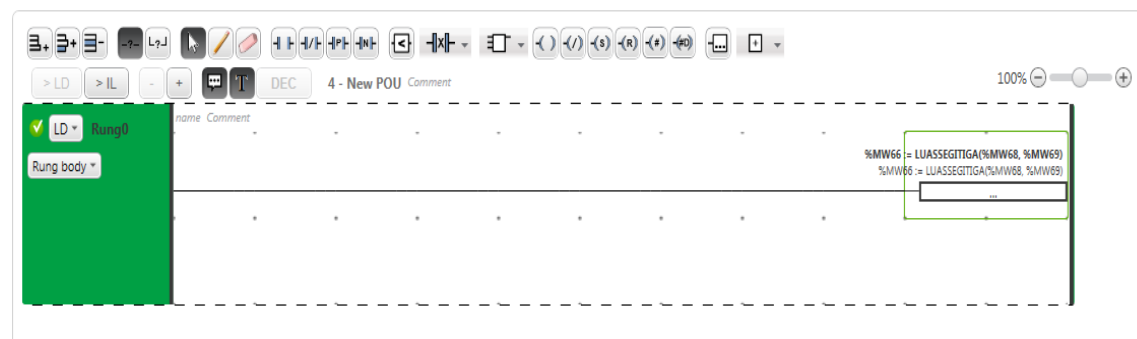
4. Untuk mengubah nama dari function yang telah dibuat, Klik kanan pada function lalu pilih Rename User-defined function.



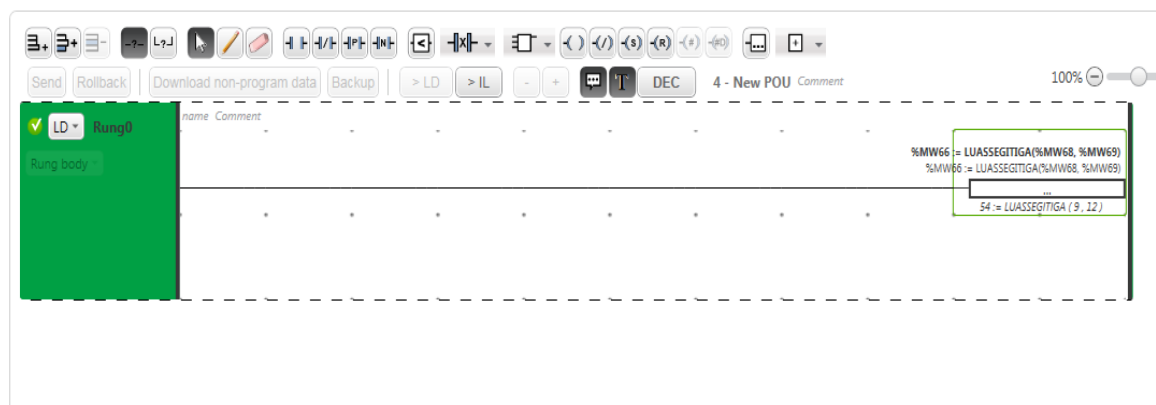
5. Contoh User-defined function untuk membuat rumus mencari luas segitiga(  $\frac{1}{2} \times a \times t$  )



6. Panggil function di POU Master Task.



7. Tampilan saat simulasi berjalan, Luas segitiga dengan panjang alas = 9, tinggi= 12 → luas = 54.

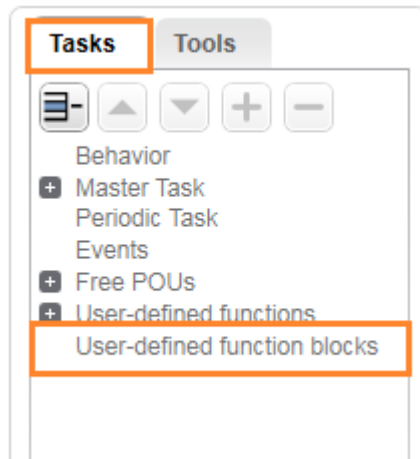


## User-defined function block

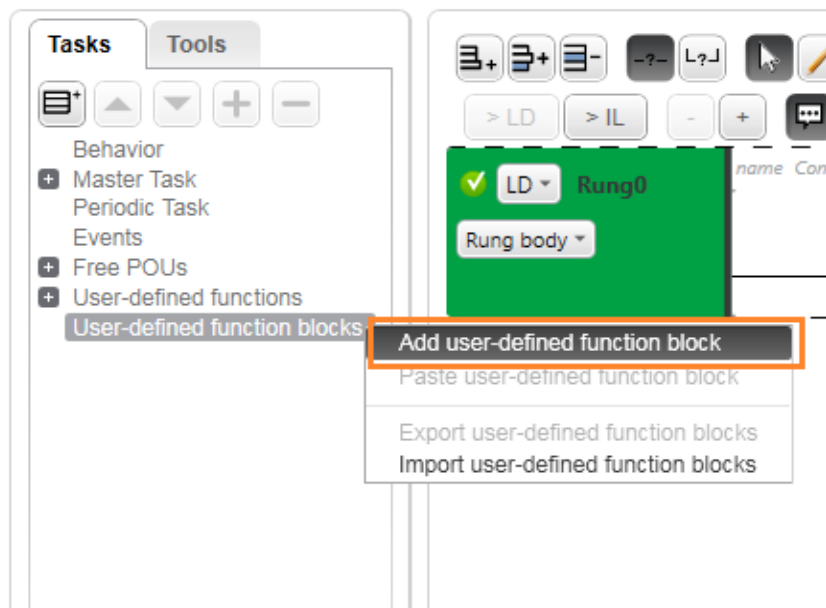
User-defined function block merupakan fitur yang digunakan untuk membuat function block sesuai keinginan. Function block yang dibuat boleh menggunakan satu atau lebih input parameter, local variabel, dan return value. User-defined function block dapat dipanggil di Master task, Periodical task, Event Task atau Free Pou's.

Langkah-langkah untuk membuat User-defined function block pada Somachine Basic adalah sebagai berikut :

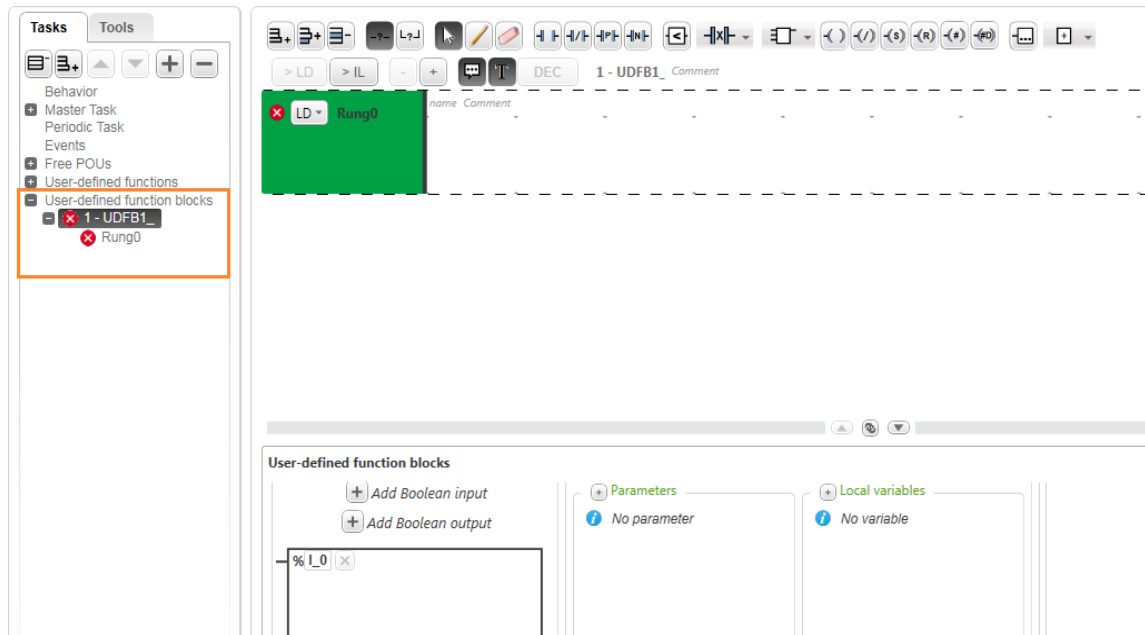
1. Masuk ke Tab Programming lalu pilih Task.



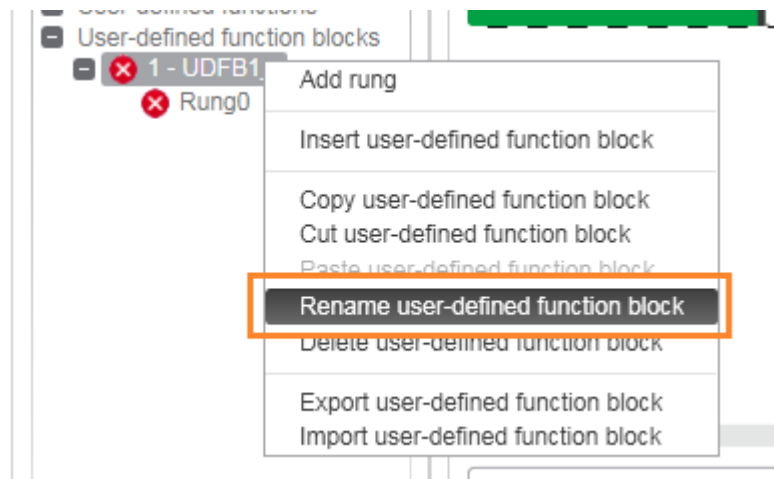
2. Untuk menambahkan User-defined function block baru maka klik kanan, dan pilih add user-defined function block.



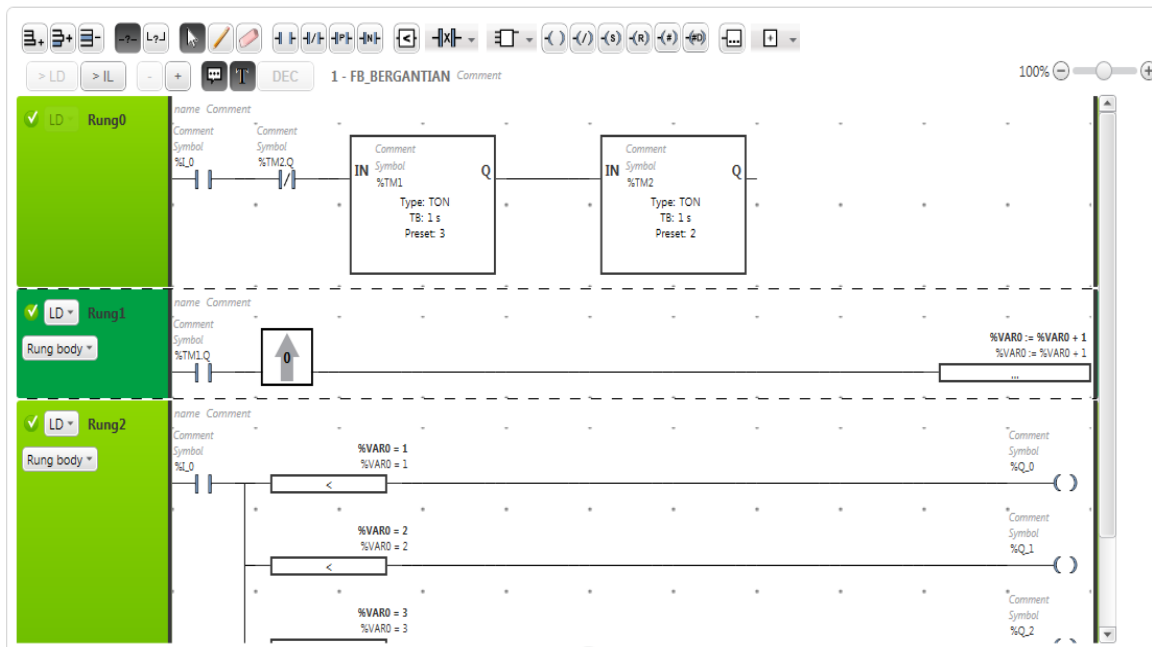
3. User-defined function block telah berhasil dibuat, sekarang tinggal menambahkan program pada user-defined function block sesuai keinginan.



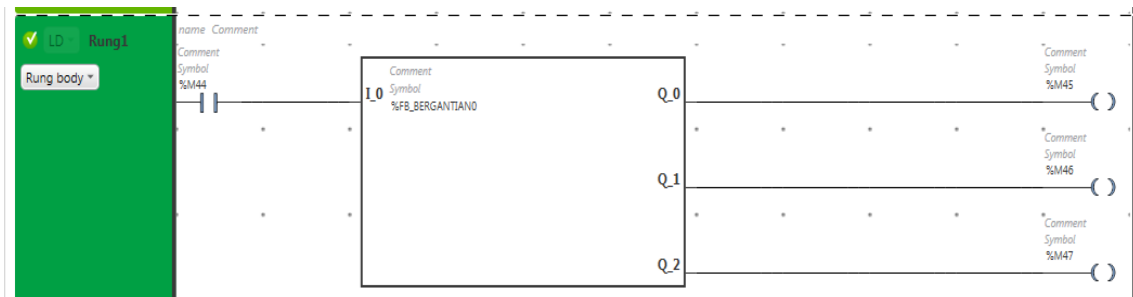
4. Untuk mengubah nama dari function block yang telah dibuat, Klik kanan pada function block lalu pilih Rename User-defined function block.



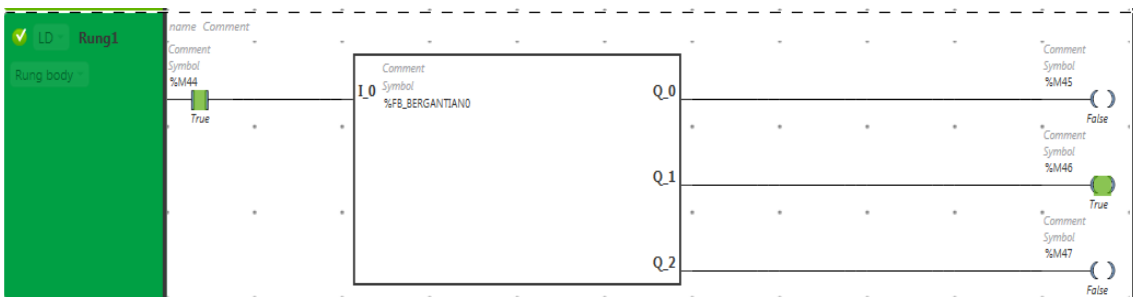
5. Contoh User-defined function block untuk menghidupkan output bergantian.



6. Panggil function block yang telah dibuat di POU Master Task.



7. Tampilan saat simulasi output akan menyala bergantian apabila input aktif.



## Programming Rungs

- **Link** : merupakan garis yang menghubungkan antara object satu dengan yang lainnya, biasanya link ini akan terbuat secara otomatis apabila menambahkan object coil pada rung. Link ini terdapat 2 mode yaitu Normal Mode dan Braching Mode. Perbedaanannya apabila menggunakan branching mode maka saat menambahkan object baru dibawah object lainnya akan langsung terbuat sebuah garis branch.



- **Automatic Links** : merupakan garis yang otomatis dibuat pada saat menambahkan object, untuk menggunakan automatic link ini maka perlu mengaktifkan Branching Mode dengan menekan ikonnya atau tombol Shift + F2.
- **Manually Drawing Links** : untuk membuat links manual maka pada SoMachine Basic terdapat 2 tool yang dapat digunakan yaitu Draw Line Tool dan Erase Line Tool untuk membuat link dan menghapusnya.





## Acces Bits pada Memory Word

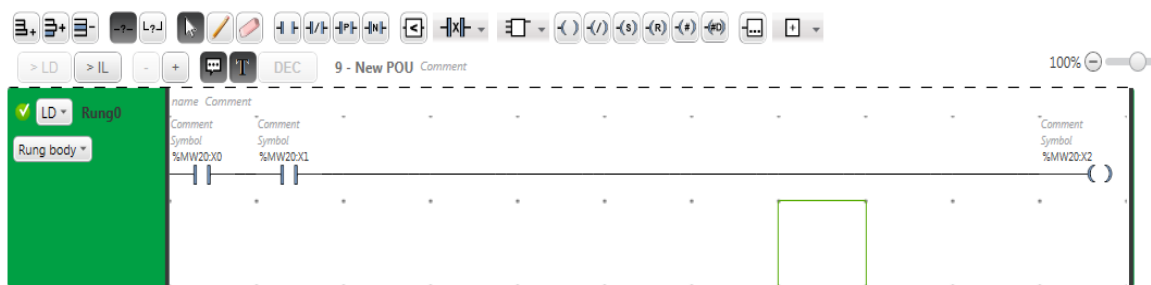
Pada SoMachine Basic Memory dengan tipe Bit yang biasa dipakai seperti %M1,%M10 tidak terhubung dengan %MW.Jadi apabila melakukan perubahan pada Memory Bit tidak akan berpengaruh terhadap Memory Word.Untuk mengakses Bit pada memory word langkahnya tidak rumit yaitu cuman mengubah cara penulisannya saja.

Penulisan untuk mengakses Bits pada Memory Word adalah sebagai berikut :

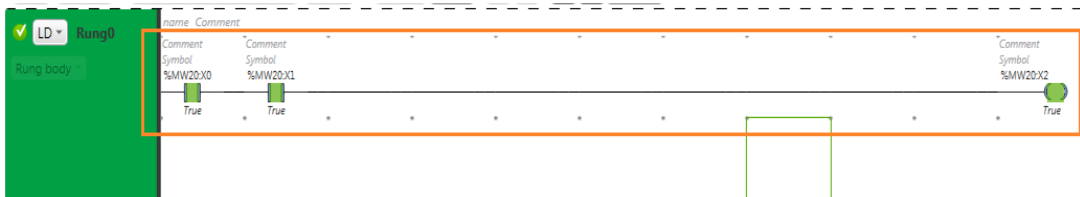
**%MWi:Xk**

Sebagai contoh ingin mengakses bit number 0 pada %MW20 maka penulisannya **%MW20:X0**.

Nanti di program apabila bit tersebut dirubah kondisinya dari False ke True atau sebaliknya maka nilai dari %MW20 akan berubah mengikuti number bit yang aktif.



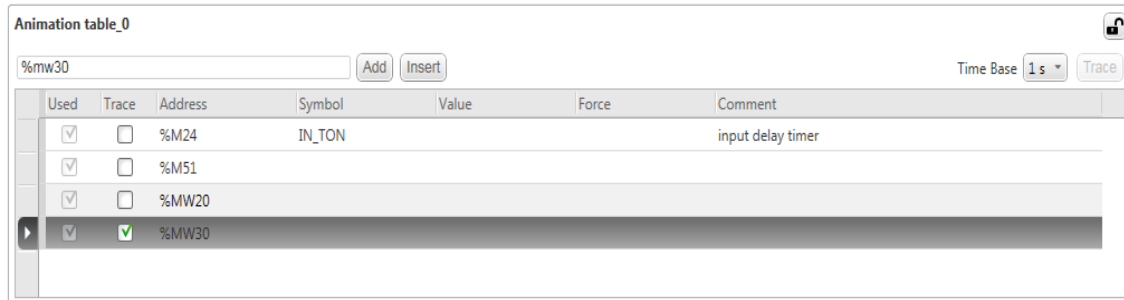
Apabila program tersebut disimulasikan maka nilai dari %MW20 akan menjadi seperti berikut ini.



Used	Trace	Address	Symbol	Value	Force	Comment
<input checked="" type="checkbox"/>	<input type="checkbox"/>	%MW20		7		

## Animations Table

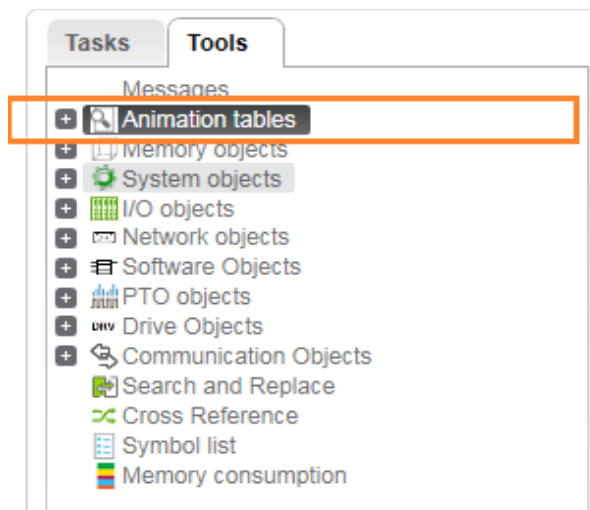
Animations Table digunakan untuk melihat symbols dan comment yang terhubung pada object memory, Monitor(view) dan Modify nilai values secara realtime dan untuk memilih object yang akan ditampilkan di Trace window. Animations table dapat dibuat dalam banyak group, jadi dapat memudahkan dalam monitoring serta modify valuenya.



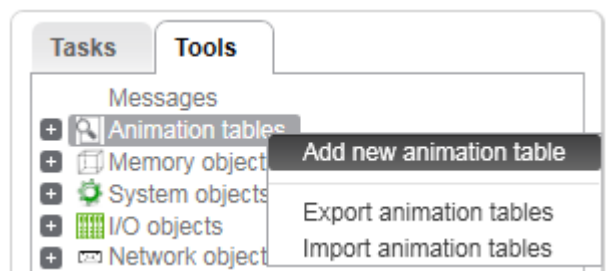
Used	Trace	Address	Symbol	Value	Force	Comment
<input checked="" type="checkbox"/>	<input type="checkbox"/>	%M24	IN_TON			input delay timer
<input checked="" type="checkbox"/>	<input type="checkbox"/>	%M51				
<input checked="" type="checkbox"/>	<input type="checkbox"/>	%MW20				
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	%MW30				

Untuk menambahkan animations table yang baru, langkah-langkahnya adalah sebagai berikut :

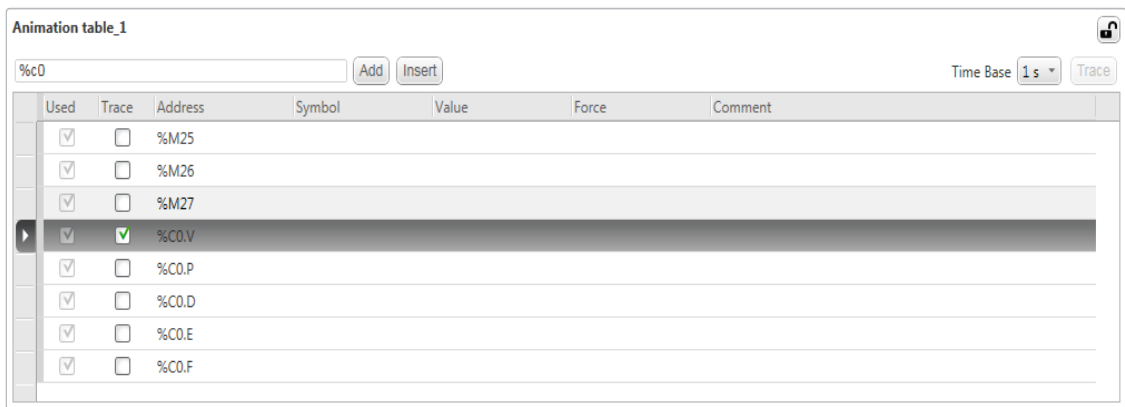
1. Buka Tools kemudian pilih animations table



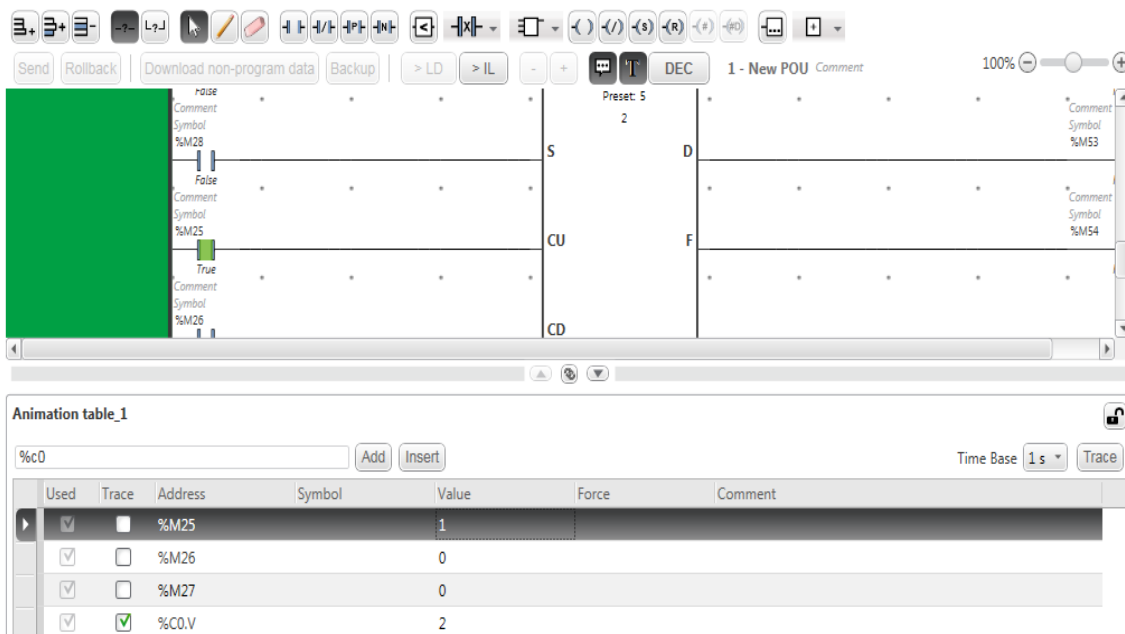
2. Klik kanan pada animations table, pilih add animations table.



3. Tambahkan memory address / IO address yang mau dimonitor.



4. Tampilan animations table pada saat runtime adalah sebagai berikut.

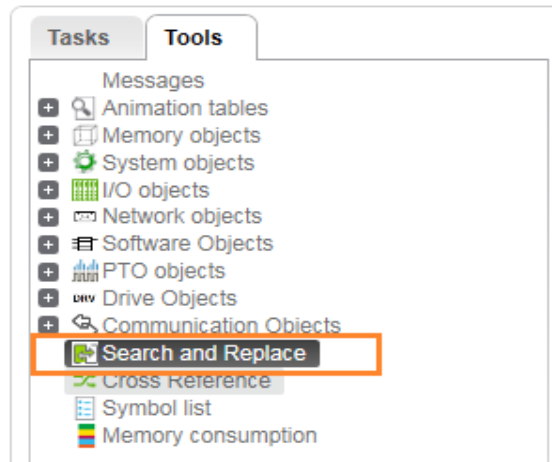


## Search dan Replace

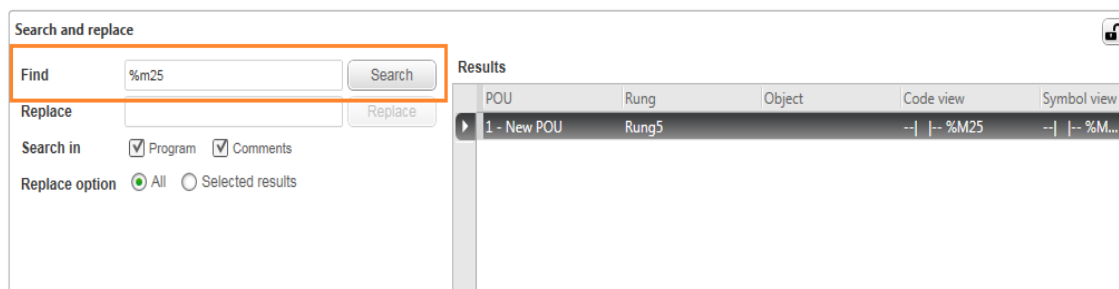
Fitur Search dan Replace digunakan untuk mencari serta mengganti address pada object. Tujuannya untuk memudahkan dalam mengganti beberapa alamat object yang kurang tepat dengan cepat.

Untuk menggunakan fitur Search dan Replace ini maka langkah-langkahnya adalah sebagai berikut :

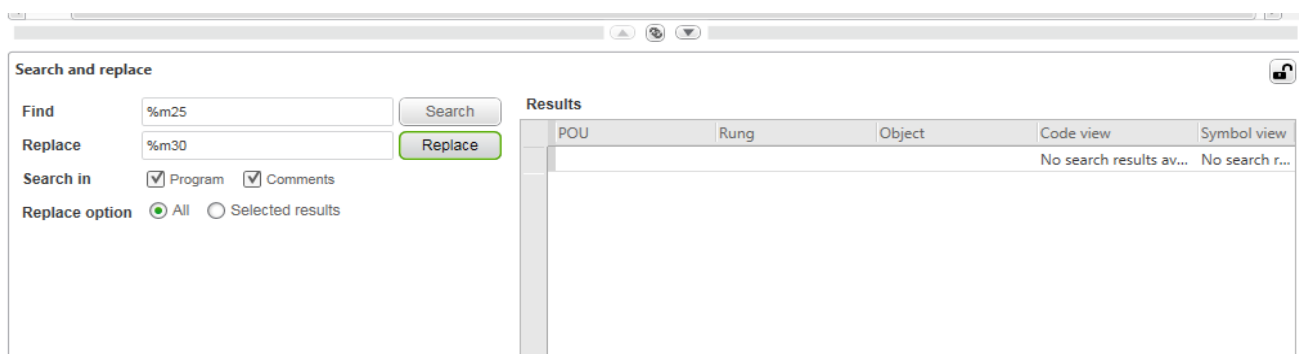
1. Masuk ke Tool kemudian pilih Search dan Replace.



2. Masukkan address yang akan dicari/direplace.



3. Masukkan address yang digunakan untuk mereplace, kemudian klik replace. Maka address %M25 jika disearch lagi tidak akan ada hasilnya.

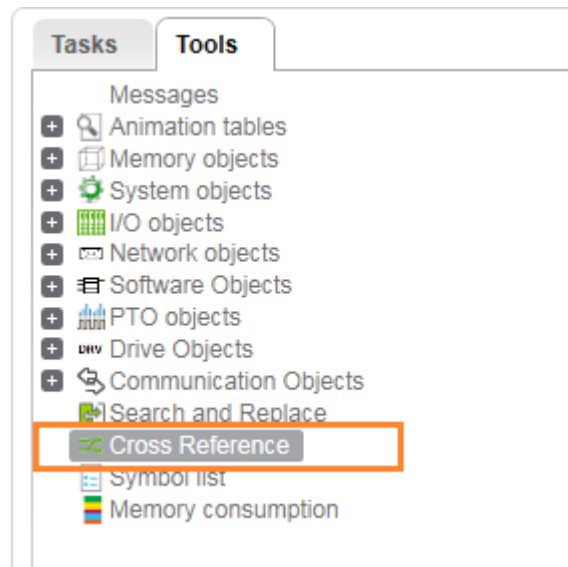


## Cross Reference

Cross Reference digunakan untuk mencari letak dimana saja address tersebut berada. Apabila ada beberapa object yang memakai address yang sama maka akan muncul letak rung dan detailnya di bagian cross reference.

Untuk menggunakan fitur cross reference ini maka langkah-langkahnya adalah sebagai berikut :

1. Masuk ke Tools kemudian pilih cross reference.



2. Pilih object yang akan dicari addressnya menggunakan cross reference. Sebagai contoh address %M21

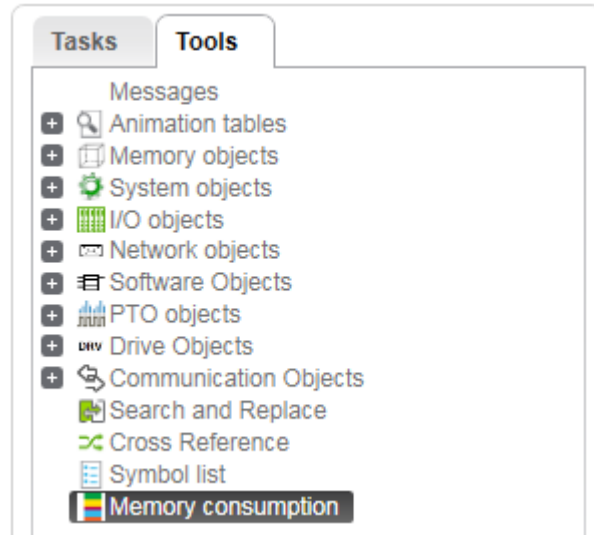
POU	Rung	Object	Code view	Symbol view
<b>%M21</b>				
1 - New POU	Rung1	--   -- %M21	--   -- %M21	Symbol view
1 - New POU	Rung2	--   -- %M21	--   -- %M21	

## Memory Consumption

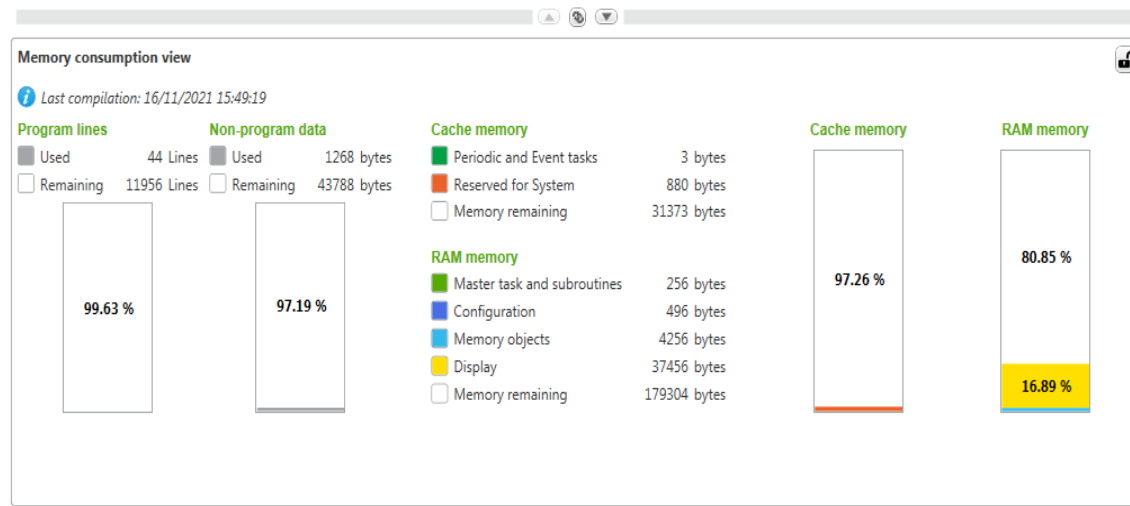
Memory Consumption digunakan untuk melihat memory yang telah digunakan pada saat mengedit program. Memory consumption ini dapat menjadi patokan berapa besar memori yang masih tersisa dan belum terpakai.

Untuk melihat berapa memory yang telah terpakai, serta memory sisa, langkah-langkahnya adalah sebagai berikut :

1. Masuk ke Tool lalu pilih Memory Consumption.



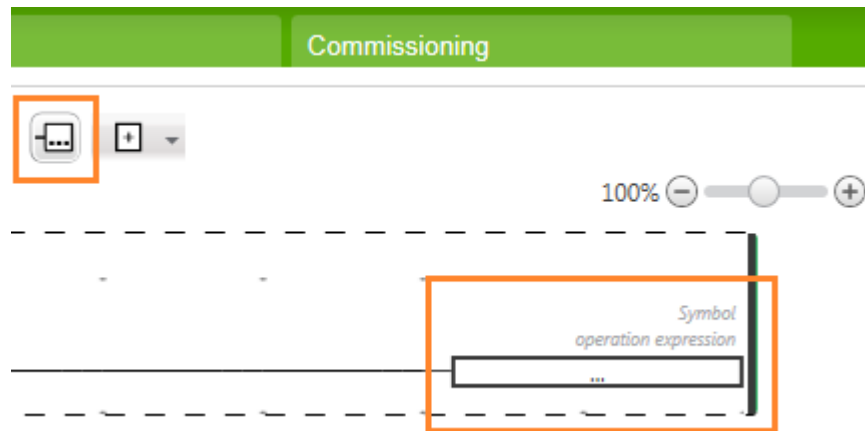
2. Lihat detail memory yang telah ditampilkan dalam bargraph. Pastikan bahwa sebelumnya program telah dicompile agar data yang tampil dapat akurat.



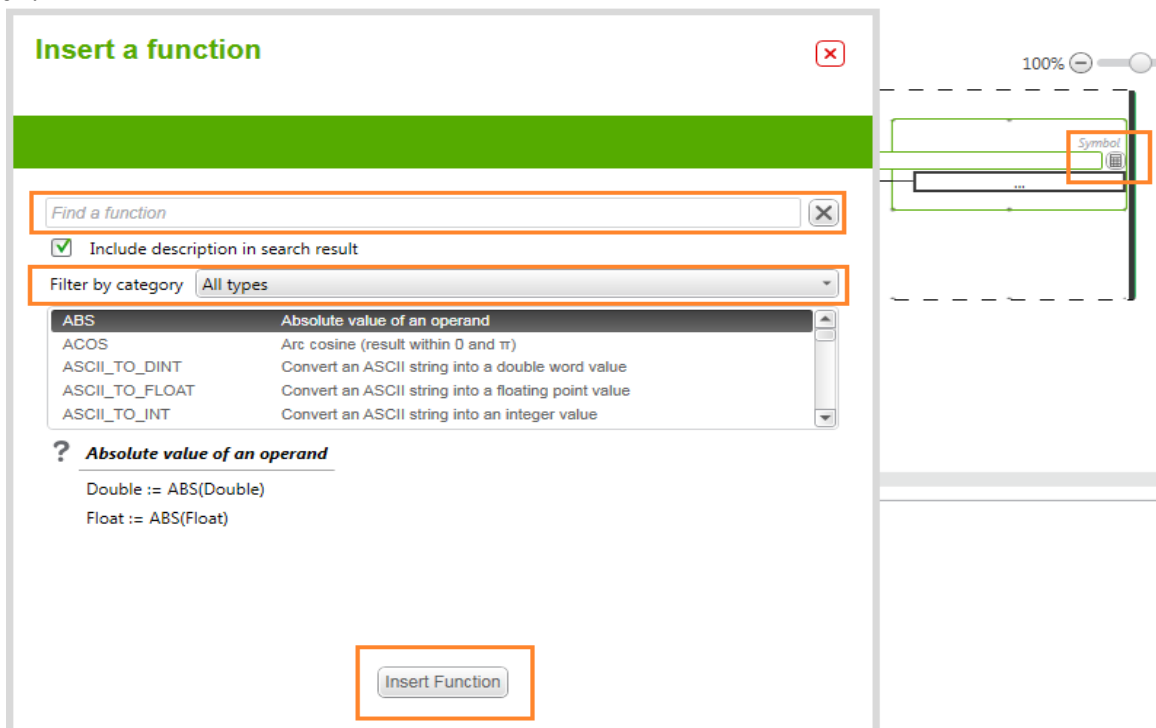
## Operation Block

Operation Block digunakan untuk menambahkan block ke dalam program. Block ini dapat diisi instruksi berupa function yang berfungsi untuk membantu mengerjakan kalkulasi pada program. Instruksi yang terdapat pada group Operation Block dibagi menjadi beberapa kategori yaitu : ASCII, Communication, Floating Point, I/O Object, Numerical processing, PID, Table, User defined function.

Tampilan Operation Block sebelum ditambahkan function yang akan digunakan.

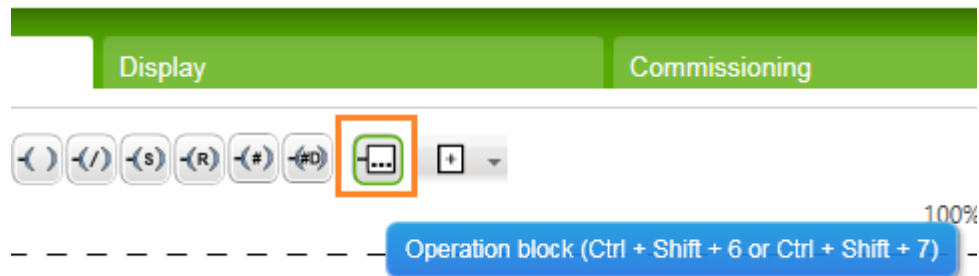


Untuk membantu memudahkan menambah function pada program, maka dapat membuka Dialog Insert Function.

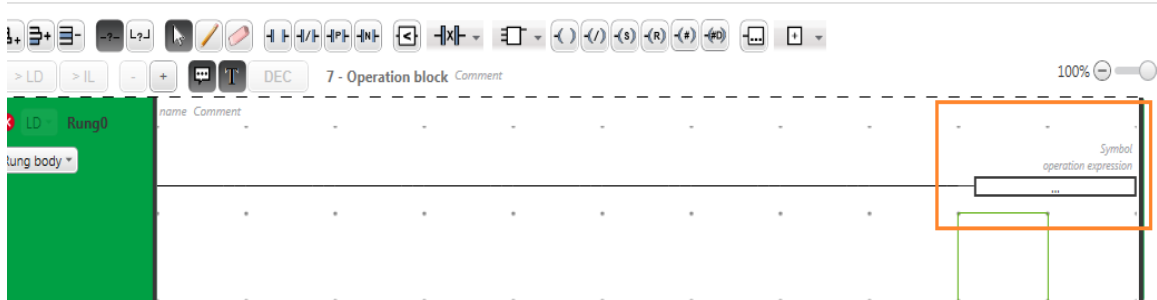


Langkah-langkah untuk menambahkan Function berupa Operation Block pada SoMachine Basic adalah sebagai berikut.

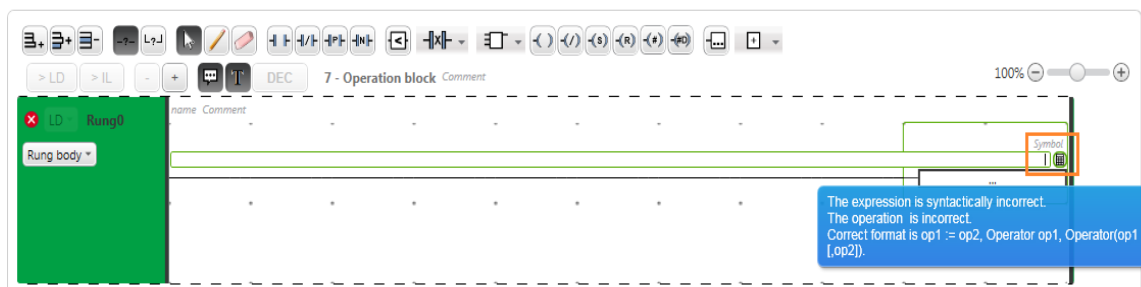
1. Masuk ke Tab Programming, lalu pilih ikon Operation Block.



2. Klik pada ikon lalu drag ke rung yang diinginkan. Klik untuk menempatkan object ke rung.

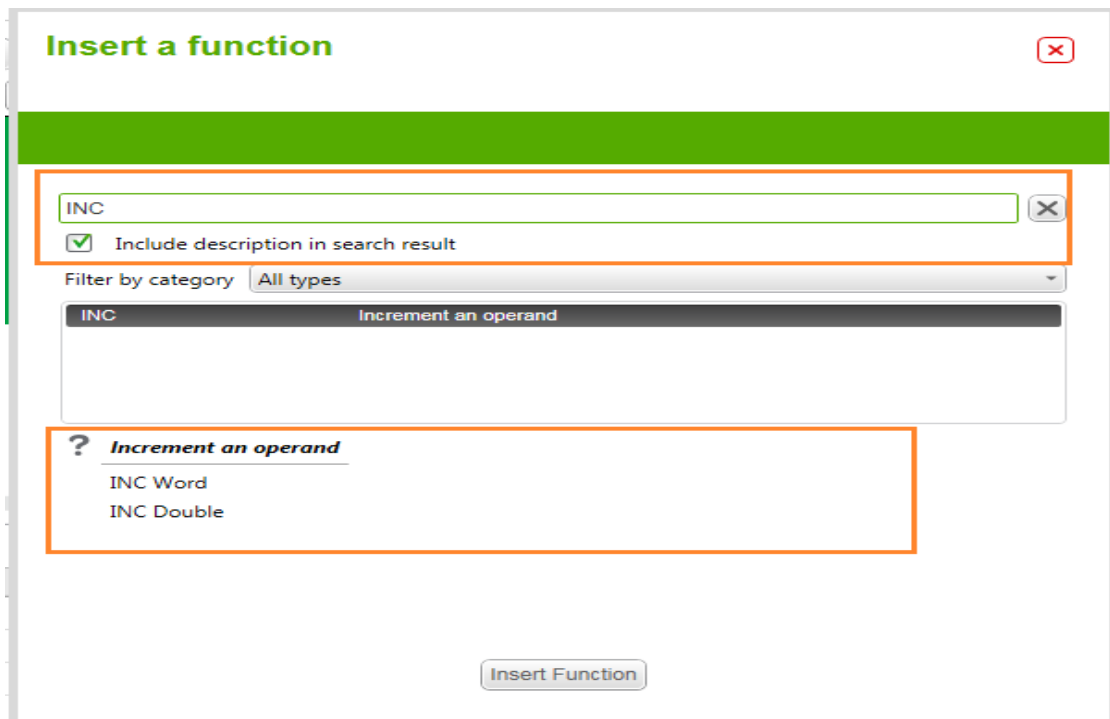


3. Klik tulisan “*operation expression*” pada operation block untuk menambahkan function. Function dapat langsung ditambahkan, apabila ingin membuka dialog insert function maka pilih ikon dibagian paling kanan pada kolom.



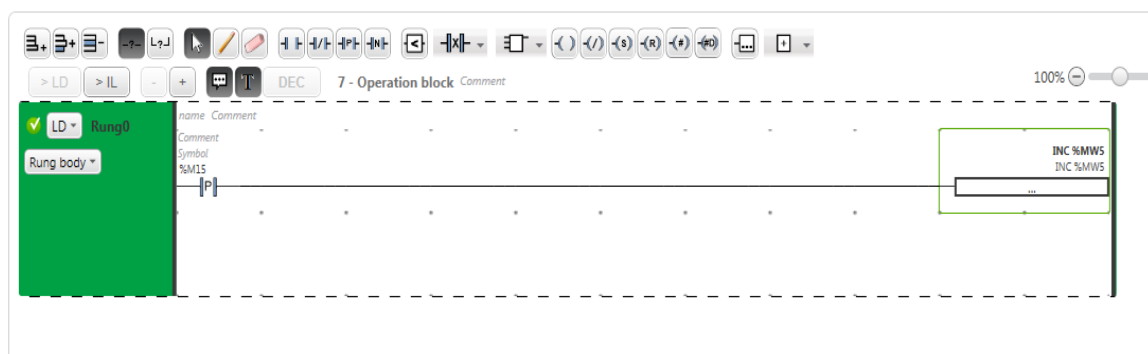


4. Dialog Insert Function telah muncul, cari function yang akan digunakan. Sebagai contoh disini akan ditambahkan instruksi INC (increment = untuk menambahkan nilai pada register dengan nilai 1 setiap kali instruksi dijalankan).



Untuk mencari function yang dibutuhkan maka cukup menuliskan namanya pada kolom search yang tersedia, kemudian enter. Di bagian paling bawah terdapat bantuan tentang penulisan function pada operation block. Jika function telah ditemukan maka klik insert function.

5. Tampilan program setelah function berhasil ditambahkan.



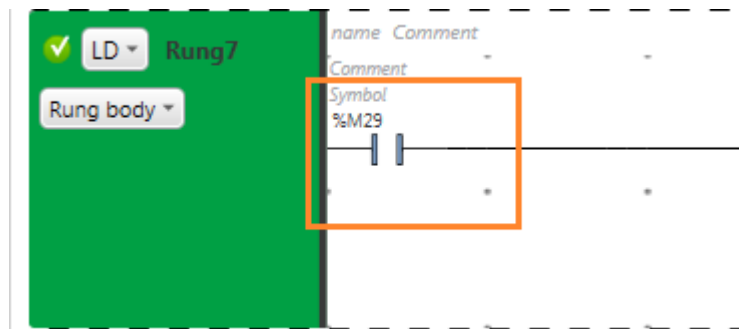
## Instructions

Instruction atau instruksi digunakan dalam pembuatan sebuah program agar program dapat berjalan sesuai dengan fungsi yang diinginkan. Pada SoMachine Basic terdapat bermacam-macam instruksi yang dapat digunakan. Instruksi tersebut antara lain adalah sebagai berikut :

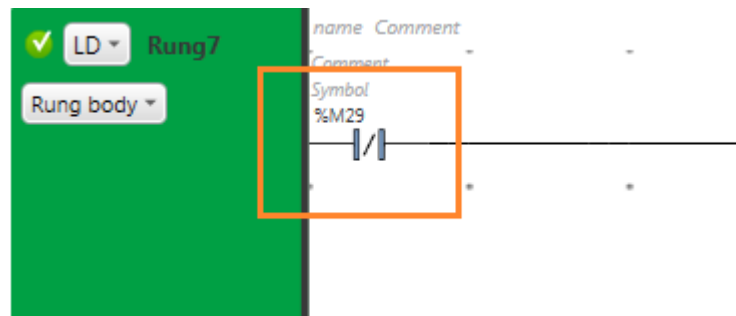
### Contacts dan Coils

Contact dan Coil adalah instruksi yang digunakan untuk memproses nilai boolean(boolean processing). Contacts terdiri dari beberapa macam yaitu :

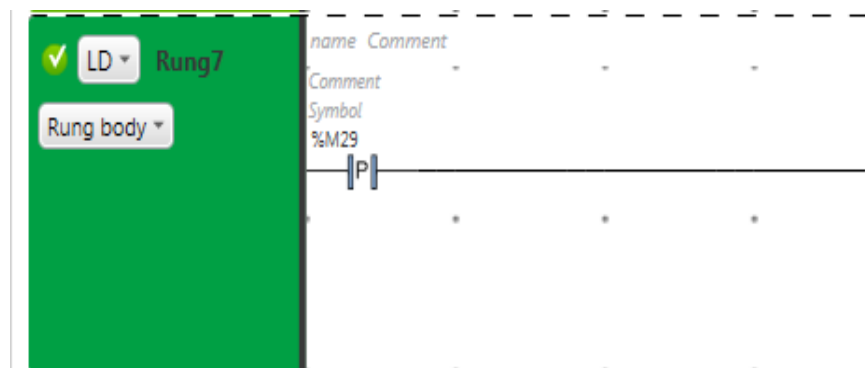
- **NO (Normaly Open)** : Contact yang pada saat kondisi normal dia membuka(open).



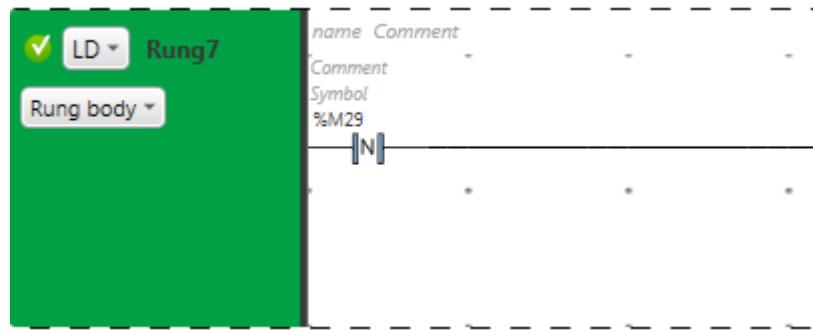
- **NC (Normaly Close)** : Contact yang pada saat kondisi normal dia menutup(close).



- **Rising Edge Contact** : Contact hanya akan aktif sekali pada saat Rising(Off → On).

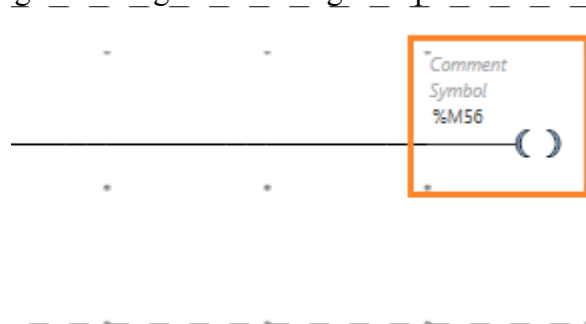


- **Falling Edge Contact** : Contact akan menyala sekali pada saat falling(On → Off)

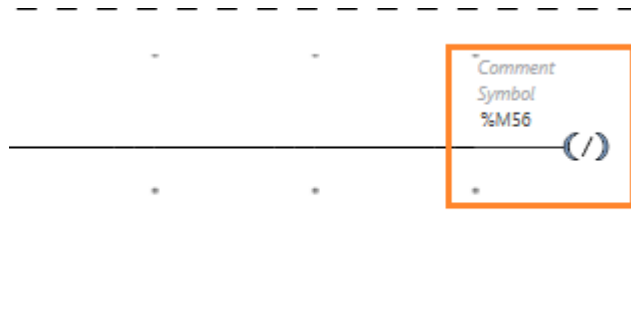


Coils terdiri dari beberapa macam yaitu :

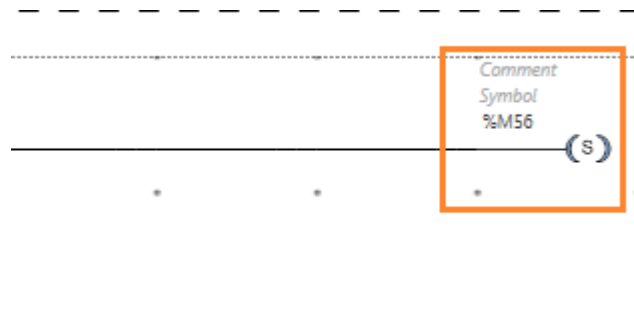
- **Direct Coil** : koil yang umum digunakan sebagai outputan.



- **Negated/Inverse Coil** : koil yang memiliki fungsi invert / NOT.



- **Set Coil** : merupakan koil yang latching terus menerus sampai ada yang mereset.



- **Reset Coil** : digunakan untuk mereset koil yang telah aktif ( Set).



## Exclusive OR Operator

Exclusive OR Operator merupakan instruksi berupa contact yang memiliki fungsi untuk memproses operand dan menghasilkan hasil outputan sesuai logika seperti tabel kebenaran milik gerbang logika Exclusive OR.



Tabel kebenaran Exclusive OR adalah sebagai berikut :

Input 1	Input 2	Result / Hasil
0	0	0
1	0	1
0	1	1
1	1	0

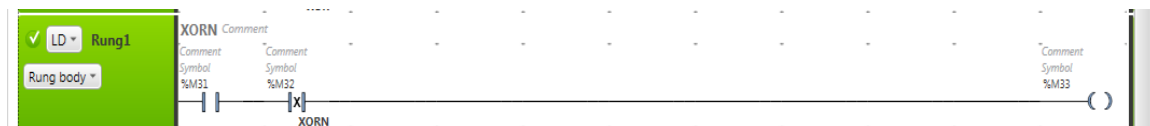
Penggunaan Exclusive OR pada program tidak boleh di paralel dengan diagram ladder element yang lain, serta peletakan posisi instruksi ini tidak boleh berada pada posisi awal rung.

Pada SoMachine Basic terdapat 4 macam instruksi Exclusive OR yang dapat digunakan.

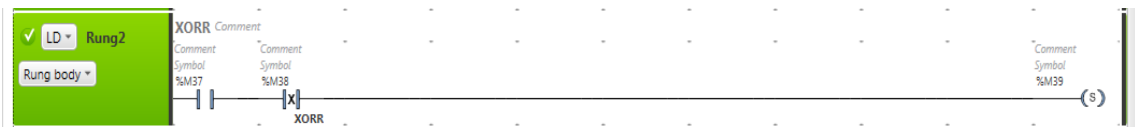
- **XOR** = Exclusive OR



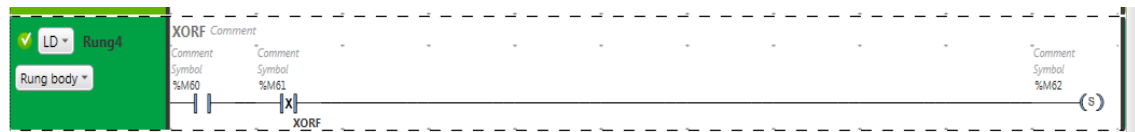
- **XORN** = Exclusive OR Inverse



- ***XORR*** = Exclusive OR Rising Edge



- ***XORF*** = Exclusive OR Falling Edge



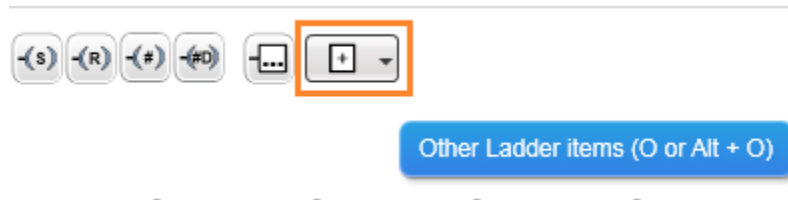
Dengan menggunakan instruksi Exclusive OR ini pada program maka penulisan program akan lebih ringkas dan mudah dilihat saat proses monitoring.

## Rising dan Falling

Rising dan Falling instruction merupakan instruksi yang fungsinya mirip dengan rising edge contact dan falling edge contact. Perbedaan Rising dan Falling ini merupakan sebuah function yang tidak memerlukan address memory. Jadi untuk menggunakannya cukup dengan menambahkan tanpa mengatur address yang mentrigger. Maksimal penggunaannya pada program yaitu : 32. Penempatan instruksi ini tidak dapat diposisikan di awal kolom pada Ladder / instruksi pertama pada IL.

Untuk menggunakan instruksi pada program langkah-langkahnya adalah sebagai berikut :

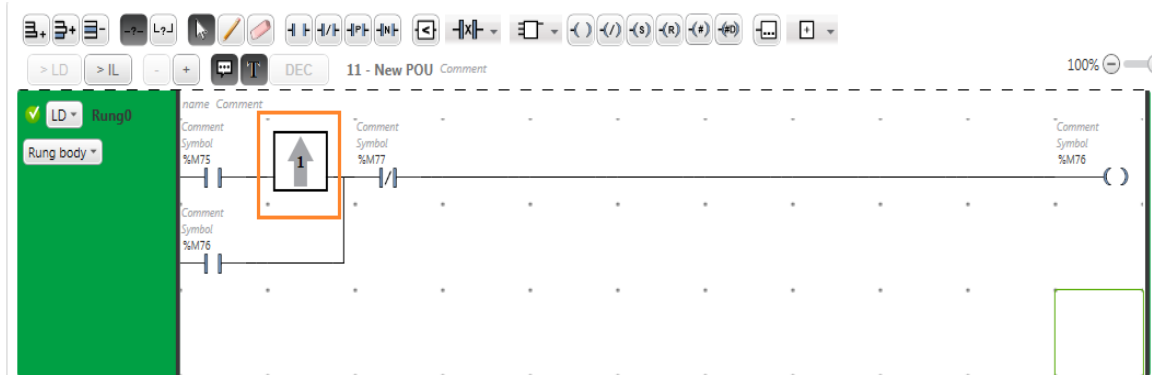
1. Buka Tab Programming, kemudian klik ikon Other Ladder Items.



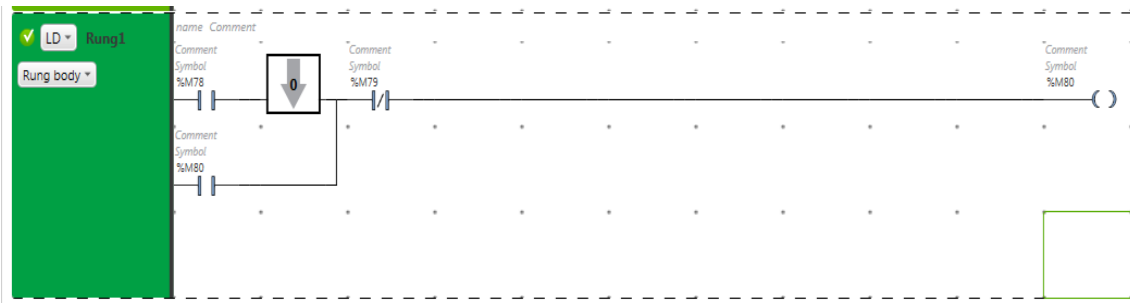
2. Cari Rising atau Falling lalu klik untuk memilih.



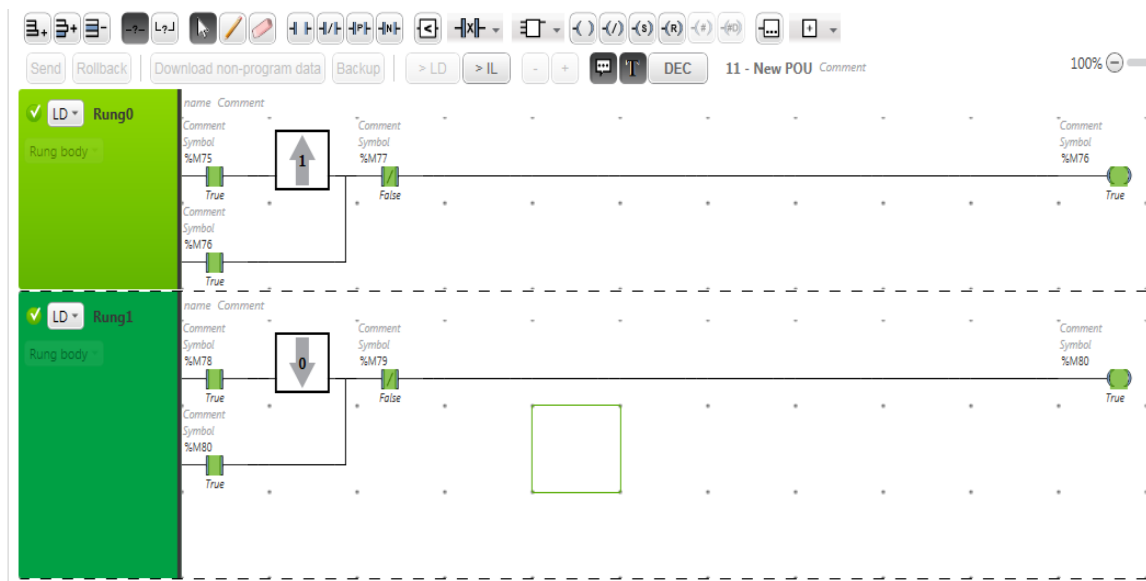
3. Drag object ke rung yang diinginkan. Klik untuk menempatkan object pada rung.



4. Untuk menambahkan instruksi Falling maka langkah-langkahnya sama seperti menambahkan instruksi Rising.



5. Jalankan simulasi untuk memonitor program yang dibuat.



Untuk instruksi Rising maka ia akan mentrigger sekali saja pada saat kondisi dari Off ke On, Sedangkan instruksi Falling akan mentrigger sekali pada saat kondisi dari On ke Off.



## Timers

Timer digunakan untuk membuat delay waktu pada program. Timer pada Somachine Basic terdapat 3 tipe yaitu TON(On Delay Timer),TOF(Off Delay Timer),TP(Pulse Timer).Terdapat 255 timer yang dapat digunakan pada program.Untuk melihat daftar timer yang sudah/belum digunakan maka pilih Tools → Software Object → Timers.

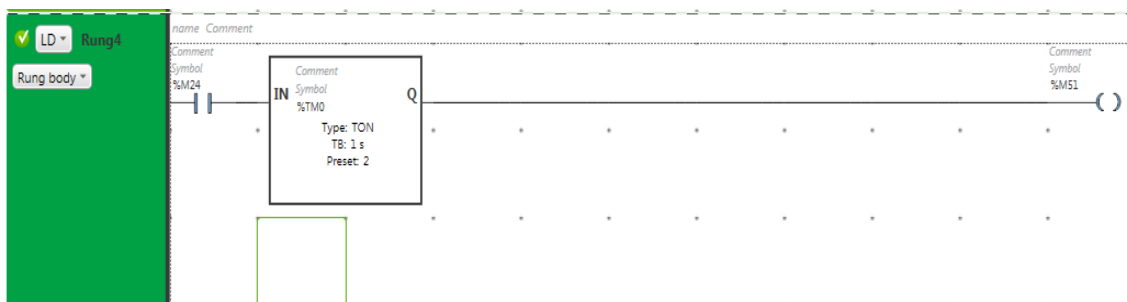
Used	Address	Symbol	Type	Retentive	Time Base	Preset	Comment
<input checked="" type="checkbox"/>	%TM0		TON	<input type="checkbox"/>	1 s	2	
<input type="checkbox"/>	%TM1		TON	<input type="checkbox"/>	1 min	9999	
<input checked="" type="checkbox"/>	%TM2		TON	<input checked="" type="checkbox"/>	1 min	9999	
<input type="checkbox"/>	%TM3		TON	<input type="checkbox"/>	1 min	9999	
<input type="checkbox"/>	%TM4		TON	<input type="checkbox"/>	1 min	9999	

Untuk menambahkan Timer pada program, langkah-langkahnya adalah sebagai berikut :

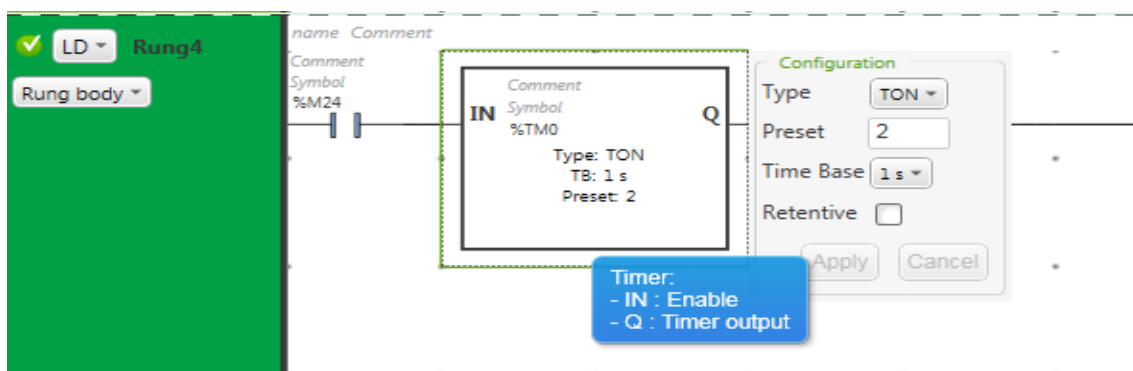
1. Klik pada ikon function block lalu pilih timer.



2. Drag Object ke rung, klik pada rung yang diinginkan, klik Esc.

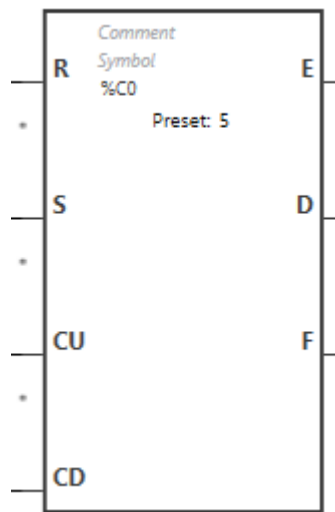


3. Untuk mengatur tipe, time base, serta preset maka arahkan cursor pada object timer maka akan muncul window untuk mengatur.



## Counters

Counters digunakan untuk menghitung inputan yang masuk. Counter ini ada 2 macam yaitu Counter Up dan Counter Down. Terdapat 255 counter yang dapat dibuat pada program.



Pada Block Counter terdapat beberapa Operation input yang dapat digunakan yaitu :

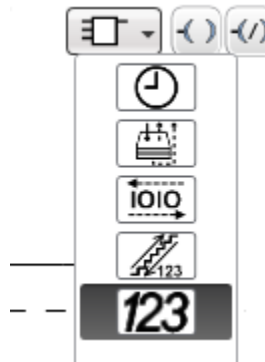
- R : Reset, digunakan untuk mereset nilai value dari counter menjadi 0.
- S : Set, digunakan untuk set nilai value dari counter sesuai dengan nilai preset.
- CU : Count Up, digunakan untuk menghitung hitungan naik / increment.
- CD : Count Down, digunakan untuk menghitung hitungan turun / decrement.

Sedangkan untuk Operation Output yang dapat digunakan yaitu :

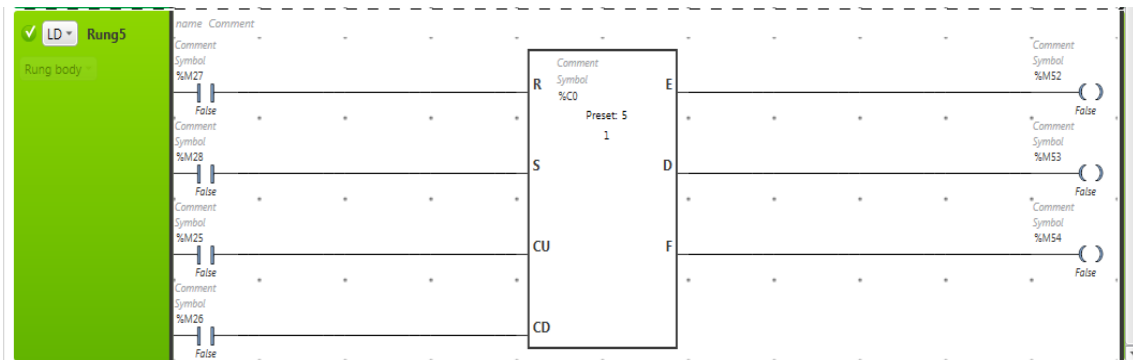
- E : Down Counter Overflow, aktif apabila nilai value counter 9999.
- D : Preset Output tercapai, aktif apabila nilai value counter = nilai preset.
- F : Up Counter Overflow, aktif apabila nilai value counter naik dan sampai ke 0.

Untuk menambahkan counter pada program, langkah-langkahnya adalah sebagai berikut :

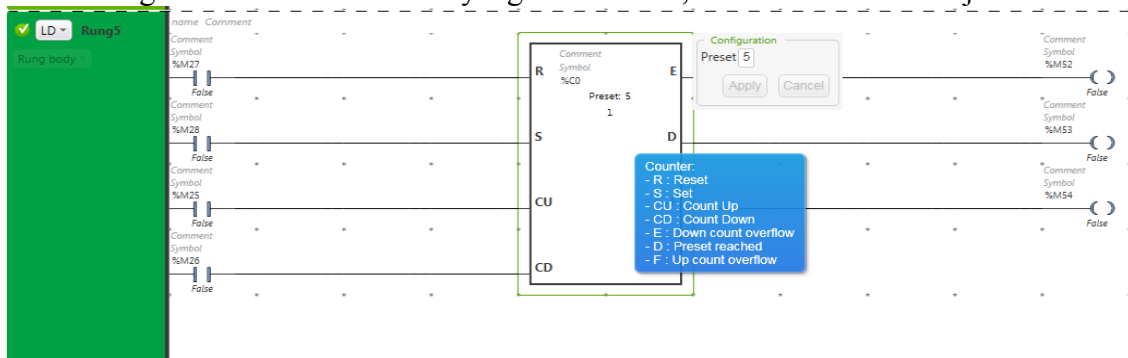
1. Klik pada ikon Function Block lalu pilih Counter.



2. Drag Object ke rung yang diinginkan, kemudian klik untuk menempatkan object pada rung.



3. Untuk mengatur Preset dari counter yang telah dibuat, arahkan cursor ke object.



## Comparison

Comparison atau compare digunakan untuk membandingkan nilai value dari satu memori address dengan memory yang lain atau dengan konstan value. Memory address yang akan dibandingkan harus sama datatypenya, misalkan Word harus dengan Word juga. Operator compare yang dapat digunakan pada program yaitu : < , <= , = , > , >= , <> .

Untuk menambahkan Comparison pada program, langkah-langkahnya adalah sebagai berikut :

1. Klik ikon Compare lalu drag object ke rung yang akan ditambahkan.



2. Klik pada rung yang dipilih untuk menempatkan object comparison.



3. Tulis instruksi compare yang akan dipakai, sebagai contoh yaitu : <(kurang dari).



4. Tampilannya pada saat runtime, apabila nilai %MW 20 < 10 maka %M55 akan On.

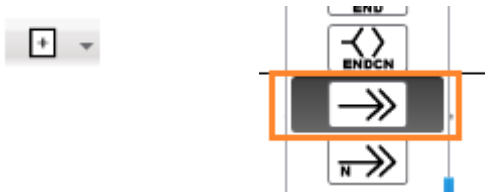


## Jump

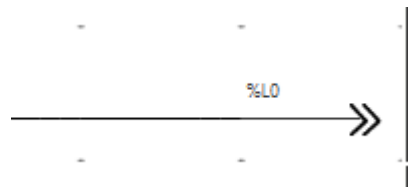
Jump merupakan instruksi yang digunakan untuk melompati sebuah program yang tidak ingin discanning atau dijalankan. Jump menggunakan label untuk menentukan batas area rung mana saja yang mau dilompati. Pada SoMachine Basic terdapat 2 tipe jump yaitu JMPC (jump apabila RLO = 1) dan JMPCN (jump apabila RLO = 0).

Langkah-langkah untuk menambahkan jump pada program adalah sebagai berikut :

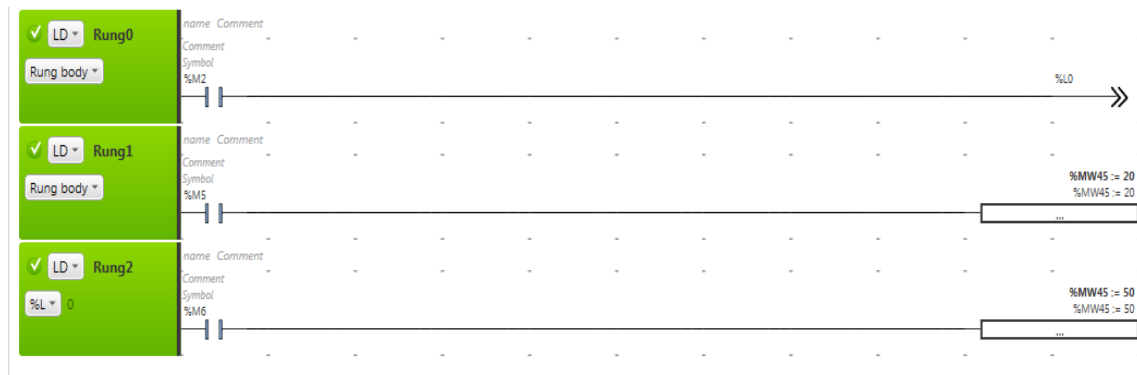
1. Pilih ikon Other Ladder item pada Tab Programming, kemudian cari Jump.



2. Drag object ke rung yang dikehendaki, lalu klik pada rung untuk menempatkan object jump. Masukkan Label jump %L0.



3. Buat program untuk mencoba instruksi jump.



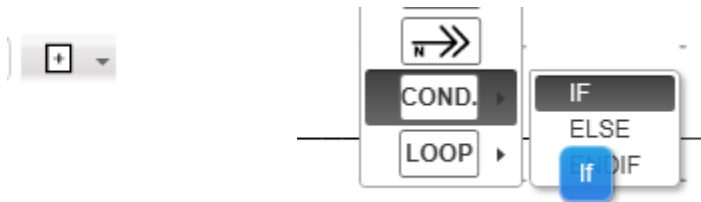
Perhatikan pada Rung ke 2 , disana dipilih %L artinya label untuk instruksi jump. Maka apabila program dijalankan dengan mengaktifkan %M2 , rung 1 akan dilompati dan walaupun %M5 diaktifkan tidak akan mempengaruhi nilai dari %MW45. Untuk meletakkan label penanda jump ini dapat diletakkan di rung mana saja sesuai keinginan.

## Conditional Elements

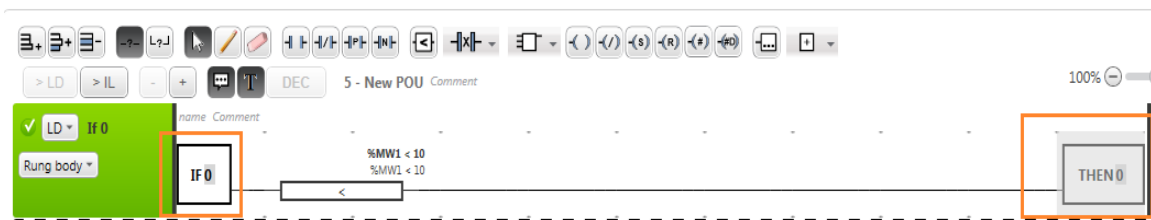
Conditional Elements merupakan instruksi yang digunakan untuk menambahkan kode pengkondisian pada program. Kode pengkondisian ini antara lain adalah **IF**, **ELSE**, **ENDIF**. Pada SoMachine Basic instruksi pengkondisian ini dapat ditambahkan sebanyak 128. Setiap rung hanya bisa diisi dengan 1 element instruksi/ 1 pengkondisian.

Untuk menggunakan instruksi pengkondisian ini pada program, langkah-langkahnya adalah sebagai berikut :

1. Masuk ke Tab Programming, lalu pilih ikon **Other Ladder items**, kemudian pilih **COND.**

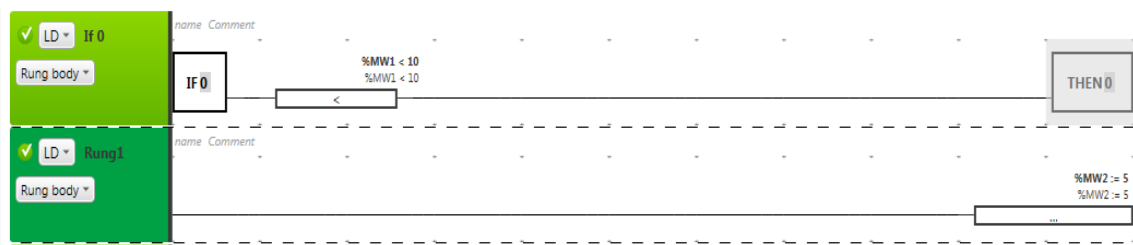


2. Pilih item yang akan digunakan, untuk pertama kali maka pilih **IF** dengan cara klik lalu drag object ke rung yang diinginkan. Klik untuk menempatkan object pada rung.

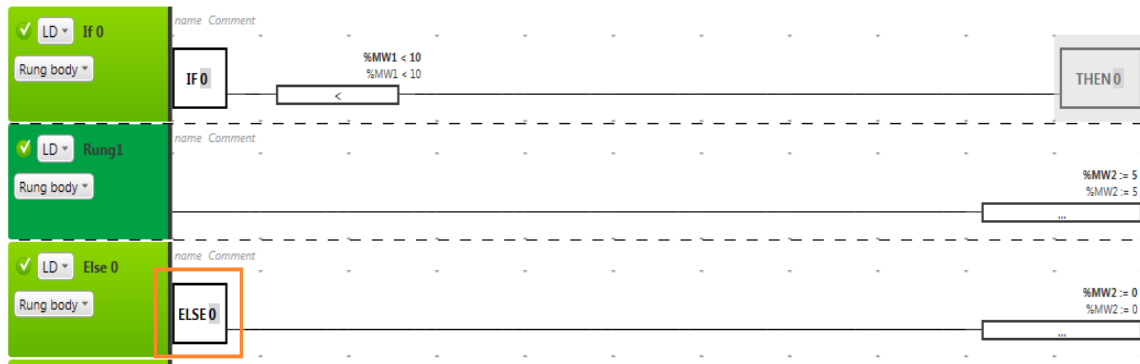


Pada saat menambahkan item object IF maka akan diikuti juga dengan item THEN, karena instruksi ini sepasang.

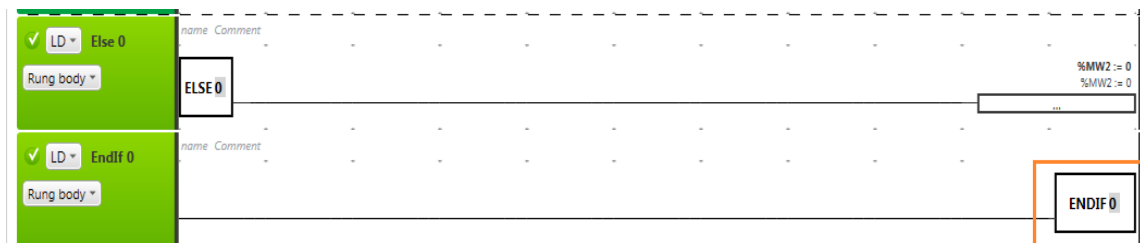
3. Tambahkan rung untuk **THEN**, rung program yang mau dijalankan apabila kondisi IF terpenuhi.



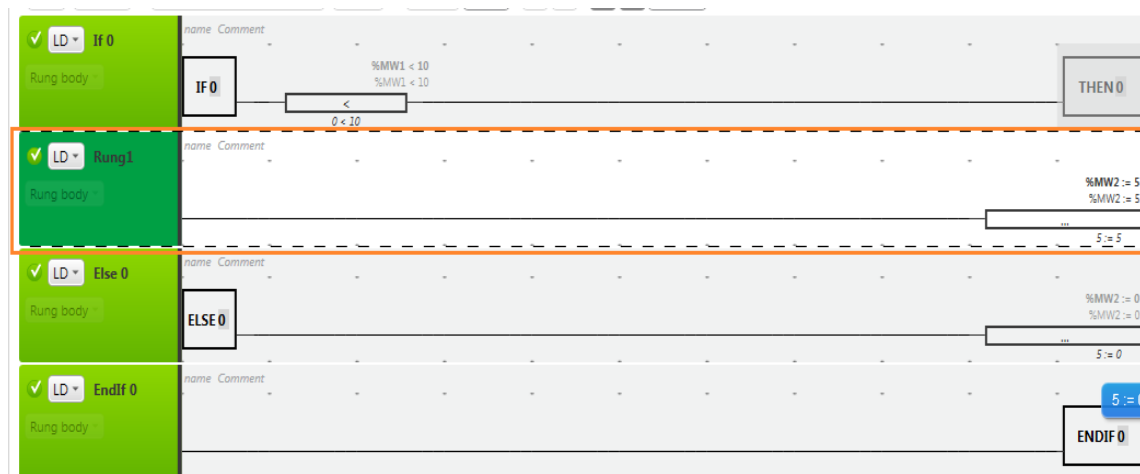
- Tambahkan **ELSE** (optional), jika program yang diinginkan harus menggunakan ELSE maka tambahkan object tersebut dengan cara yang sama seperti menambahkan object IF.



- Tambahkan **END IF** untuk menutup program pengkondisian. Tambahkan object tersebut dengan cara yang sama seperti menambahkan object ELSE.



- Tampilan pada saat runtime apabila nilai dari  $\%MW1 < 10$  maka nilai  $\%MW2 = 5$ .

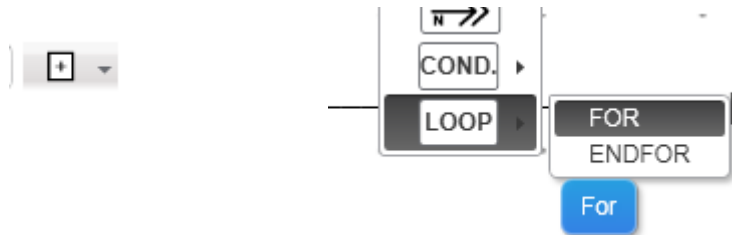


## Loop Elements

Loop Element merupakan instruksi yang digunakan untuk menambahkan kode sequence pada program. Kode sequence ini antara lain adalah **FOR**, **ENDFOR**. Pada SoMachine Basic instruksi Loop element ini dapat ditambahkan sebanyak 128. Setiap rung hanya bisa diisi dengan 1 element instruksi/ 1 Loop elements.

Untuk mulai menggunakan instruksi Loop pada program, maka langkah-langkahnya adalah sebagai berikut :

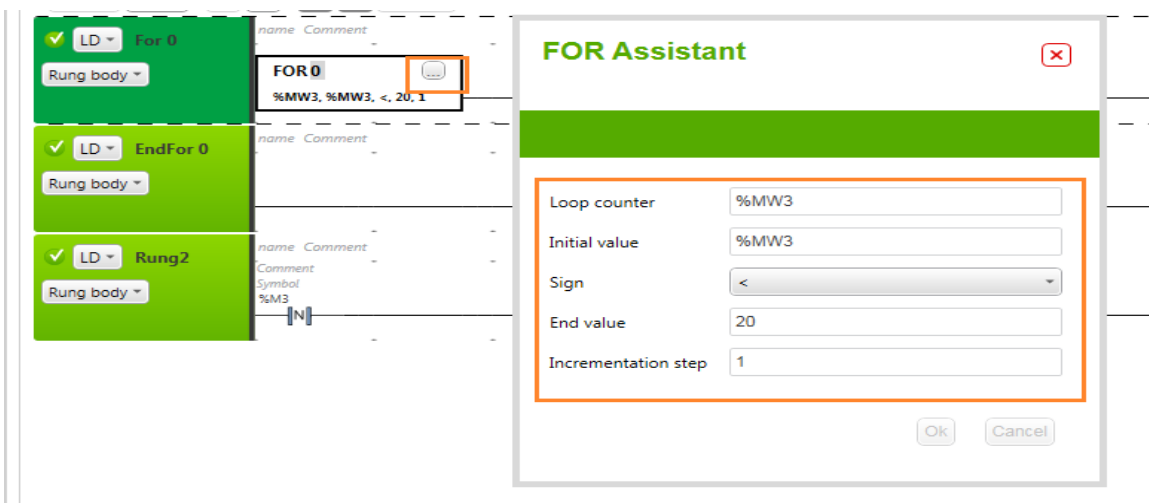
1. Masuk ke Tab Programming lalu pilih ikon **Other Ladder Items**. kemudian pilih **LOOP**.



2. Pilih item yang akan digunakan, pada Loop elements ini item yang perlu ditambahkan ke program terlebih dahulu yaitu **FOR**. Klik pada item For lalu drag object ke rung yang diinginkan. Klik untuk menempatkan object ke rung.

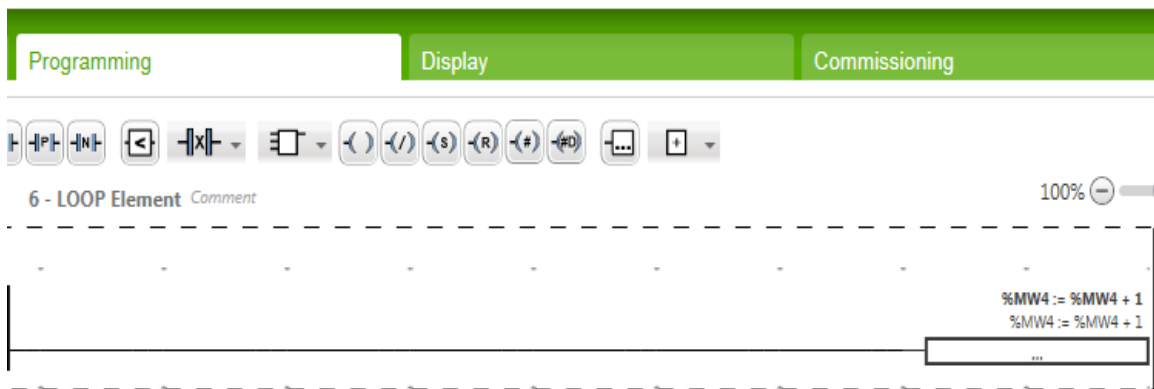


3. Untuk mengatur konfigurasi step serta counter pada Loop Elements maka klik ikon (...) di samping kanan instruksi **FOR**.



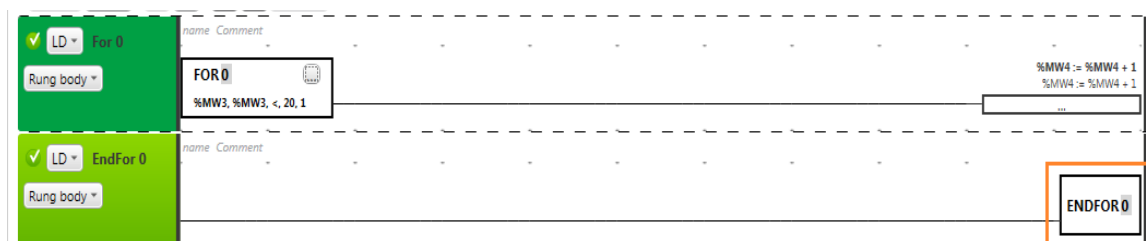


- Tambahkan kode yang mau dijalankan dengan instruksi **FOR**.

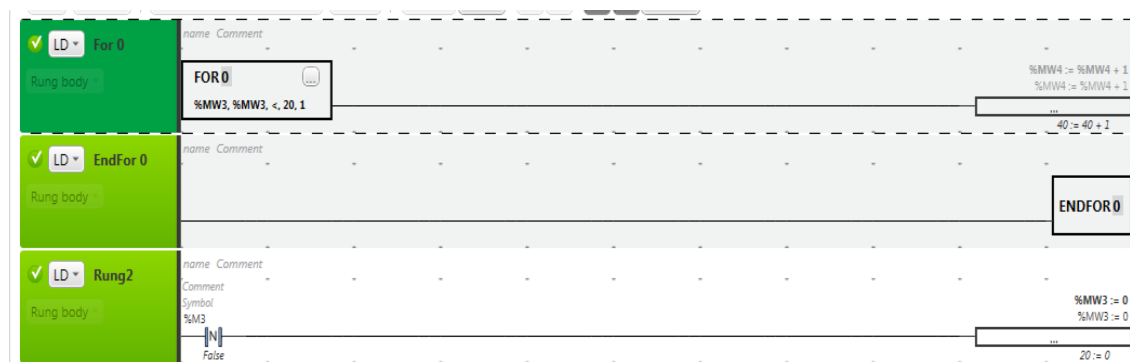


Kode ini artinya apabila loop counter pada instruksi **FOR** belum mencapai nilai end value maka nilai %MW 4 akan bertambah 1.

- Tambahkan instruksi **ENDFOR** agar program tidak error dan dapat dijalankan. Untuk menambahkan instruksi **ENDFOR** langkahnya sama seperti menambahkan instruksi **FOR**.



- Tampilan program pada saat runtime, apabila nilai %MW3(loop counter) < 20 maka nilai %MW4 akan bertambah. Jika nilai %MW3 direset maka akan langsung bertambah menjadi 20.



## Drum Assistant

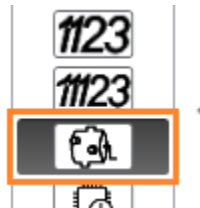
Drum Assistant merupakan instruksi step sequencer, Drum assistant dapat digunakan untuk mengaktifkan bit berdasarkan step. Step maksimalnya yaitu 8, sedangkan bit yang dapat dikonfigurasi setiap stepnya yaitu 16 bit / 1 Word.

Untuk menambahkan instruksi Drum Assistant pada program, langkah-langkahnya adalah sebagai berikut :

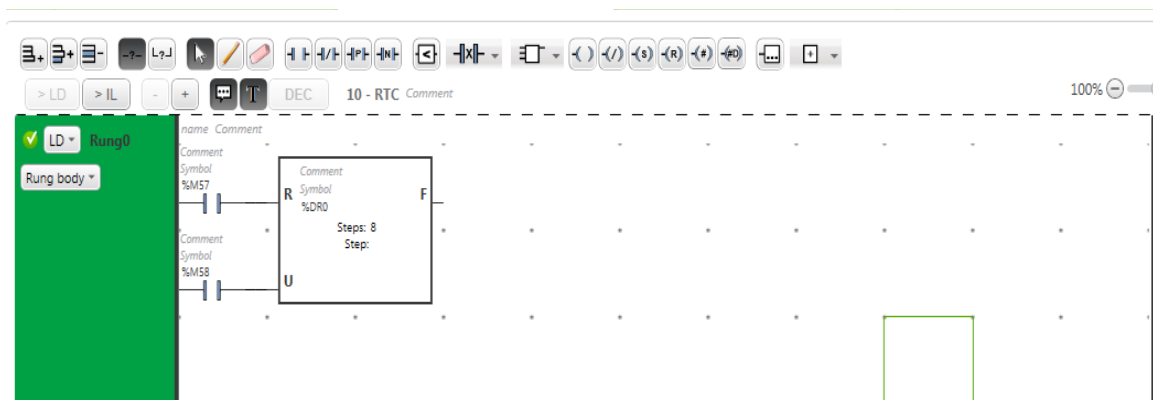
1. Masuk ke Tab Programming lalu pilih ikon function block.



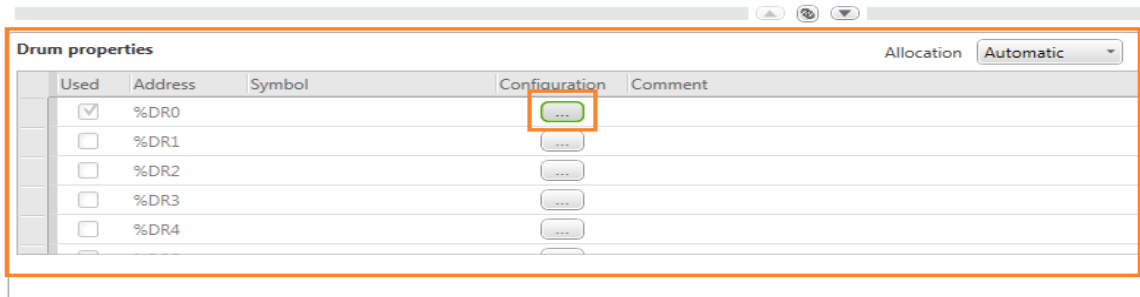
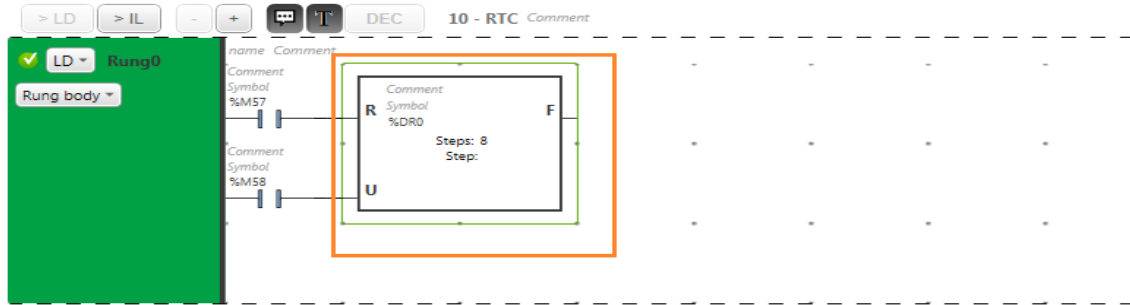
2. Cari ikon Drum untuk menambahkannya ke program.



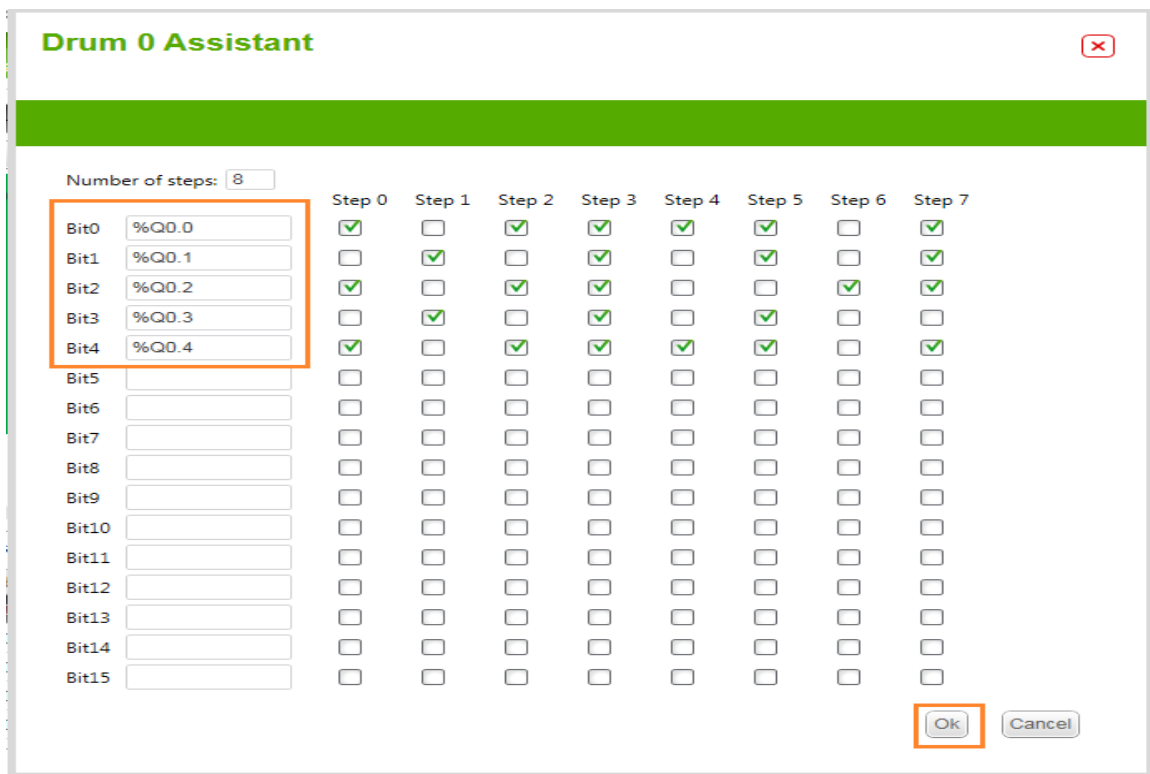
3. Klik object lalu Drag ke rung yang diinginkan. Klik untuk menempatkan object ke rung.



- Double klik pada object untuk membuka Drum Properties, kemudian pilih ikon (...) untuk membuka dialog Drum 0 Assistant.

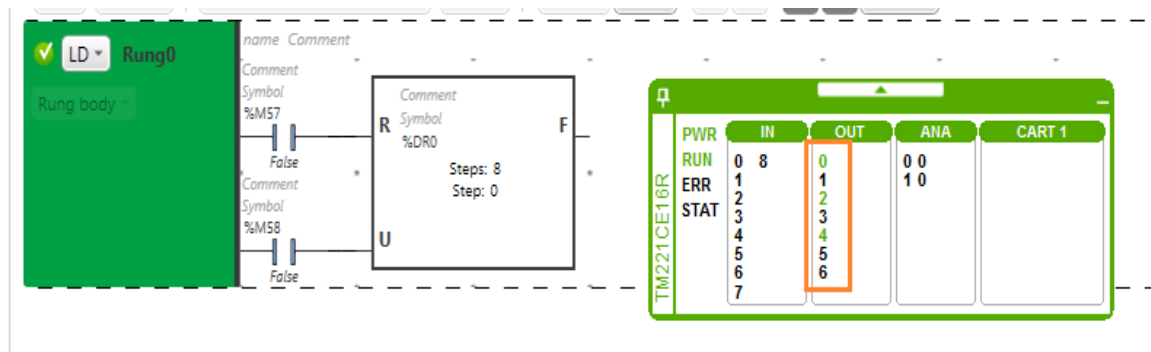


- Masukan address memory(bit) yang akan dikontrol kemudian atur ON/OFF nya menggunakan check-box di setiap step.

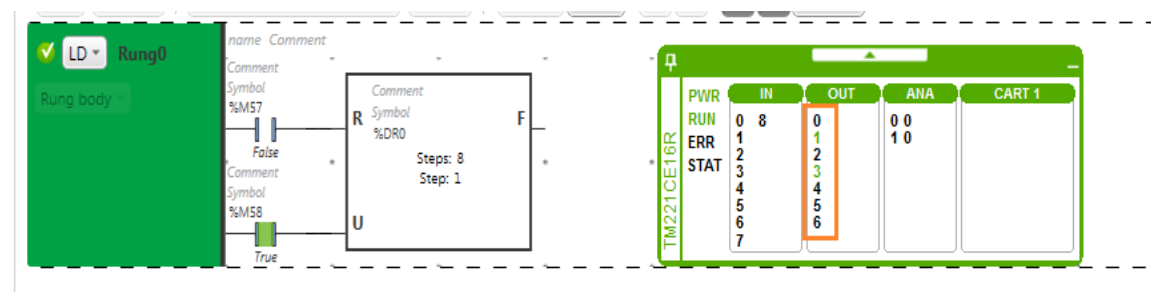


6. Jalankan simulasi untuk melihat hasilnya.

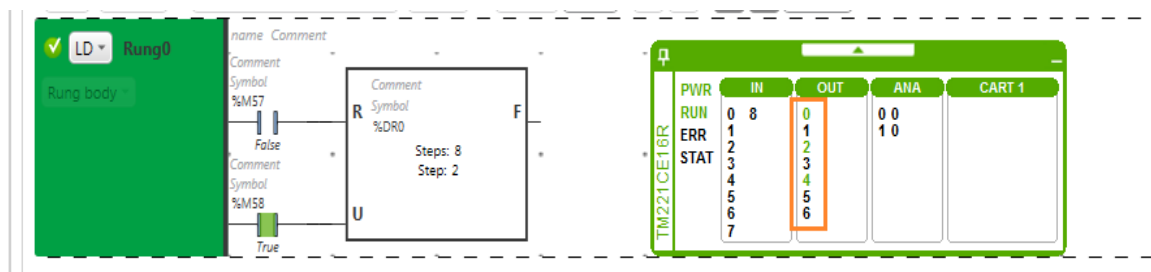
Step 0



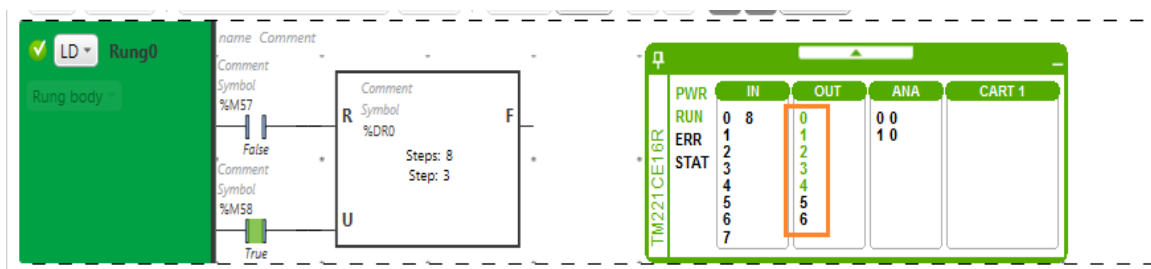
Step 1



Step 2



Step 3

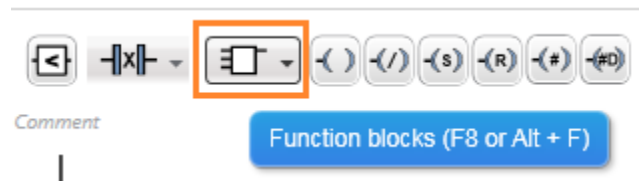


## Shift Bit Register

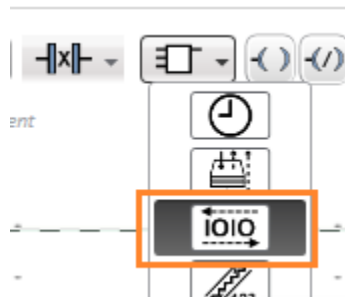
Shift Bit Register merupakan instruksi yang digunakan untuk menggeser bit ke kanan atau ke kiri. Pada SoMachine Basic instruksi Shift Register dapat menggunakan function ataupun function block. Untuk yang menggunakan function block maka maksimal instruksi yang dapat dibuat yaitu 8.

Untuk menambahkan instruksi Shift Bit Register langkah-langkahnya adalah sebagai berikut :

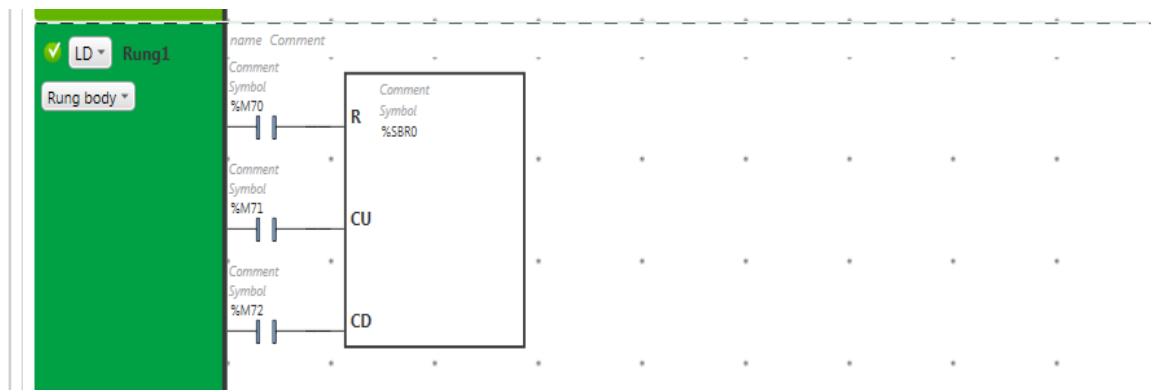
1. Masuk ke Tab Programming lalu pilih ikon function block untuk menambahkan instruksi Shift Bit Register.



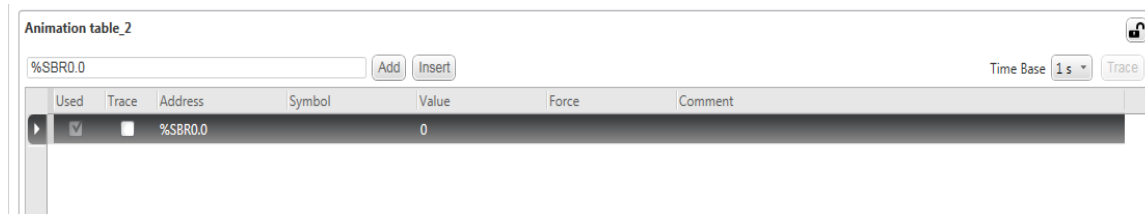
2. Pilih ikon Shift Bit Register untuk menambahkannya pada program.



3. Klik Object kemudian Drag object ke rung yang diinginkan. Klik untuk menempatkan object pada rung.

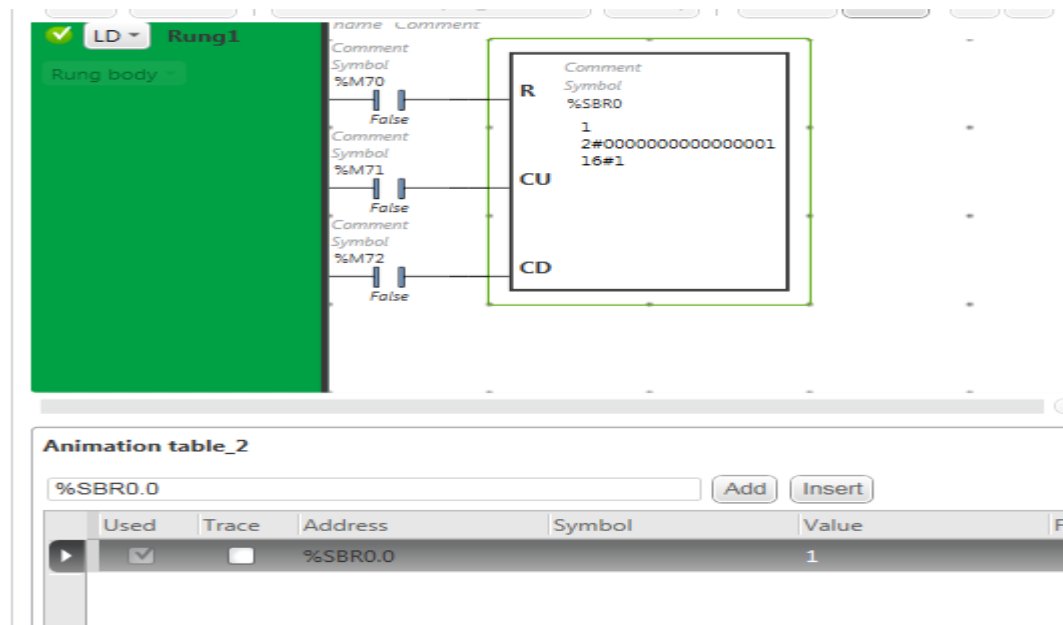


4. Tambahkan Animations table untuk mengubah nilai dari bit SBR0.



Bit 0 pada SBR0 maka penulisannya adalah %SBR0.0

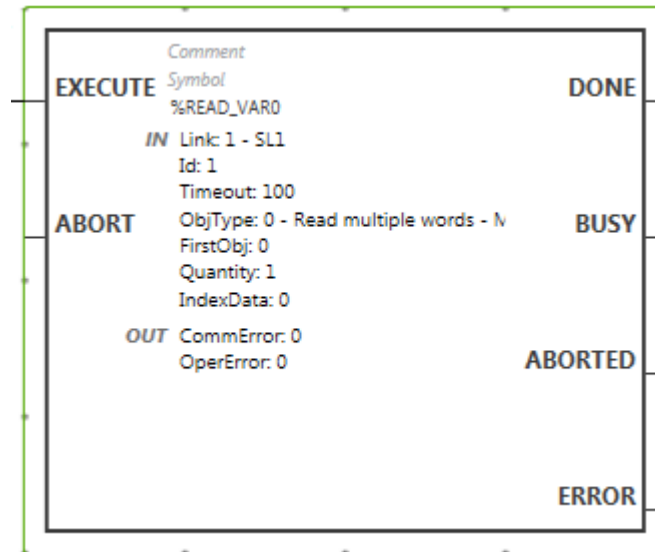
5. Jalankan Simulasi untuk memonitor program.



Apabila nilai bit dari SBR0 sudah diubah dan Input CU mendapatkan trigger maka bit akan bergeser ke kiri, sebaliknya apabila Input CD yang mendapatkan trigger maka bit akan begeser ke kanan.

## Read Var

Read Var merupakan instruksi yang digunakan untuk membaca data dari remote device pada Modbus Serial atau Modbus TCP. Jumlah maksimal instruksi ReadVar yang dapat dibuat pada program yaitu 16 Instruksi, dan apabila menggunakan 1 port yang sama maka eksekusi instruksi Readvar tidak boleh berjalan bersamaan pada satu waktu ( harus bergantian ).



Read Var function block memiliki input sebagai berikut :

- **Execute** : digunakan untuk mengeksekusi atau menjalankan instruksi.
- **Abort** : digunakan untuk membatalkan atau menggagalkan instruksi, jika fungsi Abort berhasil dijalankan maka akan muncul nilai 02Hex pada CommError.

Read Var function block memiliki output sebagai berikut ;

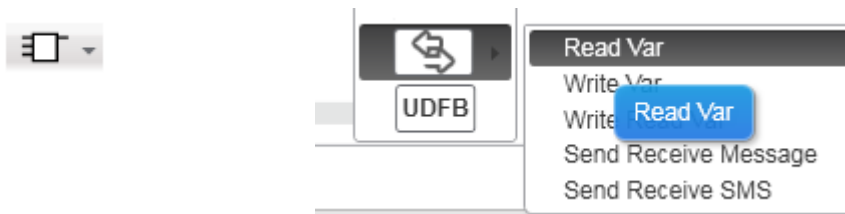
- **Done** : jika True artinya indikasi bahwa Read Var function block berhasil dieksekusi dan tanpa ada error yang muncul.
- **Busy** : jika True artinya indikasi bahwa Read Var function block sedang dieksekusi.
- **Aborted** : jika True artinya bahwa Read Var function block digagalkan melalui input Abort.
- **Error** : jika True artinya eksekusi Read Var function block terhenti karena adanya error, untuk melihat penyebab errornya maka dapat dilihat kode CommError(Communication Error) dan OperError(Operation Error).

Bagian configuration dari Read Var function block sebagai berikut :

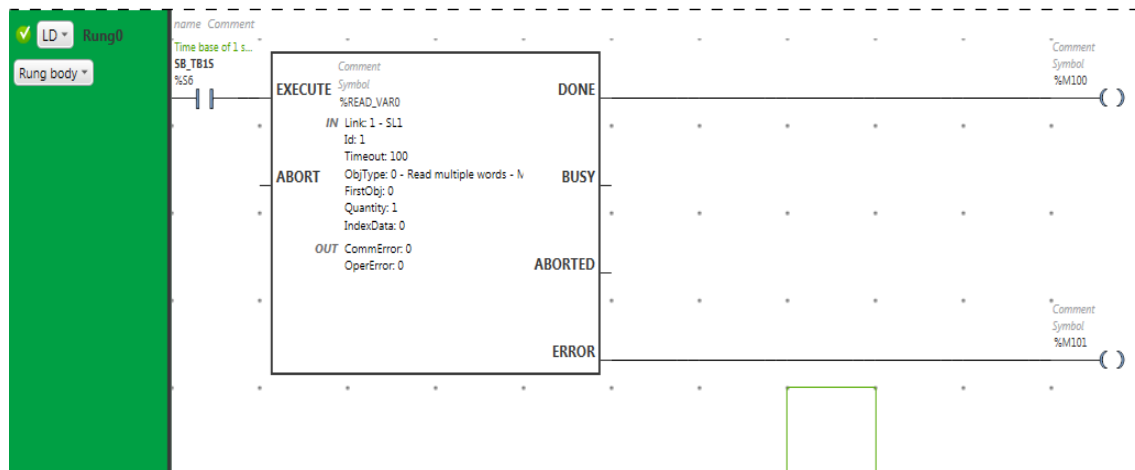
- **Link** : Pilihan port yang dapat dipilih.
- **Id** : Device identifier (1..247 untuk serial) dan (1..16 untuk Ethernet).
- **Timeout** : Waktu maksimum untuk menerima jawaban dari remote device.
- **ObjType** : Tipe kode untuk membaca data lewat modbus(0 = Holding Register, 1=Digital input, 2=Digital output, 3=Input Register).
- **FirstObj** : Alamat first object pada remote device yang mau dibaca.
- **Quantity** : Jumlah object yang mau di baca pada remote device.
- **IndexData** : alamat memory untuk menyimpan hasil pembacaan dari remote device.
- **Comment** : komentar untuk Readvar function block.

Untuk menambahkan Read Var function block pada program, langkah-langkahnya adalah sebagai berikut :

1. Klik ikon function block lalu cari ikon Communication dan pilih Read Var.

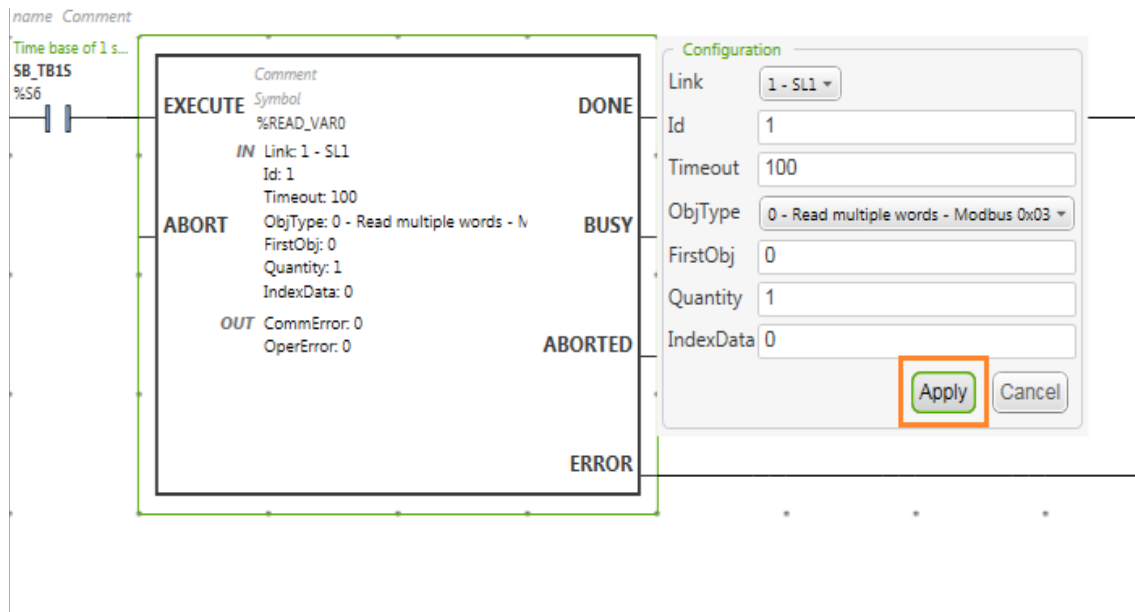


2. Drag object ke rung, lalu klik pada posisi yang diinginkan untuk meletakkan Read Var function block.





- Untuk mengatur configuration pada Read Var function block maka klik object. Klik Apply untuk menyetujui perubahan.

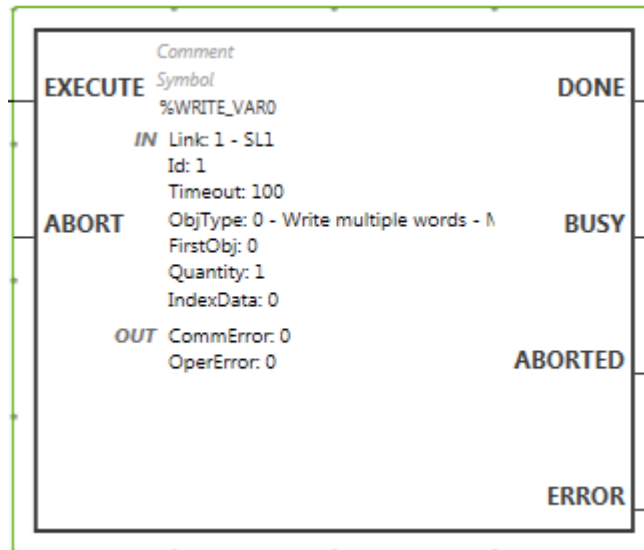


- Untuk melihat Read Var berapa saja yang sudah digunakan, dapat masuk ke Tools → Communication Object → Read Var . Tanda Used artinya sudah terpakai.

Used	Address	Symbol	Link	Id	Timeout	ObjType	FirstObj	Quantity	IndexData	Comment
<input checked="" type="checkbox"/>	%READ_VAR0		1 - SL1	1	100	0 - Read multiple words - Modbus 0x03	0	1	0	
<input type="checkbox"/>	%READ_VAR1		1 - SL1	1	100	0 - Read multiple words - Modbus 0x03	0	1	0	
<input type="checkbox"/>	%READ_VAR2		1 - SL1	1	100	0 - Read multiple words - Modbus 0x03	0	1	0	
<input type="checkbox"/>	%READ_VAR3		1 - SL1	1	100	0 - Read multiple words - Modbus 0x03	0	1	0	
<input type="checkbox"/>	%READ_VAR4		1 - SL1	1	100	0 - Read multiple words - Modbus 0x03	0	1	0	
<input type="checkbox"/>	%READ_VAR5		1 - SL1	1	100	0 - Read multiple words - Modbus 0x03	0	1	0	
<input type="checkbox"/>	%READ_VAR6		1 - SL1	1	100	0 - Read multiple words - Modbus 0x03	0	1	0	
<input type="checkbox"/>	%READ_VAR7		1 - SL1	1	100	0 - Read multiple words - Modbus 0x03	0	1	0	

## Write Var

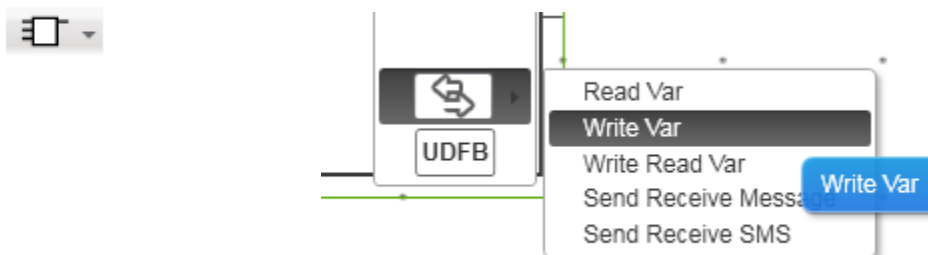
Write Var merupakan instruksi yang digunakan untuk menulis data dari PLC ke remote device menggunakan Modbus Serial atau Modbus TCP. Jumlah maksimal instruksi yang dapat dibuat yaitu 16, sama seperti Read Var apabila menggunakan 1 port yang sama tidak boleh eksekusi Write Var function block secara bersamaan (harus bergantian). Eksekusi Read Var dan Write Var secara bersamaan juga tidak bisa karena akan menimbulkan error.



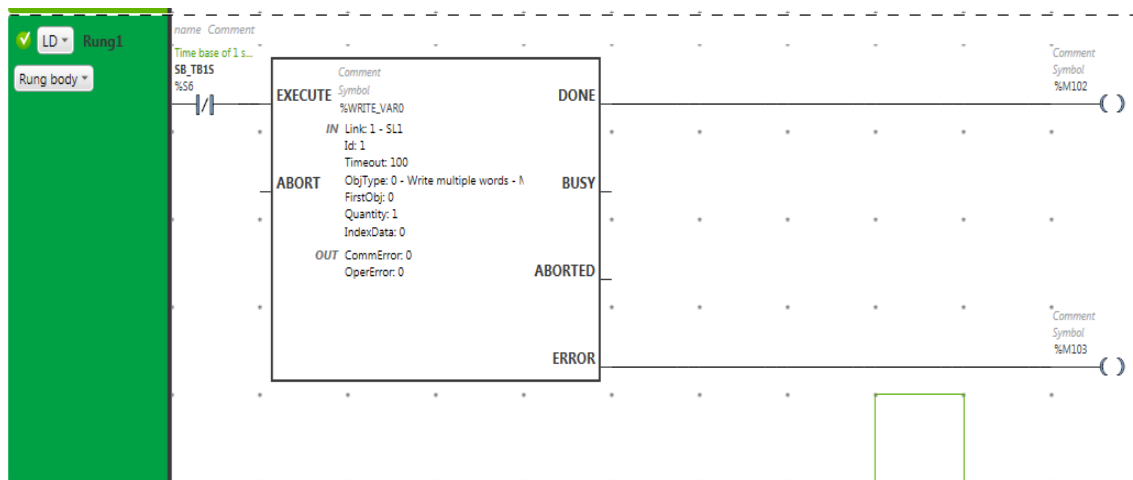
Untuk bagian input dan output pada Write Var sama seperti Read Var perbedaannya pada bagian configuration **ObjType** (0= Write Multiple Word register, 2=Write Multiple Bits(digital output), 4=Write Single Bits(digital output), 5=Write Single Word register).

Langkah-langkah untuk menambahkan Write Var function block ke program sebagai berikut :

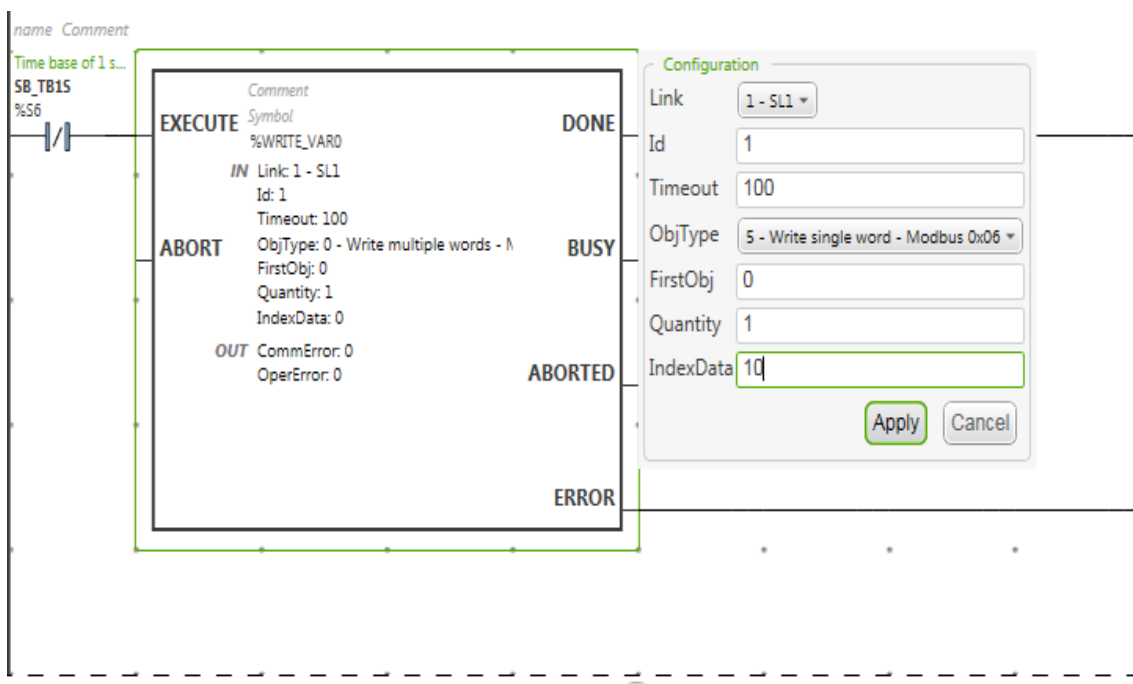
1. Klik ikon function block lalu cari ikon Communication dan pilih Write Var.



2. Drag object ke rung, lalu klik pada posisi yang diinginkan untuk meletakkan Write Var function block.

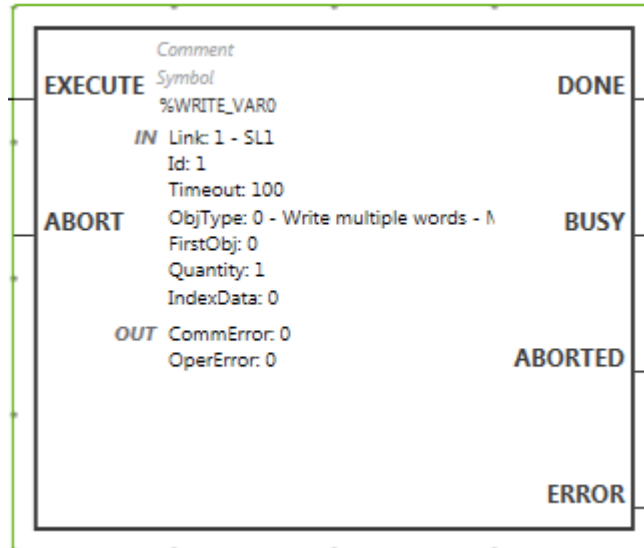


3. Untuk mengatur configuration maka klik pada object. Klik Apply untuk menyetujui perubahan.



## Write Read Var

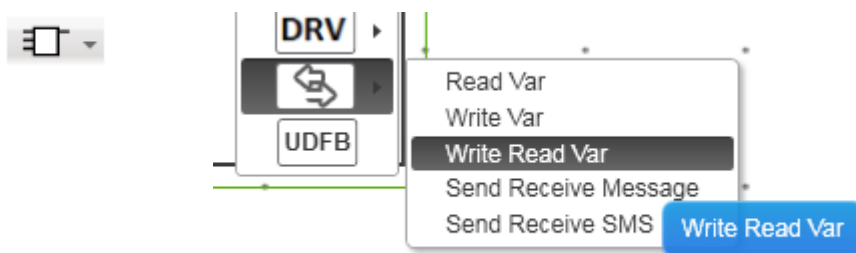
Write Read Var merupakan instruksi yang digunakan untuk membaca sekaligus menulis data pada remote device menggunakan Modbus Serial atau Modbus TCP. Terdapat 16 block instruksi yang dapat digunakan pada PLC M221, instruksi ini tidak boleh berjalan bersamaan antara satu function block dengan yang lainnya (harus bergantian). Dengan instruksi ini maka dalam sekali request dapat langsung menerima dan mengirim data.



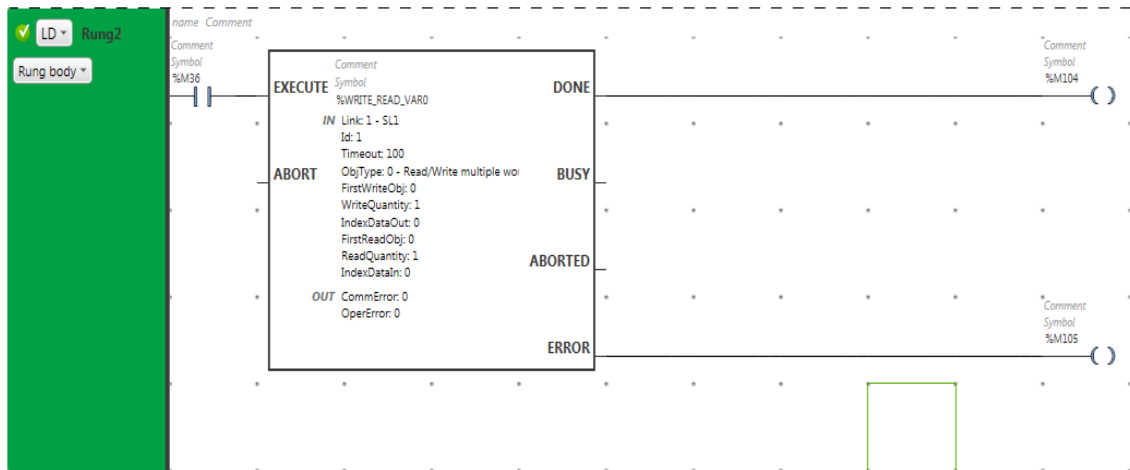
Untuk bagian input dan output pada Write Read Var sama seperti Read Var dan Write Var perbedaannya pada bagian configuration **ObjType** (0= Read Write Multiple Word register).

Langkah-langkah untuk menambahkan Write Read Var function block ke program sebagai berikut :

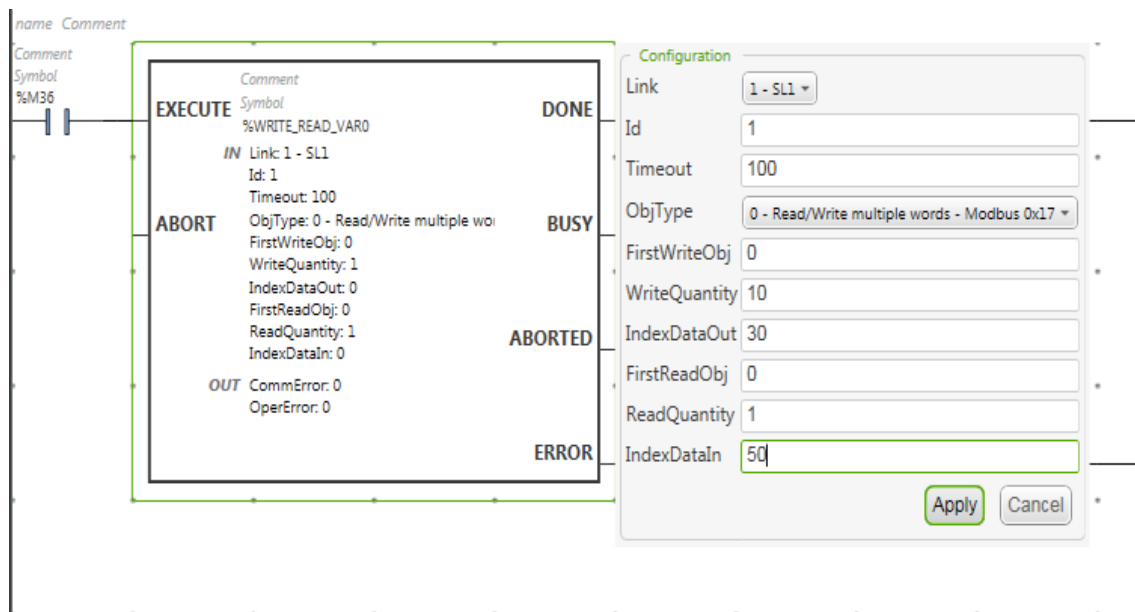
1. Klik ikon function block lalu cari ikon Communication dan pilih Write Read Var.



2. Drag object ke rung, lalu klik pada posisi yang diinginkan untuk meletakkan Write Read Var function block.

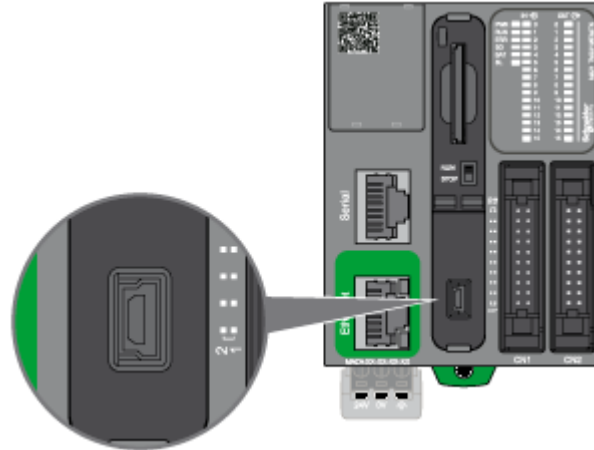


3. Untuk mengatur configuration maka klik pada object. Klik Apply untuk menyetujui perubahan.



## Communication

PLC M221 memiliki port Ethernet, Serial, dan USB Port. USB Port ini digunakan untuk mendownload program ke PLC pertama kali, hal ini untuk memudahkan apabila tidak mengetahui alamat IP dari PLC. Selanjutnya apabila port Ethernet sudah diatur IP-nya maka dapat melakukan transfer program menggunakan port Ethernet.



USB Port yang tersedia pada PLC M221 menggunakan connector tipe Mini-B.

### Ethernet Port

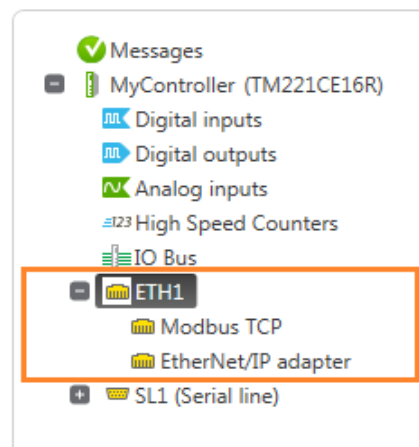
Untuk mulai menggunakan port Ethernet maka harus mengatur terlebih dahulu IP-nya. PLC M221 memiliki 3 tipe pengaturan IP yaitu :

- DHCP
- BOOTP
- Fixed IP

Untuk memudahkan maka gunakan tipe Fixed IP agar kita hanya perlu mengatur IP Address, Subnet Mask, dan Gateway Address.

Untuk mengatur IP tersebut langkah-langkahnya adalah sebagai berikut :

1. Masuk ke tab Configuration, pilih **ETH1**.



2. Tampilan jika belum diatur maka IP-nya.

**Ethernet**

Device name: M221

☐ IP address by DHCP  
☐ IP address by BOOTP  
☒ Fixed IP address

IP address: 0 . 0 . 0 . 0  
Subnet mask: 0 . 0 . 0 . 0  
Gateway address: 0 . 0 . 0 . 0

Transfer Rate: Auto

**Security Parameters**

☒ Programming protocol enabled  
☒ EtherNet/IP protocol enabled  
☒ Modbus server enabled  
☒ Auto discovery protocol enabled

Apply Cancel

3. Atur IP address, Subnet mask, Gateway address sesuai keinginan. Klik apply untuk menyetujui perubahan.

**Ethernet**

Device name: M221

☐ IP address by DHCP  
☐ IP address by BOOTP  
☒ Fixed IP address

IP address: 192 . 168 . 0 . 1  
Subnet mask: 255 . 255 . 255 . 0  
Gateway address: 0 . 0 . 0 . 0

Transfer Rate: Auto

**Security Parameters**

☒ Programming protocol enabled  
☒ EtherNet/IP protocol enabled  
☒ Modbus server enabled  
☒ Auto discovery protocol enabled

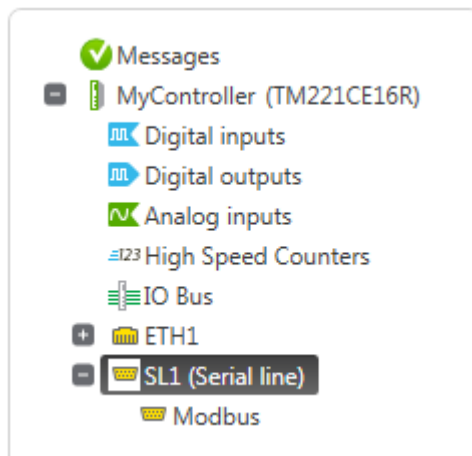
Apply Cancel

## Serial Port

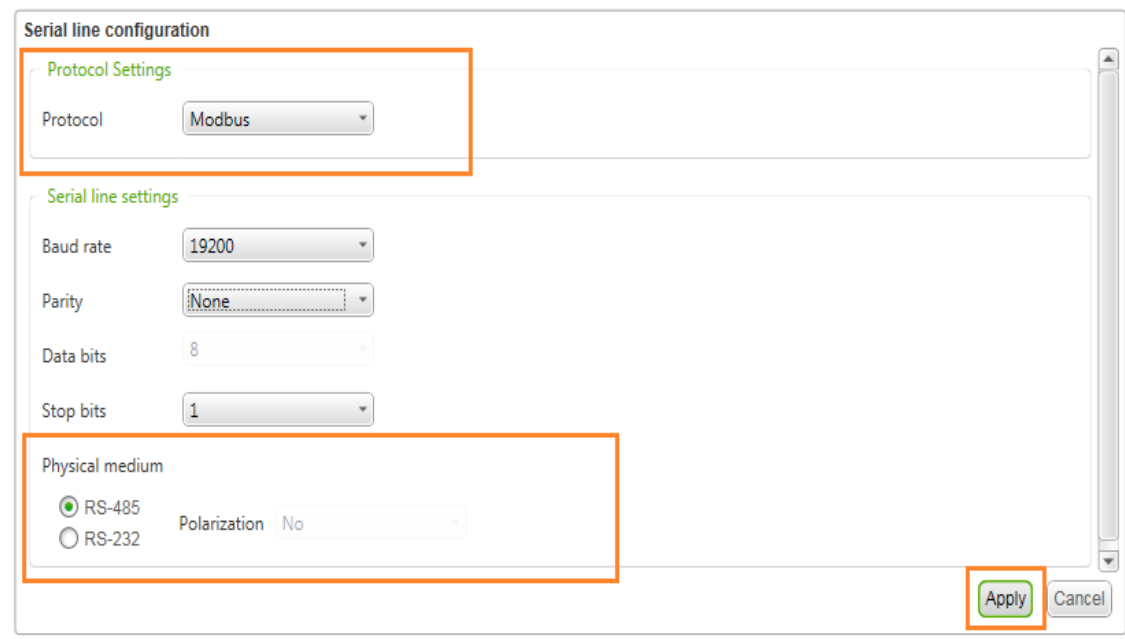
PLC M221 memiliki setidaknya 1 Serial Port yang dapat digunakan untuk komunikasi, port ini menggunakan connector RJ45. Physical medium bisa memilih RS232 atau RS485, Protocol yang tersedia yaitu : Modbus RTU, Modbus ASCII, Modbus Serial IOScanner.

Untuk mulai menggunakan port Serial maka langkah-langkahnya sebagai berikut :

1. Masuk ke Tab Configuration, lalu pilih **SL1(Serial Line)**.



2. Pilih protocol yang mau digunakan serta physical mediumnya. Klik Apply untuk menyetujui perubahan.



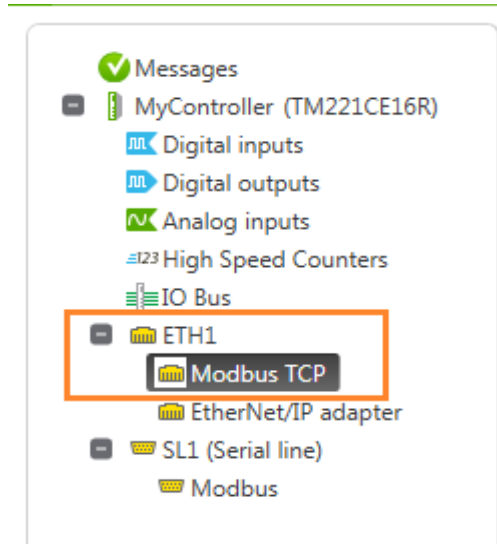


## Modbus TCP

Modbus TCP merupakan modbus yang digunakan pada jaringan TCP/IP. Jika mau menggunakan pada PLC M221 maka harus memilih yang ada port Ethernetnya. Jumlah device yang dapat terhubung ke PLC menggunakan Modbus TCP maksimal 16 device.

Langkah-langkah untuk menggunakan Modbus TCP adalah sebagai berikut :

1. Masuk ke Tab Configuration, Pilih Modbus TCP.



2. Tampilan konfigurasi Modbus TCP, pada kolom IP address isikan dengan address client yang akan dihubungkan melalui Modbus TCP. Klik Apply untuk menyetujui perubahan.

**Modbus TCP**

**Modbus mapping**

☐ Enabled   Unit ID    Output registers (%IWM)    Input registers (%QWM)

**Client mode: remote device table (max 16)**

☐ Enable Modbus TCP IOScanner

IP address    ☒ Generic   ☐ Drive   ☐ ATV12   ☐ Predefined     

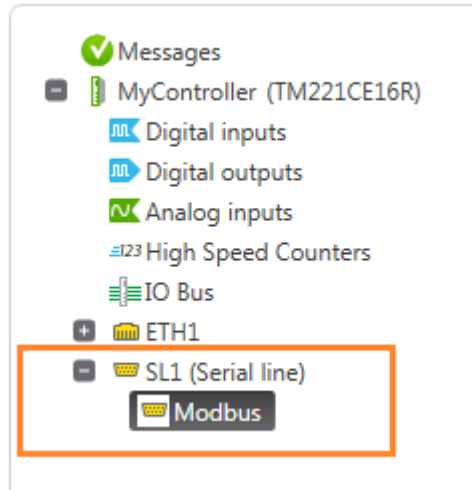
ID	Name	Address	Type	Index	IP address	Response	Reset variable	Scanned	Init Res
0	Device 0		Generic device	1	192.168.0.5	10		<input type="checkbox"/>	255

## Modbus Serial

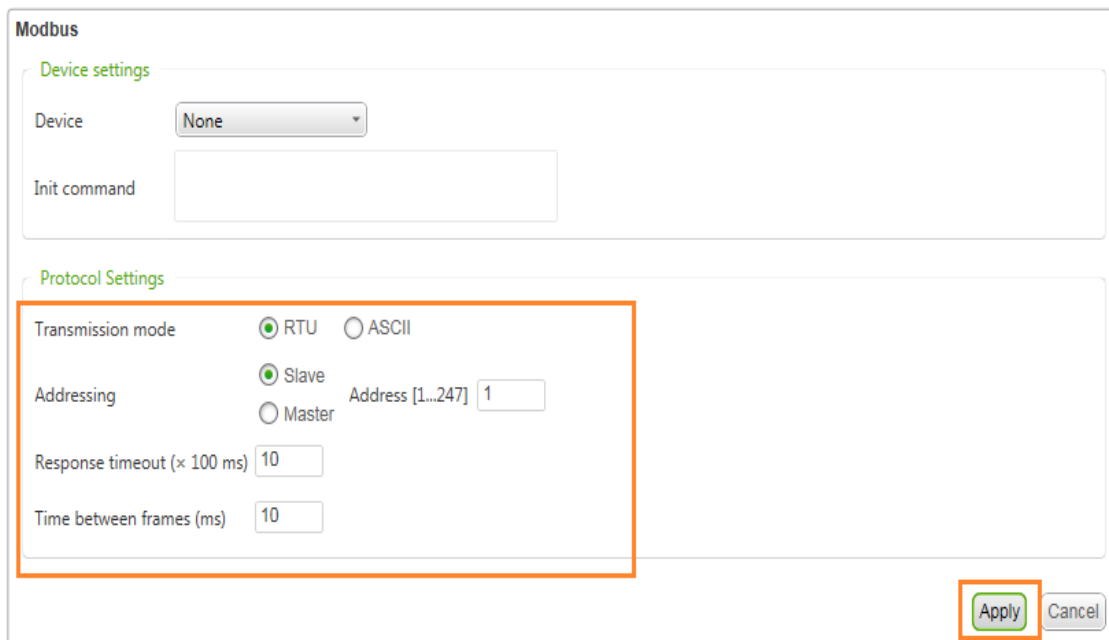
Modbus Serial terbagi menjadi 2 macam berdasarkan mode yaitu Modbus RTU dan Modbus ASCII. Menggunakan Modbus Serial ini perlu menentukan Master dan Slave, serta mengatur address masing – masing slave dengan tepat. PLC M221 dapat terhubung dengan maksimal 16 devices menggunakan modbus Serial.

Langkah-langkah untuk menggunakan modbus Serial sebagai berikut :

1. Masuk ke Tab Configuration, lalu pilih Modbus pada SL1(Serial Line).



2. Pilih mode transiminya(RTU/ASCII), tentukan addressingnya(Master/Slave), atur response timeoutnya.Klik Apply untuk menyetujui perubahan.

A screenshot of the 'Modbus' configuration window. It has two main sections: 'Device settings' and 'Protocol Settings'. In 'Device settings', 'Device' is set to 'None' and 'Init command' is empty. In 'Protocol Settings', 'Transmission mode' has 'RTU' selected (radio button). 'Addressing' has 'Slave' selected (radio button) and 'Address [1...247]' is set to '1'. 'Response timeout (× 100 ms)' is set to '10' and 'Time between frames (ms)' is set to '10'. The 'Protocol Settings' section is highlighted with an orange box. At the bottom right, there are 'Apply' and 'Cancel' buttons, with 'Apply' highlighted by an orange box.

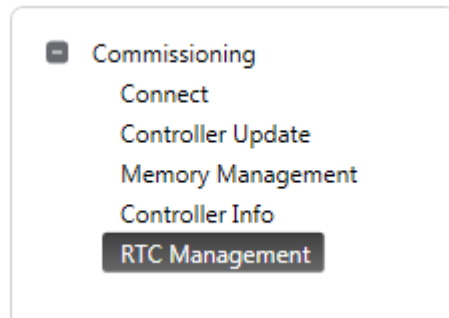
## Mengatur Jam RTC

PLC M221 memiliki jam RTC ( Real Time Clock ) yang dapat disetting nilainya sesuai dengan waktu saat ini. Jam RTC perlu diatur seakurat mungkin apabila program yang dibuat menggunakan waktu untuk menjalankan instruksi secara otomatis. Untuk mengatur jam pada PLC cara yang dapat dilakukan yaitu :

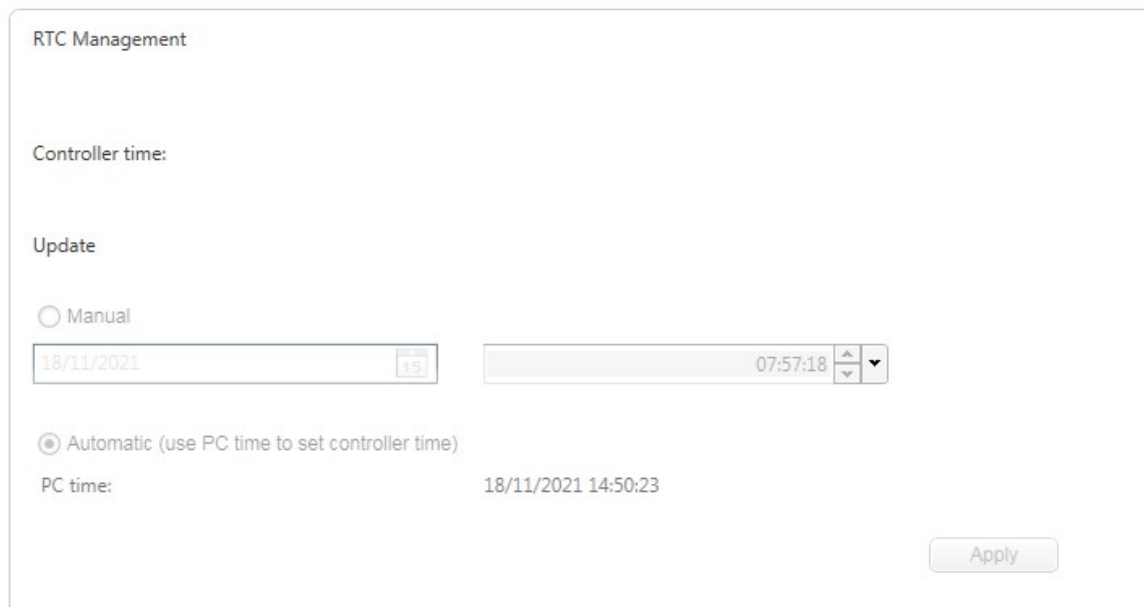
- Mengatur melalui RTC Management yang terdapat pada Commisioning tab.

Langkah-langkah untuk mengatur Jam RTC melalui Commisioning tab adalah sebagai berikut :

1. Masuk ke Commisioning tab lalu pilih RTC Management.



2. RTC dapat diatur secara manual maupun otomatis (default) apabila otomatis maka waktunya mengikuti waktu PC yang digunakan.

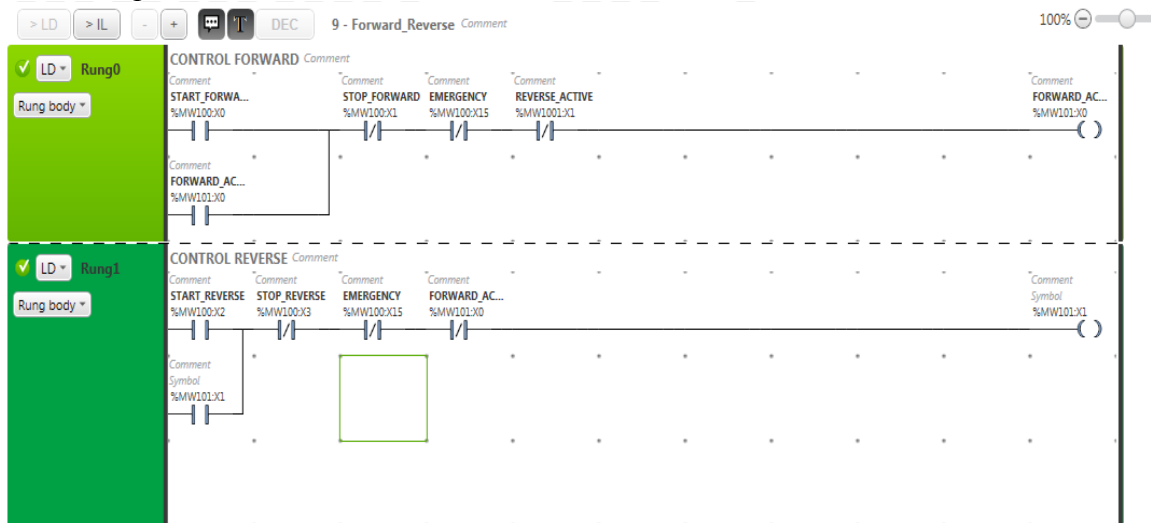
A screenshot of the 'RTC Management' configuration window. It features a 'Controller time:' label, an 'Update' section with radio buttons for 'Manual' and 'Automatic (use PC time to set controller time)'. The 'Automatic' option is selected. Below this, there is a date field showing '18/11/2021' and a time field showing '07:57:18'. At the bottom, the 'PC time:' is displayed as '18/11/2021 14:50:23'. An 'Apply' button is located in the bottom right corner.

## Simulasi SoMachine Basic ke Ecostruxure SCADA Expert

Program yang telah dibuat pada SoMachine Basic dapat disimulasikan dengan project Ecostruxure SCADA Expert tanpa harus menggunakan hardware PLC.

Langkah-langkah untuk menghubungkan keduanya adalah sebagai berikut :

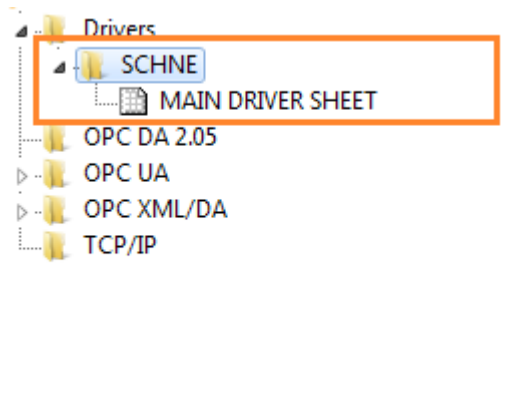
1. Buat program baru pada SoMachine. Sebagai contoh disini yaitu program untuk mengotrol motor berputar Forward dan Reverse.



2. Buat Tag untuk mengontrol program melalui Scada.

53	Start_Forward	0	Boolean
54	Start_Reverse	0	Boolean
55	Stop_Forward	0	Boolean
56	Stop_Reverse	0	Boolean
57	Emergency	0	Boolean
58	Forward_Active	0	Boolean
59	Reverse_Active	0	Boolean

3. Tambahkan Driver untuk komunikasi antara SoMachine dengan Scada Expert.



Untuk **PLC M221** Communication Drivernya dapat menggunakan **SCHNE**.

4. Hubungkan Tag yang telah dibuat tadi dengan address memory pada program SoMachine Basic.

Description:  
MAIN DRIVER SHEET

Disable:

Read Completed: readCompleted      Read Status: readStatus

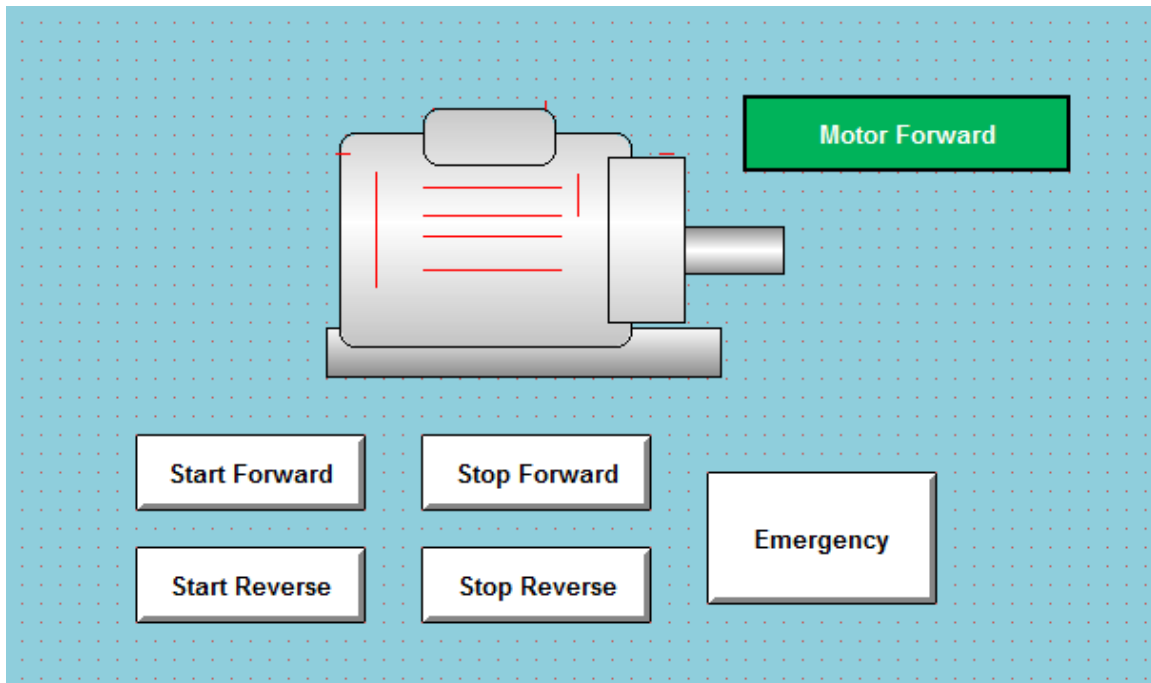
Write Completed: writeCompleted      Write Status: writeStatus

Min:       Max:

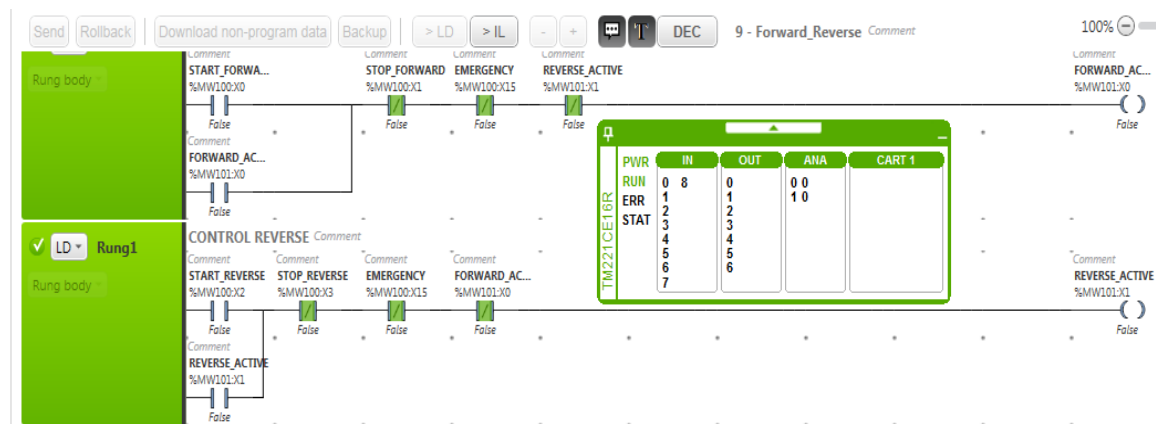
	Tag Name	Station	I/O Address	
	<input type="text" value="Filter text"/>	<input type="text" value="Filter text"/>	<input type="text" value="Filter text"/>	
1	Start_Forward	192.168.0.1	%MW100.0	F
2	Start_Reverse	192.168.0.1	%MW100.2	F
3	Stop_Forward	192.168.0.1	%MW100.1	F
4	Stop_Reverse	192.168.0.1	%MW100.3	F
5	Emergency	192.168.0.1	%MW100.15	F
6	Forward_Active	192.168.0.1	%MW101.0	F
7	Reverse_Active	192.168.0.1	%MW101.1	F

Pada Kolom Station / IP PLC apabila hanya menggunakan simulator maka isikan dengan IP PC yang dipakai.

5. Buat tampilan untuk mengontrol program yang telah dibuat.

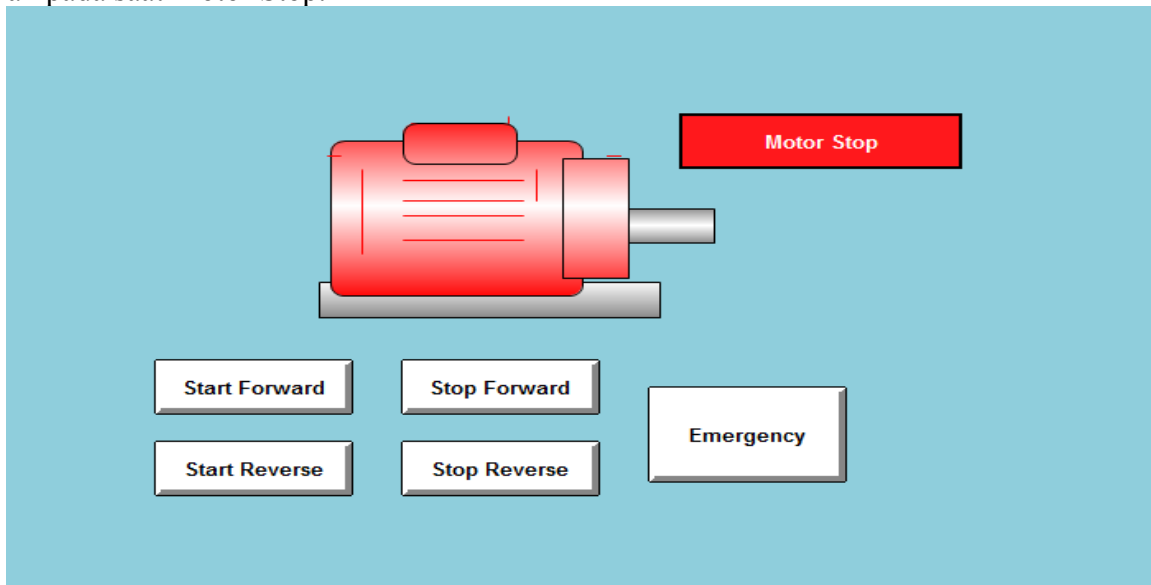


6. Jalankan simulasi pada Software SoMachine.

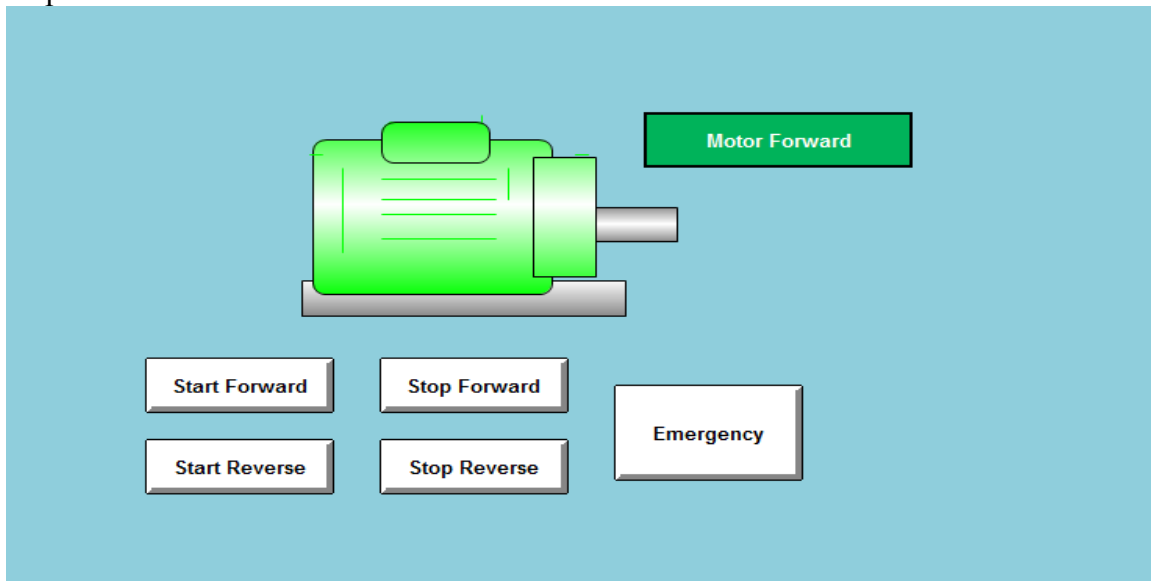


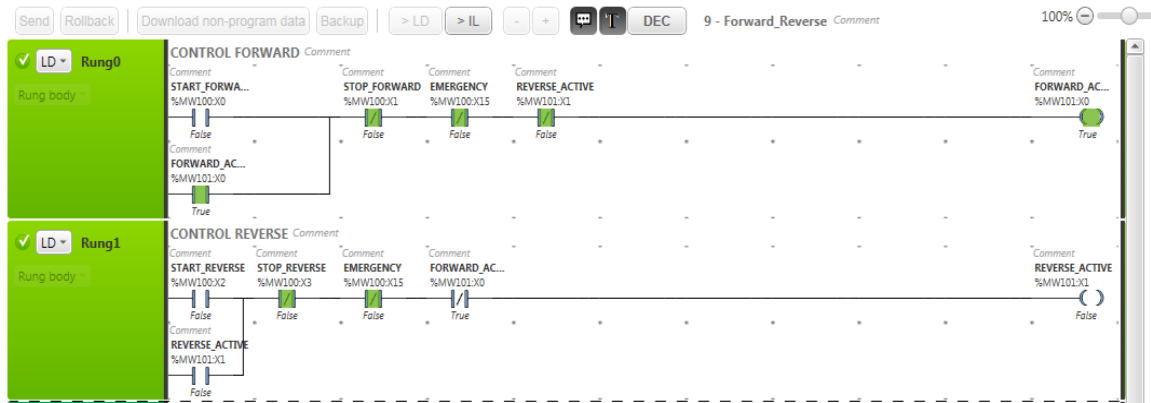
7. Jalankan runtime pada Scada.

Tampilan pada saat Motor Stop.



Tampilan pada saat Motor Forward.





Tampilan pada saat Motor Reverse.

