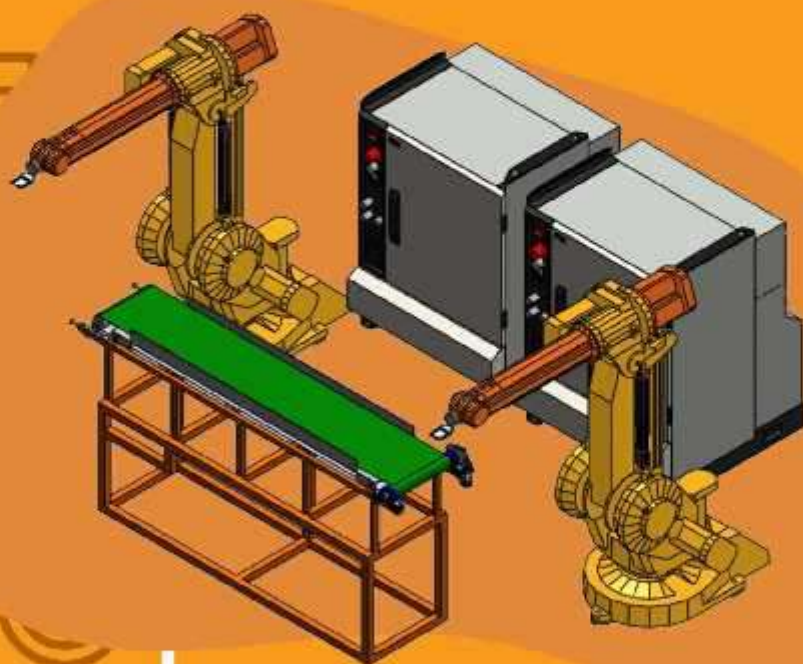




KOMUNIKASI ROBOT DAN PLC



IGNATIUS DERADJAD PRANOWO



KOMUNIKASI ROBOT DAN PLC

UU No 28 tahun 2014 tentang Hak Cipta

Fungsi dan sifat hak cipta Pasal 4

Hak Cipta sebagaimana dimaksud dalam Pasal 3 huruf a merupakan hak eksklusif yang terdiri atas hak moral dan hak ekonomi.

Pembatasan Pelindungan Pasal 26

Ketentuan sebagaimana dimaksud dalam Pasal 23, Pasal 24, dan Pasal 25 tidak berlaku terhadap:

- i. Penggunaan kutipan singkat Ciptaan dan/atau produk Hak Terkait untuk pelaporan peristiwa aktual yang ditujukan hanya untuk keperluan penyediaan informasi aktual;
- ii. Penggandaan Ciptaan dan/atau produk Hak Terkait hanya untuk kepentingan penelitian ilmu pengetahuan;
- iii. Penggandaan Ciptaan dan/atau produk Hak Terkait hanya untuk keperluan pengajaran, kecuali pertunjukan dan Fonogram yang telah dilakukan Pengumuman sebagai bahan ajar; dan
- iv. Penggunaan untuk kepentingan pendidikan dan pengembangan ilmu pengetahuan yang memungkinkan suatu Ciptaan dan/atau produk Hak Terkait dapat digunakan tanpa izin Pelaku Pertunjukan, Produser Fonogram, atau Lembaga Penyiaran.

Sanksi Pelanggaran Pasal 113

1. Setiap Orang yang dengan tanpa hak melakukan pelanggaran hak ekonomi sebagaimana dimaksud dalam Pasal 9 ayat (1) huruf i untuk Penggunaan Secara Komersial dipidana dengan pidana penjara paling lama 1 (satu) tahun dan/atau pidana denda paling banyak Rp100.000.000 (seratus juta rupiah).
2. Setiap Orang yang dengan tanpa hak dan/atau tanpa izin Pencipta atau pemegang Hak Cipta melakukan pelanggaran hak ekonomi Pencipta sebagaimana dimaksud dalam Pasal 9 ayat (1) huruf c, huruf d, huruf f, dan/atau huruf h untuk Penggunaan Secara Komersial dipidana dengan pidana penjara paling lama 3 (tiga) tahun dan/atau pidana denda paling banyak Rp500.000.000,00 (lima ratus juta rupiah).

KOMUNIKASI ROBOT DAN PLC

Ignatius Deradjad Pranowo



Cerdas, Bahagia, Mulia, Lintas Generasi.

KOMUNIKASI ROBOT DAN PLC

Ignatius Deradjad Pranowo

Desain Cover :
Rulie Gunadi

Sumber :
<https://www.shutterstock.com/>

Tata Letak :
Gofur Dyah Ayu

Proofreader :
Mira Muarifah

Ukuran :
x, 112 hlm, Uk: 15.5x23 cm

ISBN :
978-623-02-3363-0

Cetakan Pertama :
September 2021

Hak Cipta 2021, Pada Penulis

Isi diluar tanggung jawab percetakan

Copyright © 2021 by Deepublish Publisher
All Right Reserved

Hak cipta dilindungi undang-undang
Dilarang keras menerjemahkan, memfotokopi, atau
memperbanyak sebagian atau seluruh isi buku ini
tanpa izin tertulis dari Penerbit.

PENERBIT DEEPUBLISH
(Grup Penerbitan CV BUDI UTAMA)

Anggota IKAPI (076/DIY/2012)

Jl.Rajawali, G. Elang 6, No 3, Drono, Sardonoharjo, Ngaglik, Sleman

Jl.Kaliurang Km.9,3 - Yogyakarta 55581

Telp/Faks: (0274) 4533427

Website: www.deepublish.co.id

www.penerbitdeepublish.com

E-mail: cs@deepublish.co.id

PRAKATA

Kita tentu masih ingat pada masa ketika kereta kuda digantikan oleh adanya kendaraan berbahan bakar bensin yaitu mobil. Hal itu terjadi pada sekitar awal abad ke-20 di mana telah terjadi banyak perubahan dan pergeseran fungsi-fungsi. Dengan adanya pergeseran tersebut menyebabkan banyak bengkel kereta kuda tutup dan digantikan dengan munculnya banyak bengkel mobil. Peran dan pekerjaan para pekerja bengkel kereta kuda tergantikan oleh teknisi bengkel otomotif. Demikianlah situasi disrupsi yang tidak bisa kita hindari lagi pada jaman ini.

Kita juga tahu sejarah perkembangan terjadinya revolusi di dunia industri yang dipicu oleh adanya disrupsi tadi. Revolusi industri 1.0 dengan perubahan melalui penggunaan secara besar-besaran sistem tenaga uap menggantikan tenaga manusia dengan penemuan mesin uap oleh James Watt pada abad ke-18 untuk industri tekstil; biasa disebut dengan era mekanisasi. Revolusi industri 2.0 muncul awal abad ke-20 dengan penemuan tenaga listrik untuk sistem penggerak mesin; biasa disebut dengan era industri. Revolusi industri 3.0 ditandai dengan sistem komputasi data dan maraknya penggunaan komponen semikonduktor dan transistor hingga microchip; biasa disebut dengan era komputerisasi. Revolusi industri 4.0 sudah dicetuskan sejak tahun 2000-2005 saat internet berkembang dengan kecepatan tinggi, serta sistem *cloud* dan *Big Data*.

Antara kebutuhan dan tuntutan perubahan di bidang teknologi otomasi industri. Perubahan yang demikian pesatnya itu tentu membawa konsekuensi tersendiri. Maka perlu ada semacam upaya sinkronisasi antara tuntutan perubahan yang cepat tersebut dengan kebutuhan yang ada. Apakah kemudian memang kita siap dan sungguh membutuhkan perubahan sedemikian itu atau malah sebaliknya; perubahan itu justru menyeret kita dan/atau menciptakan kebutuhan yang baru. Buku ini

mencoba memberikan semacam bentuk kontribusi khusus dalam menghadapi tuntutan perubahan yang demikian cepat di bidang otomasi industri. Kebutuhan efisiensi dan kecepatan produksi menuntut diterapkannya sistem robot industri pada banyak industri manufaktur.

Buku ini memberikan informasi yang sangat memadai bagi mahasiswa yang tertarik untuk mempelajari aplikasi pemrograman robot industri. Bahkan bagi mereka yang belum memiliki pengetahuan dasar robotika akan mampu untuk melakukan secara bertahap pemrograman robot hingga komunikasi dua robot industri.

Apa sebenarnya yang ada dalam buku ini? Buku ini merupakan materi pembelajaran mengenai penggunaan dan pemrograman robot industri yang dikemas sedemikian rupa sehingga pembelajar tidak akan mengalami kesulitan. Alur pembahasan materi disajikan dalam Pertemuan ke-1 hingga Pertemuan ke-4. Pada Pertemuan ke-1 disampaikan mengenai dasar-dasar aplikasi Robot ABB. Selanjutnya pada Pertemuan ke-2 materi yang disampaikan adalah mengenai komunikasi dan pemrograman Robot ABB. Berikutnya pada Pertemuan ke-3 disajikan aplikasi pengintegrasian antara Robot ABB dengan Konveyor. Pada Pertemuan ke-4 yang merupakan puncak materi, akan diuraikan komunikasi dua Robot ABB menggunakan metode input-output dengan aplikasi konveyor. Buku ini tidak menyajikan hal-hal yang rumit terkait dengan pemrograman robot, misalnya rumus algoritma tingkat tinggi dalam menentukan *path planning and trajectory* sebagaimana halnya bila kita ingin mempelajari teori robot. Namun disajikan dalam bentuk tutorial lengkap dan mudah dipahami, serta pada akhir setiap pertemuan diberikan latihan dan soal sebagai bentuk pengukuran terhadap pemahaman materi.

Dengan terbitnya buku ini, saya ingin menyampaikan penghargaan dan ucapan terima kasih yang setinggi-tingginya kepada semua pihak yang telah membantu dan mendukung. *Pertama*, kepada Aliya dan Orlando yang telah berupaya untuk menghidupkan kembali kedua Robot ABB IRB1410. Karena kedua robot yang digunakan dalam materi pembelajaran ini semula dalam kondisi mati dan teronggok di gudang

sebuah perusahaan sampai akhirnya robot dihibahkan kepada kami. *Kedua*, kepada Fachrudin dan Sena yang telah melanjutkan dan melengkapi kedua robot ABB ini dengan *gripper* yang berfungsi baik dan melakukan penambahan fungsi *pick and place* yang membuat kedua robot ini dapat dipelajari lebih lanjut untuk integrasi Robot ABB dengan PLC (*Programmable Logic Controller*). *Ketiga*, kepada Wahyu, Yudha, dan Dita yang telah melengkapi sistem ini dengan konveyor dan memungkinkan terjadinya Komunikasi dua Robot ABB berbasis PLC menggunakan metode *input-output*.

Terima kasih kepada mereka yang sudah saya sebut karena telah membantu mengumpulkan materi serta melakukan uji coba sehingga buku ini tidak hanya sekedar berupa tulisan saja, namun telah terbukti dan dapat dipraktikkan secara langsung guna mendukung pencapaian kompetensi pembelajaran seperti yang ditargetkan.

Semoga buku ini bisa menjawab kebutuhan akan materi pembelajaran mengenai pemrograman robot industri dan penerapannya untuk kalangan mahasiswa dan siapa saja yang tertarik di bidang ini.

Ignatius Deradjad Pranowo
dradjad@pmsd.ac.id

DAFTAR ISI

PRAKATA	v
DAFTAR ISI.....	viii
PERTEMUAN 1 DASAR-DASAR APLIKASI ROBOT ABB	1
1. PENGENALAN ROBOT ABB IRB1410	1
2. SPESIFIKASI TEKNIS ROBOT ABB IRB1410	2
3. PENGENALAN ROBOTSTUDIO	9
4. PEMROGRAMAN MENGGUNAKAN ROBOTSTUDIO	10
5. TAHAP PENGOPERASIAN	29
LATIHAN 1	34
SOAL 1.....	36
PERTEMUAN 2 KOMUNIKASI DAN PEMROGRAMAN ROBOT	39
1. KOMUNIKASI ROBOT	39
2. TRANSFER PROGRAM DARI ROBOTSTUDIO KE ROBOT ABB	46
LATIHAN 2	53
SOAL 2.....	55
3. PROGRAM LOOPING	57
4. PROGRAM COUNTER	57
SOAL 3.....	61
PERTEMUAN 3 ROBOT ABB DAN KONVEYOR	64
1. INTEGRASI ROBOT DAN KONVEYOR	64
2. PEMROGRAMAN POSISI	65
LATIHAN 3	66
SOAL 4.....	68
PERTEMUAN 4 KOMUNIKASI ANTARA DUA ROBOT ABB DAN PLC.....	70
1. Komunikasi 2 Robot dengan 2 PLC	70
SOAL 5.....	71
LAMPIRAN.....	75

REFERENSI.....112

deepublish / publisher

DASAR-DASAR APLIKASI ROBOT ABB

1. PENGENALAN ROBOT ABB IRB1410



Robot ABB IRB1410



Controller IRC5



FlexPendant

Lokasi: Lab. Robot–Mekatronika, Fak. Vokasi USD

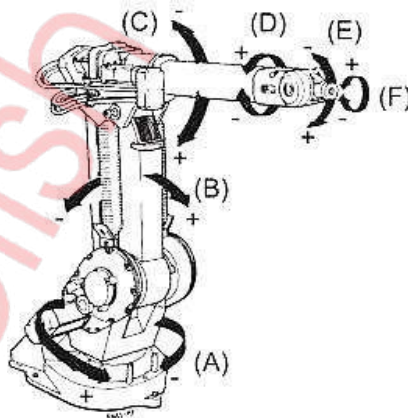
Pada gambar di atas berturut-turut adalah: Robot ABB IRB1410, Controller IRC5, dan FlexPendant yang digunakan dalam materi pembelajaran ini. Robot ABB IRB1410 adalah jenis robot dengan *link & joint* sebanyak 6 axis. Robot ABB ini merupakan tipe robot industri yang digunakan dalam proses produksi khususnya untuk pengelasan. Seperti halnya robot ABB pada umumnya, seri IRB1410 ini memerlukan Controller IRC5 beserta FlexPendant untuk pengoperasiannya. Berikut ini adalah spesifikasi teknisnya.

Spesifikasi teknis

ROBOT ABB IRB 1410	
Number of axes	6 axis
Weight	225 kg
Handling capacity	5 kg
Made in	China
Manufactured year	2008
Serial No.	14-63096
CONTROLLER IRC5 M2004	
Supply voltage	380-400 VAC
Weight	150 kg
Controller Dimension	970 x 725 x 710 mm
Made in	China
Serial No.	14-63096

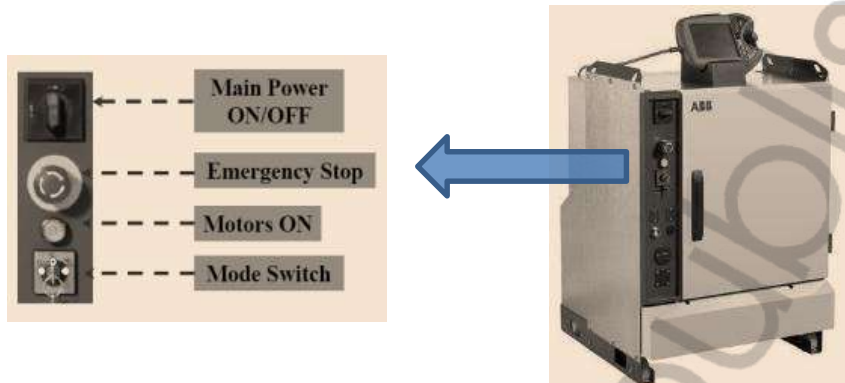
2. SPESIFIKASI TEKNIS ROBOT ABB IRB1410

2.1. DOF (Degree of Freedom)

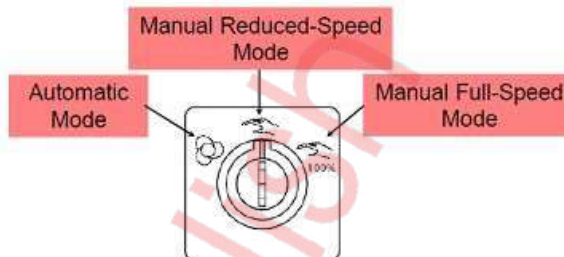


A	Axis 1	D	Axis 4
B	Axis 2	E	Axis 5
C	Axis 3	F	Axis 6

2.2. CONTROLLER (IRC5)

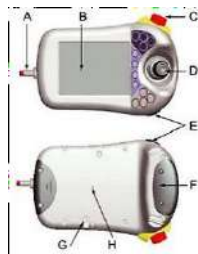


Saklar Main Power	Menyalakan/mematikan <i>controller</i>
Emergency Stop	Menghentikan sistem dalam keadaan darurat
Motor ON	Menghidupkan motor servo untuk axis robot
Mode Switch	Mode pergerakan robot



<i>Automatic Mode</i>	<i>Manual Reduced - Speed Mode</i>	<i>Manual Full - Speed Mode</i>
Untuk proses produksi!	Untuk pemrograman, menggerakkan robot secara manual, <i>parameter setting</i>	Untuk menjalankan program dengan <i>full speed</i>
Tidak bisa dijalankan menggunakan <i>joystick</i>	Perlu menekan <i>enabling device</i> untuk mengaktifkan motor robot.	Perlu menekan <i>enabling device</i> untuk mengaktifkan motor robot.
 Robot in Automatic Mode	 Programming Mode	 Testing Mode

2.3. FLEXPENDANT

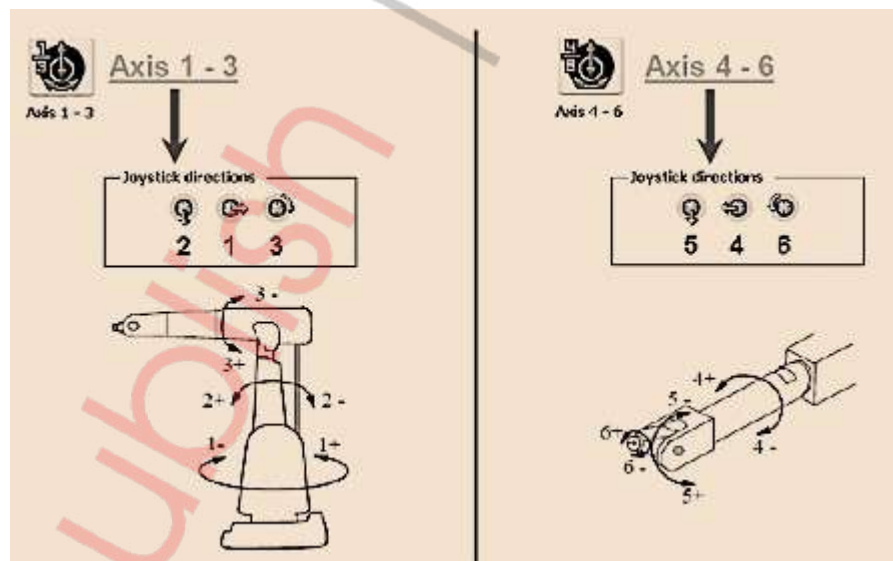


A	Connector
B	Touch Screen
C	Emergency stop button
D	Joystick
E	USB port
F	Enabling device
G	Stylus pen
H	Reset button

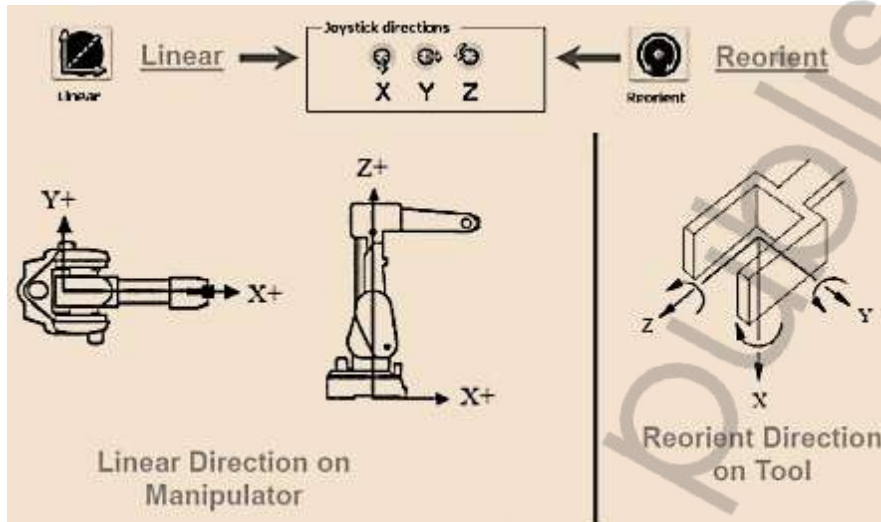
2.3.1. Joystick

Fungsi *joystick* pada FlexPendant adalah untuk menggerakkan manipulator atau yang disebut juga dengan *jogging*.

2.3.1.1. Joystick Direction (*axis by axis*)



2.3.1.2. Joystick Direction (*linear and reorient*)



2.3.2. Hard Button FlexPendant

"Programmable keys", 1-4	A-D
Tombol fungsi "mechanical unit"	E
"Mode Motion" reorientasi atau linear	F
"Mode Motion" axis 1-3 atau axis 4-6	G
Tombol fungsi "increments"	H
"Step Backward" menjalankan line program sebelumnya	J
"Start" menjalankan line program berurutan	K
"Step Forward" menjalankan line program selanjutnya	L
"Stop" menghentikan eksekusi program	M



2.3.3. Fungsi Tombol Enabling Device



Fungsi dari **enabling device** adalah untuk menghidupkan motor pada robot, sehingga setiap pengoperasian atau saat menjalankan program tombol ini harus ditekan. Robot akan otomatis berhenti apabila enabling device dilepas atau *released*.

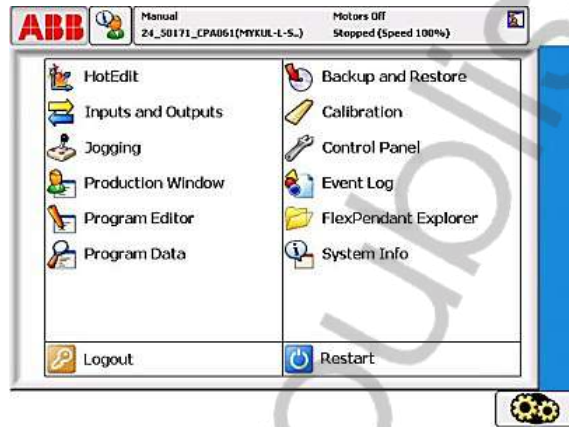
2.3.4. Tampilan Stand By FlexPendant



Touch Screen Elements

A	ABB Menu
B	Operator Windows
C	Status Bar
D	Close Button
E	Task Bar
F	Quickset Menu

2.3.4.1. ABB Menu



2.3.4.2. Operator Windows

Operator windows menampilkan pesan dari program robot. Biasanya muncul apabila suatu program membutuhkan respons/persetujuan operator untuk melanjutkan suatu proses.

2.3.4.3. Status Bar

Status bar menampilkan informasi penting terkait dengan status sistem, seperti mode operasi, kondisi motor (*on/off*), dan lain sebagainya.

2.3.4.4. Close Button

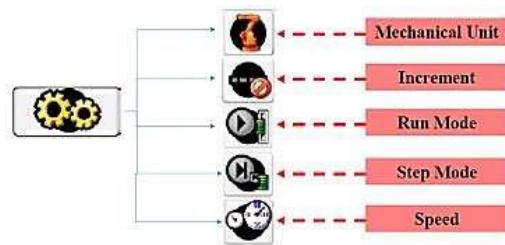
Menutup jendela/aplikasi yang sedang digunakan.

2.3.4.5. Task Bar

Digunakan untuk melihat aplikasi/jendela apa saja yang sedang digunakan, serta berpindah dari jendela satu ke jendela yang lain.

2.3.4.6. Quickset Menu

Menu ini menyediakan pengaturan cepat untuk jogging dan pengeksekusian program.



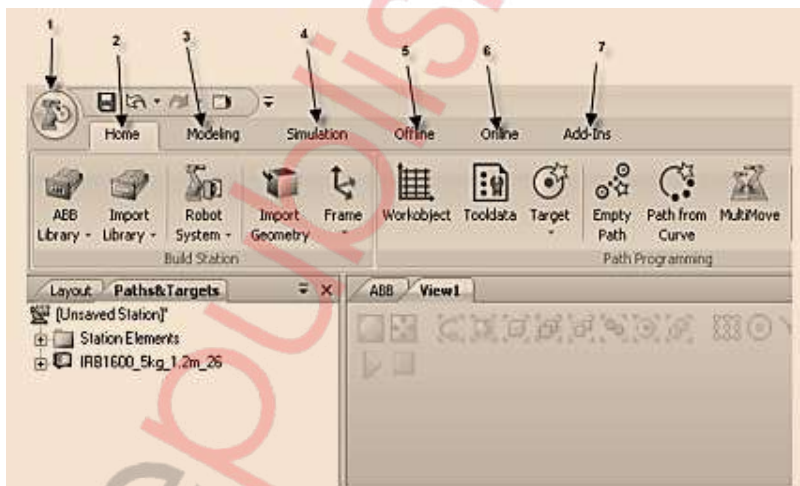
3. PENGENALAN ROBOTSTUDIO

3.1. Apa itu Robot Studio?

Robot studio merupakan salah satu software yang digunakan untuk melakukan simulasi pemrograman robot khusus ABB dan program tersebut dapat ditransfer ke robot nyata, sehingga kita tidak perlu memprogram melalui flexpendant pada kontroler. Selain itu program dari kontroler juga dapat kita ambil untuk disimpan pada komputer.



3.2. Toolbar Robot Studi



3.2.1. Application Menu

Diakses dari tombol RobotStudio di sudut kiri atas GUI.

3.2.2. Home Tab

Ini berisi kontrol yang diperlukan untuk membangun *station*, membuat sistem, jalur pemrograman, dan menempatkan *item*.

3.2.3. Modeling Tab

Ini berisi kontrol untuk membuat dan mengelompokkan komponen, membuat *frame*, pengukuran, dan operasi CAD.

3.2.4. Simulation Tab

Ini berisi kontrol untuk mengatur, mengonfigurasi, mengendalikan, memantau, dan merekam simulasi.

3.2.5. Offline Tab

Berisi kontrol untuk sinkronisasi, konfigurasi, dan tugas yang ditetapkan untuk Kontroler Virtual (VC).

3.2.6. Online Tab

Ini berisi kontrol untuk mengelola pengendali nyata.

3.2.7. Add-Ins Tab

Berisi kontrol untuk PowerPacs dan VSTA.

4. PEMROGRAMAN MENGGUNAKAN ROBOTSTUDIO

4.1. Hidupkan komputer yang akan digunakan

4.2. Klik *shortcut* RobotStudio pada desktop

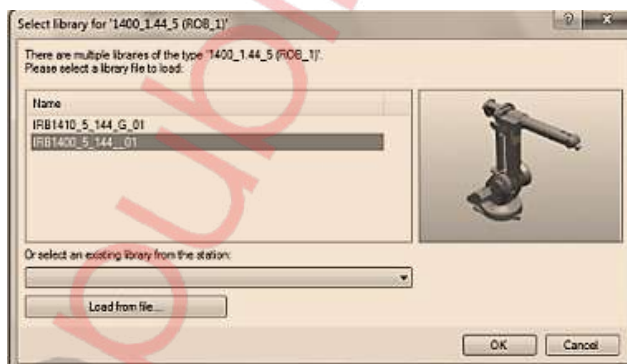


- 4.3. Pilih perintah **New**, lalu pilih **Solution with Station and Robot Controller**. Setelah itu isi nama program pada **Solution Name** dan pilih folder penyimpanan.



- 4.4. Atur robot model ke **IRB 1410 5kg 1.44m**, lalu klik **Create** dan ditunggu beberapa saat.

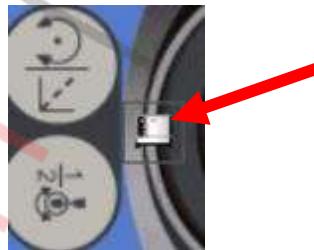
- 4.5. Setelah menunggu beberapa saat akan muncul tampilan seperti di bawah ini. Silahkan pilih robot yang akan digunakan. Lalu klik **OK**.



- 4.6. Pilih *toolbar* **Controller** dan klik pada **FlexPendant**, lalu klik **Virtual FlexPendant**.



- 4.7. Ketika sudah muncul Pendant Virtual silahkan klik pada ikon yang ditunjuk anak panah ini.



Lalu akan muncul ikon untuk memilih kondisi robot *automatic/ manual/ 100%manual*), dan pilih ke bagian manual seperti pada gambar di bawah ini.

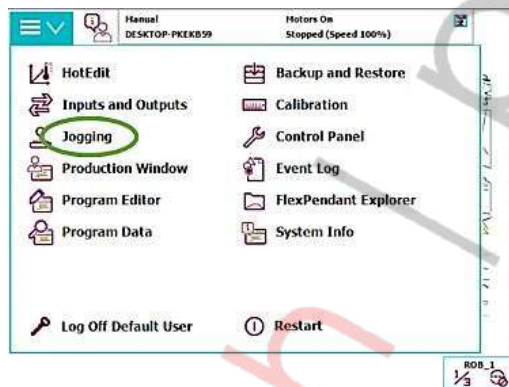


- 4.8. Klik pada **Enable** sehingga warnanya berubah menjadi hijau agar dapat digunakan untuk jogging robot.

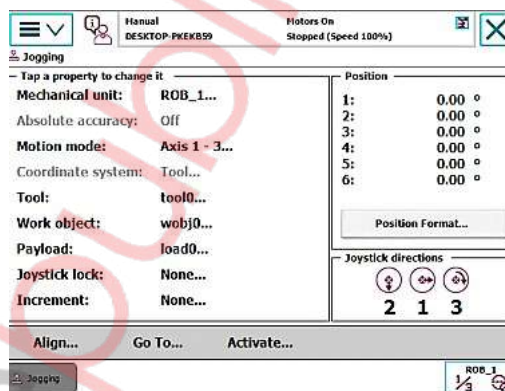


4.9. JOGGING

Jogging digunakan untuk menentukan posisi robot pada saat proses *teaching*. Untuk membuka menu *jogging*, masuk ke **ABB Menu**, lalu pilih **Jogging**.



- Maka akan muncul tampilan seperti gambar di bawah ini:



- Kemudian tombol analog pada **Virtual FlexPendant** bisa diarahkan dan robot akan mengikuti arah tersebut.



- Untuk mengubah gerakan setiap axis dan linear dapat diklik pada gambar di bawah ini.

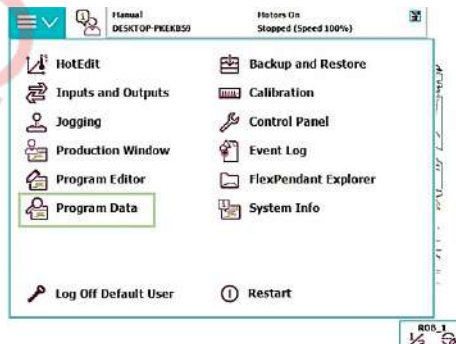


- Untuk mengubah axis yang digunakan dapat diklik pada gambar di bawah ini.

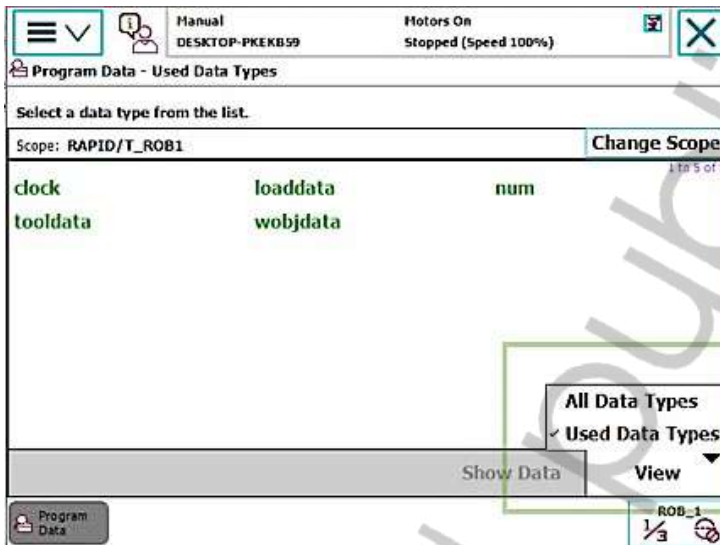


4.10. SAVING POSITION

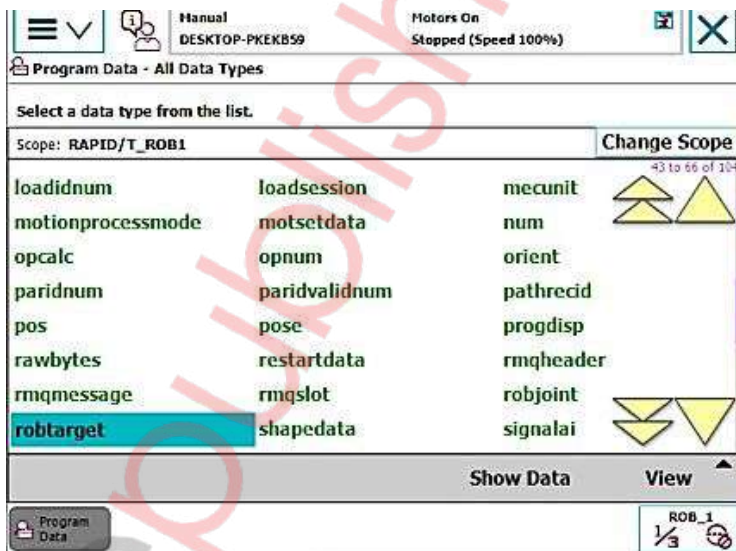
- Masuk ke **ABB Menu**, pilih **Program Data**.



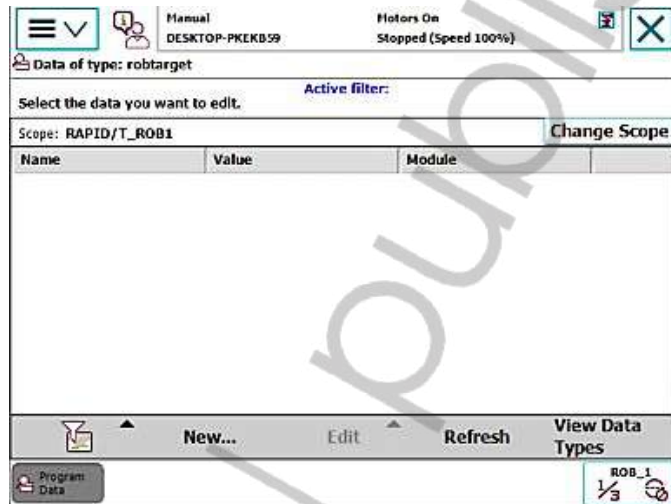
- Pada menu Program Data, klik **View** lalu pilih **All Data Types**.



- Pada daftar tipe data, pilih **robtarget** untuk masuk ke menu penyimpanan posisi robot.



- Setelah masuk *robtarget* maka akan muncul menu seperti pada gambar di bawah ini, langkah selanjutnya adalah menggerakkan robot ke posisi yang diinginkan.



- Setelah posisi robot ditentukan, klik **New** untuk membuka menu Data Declaration seperti pada gambar berikut:



- Pada kolom “Name” beri nama posisi robot sesuai dengan yang diinginkan. Pilih **OK** untuk menyimpan posisi.

4.11. CREATING A NEW PROGRAM

Step	Action
1	On the ABB menu, tap Program Editor .
2	Tap Tasks and Programs
3	Tap File , then New Program . If there was already a program loaded, a warning dialog appears. <ul style="list-style-type: none"> • Tap Save to save the loaded program. • Tap Don't save to close loaded program without saving it. i.e. delete from program memory. • Tap Cancel to leave the program loaded.
4	Use the soft keyboard to name the new program. Then tap OK .
5	Tap Back to return to the editor view or Show modules to add existing modules to the new program. Continue by adding instructions, routines, or modules.

4.12. SAVING PROGRAM

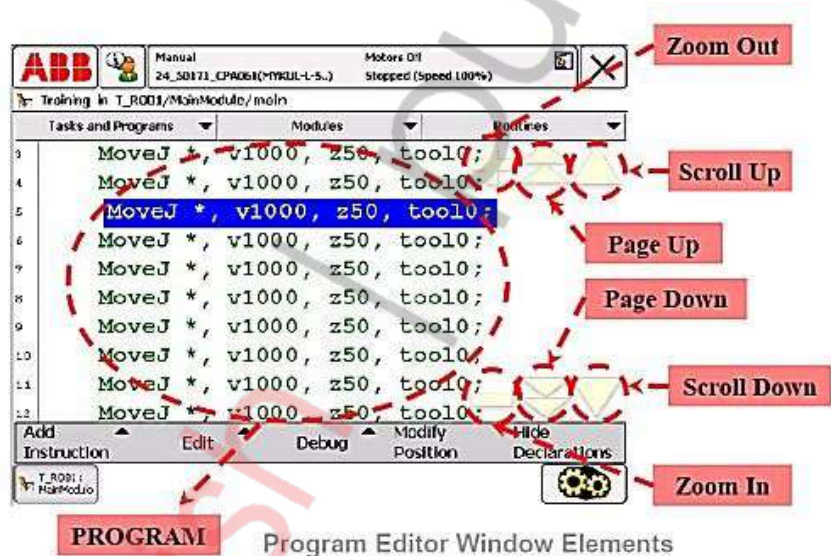
Step	Action
1	On the ABB menu, tap Program Editor .
2	Tap Tasks and Programs .
3	Tap File and select Save Program As...
4	Use the suggested program name or tap ... to open the soft keyboard and enter a new name. Then tap OK .

4.13. LOADING AN EXISTING PROGRAM

Step	Action
1	On the ABB menu, tap Program Editor .
2	Tap Tasks and Programs .
3	Tap File , then Load Program . If there was already a program loaded, a warning dialog appears. <ul style="list-style-type: none"> • Tap Save to save the loaded program.

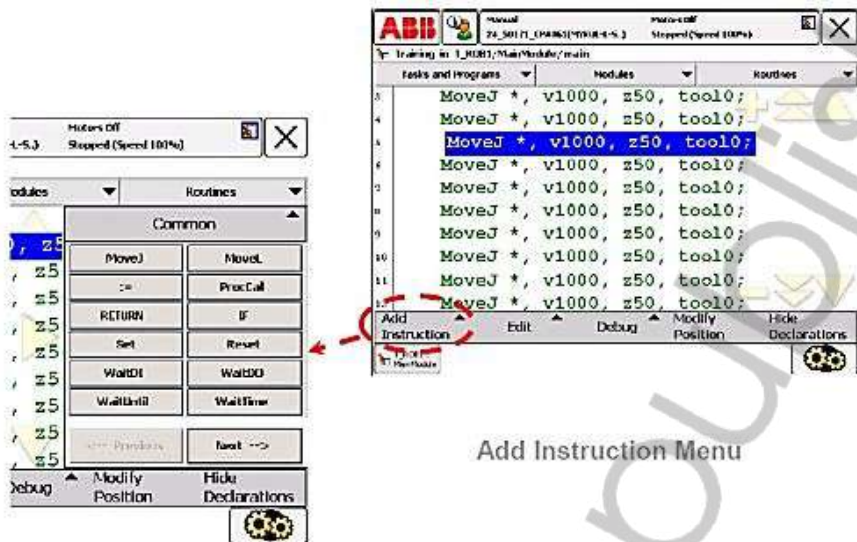
Step	Action
	<ul style="list-style-type: none"> • Tap Don't save to close loaded program without saving it. i.e. delete from program memory. • Tap Cancel to leave the program loaded.
4	Use the file searching tool to locate the program file to be loaded (file type pgf). Then tap OK . The program is loaded and the program code is displayed.

4.14. PROGRAM EDITOR



4.15. MENU "ADD INSTRUCTION"

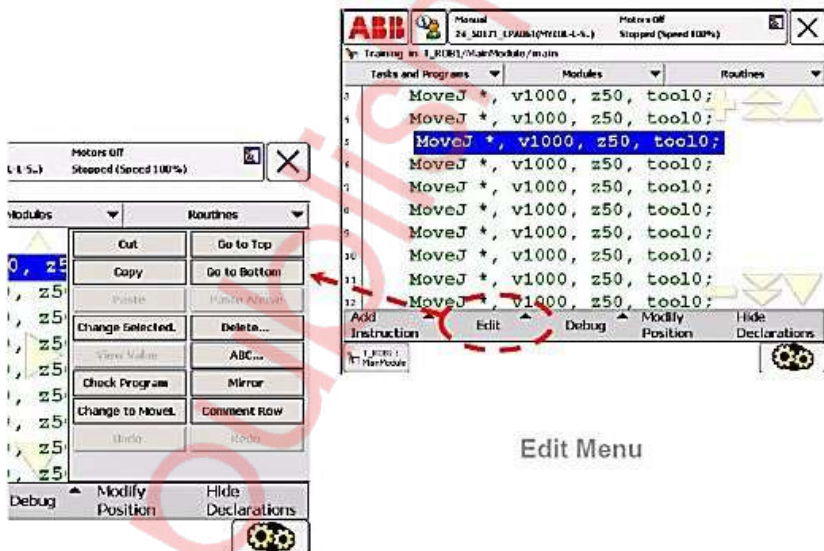
Menu ini digunakan untuk menambahkan perintah-perintah pemrograman agar gerakan robot sesuai dengan yang diinginkan.



Add Instruction Menu

4.16. MENU "EDIT"

Menu ini digunakan untuk melakukan editing perintah yang telah dibuat. Dapat untuk memindah, meng-copy, dan check perintah.



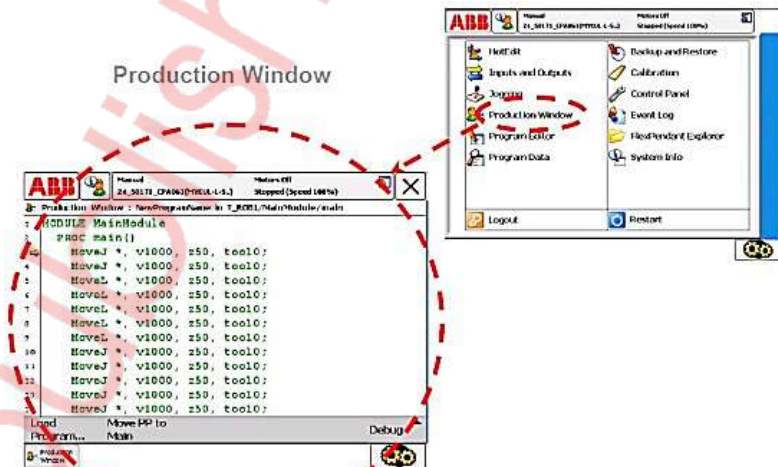
Edit Menu

4.17. MENU "DEBUG"

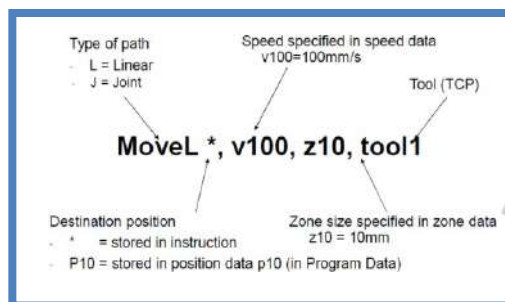


4.18. PRODUCTION WINDOWS

Windows yang menampilkan program ketika robot dijalankan secara otomatis.



4.19. DESKRIPSI PROGRAM



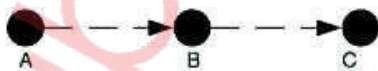
4.20. Robot Movement

Terdapat beberapa tipe pergerakan yang seringkali digunakan dalam pembuatan program robot ABB. Antara lain MoveL, MoveAbsj, MoveC dan MoveJ. Penjelasan masing-masing tipe adalah sebagai berikut:

4.21. MoveL

Adalah perintah untuk menggerakkan robot secara linear dari satu titik ke titik yang lainnya. Contoh pergerakan titik A (p10) ke titik B (p20) menggunakan perintah MoveL:

```
PROC main ();  
MoveL p10, V1000, Z50, tool0;  
MoveL p20, V1000, Z50, tool0;  
End Prog
```

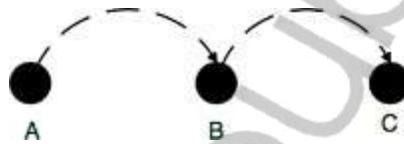


4.22. MoveAbsj

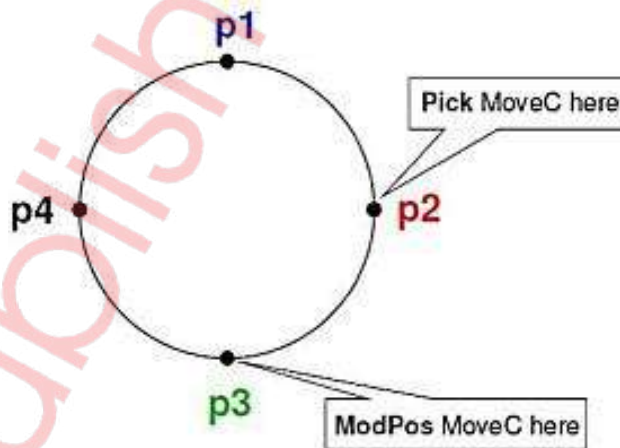
Merupakan perintah untuk menggerakkan robot kembali ke posisi Home. Sama halnya dengan MoveL, posisi Home ditentukan secara manual oleh operator sesuai dengan kondisi yang diinginkan, kemudian pilih *modify position*.

4.23. MoveC

Adalah perintah untuk menggerakkan robot membentuk sebuah radius dari satu titik ke titik yang lainnya. Perintah ini digunakan untuk benda kerja yang permukaannya cembung, cekung, maupun bentuk lain yang memiliki radius. Besarnya nilai radius dapat diatur dengan mengubah pada nilai Z. Diperlukan dua titik dalam satu line program MoveC.

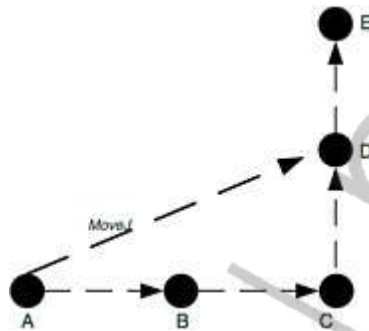


```
PROC main ();  
MoveL p1, V100, fine, tool0;  
MoveC p2, p3, v100, z1, tool0;  
MoveC p4, p1 V100, fine, tool0;  
End Prog
```



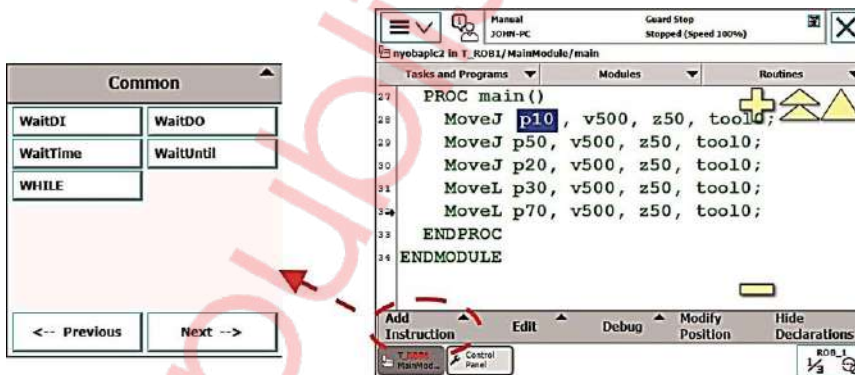
4.24. MoveJ

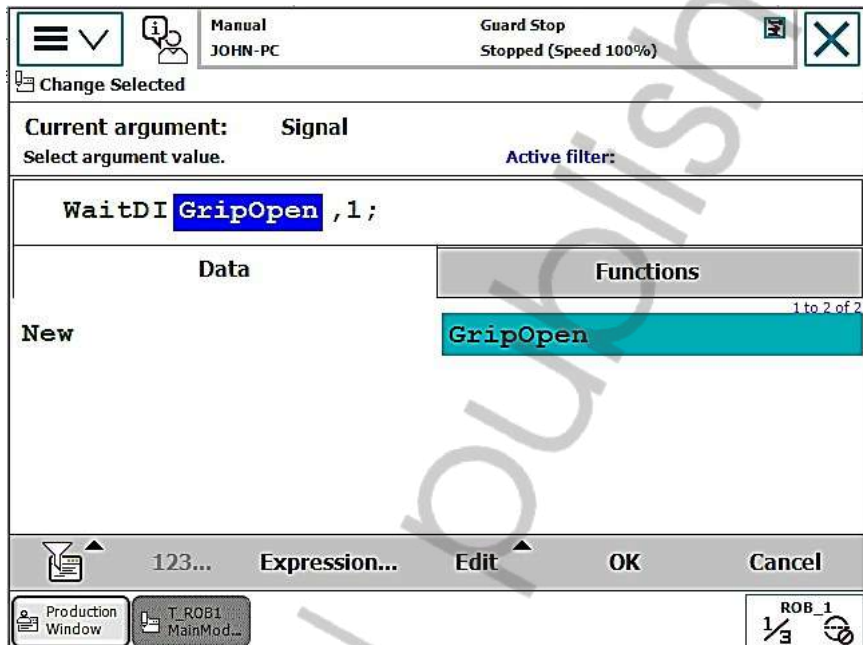
Merupakan perintah gerak untuk mencari jalan tercepat menuju titik yang telah ditentukan tanpa harus melewati titik berikutnya. Sebagai contoh, robot diprogram bergerak melewati 5 titik yang telah ditentukan. Namun pada *loop* ke-2 robot diprogram untuk bergerak dari titik A langsung menuju titik D, maka MoveJ perintah yang tepat untuk digunakan.



4.25. WAIT DI

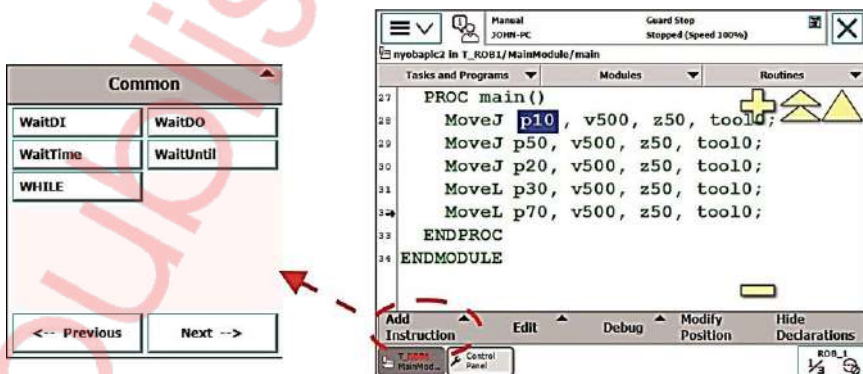
Merupakan perintah menunggu suatu *digital input* bernilai *high/low* (1/0) ketika nilai *digital input* terpenuhi maka program akan melanjutkan ke rung selanjutnya.

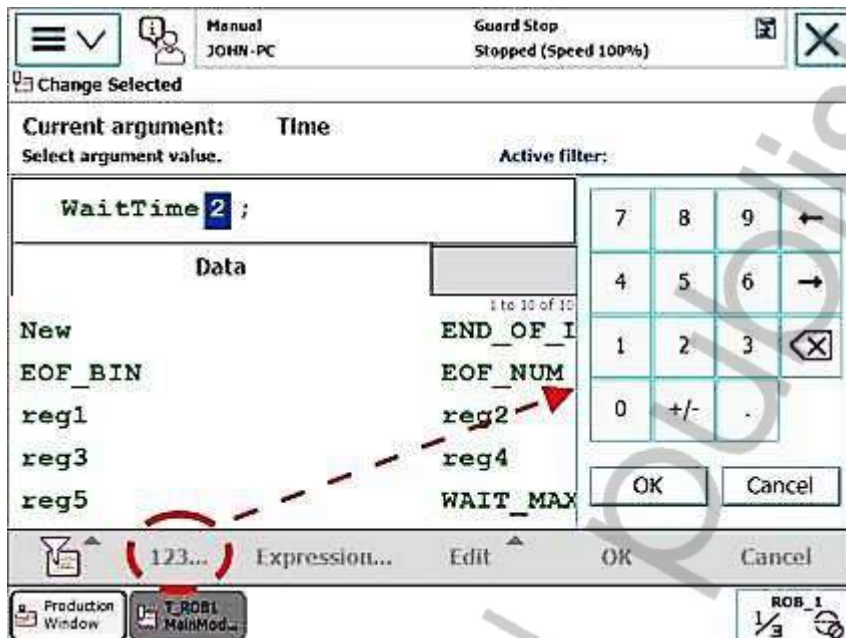




4.26. WaitTime

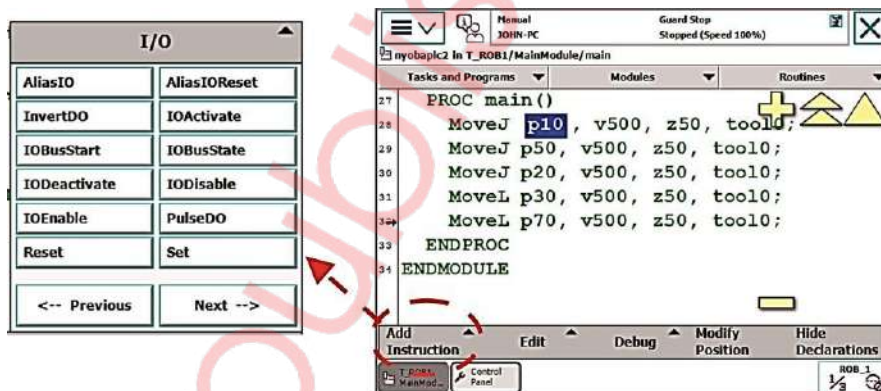
Merupakan program untuk menunggu selama waktu yang telah ditentukan. Ketika waktu sudah terpenuhi maka program akan melanjutkan ke rung selanjutnya.

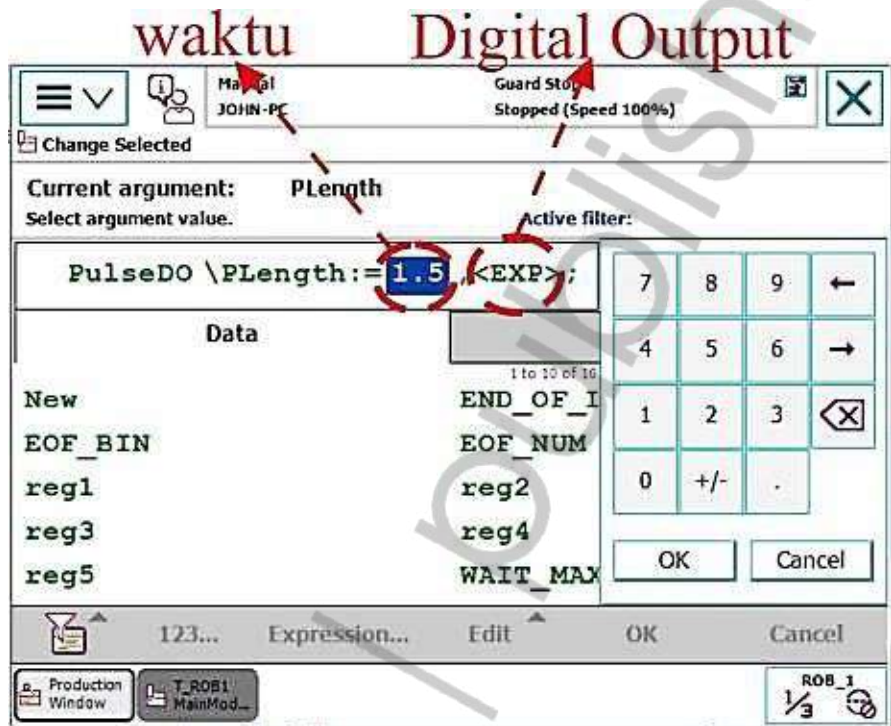




4.27. PulseDO

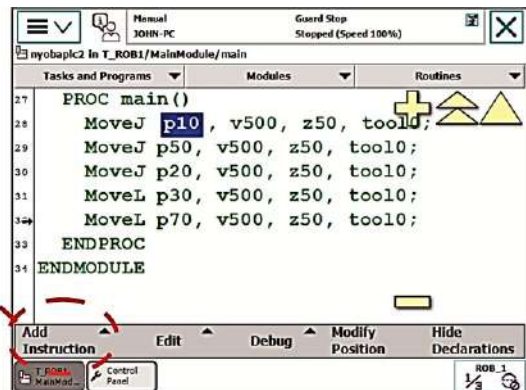
Merupakan perintah untuk membuat *Digital Output* menjadi *high* dalam waktu yang telah ditentukan. Setelah waktu habis maka *Digital Output* akan menjadi *low*.

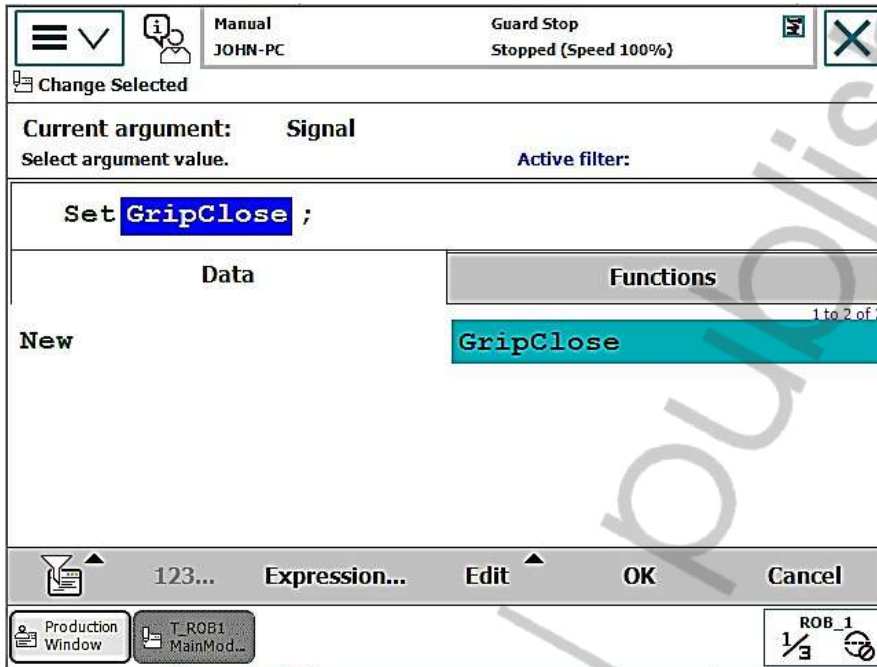




4.28. Set

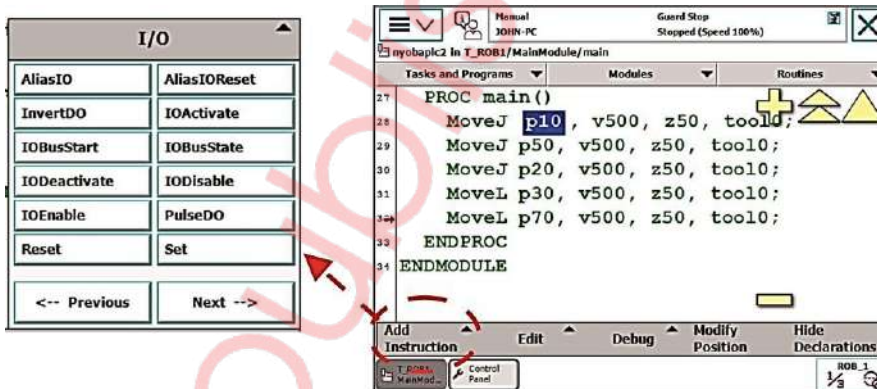
Merupakan program untuk melakukan *setting* suatu *Digital Output* atau *Digital Input* bernilai *high*.

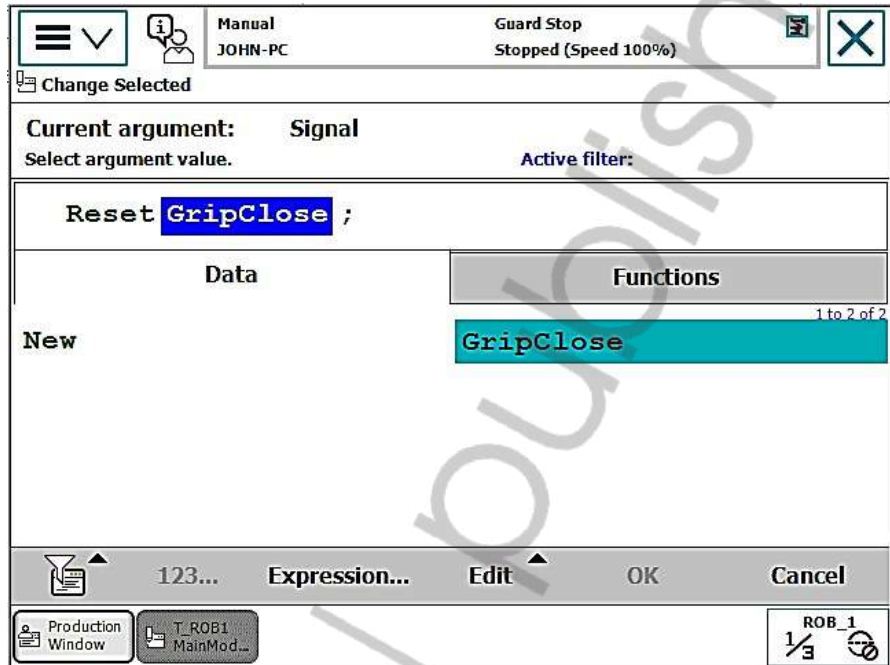




4.29. Reset

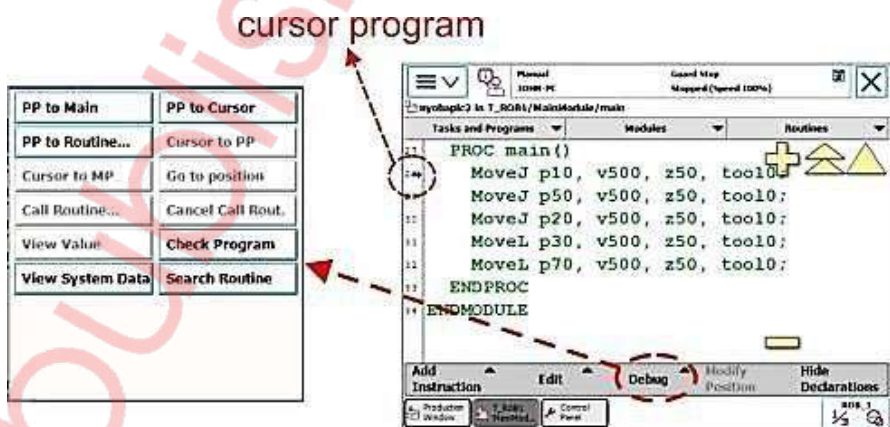
Merupakan program untuk melakukan *setting* suatu *Digital Output* atau *Digital Input* bernilai *low*.





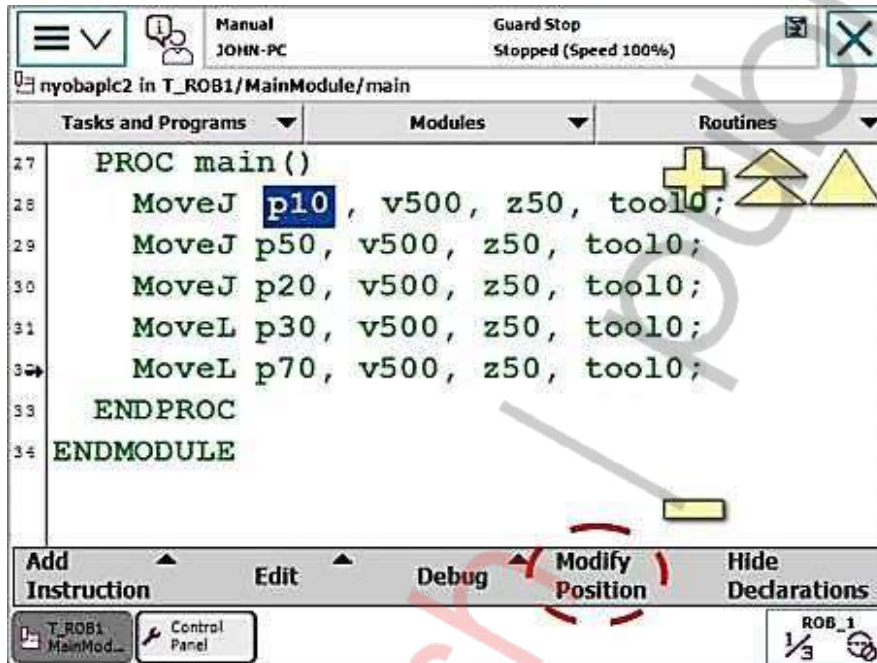
4.30. PP To Main

Merupakan perintah untuk membuat *cursor program* kembali ke *main program* (awal program)



4.31. Modify Position

Berfungsi untuk memodifikasi posisi program yang telah kita buat. Kita hanya tinggal memilih *rung program* yang akan diganti lalu tekan *Modify Position* di bagian bawah layar.

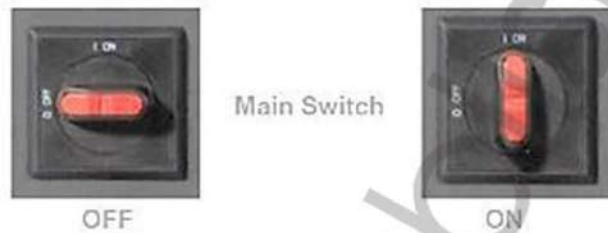


5. TAHAP PENGOPERASIAN

5.1. Cek Kelistrikan

- Pastikan *power* terhubung pada *controller*.
- Pastikan *circuit breaker* pada *controller* berada di posisi ON.
- Pastikan kabel komunikasi antara *controller* dengan robot sudah terpasang.
- Pastikan kabel FlexPendant terpasang sempurna pada *controller*.

5.2. MAIN SWITCH

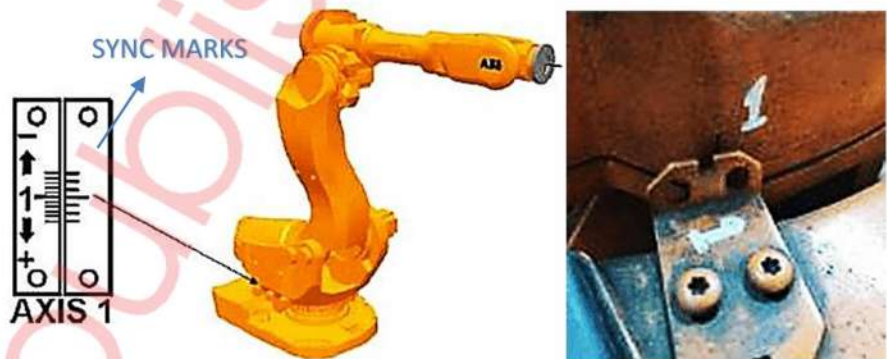


- Sebelum menghidupkan sistem, pastikan di sekeliling tidak ada yang mengganggu pergerakan robot.
- Putar saklar *controller* IRC5 pada posisi ON. FlexPendant secara otomatis akan *booting*. Tunggu hingga FlexPendant berada pada tampilan *standby*.

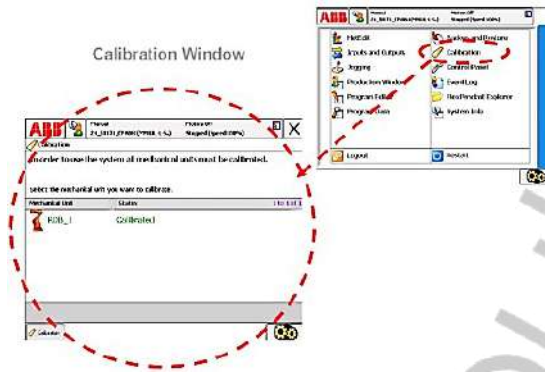
5.3. CALIBRATION

Sebelum robot dijalankan perlu dilakukan kalibrasi terlebih dahulu dengan meng-*update* posisi masing-masing *axis*. Langkah-langkah yang harus dilakukan sebagai berikut:

- Posisikan semua *axis* sesuai dengan *sync-marks* yang berada pada perpotong tiap-tiap *axis*.



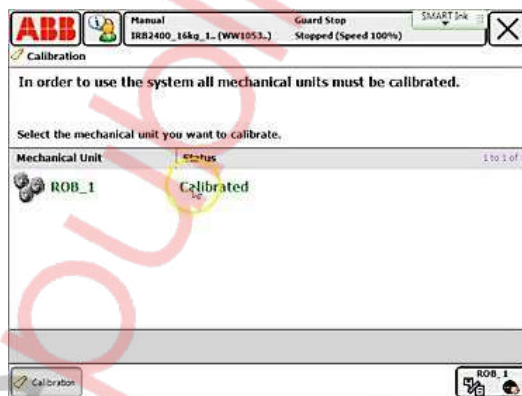
- Pada jendela **Menu ABB**, pilih menu **Calibration**.



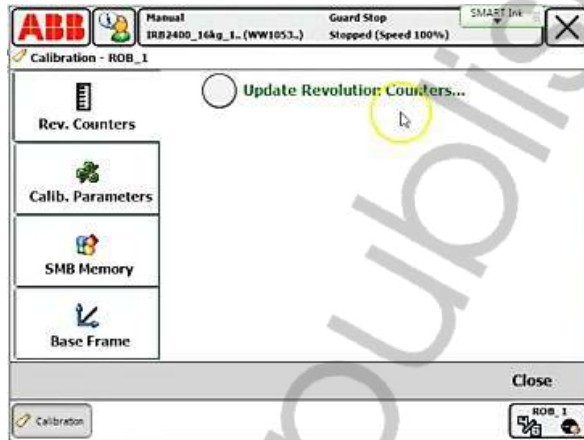
- Perhatikan status kalibrasi *mechanical unit*, apabila sudah tekalibrasi (*calibrated*) robot sudah siap untuk diprogram.

If the calibration status is...	then...
Not calibrated	The robot must be calibrated by a qualified service technician.
Rev. counter update needd	You must update the revolution counters.
Calibrated	No calibration is needed.

- Namun apabila status kalibrasi menunjukkan “*Rev.counter update needed*”. Klik pada *mechanical unit* yang akan dikalibrasi.



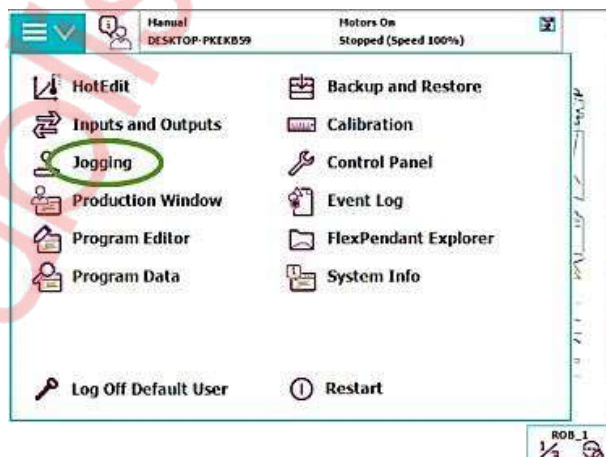
- Pada sub menu Rev. Counters pilih **Update Revolution Counters**.



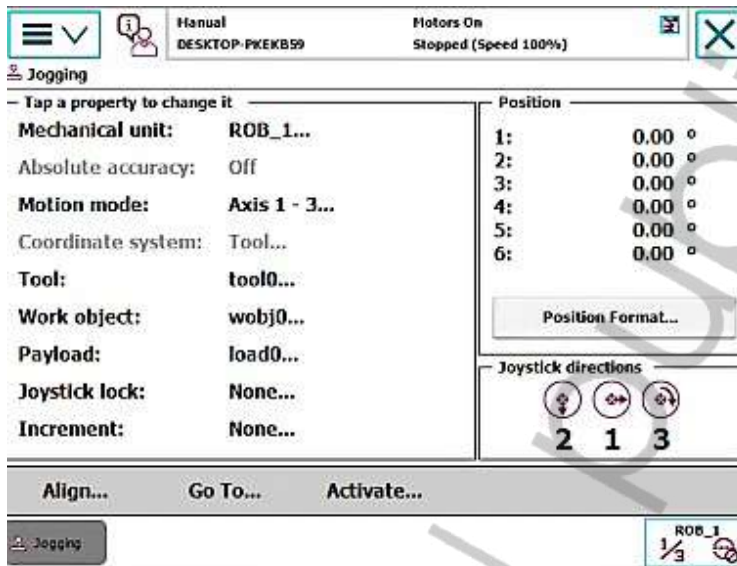
- Lakukan kalibrasi pada seluruh *axis*. Setelah selesai terkalibrasi, klik **OK**.

5.4. JOGGING

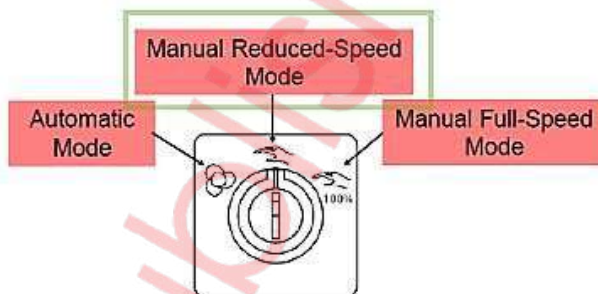
Jogging digunakan untuk menentukan posisi robot pada saat proses *teaching*. Untuk membuka menu *jogging*, masuk ke **ABB Menu**, lalu pilih **Jogging**.



- Maka akan muncul tampilan seperti gambar di bawah ini:

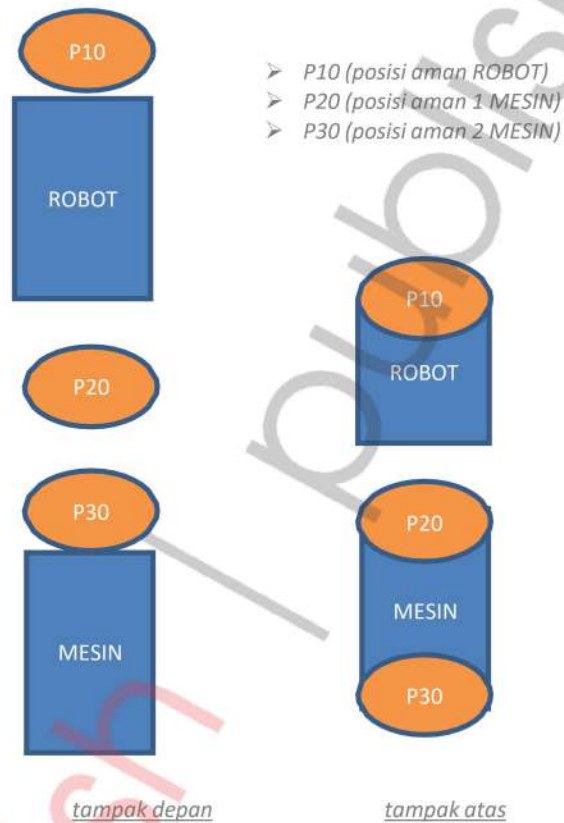


- Kemudian pada IRC5 controller, putar saklar ke mode “Manual Reduced-Speed”.



- Selanjutnya, aktifkan motor pada robot dengan menekan “Enabling device” pada FlexPendant.
- Setelah motor *on*, gerakkan robot dengan menggunakan *joystick* pada FlexPendant

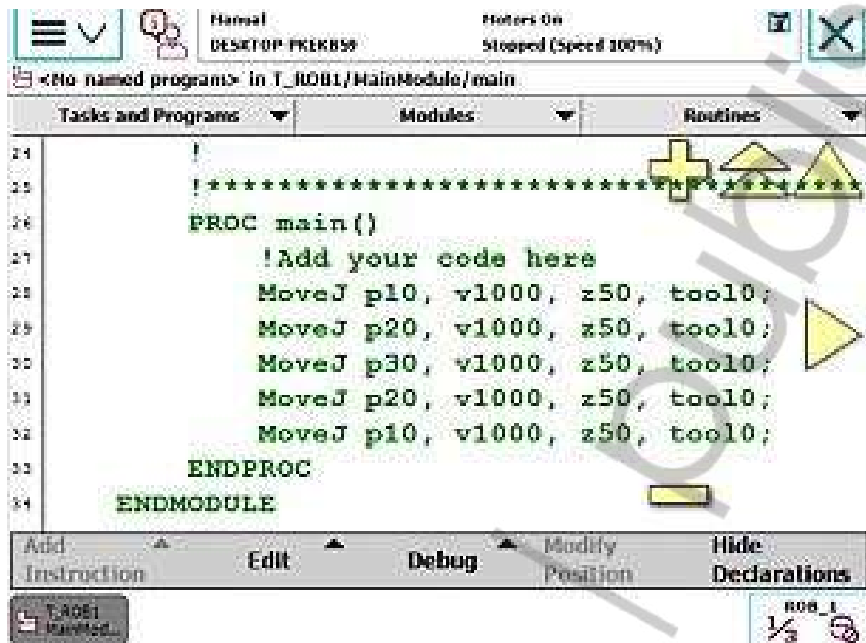
LATIHAN 1



Buatlah program robot dengan tahapan seperti berikut menggunakan RobotStudio:

1. Pertama-tama buat *home* posisi robot (P10)
2. Robot berjalan ke posisi aman 1 mesin (P20)
3. Selanjutnya robot akan berjalan ke posisi aman 2 mesin (P30), lalu kembali ke *home* posisi (P10)

Penyelesaian Latihan 1



The screenshot shows a software interface for a robot named 'ROB_1'. The main window displays a routine named 'main' with the following code:

```
24 | !  
25 | !*****  
26 | PROC main()  
27 |     !Add your code here  
28 |     MoveJ p10, v1000, z50, tool0;  
29 |     MoveJ p20, v1000, z50, tool0;  
30 |     MoveJ p30, v1000, z50, tool0;  
31 |     MoveJ p20, v1000, z50, tool0;  
32 |     MoveJ p10, v1000, z50, tool0;  
33 | ENDPROC  
34 | ENDMODULE
```

The interface includes a top status bar with 'Manual' and 'Stopped (Speed 100%)'. Below the code editor is a toolbar with buttons for 'Add Instruction', 'Edit', 'Debug', 'Modify Position', and 'Hide Declarations'. A 'T-AGE1' button is visible in the bottom left corner.

SOAL 1

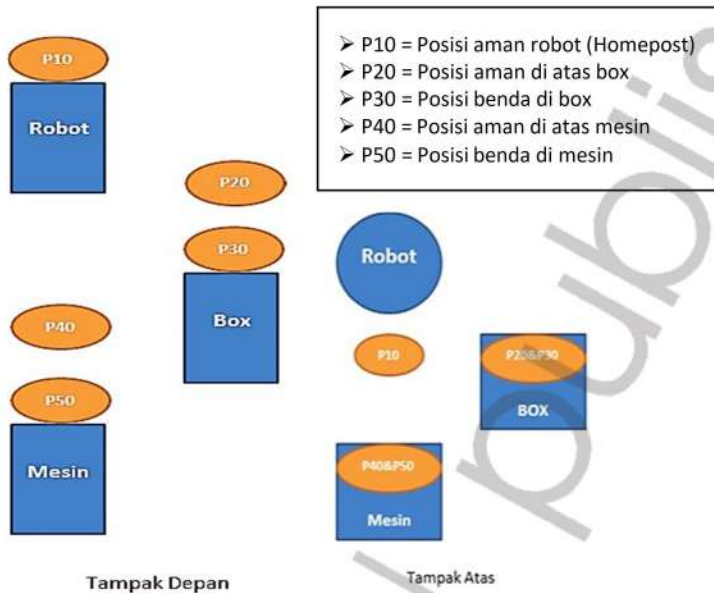
1. Sebutkan dan jelaskan secara singkat jenis-jenis robot yang digunakan di industri!
2. Jelaskan secara detail apa yang dimaksud dengan DOF?
3. Sebutkan 3 **mode switch** dan berikan penjelasannya!

- Kalibrasi seluruh axis menggunakan joystick sesuai dengan sinkmark !
- Buat program seperti berikut :
 1. Robot mengambil benda pada box lalu diletakan di mesin.
 2. Robot kembali ke posisi aman kemudian mesin akan memproses benda selama 5 detik.
 3. Robot mengambil benda di mesin kemudian diletakan di box.
 4. Robot kembali ke posisi aman.

Catatan:

1. Atur *speed* data menjadi V500.
2. Atur *speed* robot pada Quicsket Menu menjadi 70%
3. Setiap akan mengambil benda harus ada posisi aman di atas benda.

Penyelesaian Soal 1



1. Program Robot

```

Manual HPLAPTOP
Guard Stop Stopped (Speed 100%)
Latihan2 in T_ROB1/MainModule/main

Tasks and Programs Modules Routines
32 PROC main()
33 MoveJ p10, v500, z50, tool0;
34 MoveJ p20, v500, z50, tool0;
35 MoveL p30, v500, z50, tool0;
36 WaitTime 2;
37 PulseDO\PLength=-1.5, GripClose;
38 WaitTime 2;
39 MoveL p20, v500, z50, tool0;
40 MoveJ p40, v500, z50, tool0;
41 MoveL p50, v500, z50, tool0;
42 WaitTime 2;
43 PulseDO\PLength=-1.5, GripOpen;
44 WaitTime 2;
45 MoveL p40, v500, z50, tool0;
46 MoveJ p10, v500, z50, tool0;
47 WaitTime 5;
48 MoveJ p40, v500, z50, tool0;
49 MoveL p50, v500, z50, tool0;

Add Instruction Edit Debug Modify Position Hide Declarations
Production Window T_ROB1 MainMod... T_ROB1 MainMod... T_ROB1 MainMod... T_ROB1 MainMod... T_ROB1 MainMod... ROB_1 1/3
    
```


Manual HPLAPTOP Guard Stop Stopped (Speed 100%)

Latihan2 in T_ROB1/MainModule/main

Tasks and Programs	Modules	Routines
45	MoveL p40, v500, z50, tool0;	+
46	MoveJ p10, v500, z50, tool0;	
47	WaitTime 5;	▲▲▲
48	MoveJ p40, v500, z50, tool0;	
49	MoveL p50, v500, z50, tool0;	▲▲▲
50	WaitTime 2;	
51	PulseDO\PLength:=1.5, GripClose;	▲▲▲
52	WaitTime 2;	
53	MoveL p40, v500, z50, tool0;	▲▲▲
54	MoveJ p20, v500, z50, tool0;	
55	MoveL p30, v500, z50, tool0;	▲▲▲
56	WaitTime 2;	
57	PulseDO\PLength:=1.5, GripOpen;	▲▲▲
58	WaitTime 2;	
59	MoveL p20, v500, z50, tool0;	▲▲▲
60	MoveJ p10, v500, z50, tool0;	
61	ENDPROC	▬
62	ENDMODULE	

Add Instruction Edit Debug Modify Position Hide Declarations

Production Window T_ROB1 MainMod... T_ROB1 MainMod... T_ROB1 MainMod... T_ROB1 MainMod... T_ROB1 MainMod... ROB_1 1/E

KOMUNIKASI DAN PEMROGRAMAN ROBOT

Pembahasan pada bagian ini adalah sebagai berikut:

1) Komunikasi

Merupakan pengiriman data baik dari PLC ke robot maupun dari robot ke PLC.

2) Transfer *file* dari RobotStudio ke Robot ABB

Merupakan cara untuk mentransfer *file* program dari RobotStudio ke RobotABB maupun sebaliknya.

3) Program Looping

Merupakan perintah untuk melakukan pengulangan program di mana pengulangan tersebut akan berlangsung terus menerus selama syarat terpenuhi. Program *looping* ini biasanya digunakan untuk gerakan robot yang selalu sama.

4) Program Counter

Hampir sama dengan *program looping*, namun *program counter* ini bisa di-*setting* untuk pengulangan berapa kali (n).

1. KOMUNIKASI ROBOT

Pengkabelan Digital INPUT & OUTPUT



- **Digital input** memiliki 8 pin (X31 – X38)



Pada *unit mapping* terbaca (0 – 7)

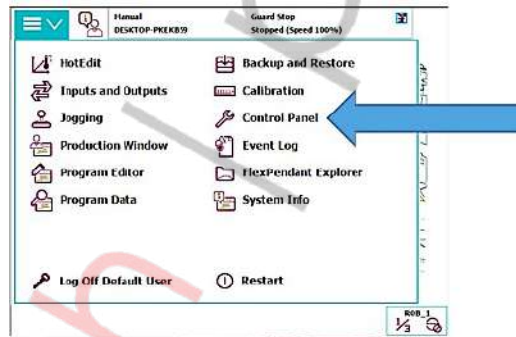
- **Digital Output** memiliki 8 pin (X11 – X18)



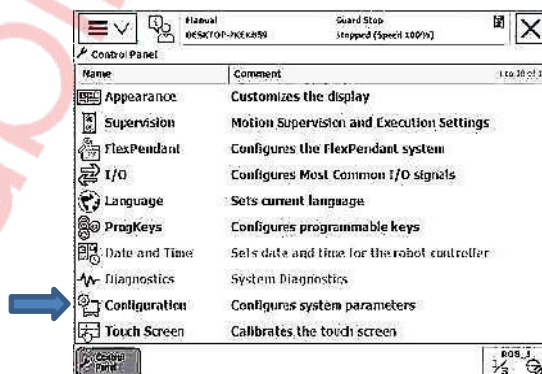
Pada *unit mapping* terbaca (32 – 39)

1.1. MENAMBAHKAN SIGNAL INPUT & OUTPUT (TIDAK DIREKOMENDASIKAN, SILAHKANGUNAKAN YANG SUDAH ADA)

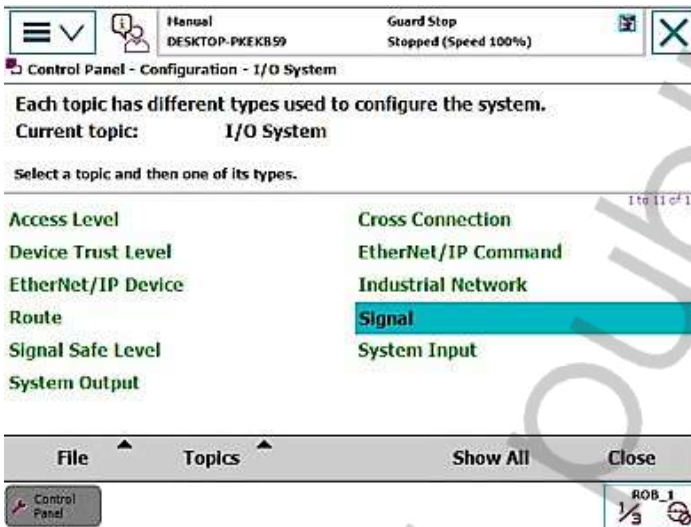
- Langkah awal yaitu dengan membuka menu **ABB** lalu pilih **Control Panel**.



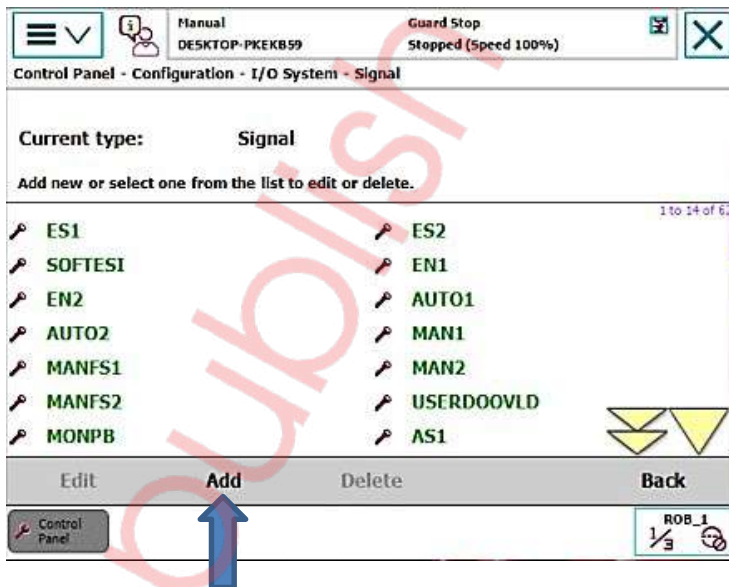
- Pilih **Configuration**



c) Pilih **Signal**



d) Pilih **Add** untuk menambahkan *signal* baru.



e) Atur nama *signal*

Manual DESKTOP-PKLB59 Guard Stop Stopped (Speed 100%)

Control Panel - Configuration - I/O System - Signal - Add

In order to add new all required inputs must be set to a value.

Tap a parameter twice in order to modify it.

Parameter Name	Value
Name	tmp0
Type of Signal	
Assigned to Device	
Signal Identification Label	
Category	
Access Level	Default

OK Cancel

Control Panel ROB_1 1/3

f) Atur Type of Signal

Manual DESKTOP-PKLB59 Guard Stop Stopped (Speed 100%)

Control Panel - Configuration - I/O System - Signal - Add

In order to add new all required inputs must be set to a value.

Tap a parameter twice in order to modify it.

Parameter Name	Value
Name	tmp0
Type of Signal	Digital Input
Assigned to Device	
Signal Identification Label	
Category	
Access Level	

Control Panel ROB_1 1/3

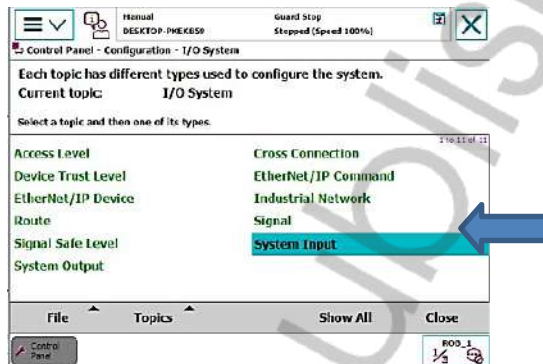
Pilihan Type of Signal:

1. *Digital Input (DI)* yaitu *output* dari PLC ke robot.
 2. *Digital Output (DO)* yaitu *output* dari robot ke PLC.
 3. *Analog Input (AI)* yaitu *output* dari PLC ke robot (berupa sinyal variabel).
 4. *Analog Output (AO)* yaitu *output* dari robot ke PLC (berupa sinyal variabel).
- g) Khusus Robot ABB IRB1410 pada kolom **Assigned to Device** pilih **“B_PROC_30”**
- h) Pilih **Signal Identification Label/Unit Mapping**

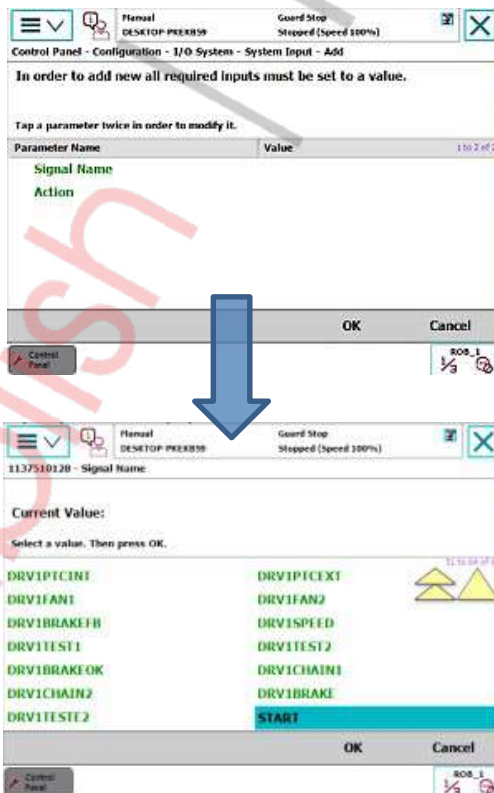
Parameter Name	Value
Name	tmp0
Type of Signal	
Assigned to Device	
Signal Identification Label	
Category	
Access Level	Default

1. Untuk Input bisa diberikan dari angka “0-7”
 2. Untuk Output bisa diberikan dari angka “32-39”
 3. Pemberian Label tidak boleh sama antara satu dengan yang lainnya.
- i) Setelah selesai lalu pilih **OK**, akan muncul peringatan **“Restart Now”**, pilih **Yes**

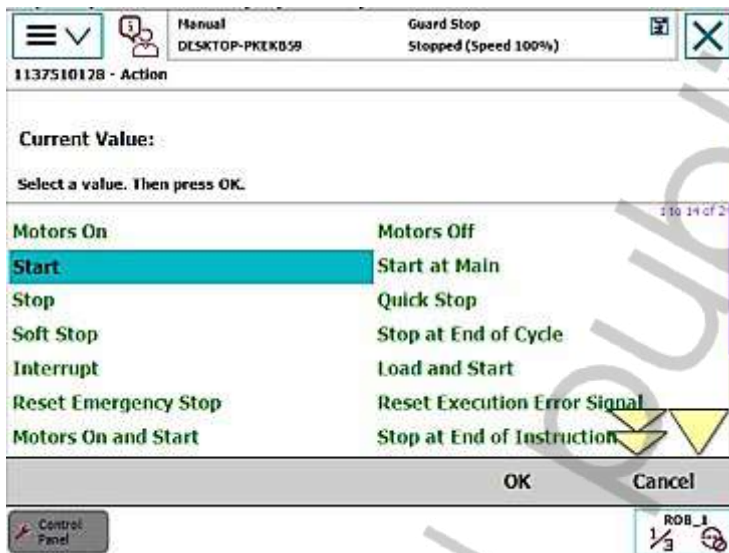
- j) Setelah selesai di-restart lanjut ke menu **System Input**, lalu pilih **Add**.



- k) Klik **Signal Name** lalu pilih *value* yang sudah dibuat.

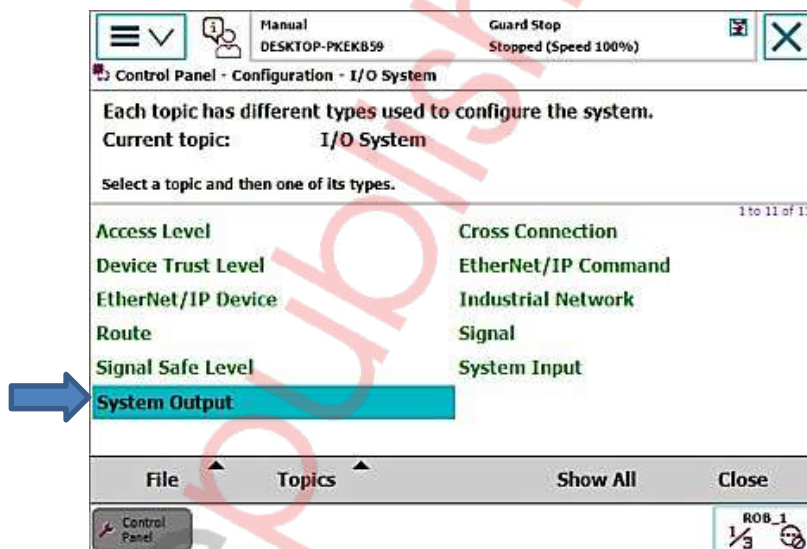


l) Pilih **Action** yang diinginkan lalu klik **OK**.

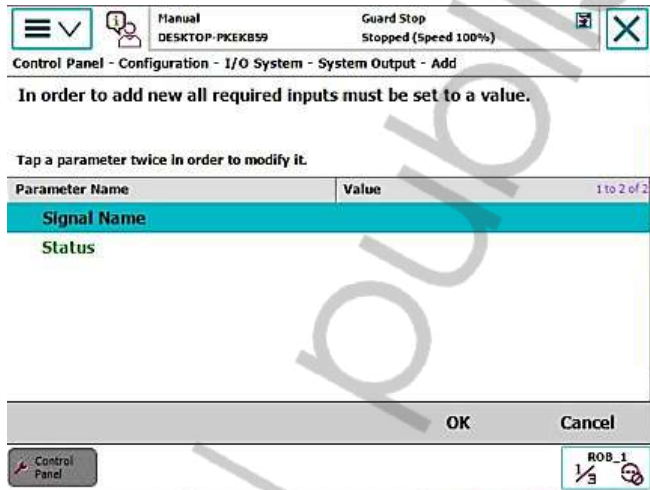


m) Maka akan muncul peringatan "Restart Now", pilih **Yes**.

n) Lalu lanjut ke menu **System Output** lalu klik **Add**.

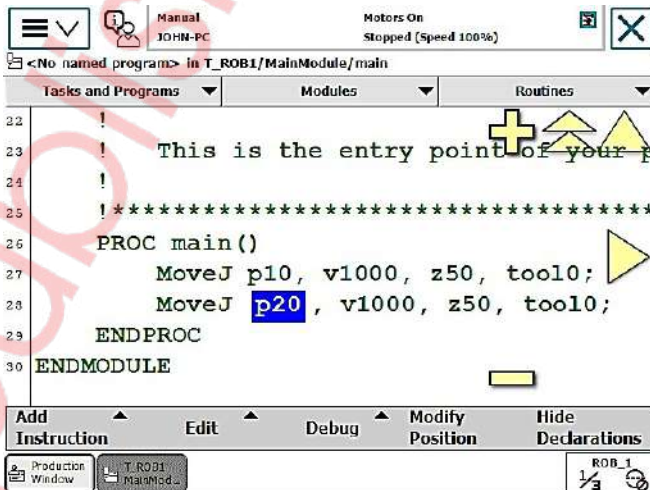


- o) Lalu masukkan **Signal Name** yang sudah dibuat.
- p) Pilih **Status/Action** yang diinginkan.
- q) Setelah selesai klik **OK**, pilih **YES** pada pesan "Restart Now".



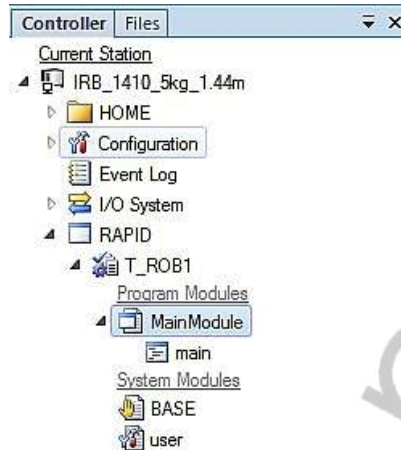
2. TRANSFER PROGRAM DARI ROBOTSTUDIO KE ROBOT ABB

- 1) Buatlah program robot pada RobotStudio dengan Virtual FlexPendant.

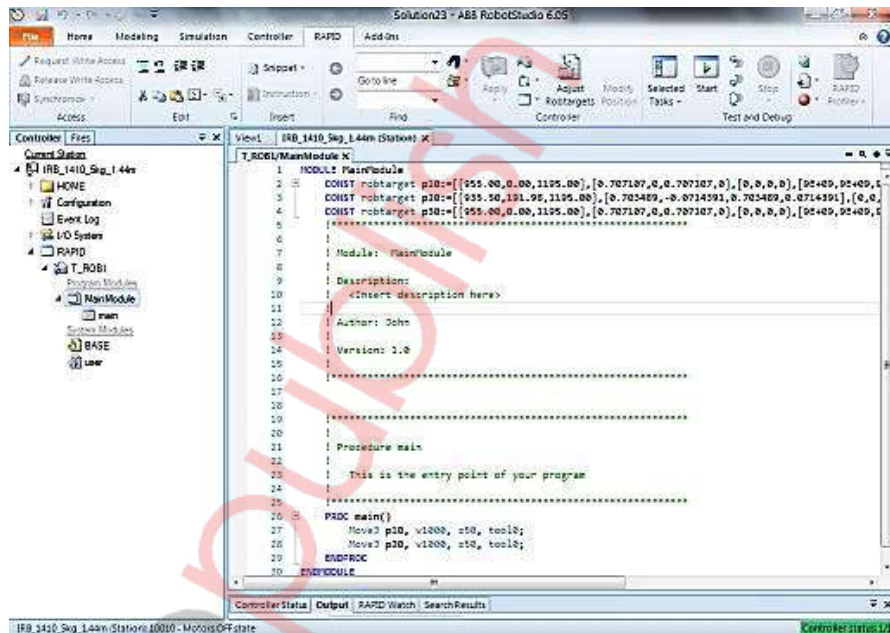


2) Kemudian Save As program.

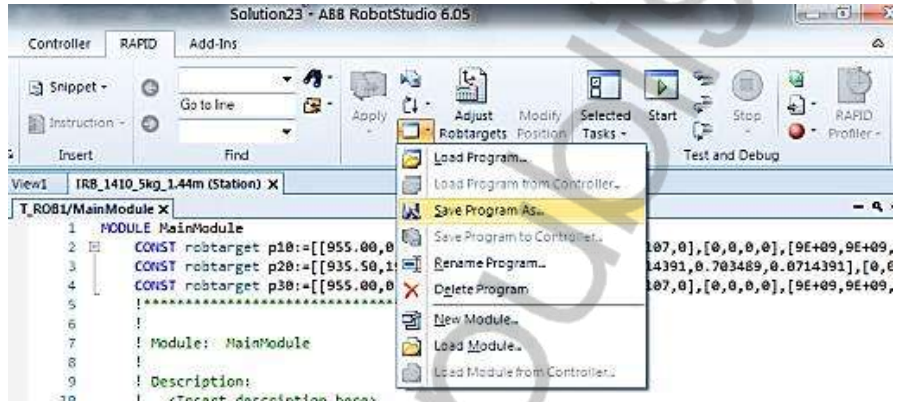
- Buka MainModule di Current Station.



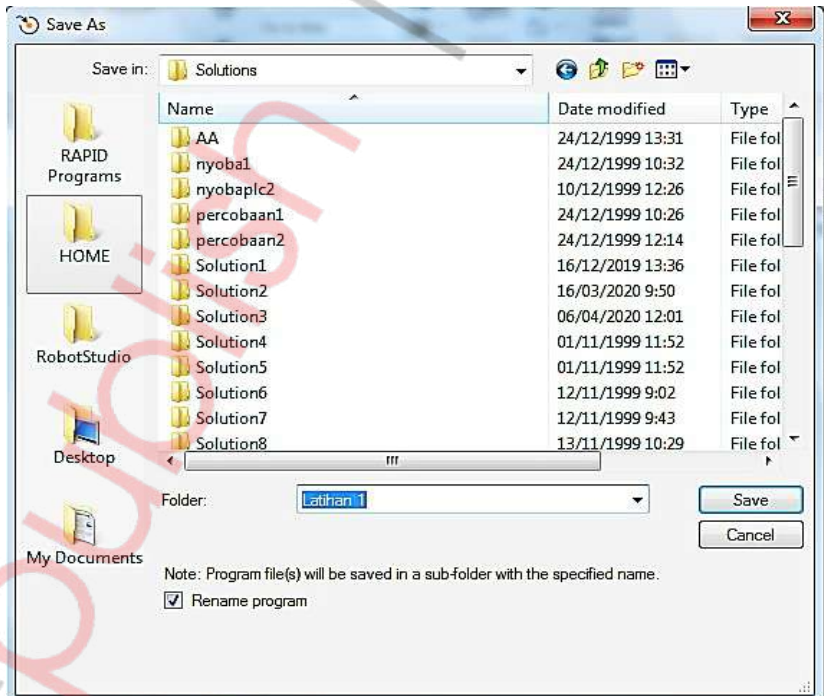
- Maka akan muncul tampilan sebagai berikut:



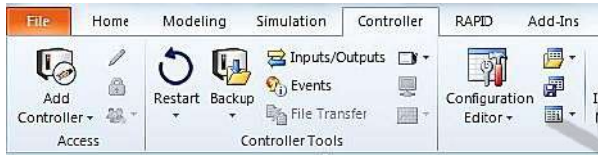
- buka *toolbar* RAPID kemudian klik Program pilih Save Program As.



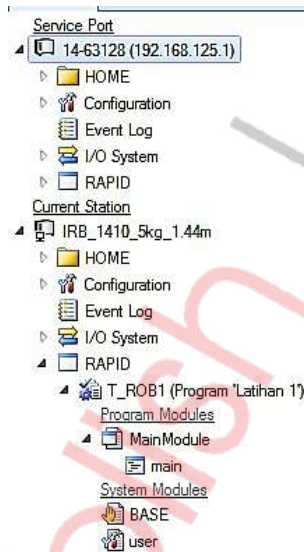
- Beri nama dan pilih lokasi penyimpanan.



- 3) Hubungkan komputer dengan robot menggunakan kabel Ethernet.
- 4) Setelah kabel dihubungkan koneksikan RobotStudio dengan robot ABB dengan cara klik *toolbar* Controller kemudian klik Add Controller.



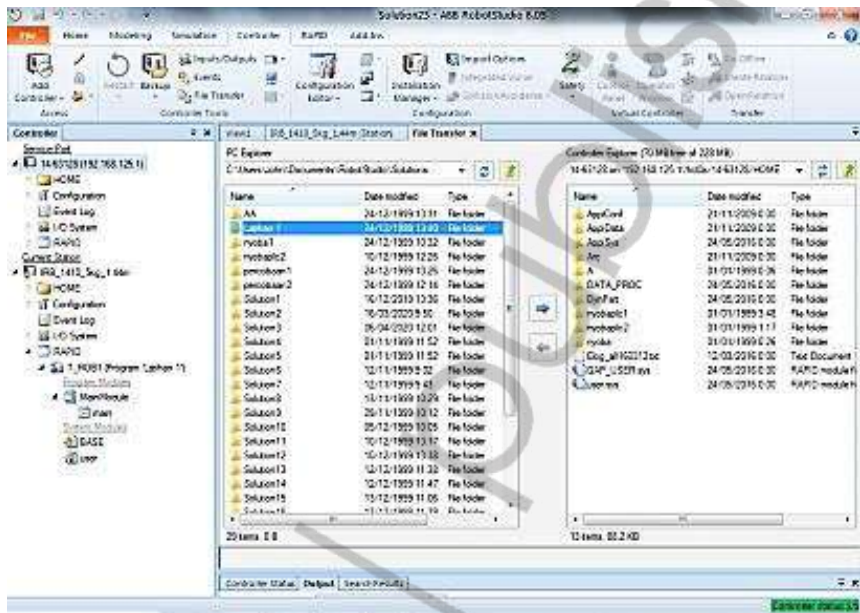
- 5) Transfer *file* program dari RobotStudio ke Robot ABB
 - Klik Controller robot di Service Port



- Klik File Transfer pada *toolbar* Controller.



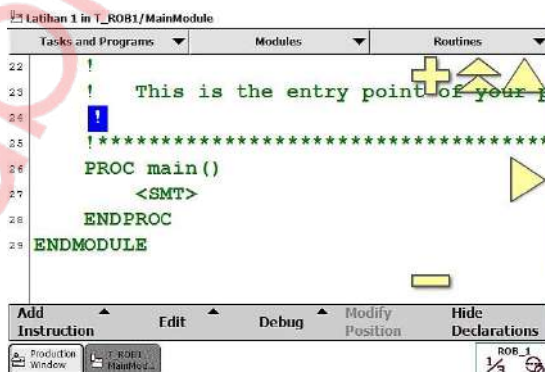
- Maka akan muncul tampilan seperti di bawah ini:



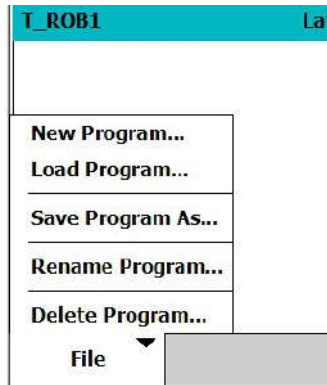
- Pilih folder *file* yang akan ditransfer kemudian klik panah ke kanan untuk mentransfer *file* dari PC ke robot.

6) Cara membuka *file* yang telah ditransfer di dalam FlexPendant

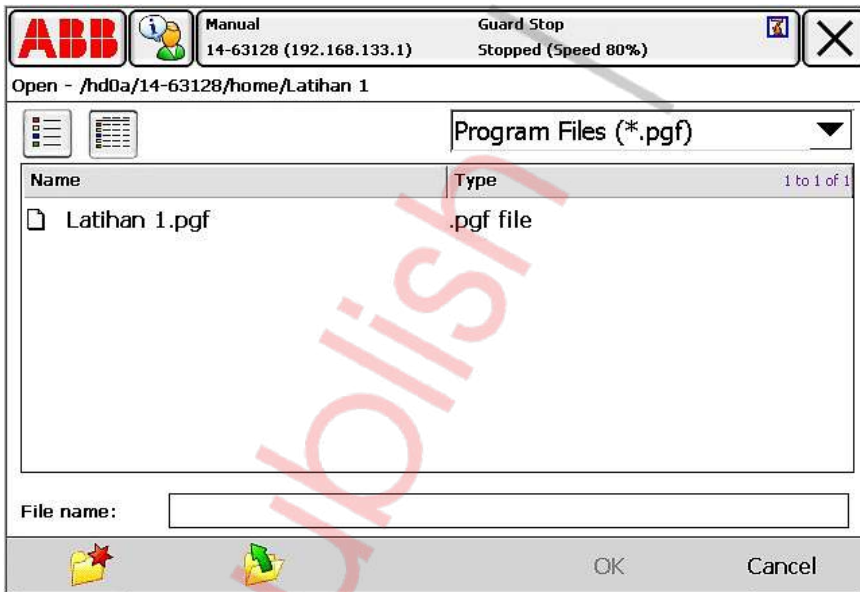
- Buka Program Data kemudian buka Tasks and Programs.



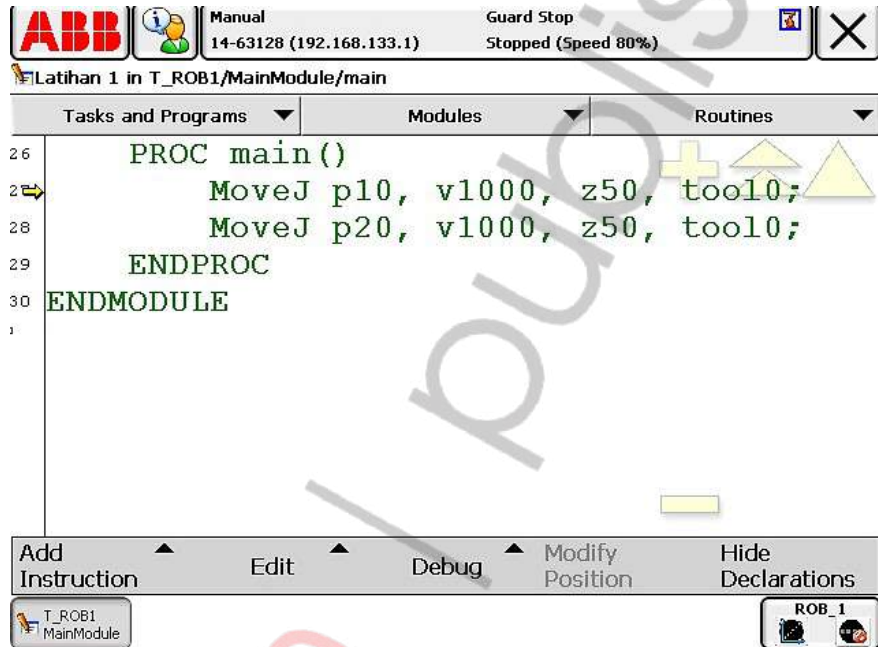
- Buka File buka Load Program



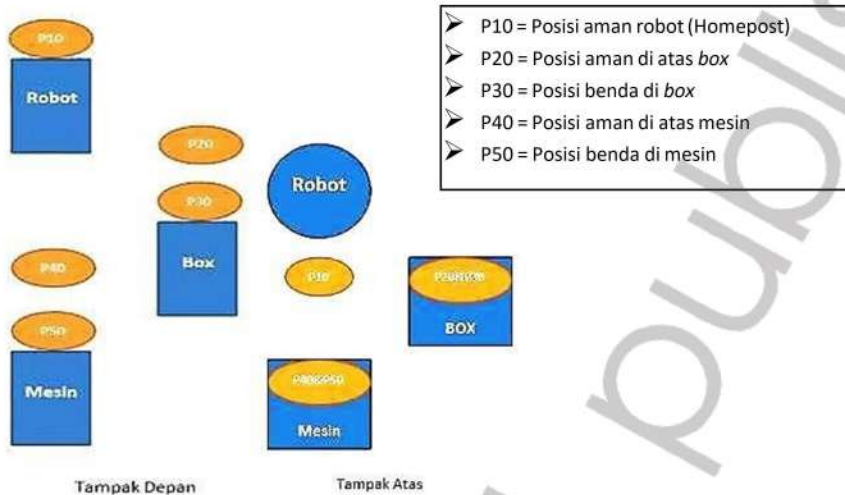
- Buka file yang tadi telah disimpan



- *File* akan terbuka, lalu kemudian *setting* posisi robot karena posisi robot di RobotStudio dan robot ABB berbeda.



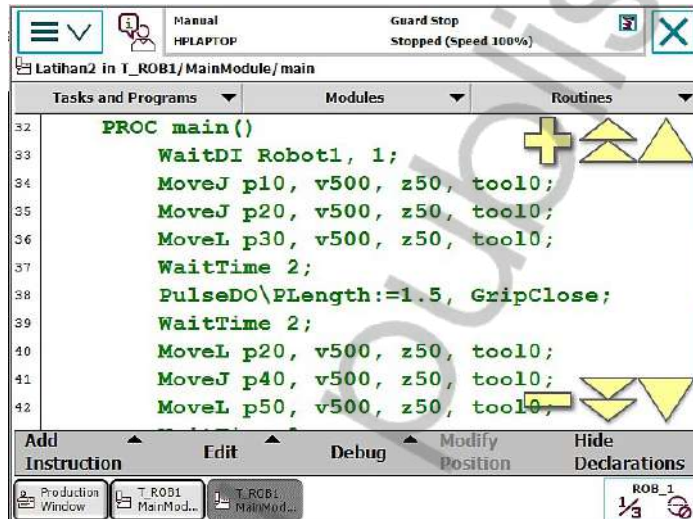
LATIHAN 2



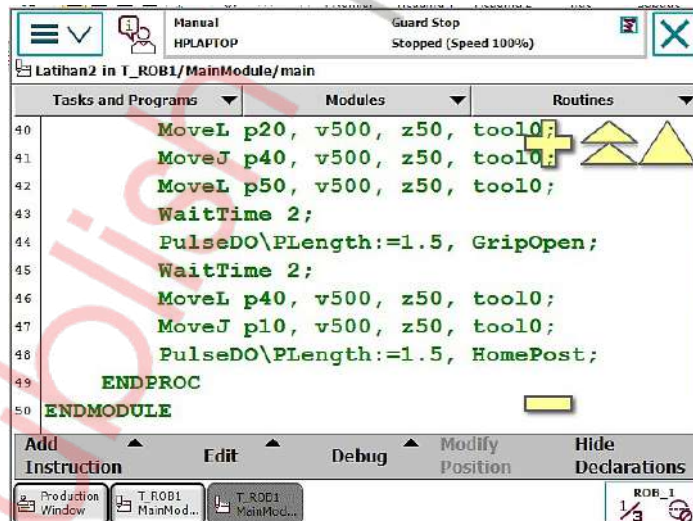
1. Buat program PLC dan program robot dengan ketentuan sebagai berikut:
 - Gunakan tombol **START** untuk memulai program.
 - Robot mengambil *part* dari **BOX** untuk diproses di **MESIN PRESS**.
 - Robot kembali ke posisi aman.
 - Mesin press diumpamakan dengan menghidupkan lampu rotary selama 5 detik.
 - Ketika tombol **START** ditekan robot akan kembali bekerja.

Penyelesaian Latihan 2

1. Program Robot



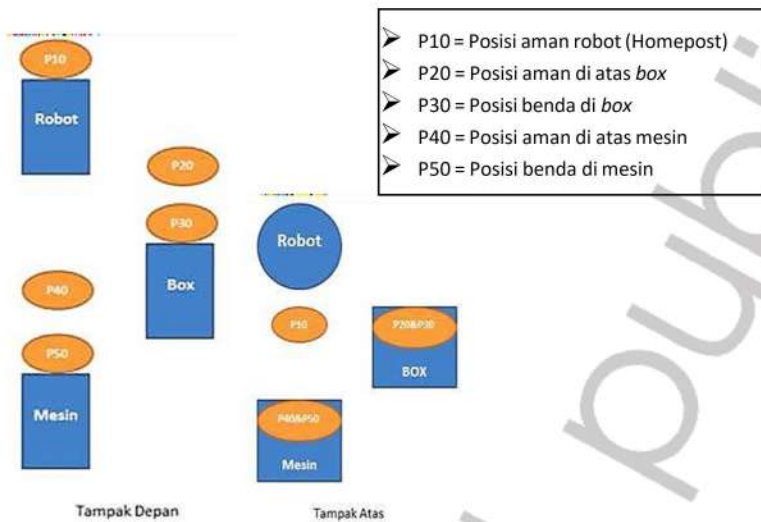
```
32 PROC main()
33     WaitDI Robot1, 1;
34     MoveJ p10, v500, z50, tool0;
35     MoveJ p20, v500, z50, tool0;
36     MoveL p30, v500, z50, tool0;
37     WaitTime 2;
38     PulseDO\PLength:=1.5, GripClose;
39     WaitTime 2;
40     MoveL p20, v500, z50, tool0;
41     MoveJ p40, v500, z50, tool0;
42     MoveL p50, v500, z50, tool0;
```



```
40     MoveL p20, v500, z50, tool0;
41     MoveJ p40, v500, z50, tool0;
42     MoveL p50, v500, z50, tool0;
43     WaitTime 2;
44     PulseDO\PLength:=1.5, GripOpen;
45     WaitTime 2;
46     MoveL p40, v500, z50, tool0;
47     MoveJ p10, v500, z50, tool0;
48     PulseDO\PLength:=1.5, HomePost;
49 ENDPROC
50 ENDMODULE
```

2. Program PLC dibuat menggunakan *software* CX-Programmer, dapat dilihat di lampiran

SOAL 2



2. Buat program PLC dan program robot dengan ketentuan sebagai berikut:
- Gunakan tombol **START** untuk memulai program.
 - Robot mengambil *part* dari **BOX** untuk diproses di **MESIN PRESS**.
 - Robot kembali ke posisi aman.
 - Mesin press diumpamakan dengan lampu *rotary* yang akan menyala selama 5detik.
 - Setelah selesai, robot akan mengambil *part* dari **MESIN PRESS** untuk kemudiandiletakkan kembali ke **BOX**.
 - Setelah itu robot kembali ke posisi aman.
 - Program akan berjalan sebanyak 2 kali.
 - Ketika tombol **STAR** ditekan robot akan kembali bekerja.
 - *Counter* dan *timer* berada pada program PLC.
 - Alamat PLC bisa di lihat di lampiran.

Penyelesaian Soal 2

1. Program robot

```
PROC main ()
32   WaitDI Robot1, 1;
34   Moved p10, v500, z50, tool0;
35   Moved p20, v500, z50, tool0;
36   Moved p30, v500, z50, tool0;
37   WaitTime 2;
38   PulseDO\PLength:=1.5, GripClose;
39   WaitTime 2;
40   Moved p20, v500, z50, tool0;
41   Moved p40, v200, z50, tool0;
42   Moved p50, v500, z50, tool0;
43   WaitTime 2;
44   PulseDO\PLength:=1.5, GripOpen;
45   WaitTime 2;
46   Moved p40, v500, z50, tool0;
47   Moved p10, v500, z50, tool0;
48   PulseDO\PLength:=1.5, HomePost;
49   WaitDI Robot1, 1;
```

```
48   PulseDO\PLength:=1.5, HomePost;
49   WaitDI Robot1, 1;
50   Moved p40, v500, z50, tool0;
51   Moved p50, v500, z50, tool0;
52   WaitTime 2;
53   PulseDO\PLength:=1.5, GripClose;
54   WaitTime 2;
55   Moved p40, v500, z50, tool0;
56   Moved p20, v500, z50, tool0;
57   Moved p30, v500, z50, tool0;
58   WaitTime 2;
59   PulseDO\PLength:=1.5, GripOpen;
60   WaitTime 2;
61   Moved p20, v500, z50, tool0;
62   Moved p10, v500, z50, tool0;
63   PulseDO\PLength:=1.5, HomePost;
64   END PROC
65   ENDMODULE;
```

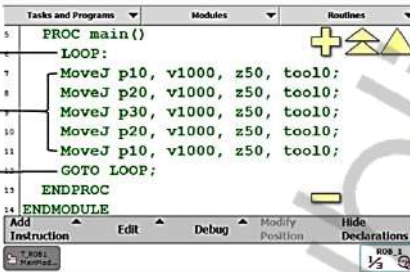
2. Program PLC dibuat menggunakan *software* CX-Programmer, dapat dilihat di lampiran

3. PROGRAM LOOPING

Buat "Label" sesuai yang Anda inginkan. Posisi harus berada di atas program yang akan di-looping.

Program yang akan di-looping.

GOTO "Label" = program akan kembali ke <Label> dan membaca program yang ada di bawahnya.

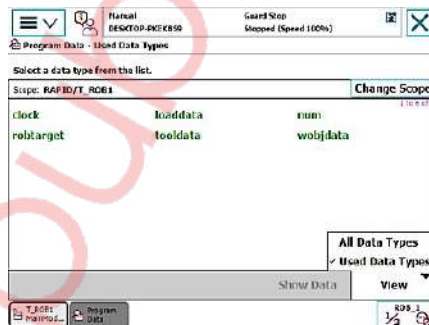


4. PROGRAM COUNTER

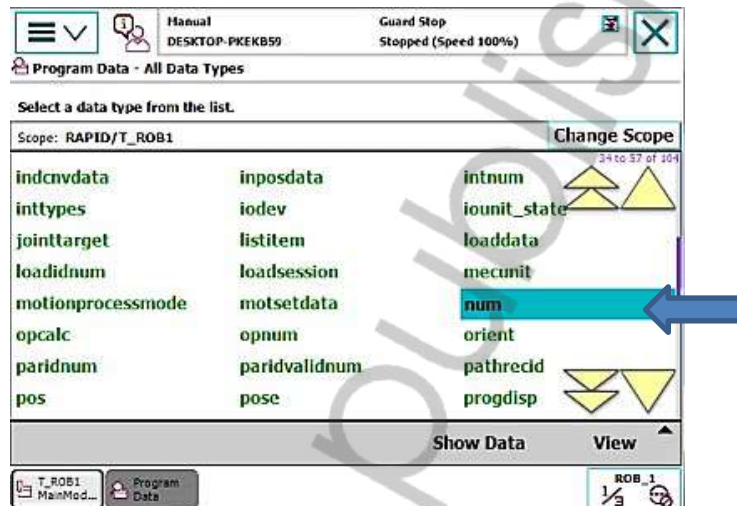
- a. Langkah pertama membuat program *counter* yaitu dengan menambahkan *variable* pada Menu **Program Data** terlebih dahulu.



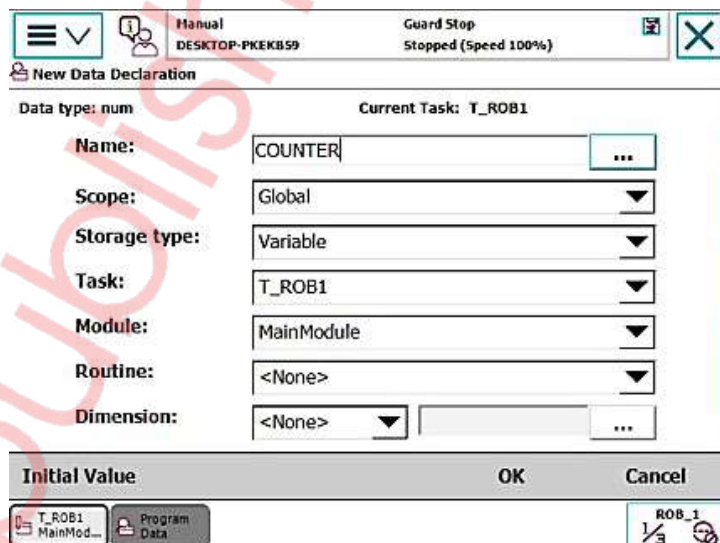
- b. Klik **View** lalu pilih **All Data Type** agar semua data dapat terlihat.



- c. Pilih tipe data **Num**, lalu klik **Show Data**.



- d. Pilih **New** untuk menambah *variable counter baru*, lalu beri nama sesuai denganyang Anda inginkan (contoh: COUNTER).
- e. Setelah selesai pilih **OK**.



- f. Variabel yang baru (contoh: COUNTER) telah muncul dan bisa dimasukkan kedalam program yang akan Anda buat.

Manual DESKTOP-PKEKB59 Guard Stop Stopped (Speed 100%)

Data of type: num

Select the data you want to edit. Active filter:

Scope: RAPID/T_ROB1 Change Scope

Name	Value	Module	Global
COUNTER	0	MainModule	Global
reg1	0	user	Global
reg2	0	user	Global
reg3	0	user	Global
reg4	0	user	Global
reg5	0	user	Global

New... Edit Refresh View Data Types

T_ROB1 MainMod... Program Data ROB_1

- g. Penggunaan variabel counter

Manual DESKTOP-PKEKB59 Guard Stop Stopped (Speed 100%)

Tasks and Programs Modules Routines

```

6 CONST robtarg P40:=[1955.00,0.00,1185.00]^(0.7071
7 VAR num COUNTER:=0;
8 PROC main()
9   MoveJ HOME, v1000, z50, tool0;
10  LOOP:
11   MoveJ P20, v1000, z50, tool0;
12   MoveJ P30, v1000, z50, tool0;
13   MoveJ P40, v1000, z50, tool0;
14   MoveJ P30, v1000, z50, tool0;
15   MoveJ P20, v1000, z50, tool0;
16   Incr COUNTER;
17   IF COUNTER < 3 THEN
18     GOTO LOOP;

```

Add Instruction Edit Debug Modify Position Hide Declarations

T_ROB1 MainMod... Program Data ROB_1

Variable counter

Program bergerak ke posisi home

Program yang akan di-counter

Incr COUNTER = penambahan nilai variabel ($x + 1$)

Jika nilai COUNTER kurang dari 3 maka program yang dibaca adalah GOTO LOOP.

Jika nilai COUNTER sama dengan 3 maka robot akan ke posisi HOME dan nilai COUNTER menjadi = 0.

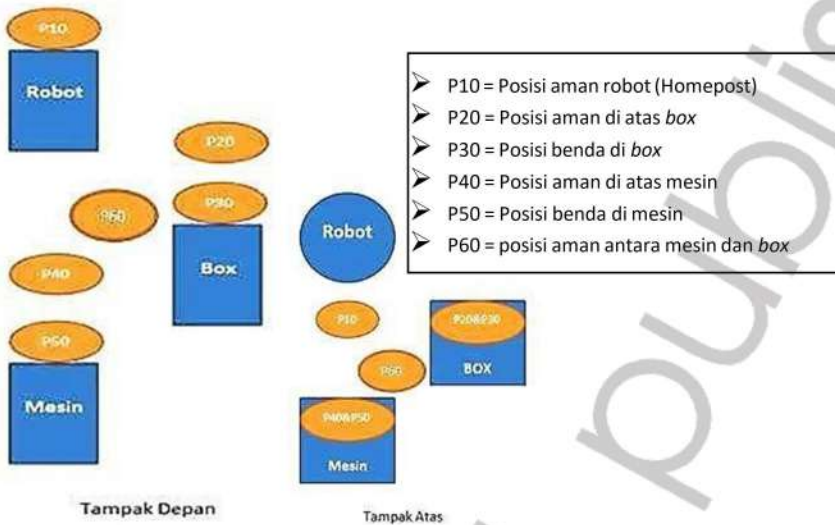
```
Manual Guard Stop
DESKTOP-PC1K839 Stopped (Speed 100%)

cobacounter1 in T_ROB1/MainModule/main

Tasks and Programs Modules Routines
14 MoveJ P30, v1000, z50, tool0;
15 MoveJ B20, v1000, z50, tool0;
16 Incr COUNTER;
17 IF COUNTER < 3 THEN
18   GOTO LOOP;
19 ENDIF
20 IF COUNTER = 3 THEN
21   MoveJ HOME, v1000, z50, tool0;
22   COUNTER := 0;
23 ENDIF
24 ENDPROC
25 ENDMODULE

Add Instruction Edit Debug Modify Hide
Instruction Position Declarations
T_ROB1 Program
MainMod... Data ROB_1
```

SOAL 3

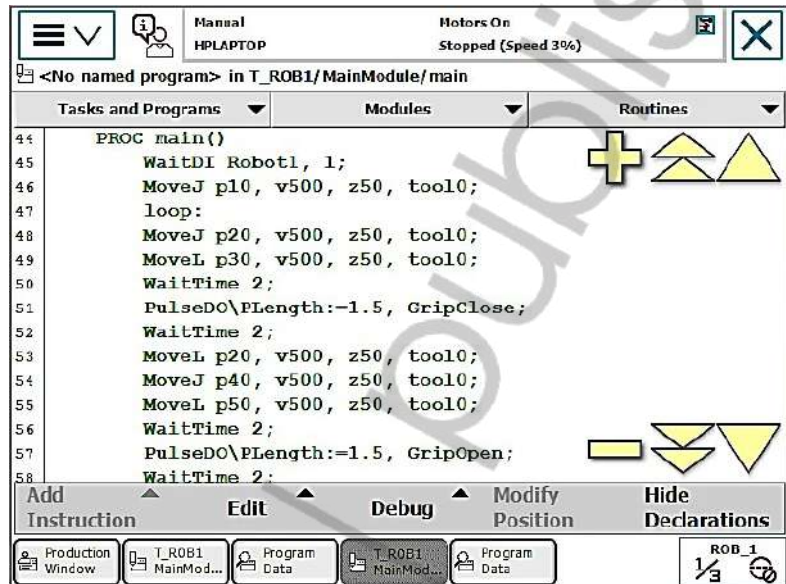


1. Buat program sebagai berikut:

- Apabila tombol **START** ditekan, robot akan bergerak dari posisi **HOME** menuju **BOX** untuk mengambil benda kerja.
- Selanjutnya robot akan bergerak menaruh benda kerja ke **MESIN**.
- Mesin akan memproses benda kerja selama 3 detik dengan menyalakan **lampu rotary**.
- Robot akan menunggu proses mesin di posisi **P60**.
- Kemudian robot mengembalikan benda di **BOX**.
- Robot akan menunggu penggantian benda selama 2 detik di posisi P60.
- Program di atas akan berulang sebanyak 2x.
- Jika sudah 2x robot akan kembali ke posisi **HOME** dan *counter* akan ter-reset secara otomatis.

Penyelesaian Soal 3

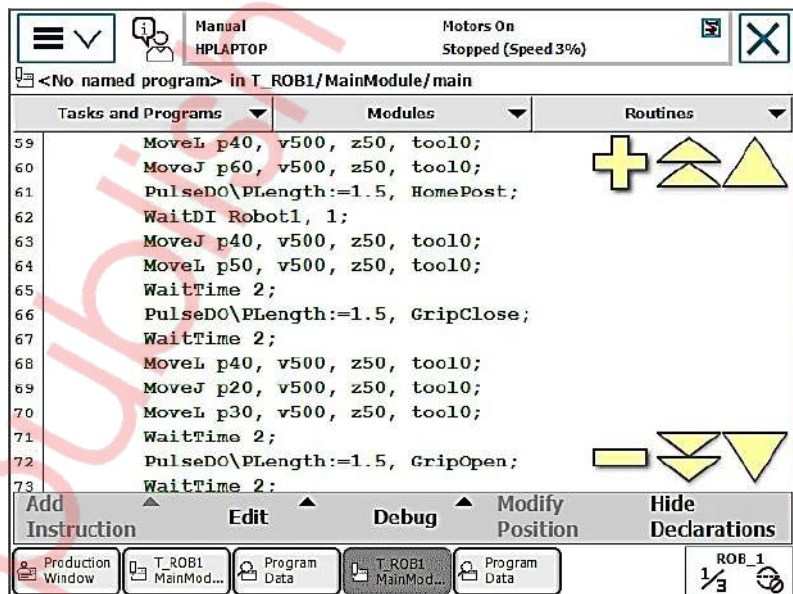
1. Program Robot



The screenshot shows a software interface for a robot program. At the top, there is a menu icon, a manual icon, and the text 'Manual HPLAPTOP'. On the right, it says 'Motors On Stopped (Speed 3%)'. Below this, it indicates '<No named program> in T_ROB1/MainModule/main'. The main area is divided into three tabs: 'Tasks and Programs', 'Modules', and 'Routines'. The 'Routines' tab is active, showing a list of instructions for a routine named 'main()'. The instructions are as follows:

```
44 PROC main()
45     WaitDI Robot1, 1;
46     MoveJ p10, v500, z50, tool0;
47     loop:
48     MoveJ p20, v500, z50, tool0;
49     MoveL p30, v500, z50, tool0;
50     WaitTime 2;
51     PulseDO\PLength:=1.5, GripClose;
52     WaitTime 2;
53     MoveL p20, v500, z50, tool0;
54     MoveJ p40, v500, z50, tool0;
55     MoveL p50, v500, z50, tool0;
56     WaitTime 2;
57     PulseDO\PLength:=1.5, GripOpen;
58     WaitTime 2;
```

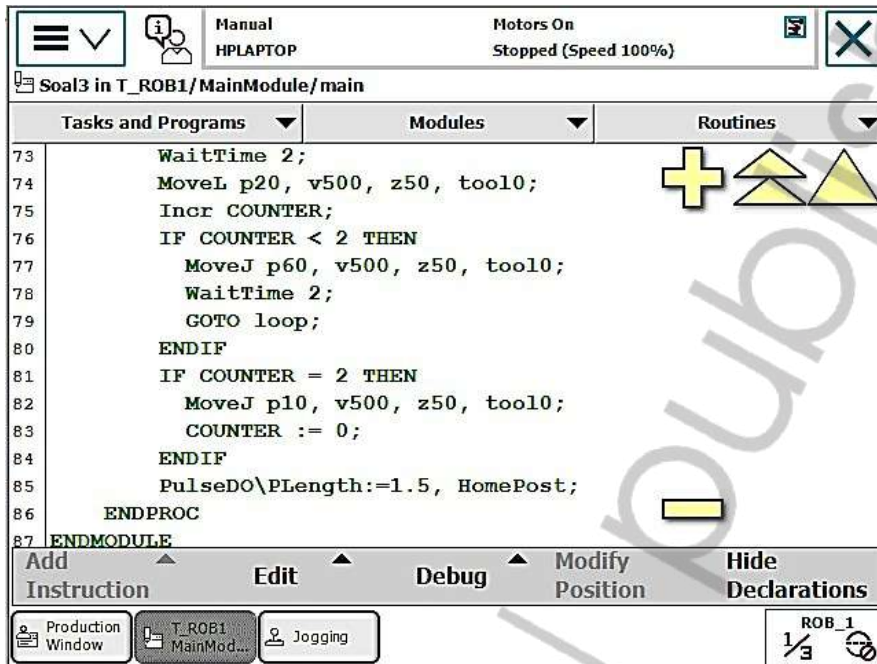
At the bottom, there is a toolbar with buttons for 'Add Instruction', 'Edit', 'Debug', 'Modify Position', and 'Hide Declarations'. Below the toolbar are several window icons: 'Production Window', 'T_ROB1 MainMod...', 'Program Data', 'T_ROB1 MainMod...', 'Program Data', and 'ROB_1'.



The screenshot shows a software interface for a robot program, similar to the one above. At the top, there is a menu icon, a manual icon, and the text 'Manual HPLAPTOP'. On the right, it says 'Motors On Stopped (Speed 3%)'. Below this, it indicates '<No named program> in T_ROB1/MainModule/main'. The main area is divided into three tabs: 'Tasks and Programs', 'Modules', and 'Routines'. The 'Routines' tab is active, showing a list of instructions for a routine named 'main()'. The instructions are as follows:

```
59 MoveL p40, v500, z50, tool0;
60 MoveJ p60, v500, z50, tool0;
61 PulseDO\PLength:=1.5, HomePost;
62 WaitDI Robot1, 1;
63 MoveJ p40, v500, z50, tool0;
64 MoveL p50, v500, z50, tool0;
65 WaitTime 2;
66 PulseDO\PLength:=1.5, GripClose;
67 WaitTime 2;
68 MoveL p40, v500, z50, tool0;
69 MoveJ p20, v500, z50, tool0;
70 MoveL p30, v500, z50, tool0;
71 WaitTime 2;
72 PulseDO\PLength:=1.5, GripOpen;
73 WaitTime 2;
```

At the bottom, there is a toolbar with buttons for 'Add Instruction', 'Edit', 'Debug', 'Modify Position', and 'Hide Declarations'. Below the toolbar are several window icons: 'Production Window', 'T_ROB1 MainMod...', 'Program Data', 'T_ROB1 MainMod...', 'Program Data', and 'ROB_1'.



2. Program PLC dibuat menggunakan *software* CX-Programmer, dapat dilihat di lampiran

ROBOT ABB DAN KONVEYOR

Pembahasan pada bagian ini adalah sebagai berikut:

1) Integrasi Robot dan Konveyor

Merupakan komunikasi 1 robot dengan konveyor sebagai *transporter*.

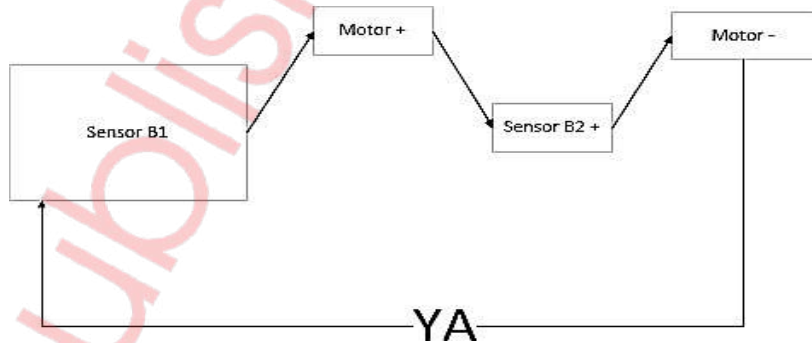
2) Pemrograman Posisi

Merupakan cara untuk menentukan posisi di robot 1 ke posisi penempatan benda di konveyor.

1. INTEGRASI ROBOT DAN KONVEYOR

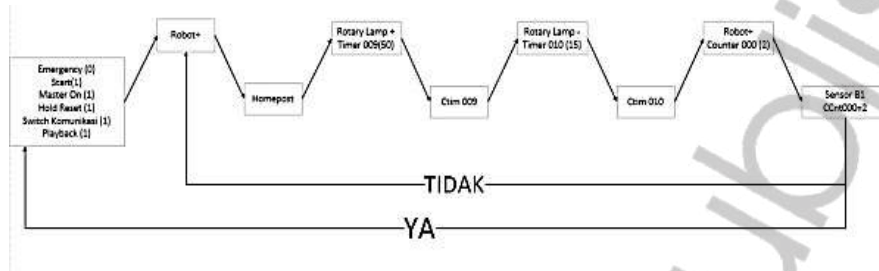
Sekuensial

- Benda terdeteksi sensor 1
- Motor konveyor berjalan aktif
- Benda terdeteksi sensor 2
- Motor konveyor berhenti



2. PEMROGRAMAN POSISI

Sekuensial



LATIHAN 3



1. Buat program sebagai berikut:
 - Ketika Tombol **START** ditekan, robot akan bergerak dari posisi **AMAN ROBOT**
 - menuju **PALLET** untuk mengambil benda kerja
 - Kemudian Robot membawa benda kerja menuju **STATION 1**, robot akan kembali ke posisi **AMAN ROBOT**, **ROTARY LAMP** akan menyala selama 5 detik untuk menandakan benda kerja sedang di proses di **STATION 1**
 - Selanjutnya robot akan bergerak dari posisi **AMAN ROBOT** menuju **STATION 1**
 - untuk mengambil benda kerja
 - Kemudian Robot membawa benda kerja menuju **KONVEYOR**
 - Program di atas akan berulang sebanyak 2x
 - Jika sudah 2x program berjalan, maka robot akan kembali ke posisi **AMAN**
 - **ROBOT** dan *counter* akan ter reset ketika tombol **START** ditekan

Penyelesaian Latihan 3

1. Program Robot–Robot kanan (A)

```
PROC Main ( )
```

```
WaitDi Robot1, 1;
```

```
WaitTime 1;
```

```
MoveJ p10, v500, z80,Sebastian;
```

```
MoveJ p20,v500, z80, Sebastian;
```

```
MoveL p30, v500, z80, Sebastian;
```

```
WaitTime 2;
```

```
PulseDo\PLength:=2, grip_close;
```

```
WaitTime 2;
```

```
MoveL p20, v500, z80,Sebastian;
```

```
MoveJ p10,v500, z80, Sebastian;
```

```
MoveJ p40, v500, z80,Sebastian;
```

```
MoveL p50, v500, z80,Sebastian;
```

```
WaitTime 2;
```

```
PulseDo\PLength:= 2, grip_open;
```

```
WaitTime 2;PulseDo\PLength:=2, HomePost;
```

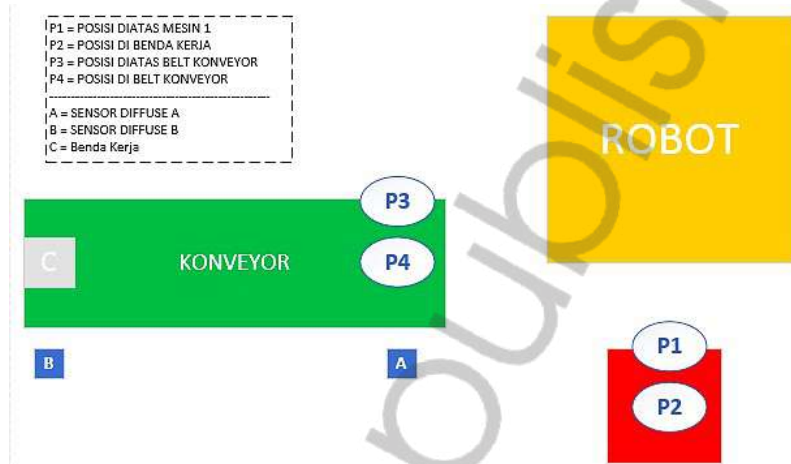
```
MoveL p40, v500, z80,Sebastian;
```

```
MoveJ p10, v500, z80,Sebastian;
```

```
ENDPROC
```

2. Program PLC dibuat menggunakan *software* CX-Programmer, dapat dilihat di lampiran

SOAL 4



1. Buat program sebagai berikut:

- Ketika Tombol **START** ditekan, maka **ROTARY LAMP** akan menyala selama 5 detik untuk menandakan benda kerja sedang di proses di **STATION 1**, robot berada di posisi **AMAN ROBOT**
- Selanjutnya robot akan bergerak dari posisi **AMAN ROBOT** menuju **STATION 1**
- untuk mengambil benda kerja.
- Kemudian Robot membawa benda kerja menuju **KONVEYOR**.
- Ketika **SENSOR B1** dan **HOMEPOST** aktif, maka **KONVEYOR** berjalan hingga benda kerja terdeteksi oleh **SENSOR B2**, **KONVEYOR** akan berhenti.
- Program di atas akan berulang sebanyak 2x.
- Jika sudah 2x program berjalan, maka robot akan kembali ke posisi **AMAN ROBOT** dan *counter* akan ter reset ketika tombol **START** ditekan.

Ket: Gerakan Robot dalam mengambil dan menaruh benda tidak boleh menunggu benda kerja dideteksi oleh **SENSOR B2**, sehingga proses produksi menjadi lebih efisien

Penyelesaian Soal 4

1. Program Robot–Robot Kanan (A)

```
PROC Main ()  
WaitDi Robot1, 1;  
WaitTime 1;  
MoveJ p10, v500, z80, Sebastian;  
MoveJ p20, v500, z80, Sebastian;  
MoveL p30, v500, z80, Sebastian;  
WaitTime 2;  
PulseDo\PLength:=2, grip_close;  
WaitTime 2;  
MoveL p20, v500, z80, Sebastian;  
MoveJ p10, v500, z80, Sebastian;  
MoveJ p40, v500, z80, Sebastian;  
MoveL p50, v500, z80, Sebastian;  
WaitTime 2;  
PulseDo\PLength:=2, grip_open;  
WaitTime 2;  
PulseDo\PLength:=2, HomePost;  
MoveL p40, v500, z80, Sebastian;  
MoveJ p10, v500, z80, Sebastian;  
ENDPROC
```

2. Program PLC dibuat menggunakan *software* CX-Programmer, dapat dilihat di lampiran

KOMUNIKASI ANTARA DUA ROBOT ABB DAN PLC

Pembahasan pada bagian ini adalah sebagai berikut:

1) Komunikasi 2 Robot ABB IRB1410 dengan 2 PLC

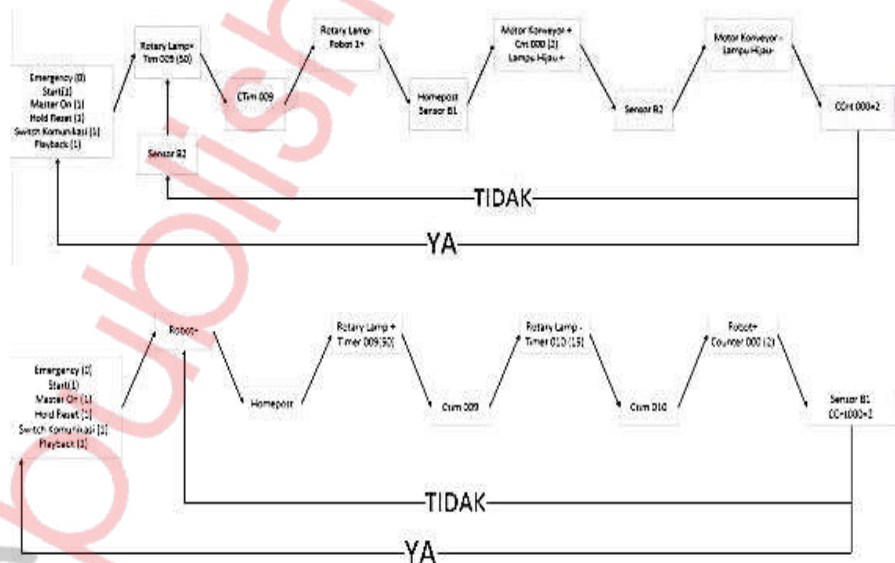
Merupakan komunikasi 2 robot ABB IRB1410 dengan 2 PLC di mana, komunikasi menggunakan I/O PLC.

2) Wiring Komunikasi

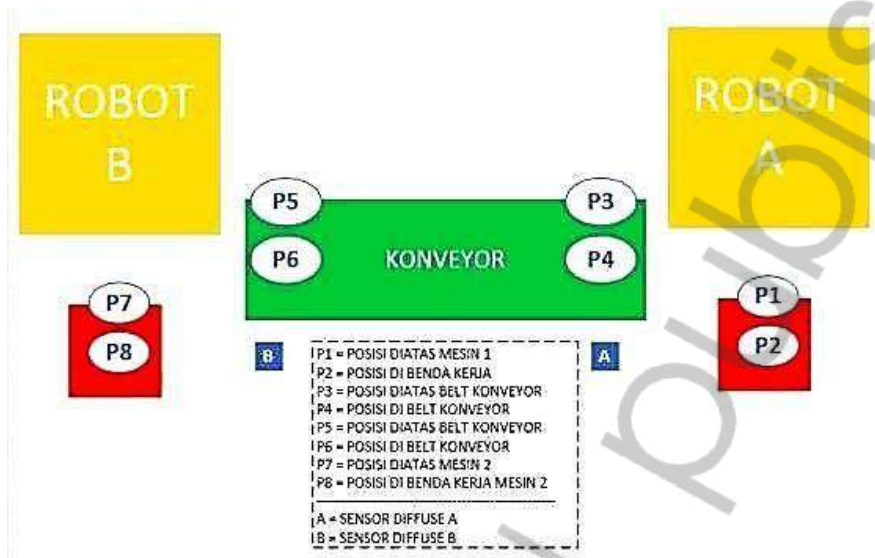
Wiring komunikasi menggunakan I/O PLC, ada beberapa port yang salingberkomunikasi, berikut daftar I/O (lihat di lampiran).

1. Komunikasi 2 Robot dengan 2 PLC

Sekuensial Robot



SOAL 5



1. Buat program berikut:

- Ketika **SWITCH KOMUNIKASI** di semua PLC di aktifkan dan semua Tombol **START** ditekan, maka **ROTARY LAMP** akan menyala selama 5 detik untuk menandakan benda kerja sedang di proses di **STATION 1**, **ROBOT A & ROBOT B** berada di posisi **AMAN ROBOT**
- Selanjutnya **ROBOT A** akan bergerak dari posisi **AMAN ROBOT** menuju **STATION 1** untuk mengambil benda kerja.
- Kemudian **ROBOT A** membawa benda kerja menuju **KONVEYOR**.
- Ketika **SENSOR B1** dan **HOMEPOST** aktif, maka **KONVEYOR** berjalan hingga benda kerja terdeteksi oleh **SENSOR B2**, **KONVEYOR** akan berhenti.
- **ROBOT B** akan mengambil benda kerja di **KONVEYOR**.
- Kemudian **ROBOT B** akan menaruh benda kerja ke **STATION 2** untuk diproses selama 5 detik, ditandai dengan **ROTARY LAMP** aktif.

- **ROBOT B** kembali ke posisi **AMAN ROBOT**.
- Program akan berulang hingga benda kerja yang ditransportasikan dari **STATION 1** menuju **STATION 2** berjumlah **2**.
- Jika sudah 2 benda yang ditransportasikan, maka **ROBOT A & ROBOT B** berada di posisi **AMAN ROBOT**.

Ket: Gerakan ROBOT A & ROBOT B dalam mengambil dan menaruh benda kerja tidak boleh saling menunggu, sehingga proses produksi menjadi lebih efisien

Penyelesaian Soal 5

1. Program Robot

Robot Kiri (B)

```
PROC main ()  
WaitDi Robot1, 1;  
WaitTime 1;  
MoveJ p10, v500, z80, Sebastian;  
MoveJ p20, v500, z80, Sebastian;  
MoveL p30, v500, z80, Sebastian;  
MoveL p60, v500, z80, Sebastian;  
WaitTime 2;  
PulseDo\PLength:=2 , grip_close;  
WaitTime 2;  
MoveL p30, v500, z80, Sebastian;  
MoveL p20, v500, z80, Sebastian;  
MoveJ p10, v500, z80, Sebastian;  
MoveJ p40, v500, z80, Sebastian;  
MoveL p50, v500, z80, Sebastian;  
WaitTime 2;  
PulseDo\PLength:=2, grip_open;  
WaitTime 2;  
PulseDo\PLength:=2,  
HomePost;  
MoveL p40, v500, z80, Sebastian;  
MoveJ p10, v500, z80, Sebastian;  
ENDPROC
```

Robot Kanan (A)

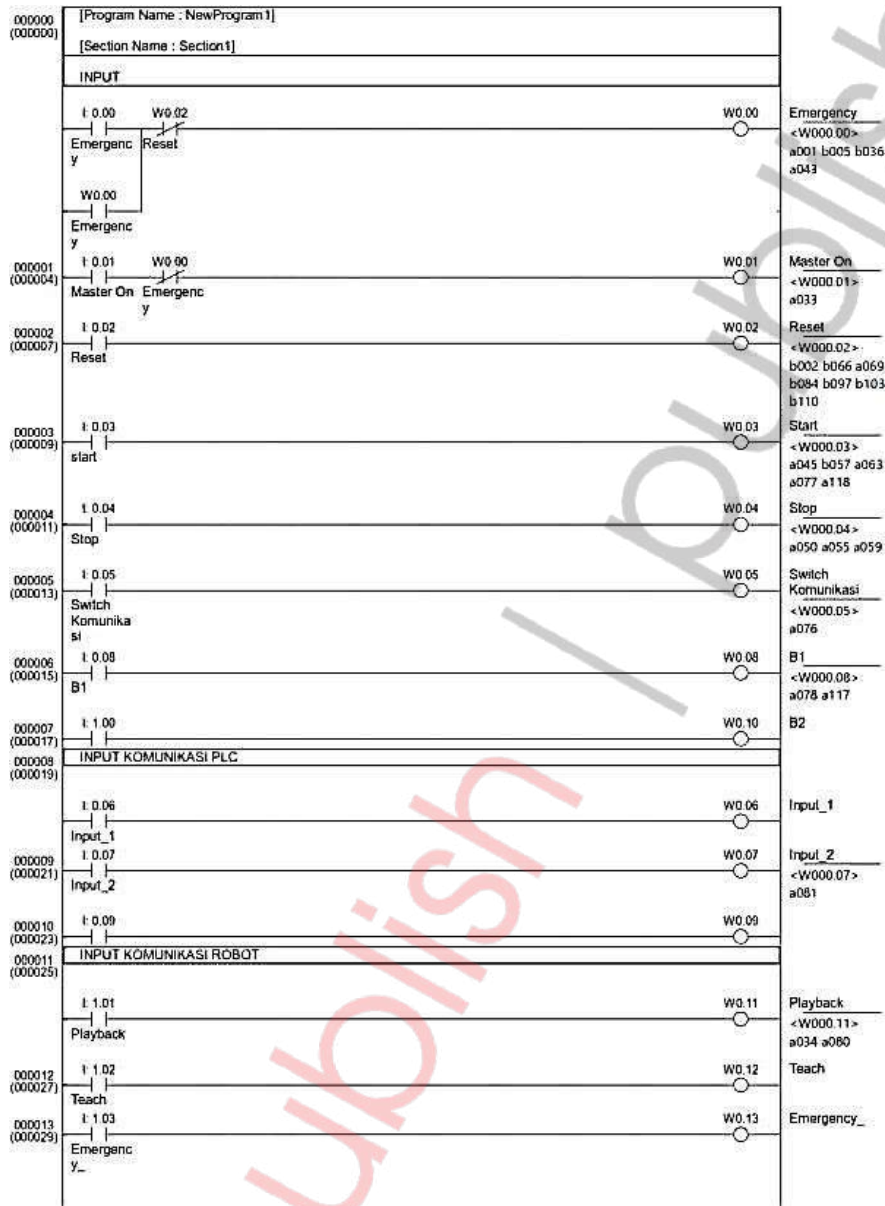
```
PROC main ()  
WaitDi Robot1, 1;  
WaitTime 1;  
MoveJ p10, v500, z80, Sebastian;  
MoveJ p20, v500, z80, Sebastian;
```

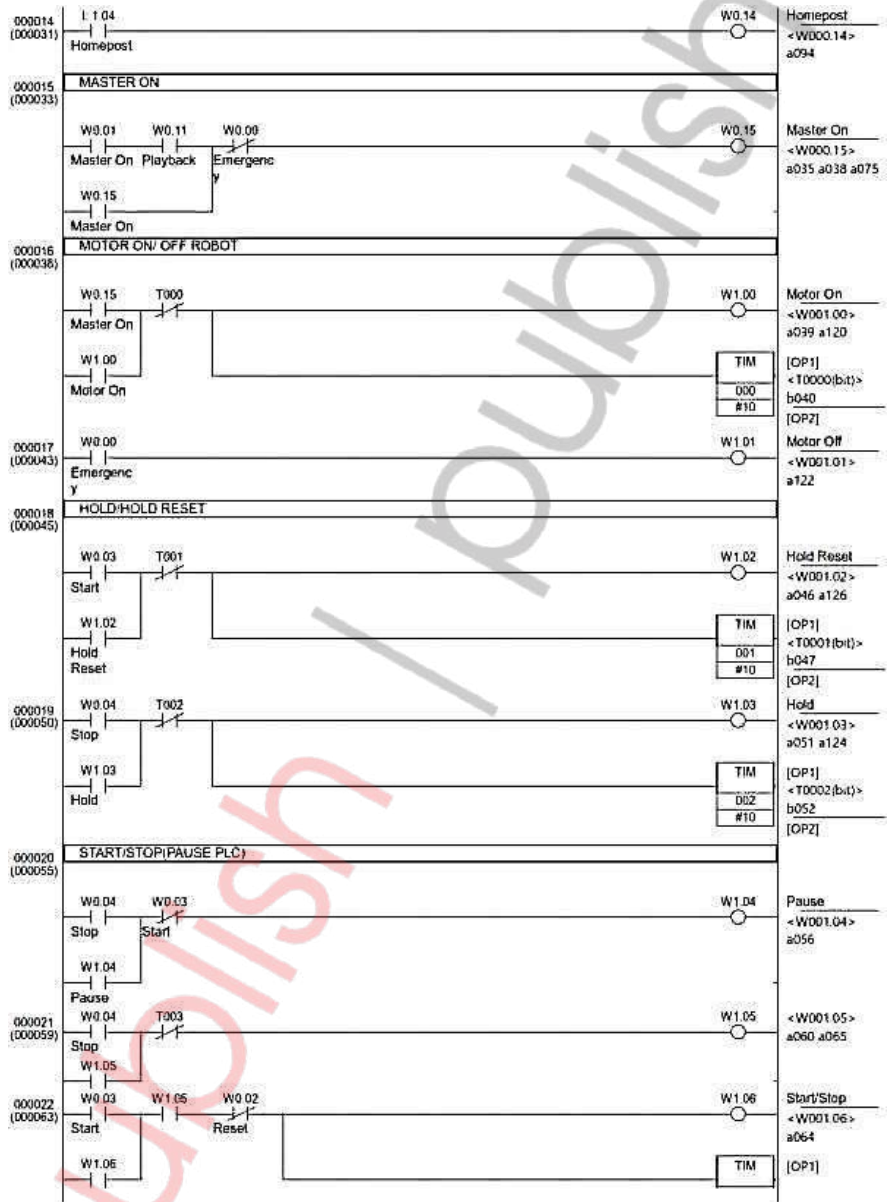
```
MoveL p30, v500, z80, Sebastian;  
WaitTime 2;  
PulseDo\PLength:=2, grip_close;  
WaitTime2;  
MoveL p20, v500, z80, Sebastian;  
MoveJ p10, v500, z80, Sebastian;  
MoveJ p40, v500, z80, Sebastian;  
MoveL p50, v500, z80, Sebastian;  
WaitTime 2;  
PulseDo\PLength:=2, grip_open;  
WaitTime 2;  
PulseDo\PLength:=2,  
HomePost;  
MoveL p40, v500, z80, Sebastian;  
MoveJ p10, v500, z80, Sebastian;  
ENDPROC
```

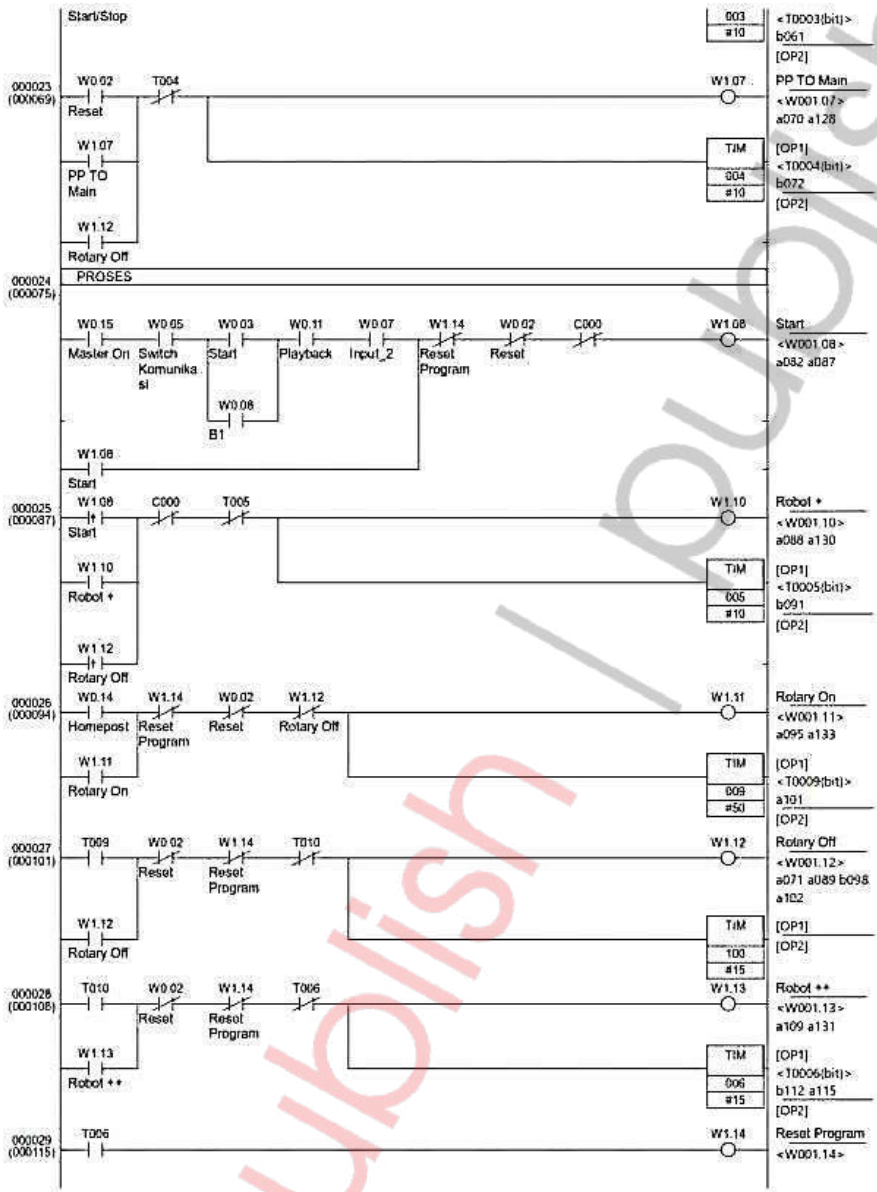
2. Program PLC dibuat menggunakan *software* CX-Programmer, dapat dilihat di lampiran

LAMPIRAN

Diagram *Ladder* Latihan & Soal 2







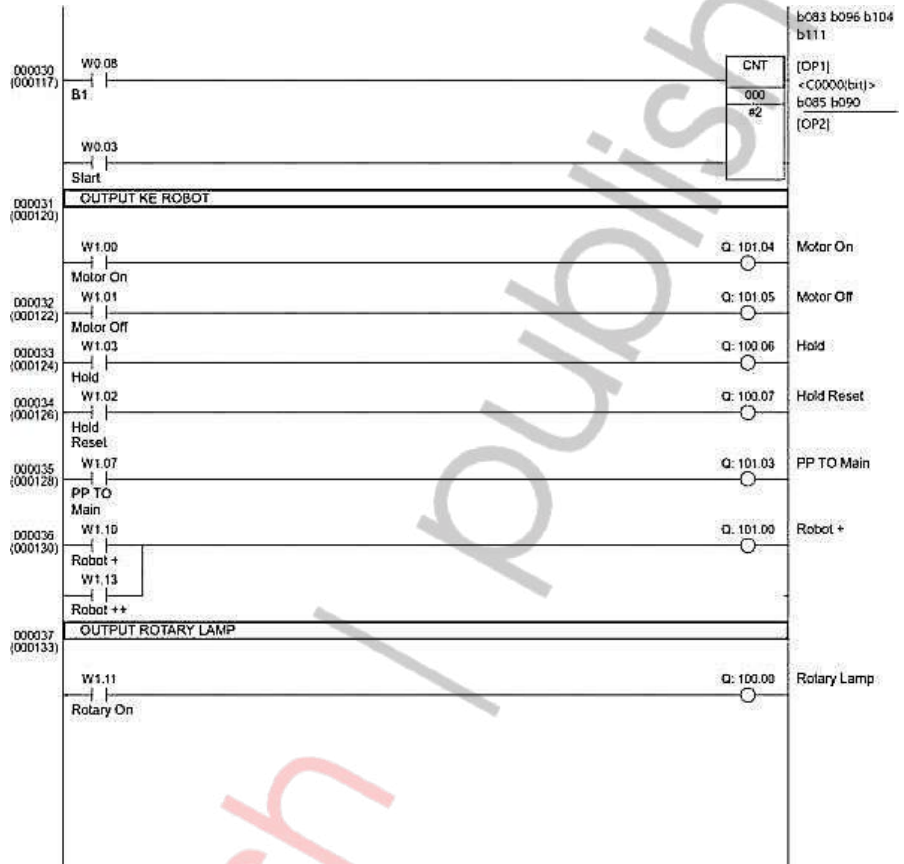
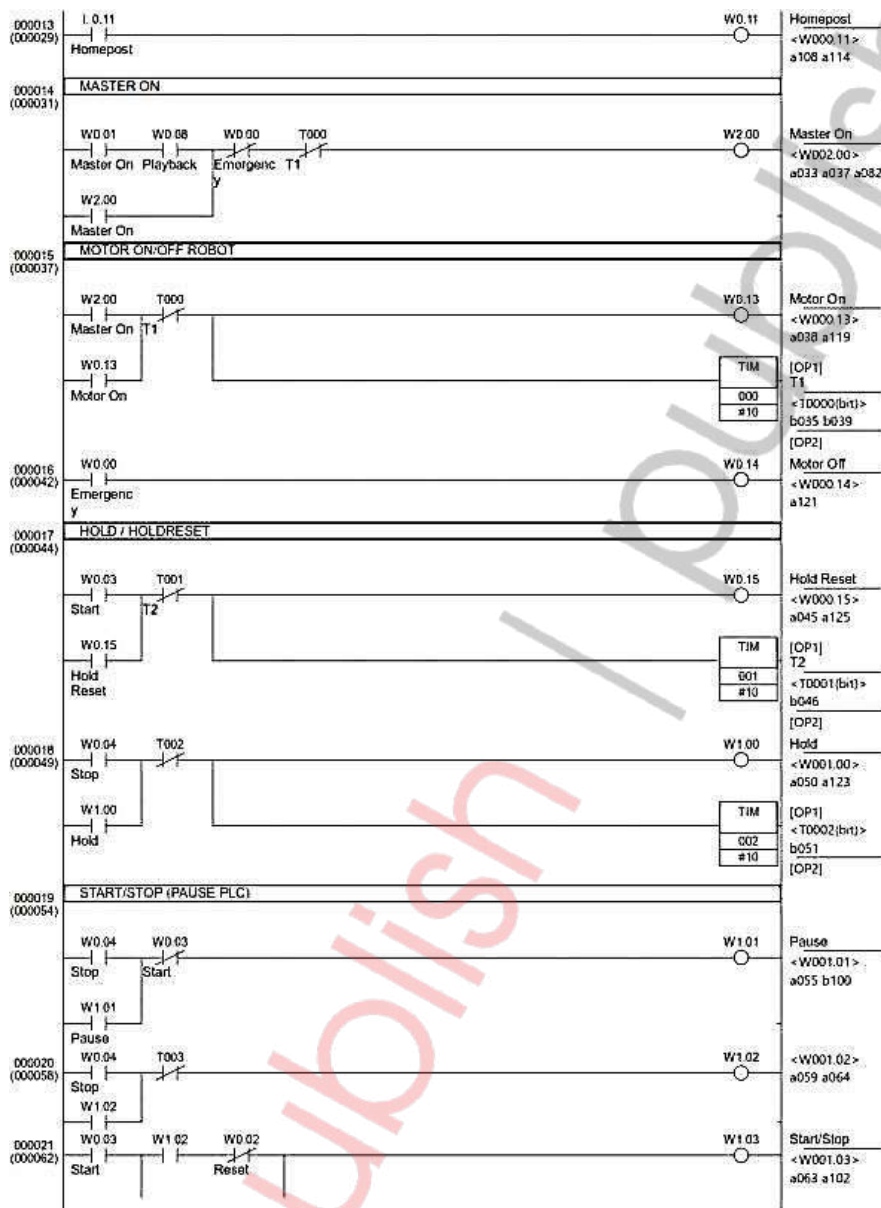
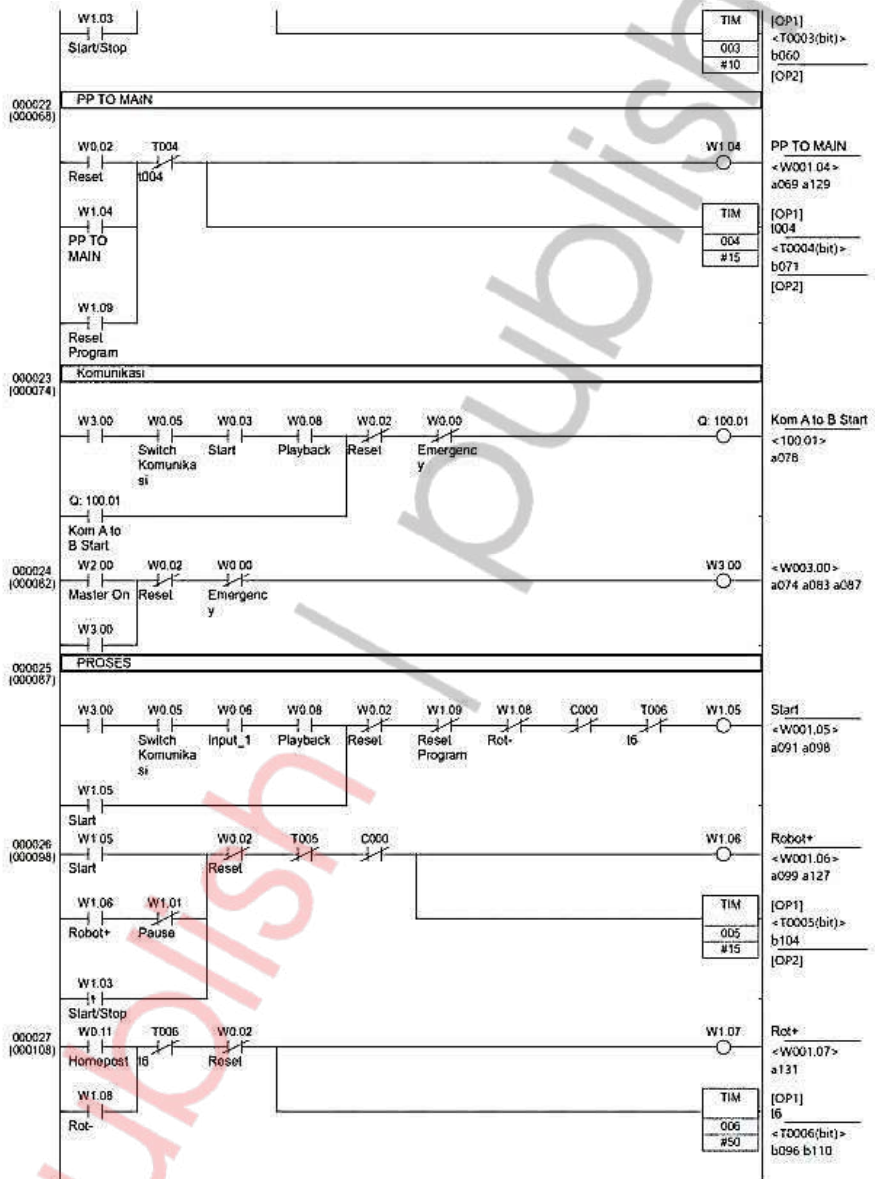
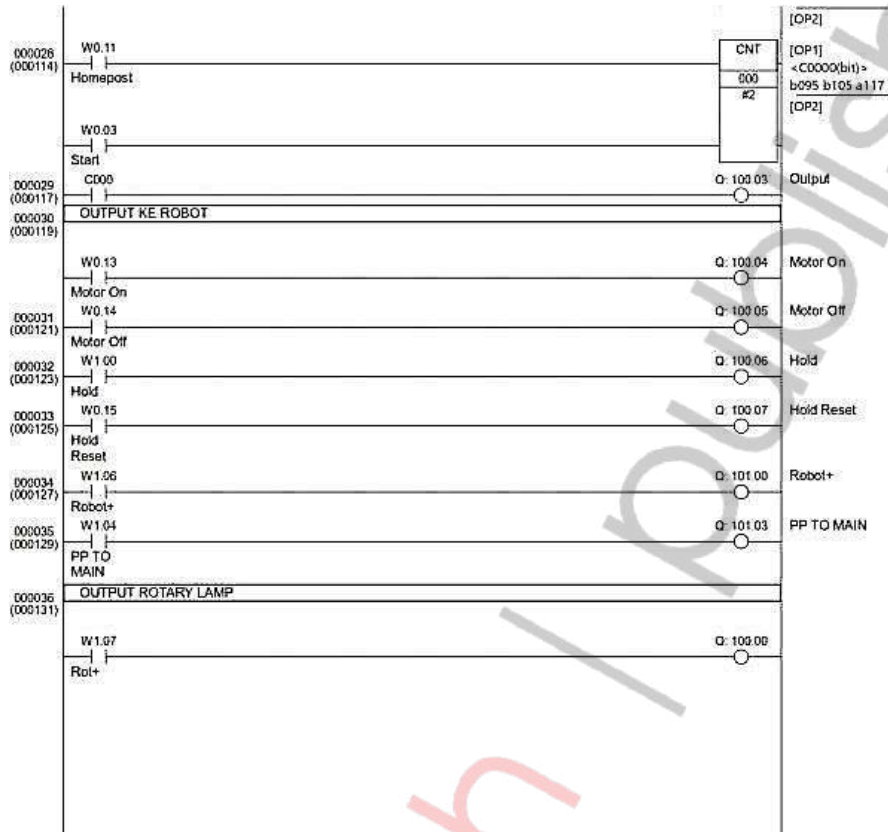


Diagram *Ladder* Soal 3

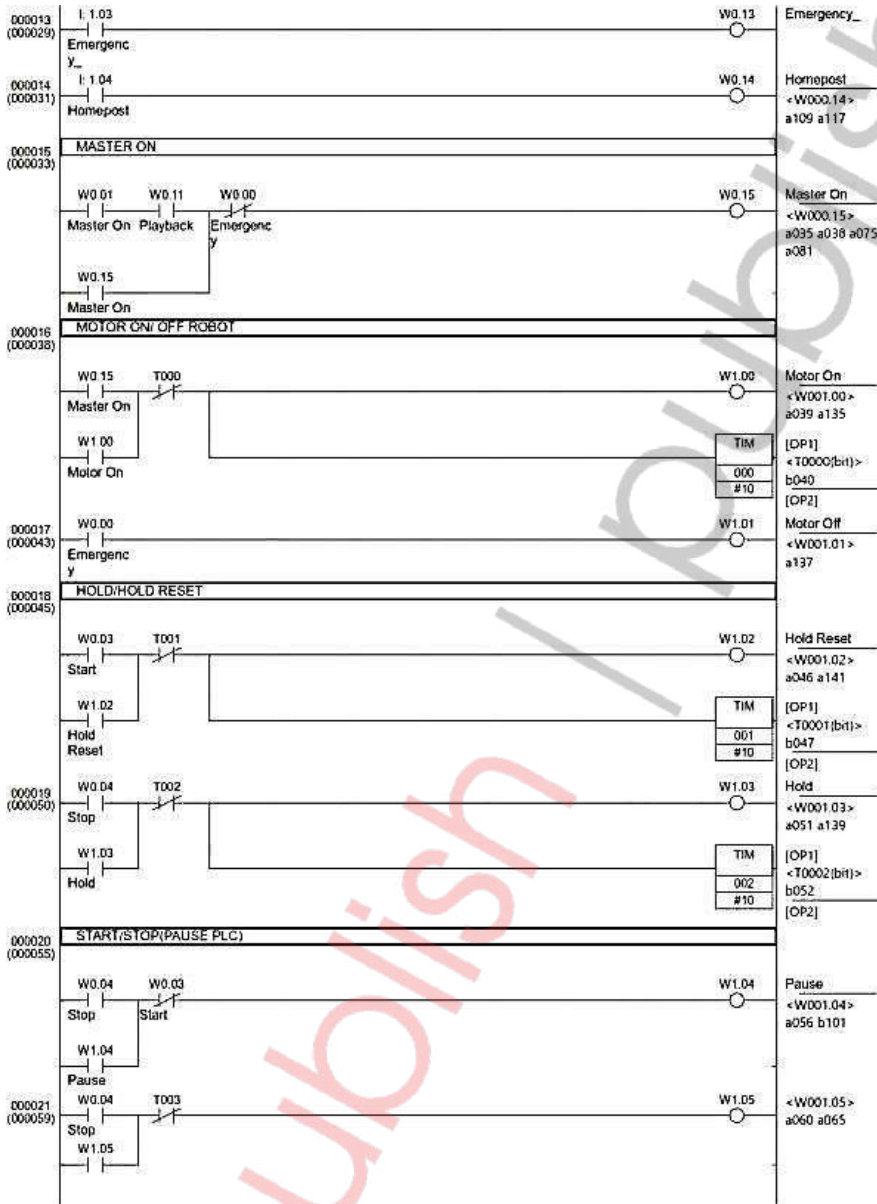
000000 (000000)		[Program Name : NewProgram1]	
		[Section Name : Section1]	
INPUT			
	I 0.00 Emergency	W0.02 Reset	W0.00 Emergency
	W0.00 Emergency		<W000.00> a001 b005 b034 a042 b080 b085
000001 (000004)	I 0.01 Master On	W0.00 Emergency	W0.01 Master On
			<W000.01> a031
000002 (000007)	I 0.02 Reset		W0.02 Reset
			<W000.02> b002 b065 a068 b079 b084 b092 b103 b111
000003 (000009)	I 0.03 start		W0.03 Start
			<W000.03> a044 b056 a062 a076 a115
000004 (000011)	I 0.04 Stop		W0.04 Stop
			<W000.04> a049 a054 a058
000005 (000013)	I 0.05 Switch Komunikasi		W0.05 Switch Komunikasi
			<W000.05> a021 a075 a088
000006 (000015) INPUT KOMUNIKASI PLC			
	I 0.06 Input_1		W0.06 Input_1
			<W000.06> a089
000007 (000017)	I 0.07 Input_2		W0.07 Input_2
000008 (000019)	I 0.08 Input_3		W0.12 Input_3
000009 (000021)	W0.05 Switch Komunikasi	Q 101.01	lamp komunikasi kuning
000010 (000023) INPUT KOMUNIKASI ROBOT			
	I 0.08 Playback		W0.08 Playback
			<W000.08> a032 a077 a090
000011 (000025)	I 0.09 Teach		W0.09 Teach
000012 (000027)	I 0.10 Emergency Robot		W0.10 Emergency Robot

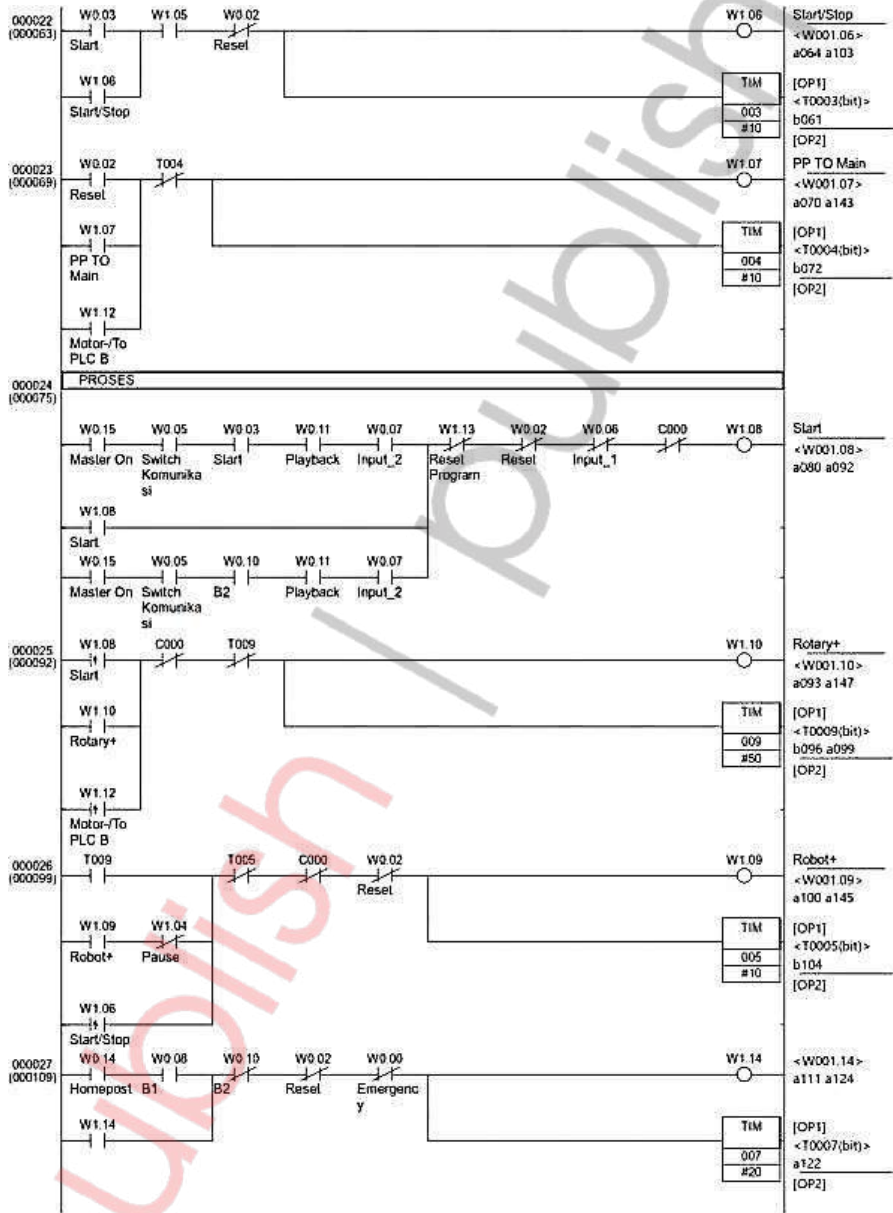


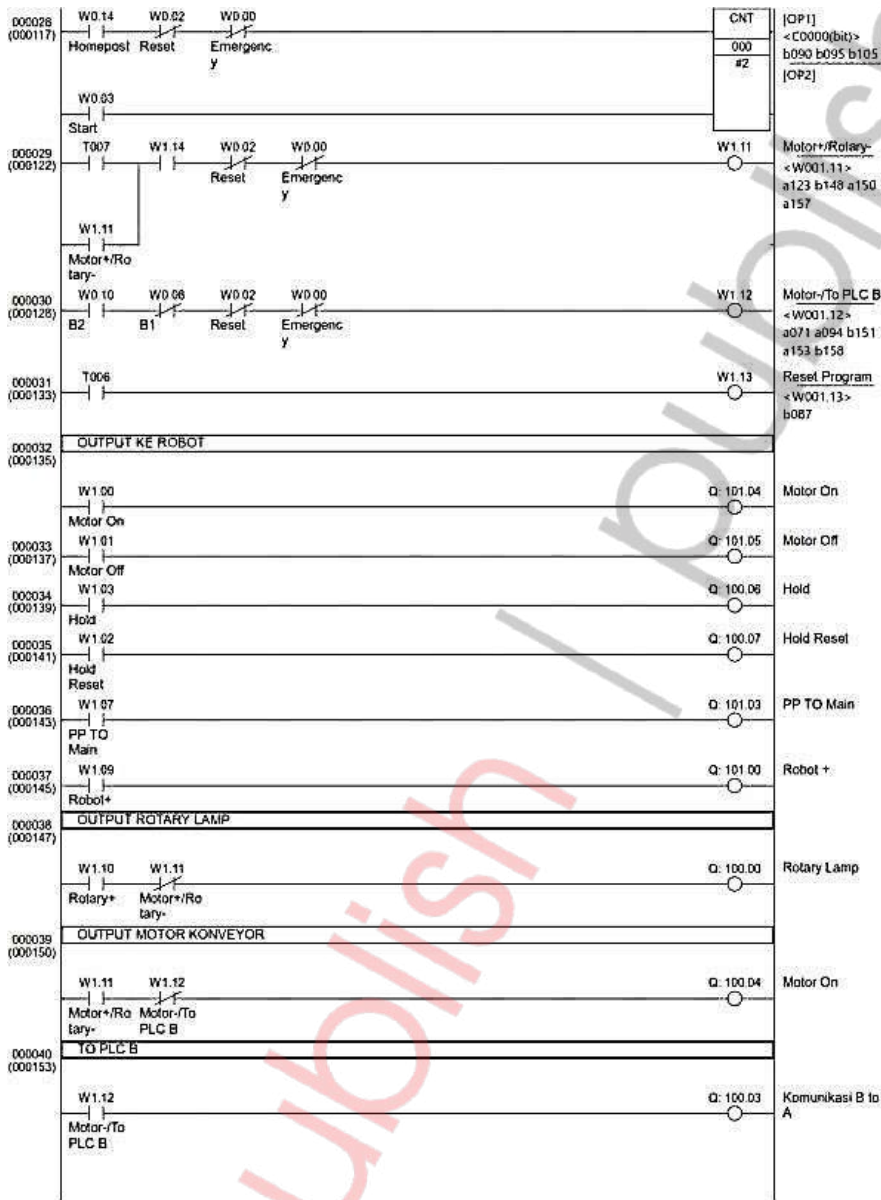




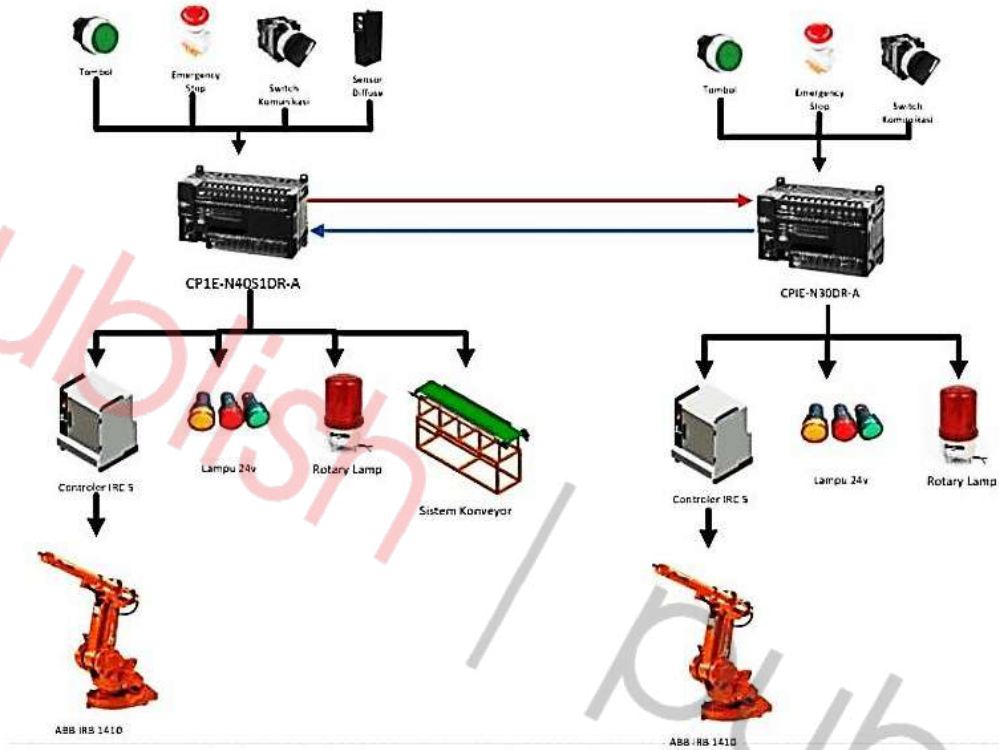
Address	Description	Symbol	Bit Range
000000 (000000)	[Program Name : NewProgram1]		
	[Section Name : Section1]		
	INPUT		
	I: 0.00 Emergency y	W0.00	Emergency <W000.00> a001 b005 b036 a043 b114 b119 b126 b131
	W0.02 Reset		
	W0.00 Emergency y		
000001 (000004)	I: 0.01 Master On	W0.01	Master On <W000.01> a033
	W0.00 Emergency y		
000002 (000007)	I: 0.02 Reset	W0.02	Reset <W000.02> b002 b066 a069 b088 b106 b113 b118 b125 b130
000003 (000009)	I: 0.03 start	W0.03	Start <W000.03> a045 b057 a063 a077 a120
000004 (000011)	I: 0.04 Stop	W0.04	Stop <W000.04> a050 a055 a059
000005 (000013)	I: 0.05 Switch Komunikasi	W0.05	Switch Komunikasi <W000.05> a076 a082 a155
000006 (000015)	I: 0.08 B1	W0.08	B1 <W000.08> a110 b129
000007 (000017)	I: 1.00 B2	W0.10	B2 <W000.10> a083 b112 a128
000008 (000019)	INPUT KOMUNIKASI PLC		
	I: 0.06 Input_1	W0.06	Input_1 <W000.06> b009
000009 (000021)	I: 0.07 Input_2	W0.07	Input_2 <W000.07> a079 a085
000010 (000023)	I: 0.09	W0.09	
000011 (000025)	INPUT KOMUNIKASI ROBOT		
	I: 1.01 Playback	W0.11	Playback <W000.11> a034 a078 a084
000012 (000027)	I: 1.02 Teach	W0.12	Teach











Rancangan Sistem Komunikasi I/O dua Robot ABB IRB1410

Tabel Pengalamatan *Input Output* PLC CP1E N40 S1DR-A

CP1E N40S1DRA

Input		Output	
Nama	Alamat	Nama	Alamat
Tombol Emergency	0.00	Output Komunikasi 1	100.03
Tombol Master On	0.01	Output Komunikasi 2	100.01
Tombol Reset	0.02	Output Komunikasi 3	100.02
Tombol Start	0.03	Rotary Lamp	100.00
Tombol Stop	0.04	Motor Konveyor CW	100.04
Switch Komunikasi	0.05	Motor Konveyor CCW	100.05
Sensor Difuse A	0.08	Komunikasi diMotorOn	101.04
Sensor Difuse B	1.00	Komunikasi diMotorOff	101.05
Input Komunikasi 1	0.06	Komunikasi diHold	100.06
Input Komunikasi 2	0.07	Komunikasi diHoldReset	100.07
Input Komunikasi 3	1.01	Komunikasi diPPtoMain	101.03
Komunikasi doPlayback	0.08	Komunikasi diRobot+	101.00
Komunikasi doTeach	0.09	Komunikasi diRobot++	101.01
Komunikasi doEmergency	0.10	Lampu 1	101.02
Komunikasi doHomepost	0.11	Lampu 2	101.06

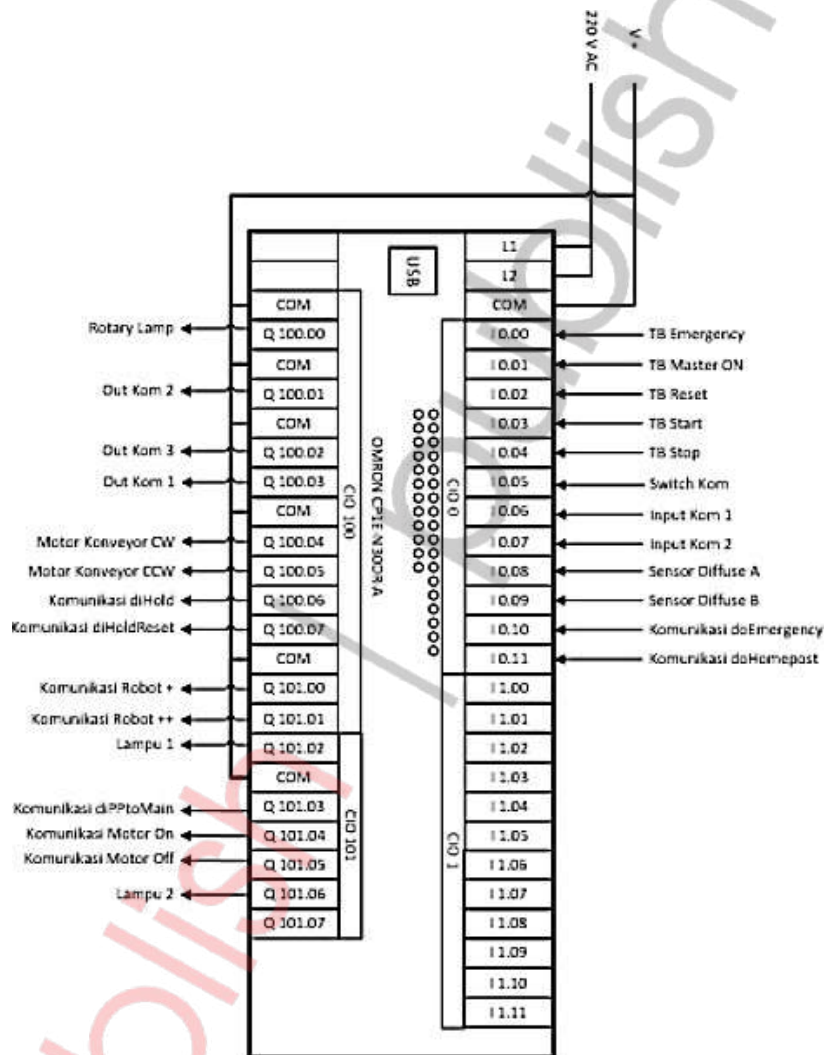
Input		Output	
Nama	Alamat	Nama	Alamat
Komunikasi diMotorOn	X3.1	Komunikasi doPlayback	X1.5
Komunikasi diMotorOff	X3.2	Komunikasi doEmergency	X1.6
Komunikasi diHold	X3.3	Komunikasi doHomepost	X1.7
Komunikasi diHoldReset	X3.4	Grip Open	X1.3
Komunikasi diPPtoMain	X3.8	Grip Close	X1.4
Komunikasi Robot+	X3.5		
Komunikasi Robot++	X3.6		
Komunikasi Robot+++	X3.7		

Tabel Pengalamatan *Input Output* PLC CP1E N30DR-A

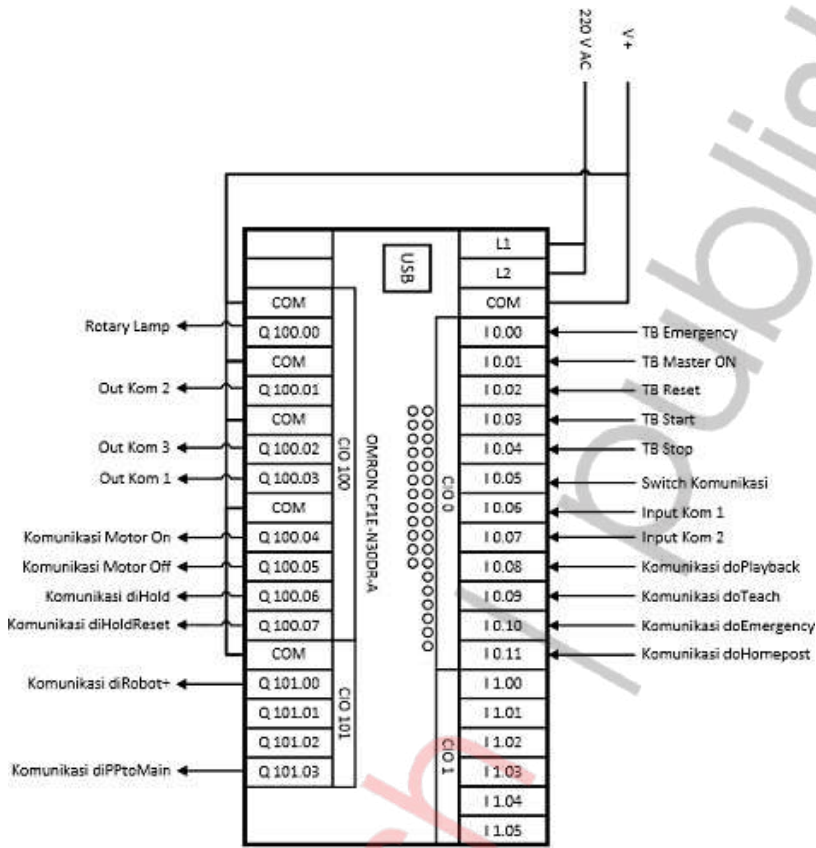
CP1E N30DRA

Input		Output	
Nama	Alamat	Nama	Alamat
Tombol Emergency	0.00	Output Komunikasi 1	100.03
Tombol Master On	0.01	Output Komunikasi 2	100.01
Tombol Reset	0.02	Output Komunikasi 3	100.02
Tombol Start	0.03	Rotary Lamp	100.00
Tombol Stop	0.04	Komunikasi diMotorOn	100.04
Switch Komunikasi	0.05	Komunikasi diMotorOff	100.05
Input Komunikasi 1	0.06	Komunikasi diHold	100.06
Input Komunikasi 2	0.07	Komunikasi diHoldReset	100.07
Input Komunikasi 3	1.01	Komunikasi diPptoMain	101.03
Komunikasi doPlayback	0.08	Komunikasi diRobot+	101.00
Komunikasi doTeach	0.09		
Komunikasi doEmergency	0.10		
Komunikasi doHomepost	0.11		

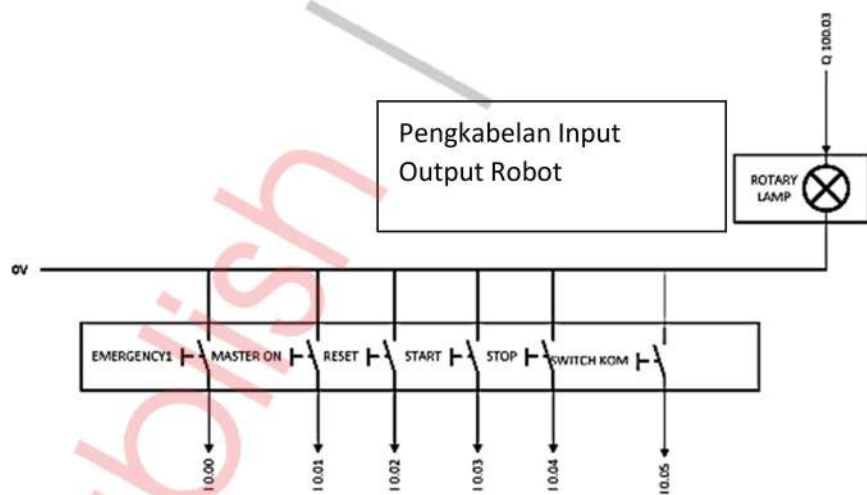
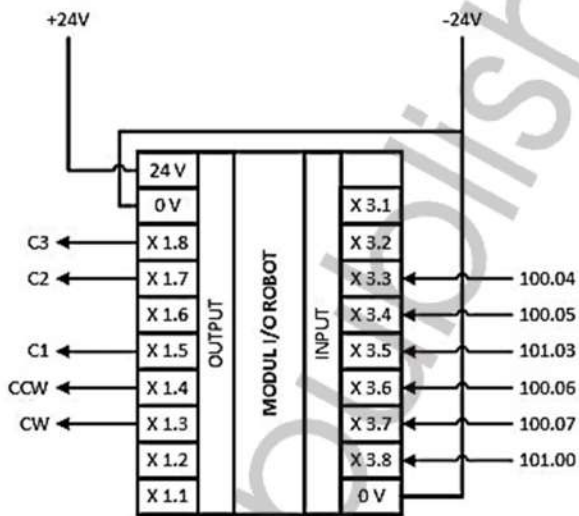
Input		Output	
Nama	Alamat	Nama	Alamat
Komunikasi diMotorOn	X3.3	Komunikasi doPlayback	X1.5
Komunikasi diMotorOff	X3.4	Komunikasi doEmergency	X1.7
Komunikasi diHold	X3.6	Komunikasi doHomepost	X1.8
Komunikasi diHoldReset	X3.7	Grip Open	X1.3
Komunikasi diPptoMain	X3.5	Grip Close	X1.4
Komunikasi Robot+	X3.8		

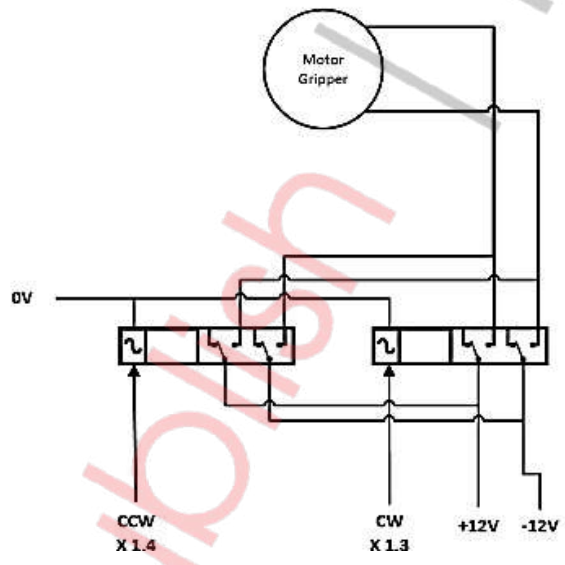


Pengkabelan Input Output PLC OMRON CP1E-N 40S1DR-A



Pengkabelan *Input Output* PLC OMRON CP1E-N 30DR-A





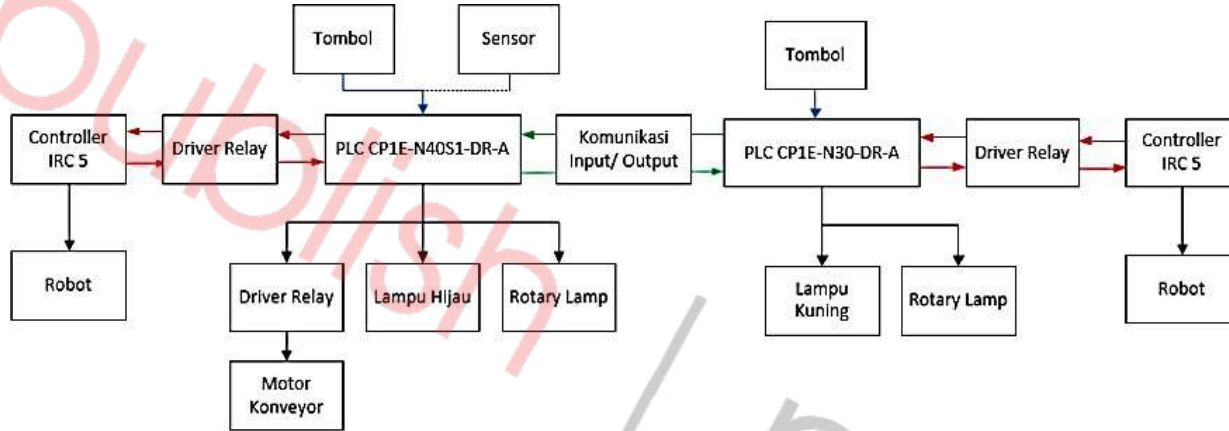
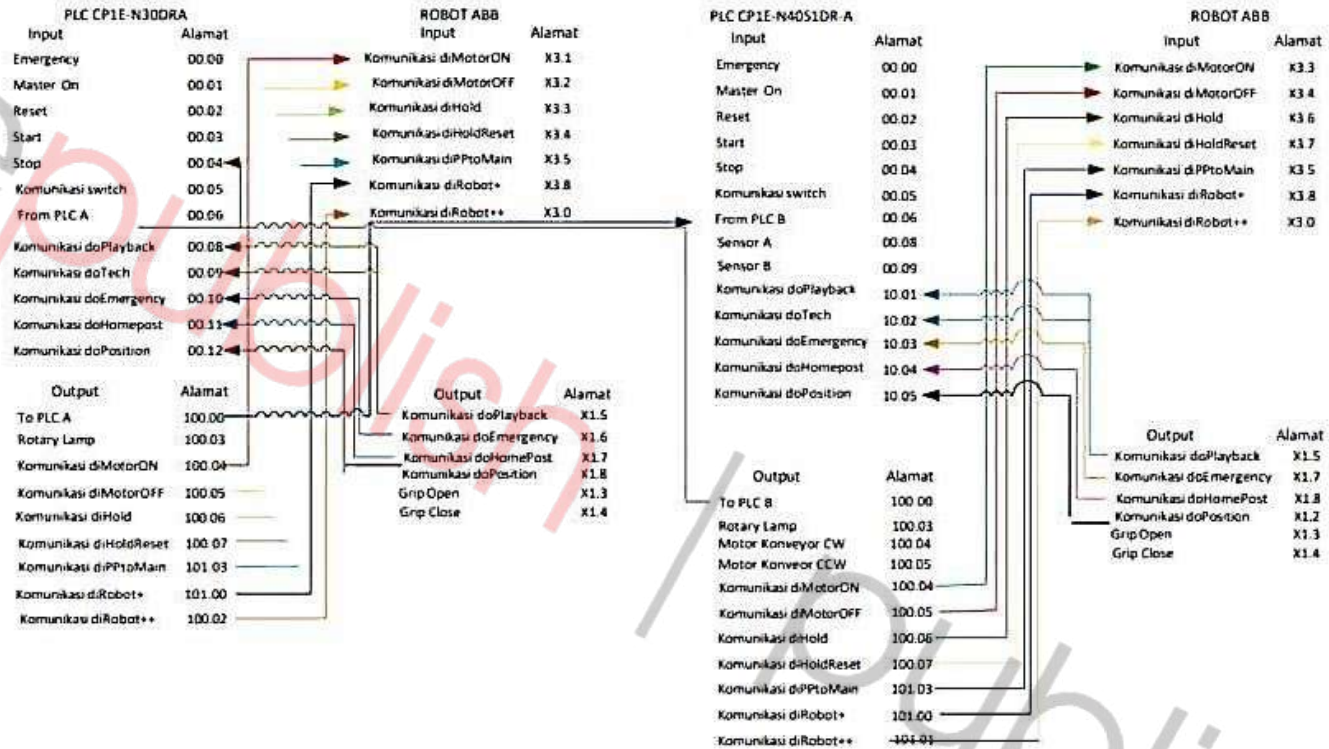


Diagram Blok Sistem Komunikasi Dua Robot ABB IRB1410

I/O PLC & ROBOT



Pengkabelan Komunikasi *Input Output* PLC & Robot

Diagram *Ladder Soal 4*

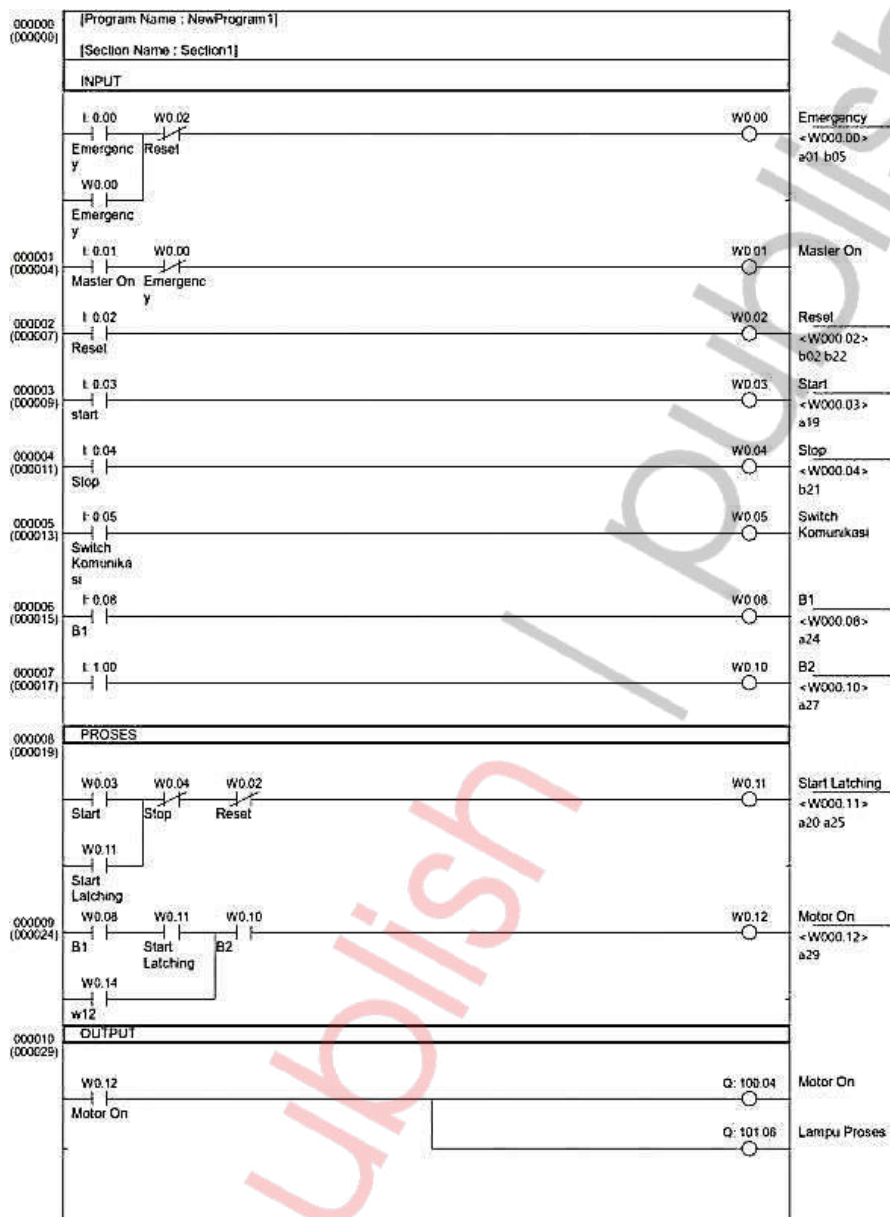
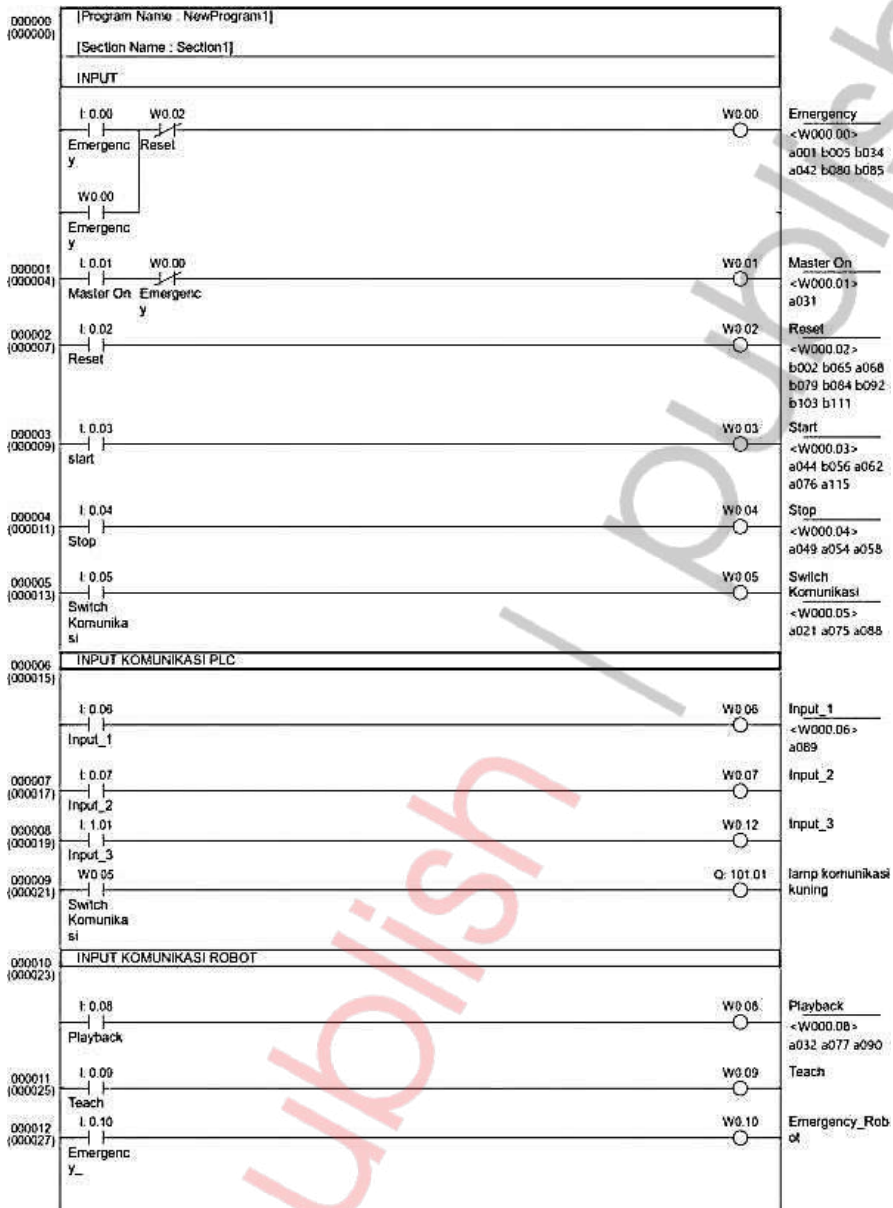
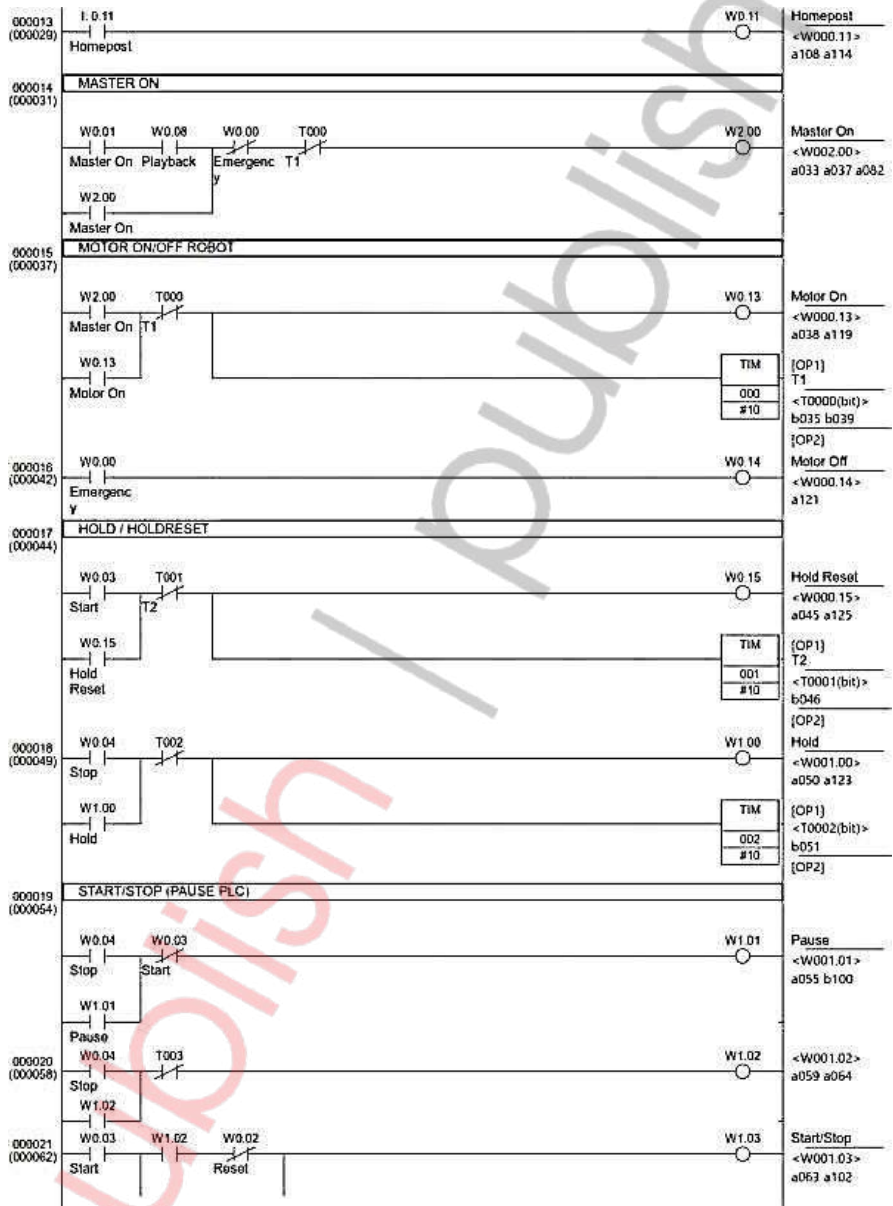
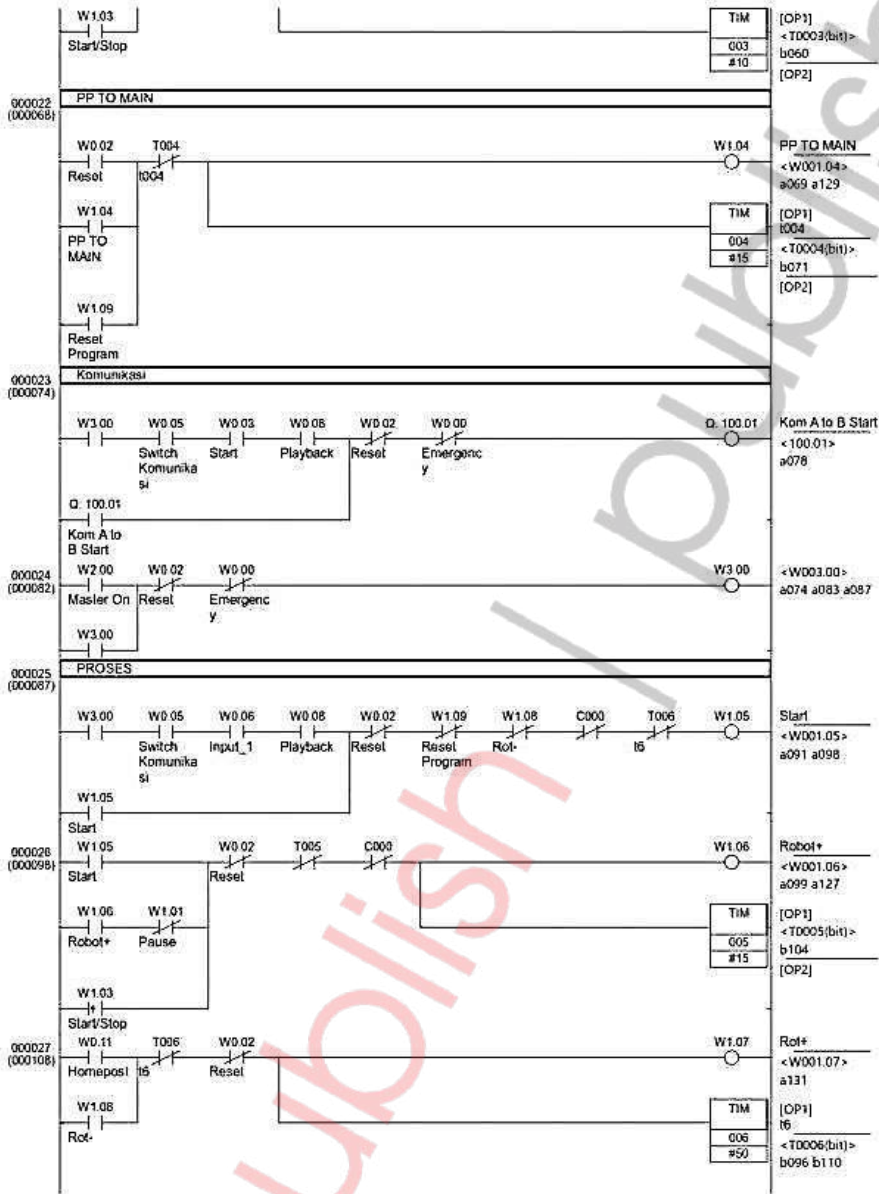
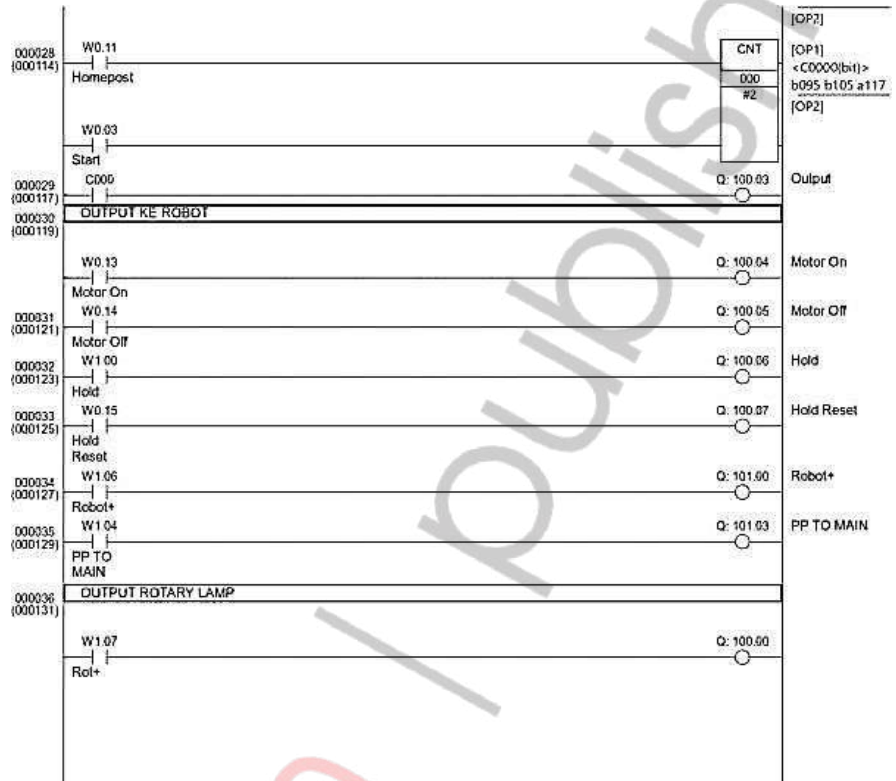


Diagram *Ladder Soal 5*

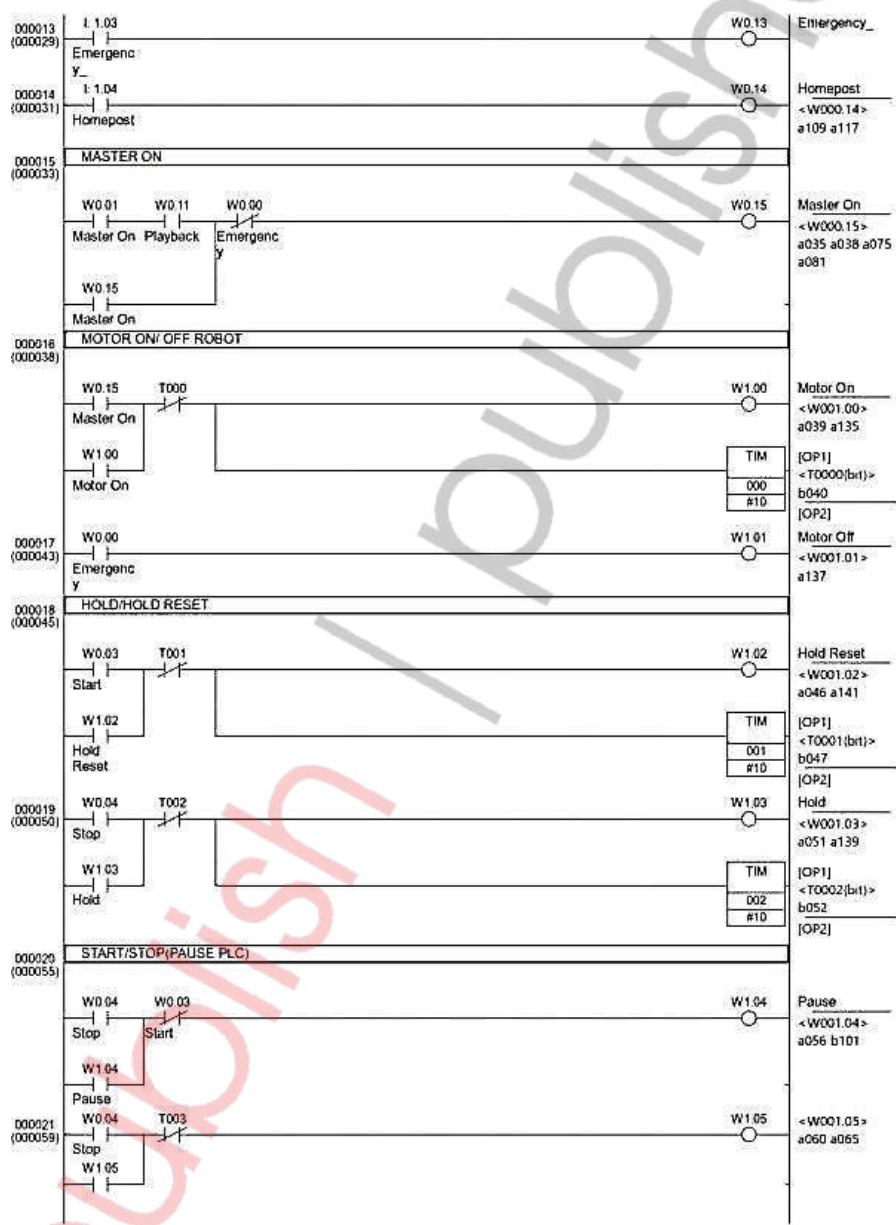




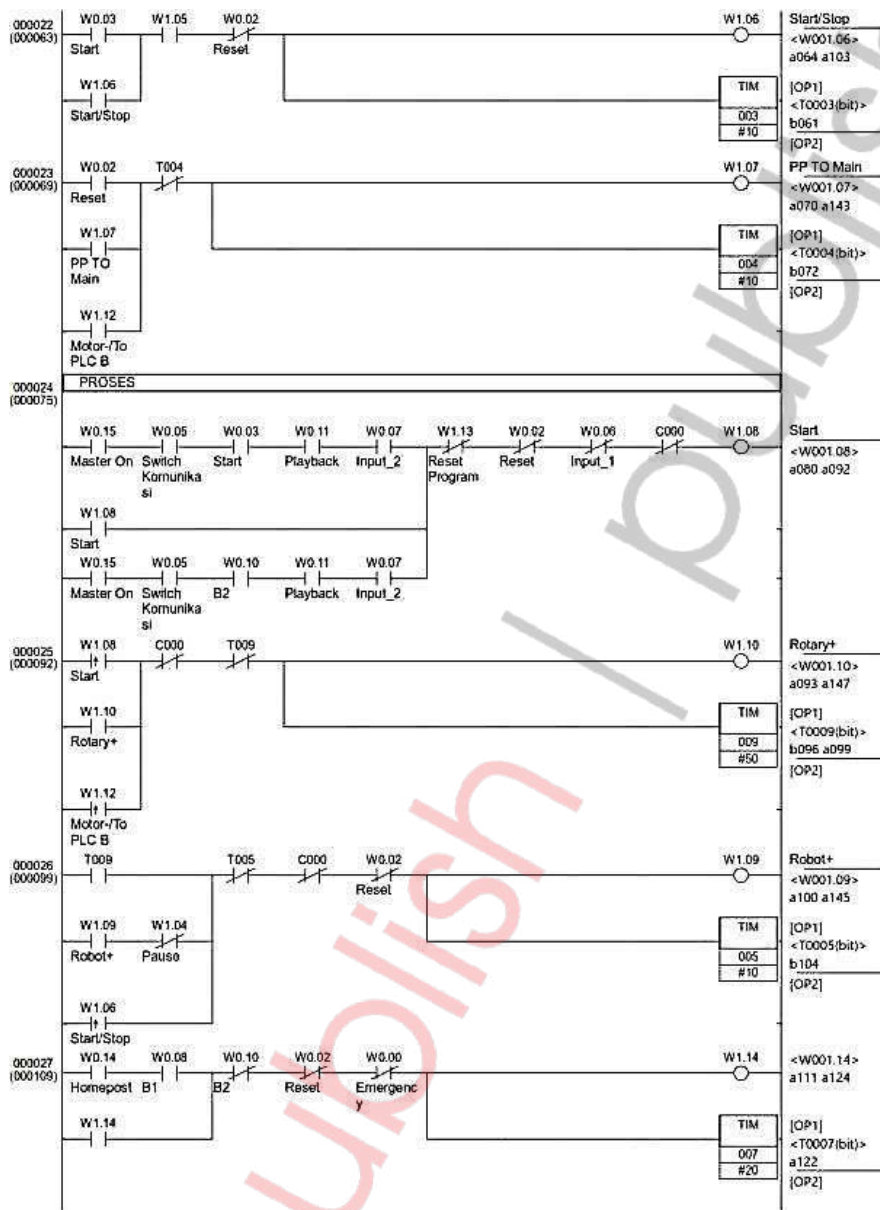


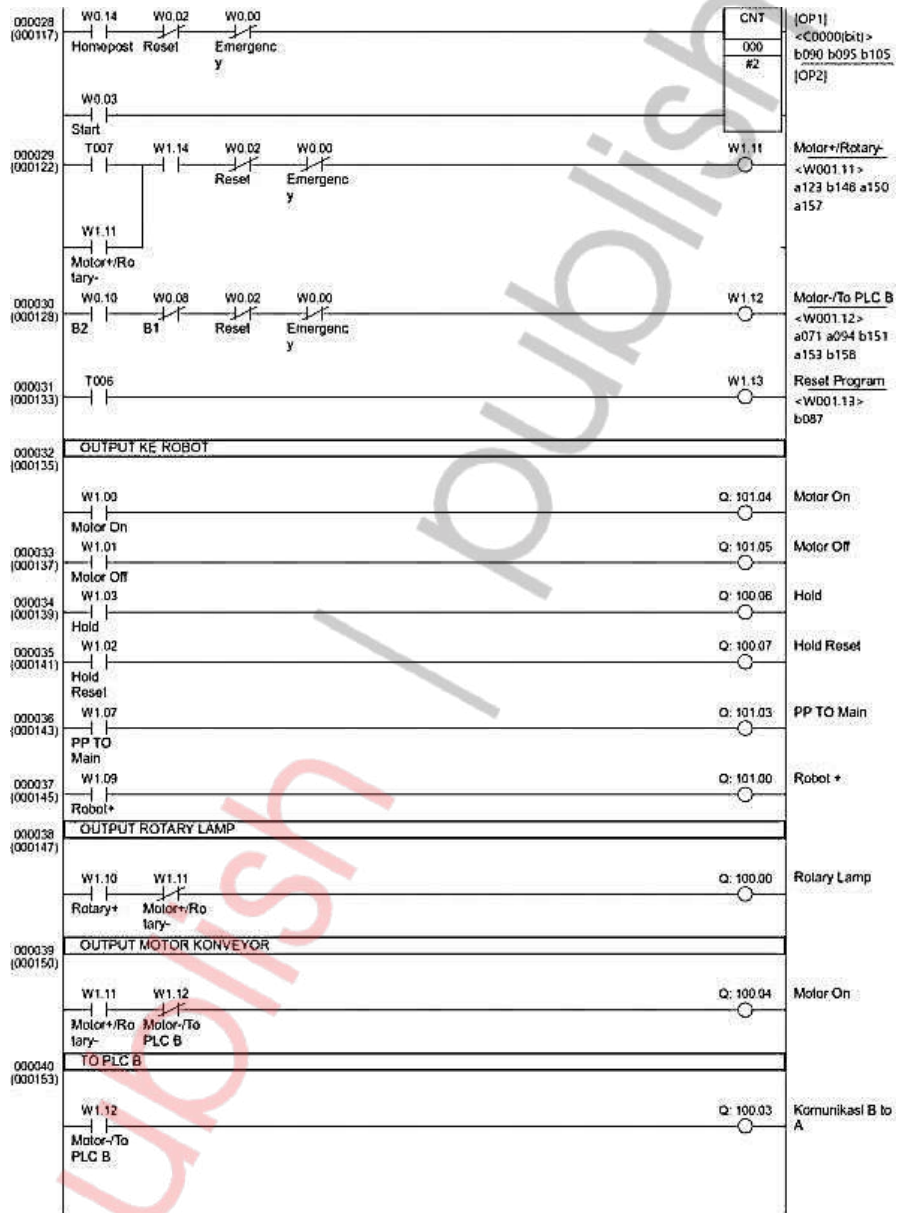


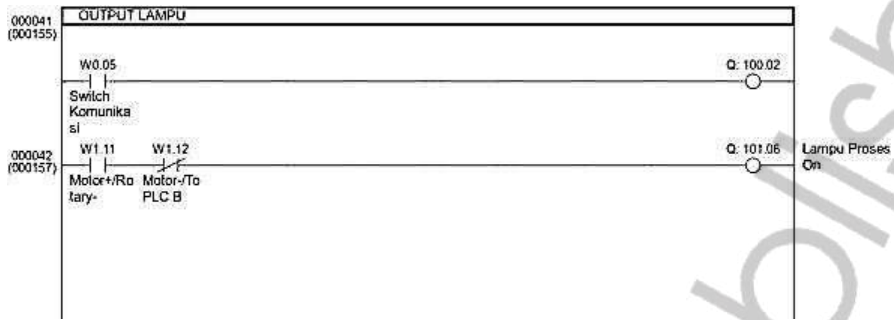
[Program Name : NewProgram1]			
[Section Name : Section1]			
INPUT			
000000 (000000)	I 0.00 Emergency y W0.02 Reset W0.00 Emergency y	W0.00	Emergency <W000.00> a001 b005 b036 a043 b114 b119 b126 b131
000001 (000004)	I 0.01 Master On W0.00 Emergency y	W0.01	Master On <W000.01> a033
000002 (000007)	I 0.02 Reset	W0.02	Reset <W000.02> b002 b066 a069 b088 b106 b113 b118 b125 b130
000003 (000009)	I 0.03 start	W0.03	Start <W000.03> a045 b057 a063 a077 a120
000004 (000011)	I 0.04 Stop	W0.04	Stop <W000.04> a050 a055 a059
000005 (000013)	I 0.05 Switch Komunika si	W0.05	Switch Komunikasi <W000.05> a076 a082 a155
000006 (000015)	I 0.08 B1	W0.08	B1 <W000.08> a110 b129
000007 (000017)	I 1.00	W0.10	B2 <W000.10> a083 b112 a128
INPUT KOMUNIKASI PLC			
000008 (000019)	I 0.06 Input_1	W0.06	Input_1 <W000.06> b089
000009 (000021)	I 0.07 Input_2	W0.07	Input_2 <W000.07> a079 a085
000010 (000023)	I 0.09	W0.09	
INPUT KOMUNIKASI ROBOT			
000011 (000025)	I 1.01 Playback	W0.11	Playback <W000.11> a034 a078 a084
000012 (000027)	I 1.02 Teach	W0.12	Teach



deepublish / publisher







REFERENSI

- [1]. ABB Robotics. *Operating manual–IRC5 with FlexPendant*. ABB AB, Robotics–Sweden, 2018.
- [2]. Division Robotics. *ABB Robotic Programing and Maintenance Training for PMSD (Politeknik Mekatronika Sanata Dharma)*. 2018.
- [3]. ABB Robotics. *Application Manual–Controller Software IRC5*. ABB AB, Robotics–Sweden, 2016.
- [4]. ABB Robotics. *Operating manual Getting started, IRC5 and RobotStudio Online*. ABB AB, Robotics–Sweden, 2007.