

## BAB III

### PELAKSANAAN KERJA MAGANG

#### 3.1 Kedudukan dan Koordinasi

Kedudukan mahasiswa magang adalah sebagai teknisi *engineering* dibawah bimbingan PM (*Preventive Maintenance*) PT LG Electronics Indonesia Tangerang, bagian pabrik lemari es satu. Mahasiswa bertanggung jawab pada PM dalam membantu kegiatan *maintenance* seperti memperbaiki dan membersihkan mesin pada *production line* kemudian menuliskannya sebagai laporan *daily task*. Kegiatan sehari-hari yang dituliskan dalam laporan *daily task*, serta evaluasi kegiatan mahasiswa akan dinilai oleh anggota staff *maintenance* sekaligus supervisor kegiatan magang Franky Sugiarto. Mahasiswa bertanggung jawab kepada supervisor untuk mengerjakan proyek yaitu mengoperasikan robot ABB IRB 6700 yang berada di area *training* perusahaan. Setiap mahasiswa magang ditugaskan membuat topik yang berbeda dari robot dan menuliskannya sebagai laporan magang.

#### 3.2 Tugas dan Uraian Kerja

Pengerjaan proyek merupakan tugas mandiri dari masing-masing mahasiswa magang di mana sebuah robot ABB IRB 6700 dibagi menjadi tiga topik. Topik pertama adalah robot yang dapat memindahkan tiga buah objek dengan bentuk yang berbeda-beda, topik kedua adalah robot yang dapat menuliskan seluruh alfabet menggunakan spidol pada sebuah papan tulis, dan topik ketiga adalah HMI yang dapat mengirim output dan menerima input robot tanpa memerlukan *hardware* dan *software* khusus, melainkan hanya menggunakan komputer biasa.

Topik yang dipilih penulis adalah robot *labelling* menggunakan spidol. Sebelum pengerjaan proyek dimulai, mahasiswa-mahasiswa magang diminta oleh supervisor untuk mempelajari secara mandiri mengenai *software* dan *hardware* yang digunakan dalam pengoperasian robot ABB. Dari pembelajaran mandiri diketahui bahwa *software* yang akan digunakan merupakan RobotStudio, dan *hardware* yang diperlukan adalah FlexPendant dan juga PLC. FlexPendant

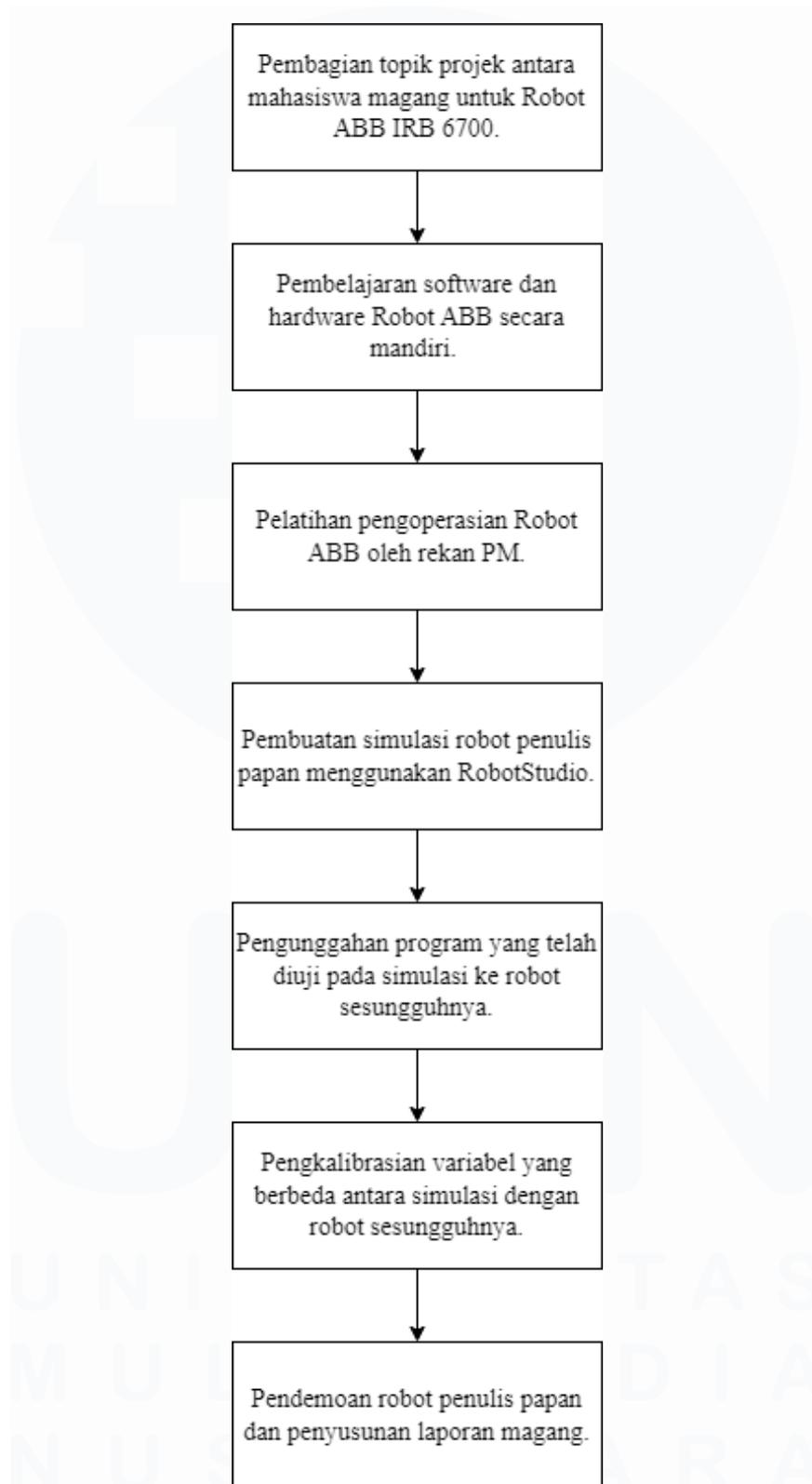
merupakan *hardware* yang digunakan untuk membuat program baru, menjalankan program lama yang tersimpan di robot, mengubah variabel-variabel robot, menggerakkan *joint* robot secara manual, dan lain-lain. PLC digunakan untuk menerima input dari HMI dan mengeluarkan output untuk robot.

Setelah pembelajaran mandiri selesai, dilakukan pelatihan pengoperasian robot oleh rekan PM. Penulis diajarkan cara menggerakkan robot secara manual menggunakan FlexPendant, cara membuat program baru beserta instruksi-intruksi apa saja yang biasanya digunakan, koneksi input dan output apa saja yang diperlukan untuk mengoperasikan *end-effector* robot, cara pemrograman PLC Mitsubishi dan koneksinya dengan robot, serta cara membuat demo robot pemindah barang sederhana sesuai dengan standar industri. Selama pelatihan, mahasiswa magang juga diajari untuk selalu menjaga keamanan seperti harusnya membatasi kecepatan maksimal robot, selalu memakai helm di area *training*, memastikan robot tidak bergerak keluar area *training*, dan bagaimana cara memprogram PLC dan robot dengan benar agar tidak terjadi *error*/kerusakan.

Setelah pelatihan, penulis memulai projek dengan pertama-tama membuat simulasi program di RobotStudio. Penulisan dan pengujian pada simulasi robot penulis dilakukan sampai robot dapat menjalankan semua fitur yang diinginkan yaitu menulis seluruf alfabet, robot dapat bergeser ke lokasi penulisan selanjutnya secara otomatis, robot dapat bergeser ke bawah ketika telah mencapai batas penulisan secara otomatis atau ketika pengguna menekan tombol “Enter”, serta robot dapat kembali ke posisi awal ketika telah mencapai batas bawah papan atau ketika pengguna menekan tombol “Done”.

Setelah simulasi berhasil dijalankan, kode yang telah dibuat akan ditambahkan ke robot sebenarnya. Kode dipindahkan ke robot menggunakan laptop kemudian dilihat apakah ada *error*. Setelah dipastikan tidak ada, kode robot yang dibuat untuk area simulasi dimodifikasi agar sesuai dengan area robot sebenarnya. Dibuat juga ladder diagram untuk PLC dan penambahan *address* koneksi IO untuk robot agar bisa saling berkomunikasi ketika dihubungkan dengan HMI. Setelah selesai robot bisa langsung didemokan.

Tugas dan uraian kerja dapat direpresentasikan sebagai sebuah *chart diagram*:



Gambar 3.1 *Chart Diagram* Tugas dan Uraian Kerja

### 3.3 Kendala yang Ditemukan

Terdapat dua masalah yang ditemukan selama pengerjaan proyek berlangsung. Saat robot pertama kali dinyalakan dari kondisi mati yang cukup lama, layar dari FlexPendant akan *freeze* sehingga pengerjaan proyek tidak bisa dilakukan. Penyelesaian dari masalah ini adalah untuk menunggu robot menyala selama beberapa saat kemudian mematikan dan menyalakan kembali robot. Penyebab dari masalah ini tidak dapat ditemukan oleh anggota *staff maintenance*.

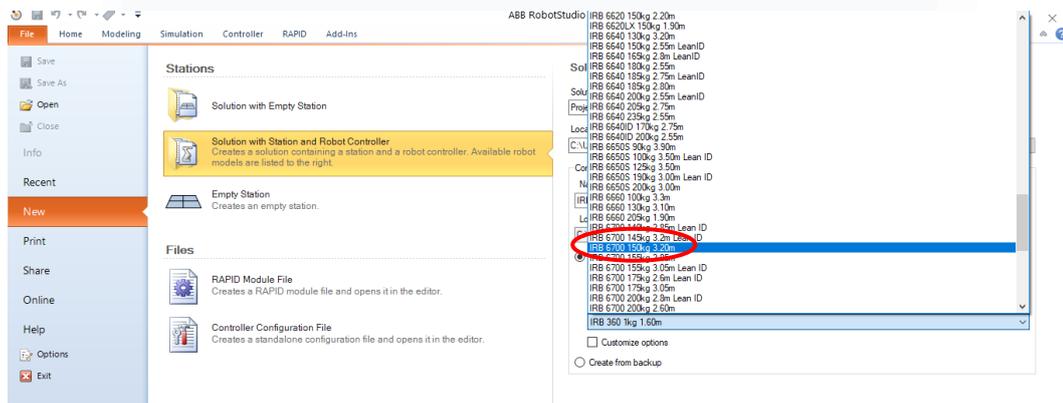
Lokasi dari papan tulis dapat menyebabkan masalah bila tidak tepat. *End-effector* robot bergerak menuju posisi dan orientasi yang spesifik sehingga lokasi papan yang tidak tepat dapat menyebabkan robot menekan spidol terlalu kencang atau sebaliknya yaitu spidol tidak mencapai papan. Penyelesaian dari masalah ini adalah menggunakan *tape* pada lantai sebagai penanda lokasi papan yang seharusnya. Bila bergeser, papan akan bisa dikembalikan ke posisi sebelumnya.

### 3.4 Solusi untuk Masalah yang Dikemukakan

Untuk mengatasi keterbatasan proses *labelling* konvensional, dapat digunakan *robotic arm* yang dilengkapi dengan laser, *drill*, *pen*, atau *end-effector* lainnya sesuai dengan material yang diproses. *Robotic arm* dapat diprogram untuk menulis atau menggambar pada produk dengan ketelitian tinggi sehingga tidak akan ada perbedaan antar produk.

*Robotic arm* memiliki *workspace* yang lebih fleksibel bila dibandingkan dengan mesin lainnya seperti contohnya *stamping machine* sehingga dapat dipakai untuk memproses *work object* dengan bentuk yang lebih kompleks. *Robotic arm* juga lebih adaptif bila terjadi perubahan desain label, misalnya operator hanya perlu memilih jenis label yang diinginkan pada HMI berdasarkan daftar label yang telah dibuat sebelumnya. Sedangkan *stamping machine* memerlukan operator untuk melakukan perubahan *hardware* agar bisa menyesuaikan dengan label baru. Untuk menunjukkan kinerja dari *robotic arm* sebagai otomasi yang dapat mempercepat proses produksi, dibuatlah robot penulis papan tulis yang akan bekerja sebagai sebuah robot *labelling*. Robot yang akan digunakan merupakan sebuah robot ABB IRB 6700 di PT LG Electronics Indonesia Tangerang.

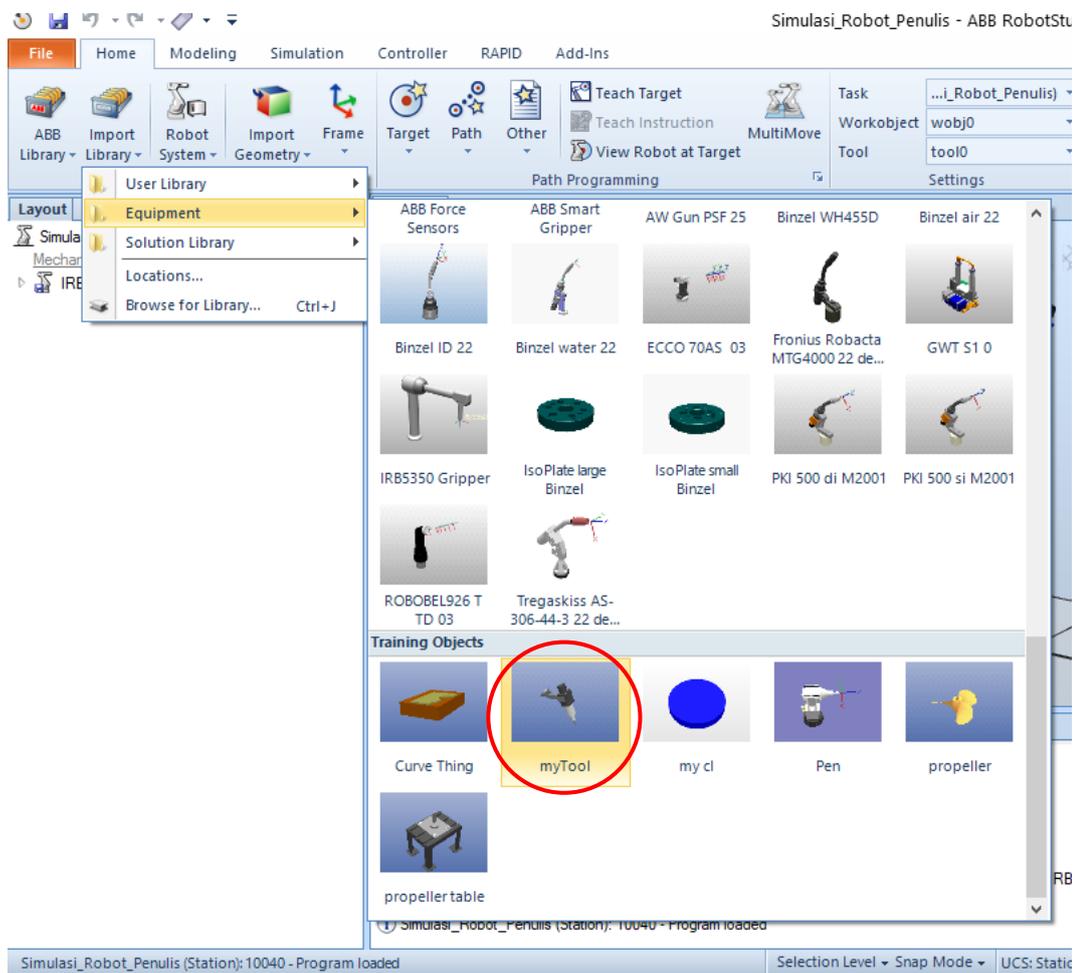
Untuk membuat robot penulis papan, penulis pertama membuat simulasi program di RobotStudio. Pada saat program pertama kali dibuka, penulis memilih jenis robot yang tersedia pada *area training* yaitu ABB IRB 6700 kemudian memilih *end-effector/work tool* tipe *pointer* bernama “myTool”. Simulasi robot memiliki *virtual controller* yang kerjanya sama seperti *software* robot ABB IRB 6700 sebenarnya, artinya robot simulasi dapat diprogram untuk bergerak, mengingat posisi, dan menerima input virtual. *End-effector* simulasi yang dipilih berjenis *pointer* karena tidak terdapat *end-effector* spidol seperti robot *training* sesungguhnya pada RobotStudio. *End-effector* tipe *pointer* dapat digunakan untuk mensimulasikan spidol karena bentuknya yang serupa.



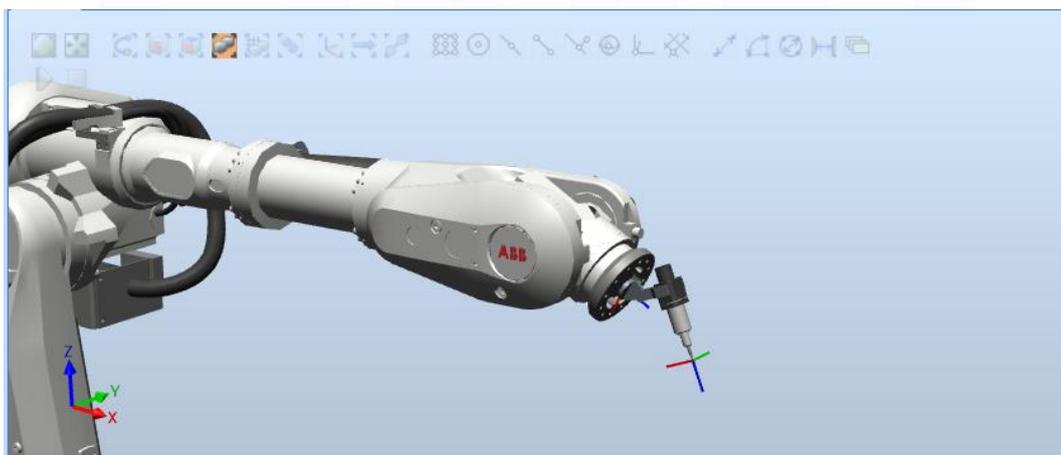
Gambar 3.2 Pemilihan Jenis Robot pada Awal Membuka RobotStudio



Gambar 3.3 Tampilan RobotStudio

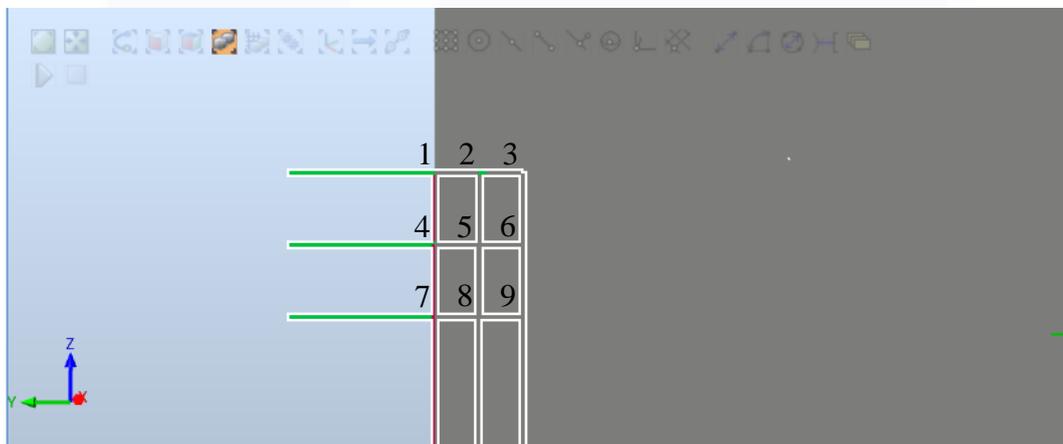


Gambar 3.4 Pemilihan *End-effector* Tipe *Pointer*



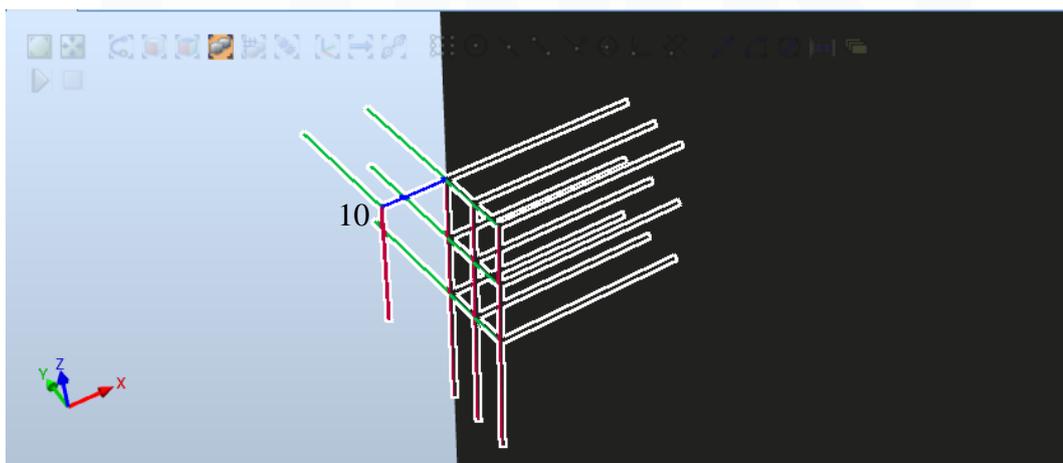
Gambar 3.5 Tampilan *End-Effector* Robot

Untuk menggerakkan robot, penulis perlu menentukan terlebih dahulu titik-titik posisi di mana robot dapat berhenti. Titik-titik ini berbentuk data posisi dan orientasi dari *end-effector* dan digunakan untuk menentukan jalur bergerak spidol pada papan. Penulis pertama membuat sembilan buah titik dengan posisi berjajar tiga ke bawah dan tiga ke samping. Titik satu berlokasi di kiri atas, titik dua berlokasi di tengah atas, titik tiga berlokasi di kanan atas, titik empat berlokasi di kiri tengah, titik lima berlokasi di tengah, titik enam berlokasi di kanan tengah, titik tujuh berlokasi di kiri bawah, titik delapan berlokasi di tengah bawah, dan titik sembilan berlokasi di kanan bawah sehingga berbentuk seperti:



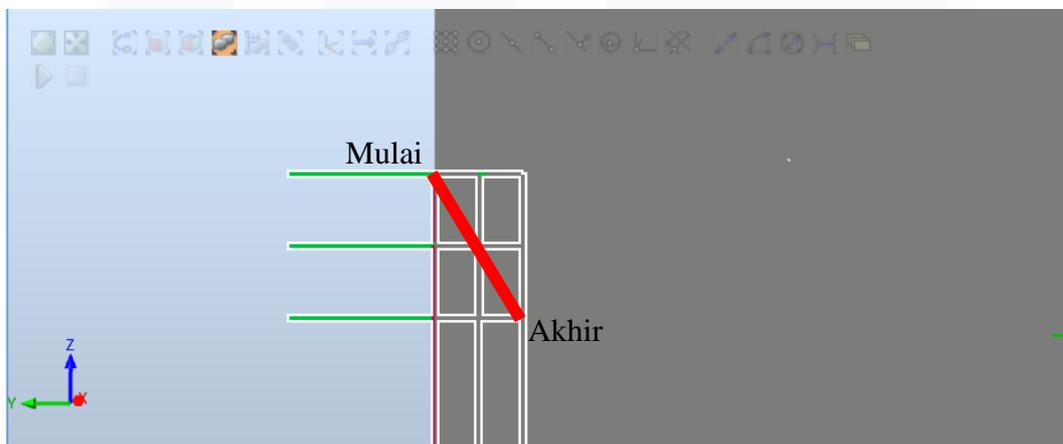
Gambar 3.6 Sembilan Titik Posisi Utama Robot Pada Sebuah Papan Virtual

Selain sembilan titik tersebut, juga terdapat satu titik posisi tambahan yaitu titik sepuluh yang berada tepat di atas titik satu.



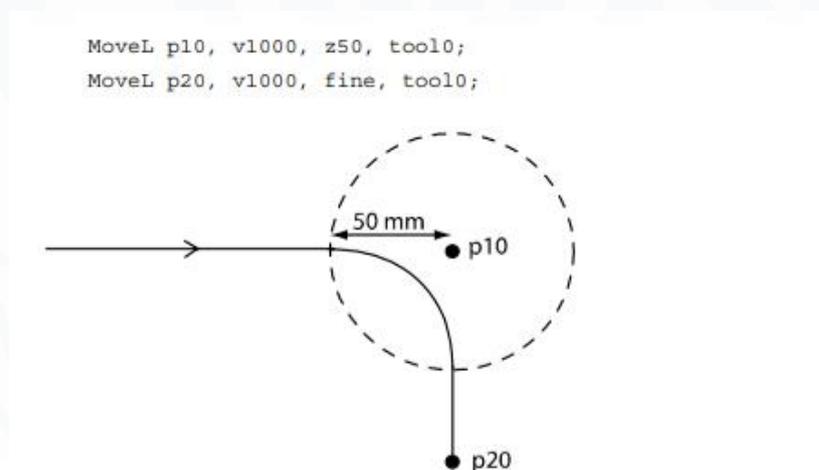
Gambar 3.7 Titik Posisi Sepuluh

Semakin besar jarak antar titik akan semakin besar huruf yang tertulis. Jarak antara titik ditentukan sendiri oleh penulis untuk menghasilkan tulisan yang tidak terlalu besar dan tidak terlalu kecil di papan. Dari sembilan titik posisi utama ini, robot kemudian akan diprogram untuk bergerak dari satu titik ke titik lain sehingga membentuk sebuah jalur. Bila misalnya robot bermula dari titik satu lalu diprogram untuk bergerak ke titik sembilan, robot akan membentuk sebuah garis diagonal dari kiri atas ke kanan bawah. Titik sepuluh akan digunakan ketika robot perlu mengangkat spidol dari papan.



Gambar 3.8 Garis Spidol dari Titik Satu ke Titik Sembilan

Jadi untuk menuliskan berbagai huruf, perbedaannya terdapat pada titik mulai dan jalur yang diambil robot. Robot juga dapat diprogram untuk berbelok sebelum mencapai titik, menghasilkan jalur yang melengkung. Ini bisa digunakan untuk menuliskan huruf seperti O, B, P, R, dan seterusnya.



Gambar 3.9 Garis Spidol Bila Berbelok 50 mm Sebelum Titik

Instruksi jalur pergerakan robot untuk setiap huruf dapat disimpan sebagai sebuah *function* (atau disebut sebagai *routine* dalam RobotStudio) yang berbeda-beda. Masing-masing *routine* berisi instruksi robot untuk bergerak ke titik posisi awal, instruksi robot untuk bergerak ke titik-titik posisi selanjutnya, lalu instruksi robot untuk kembali ke titik posisi awal. Jadi ketika *routine* dipanggil, robot akan bergerak mengikuti jalur yang telah terprogram. Penulis perlu membuat *routine* untuk setiap huruf pada alfabet dan juga beberapa karakter tambahan seperti “spasi” dan “simbol garis (-)”. Simbol garis dibuat karena penulis ingin membuat robot dapat menuliskan kalimat seperti “Kevin – UMN” pada papan.



Gambar 3.10 Contoh Jalur *End-effector* Robot untuk *Routine* B

```

2063 PROC Kevin_B()
2064     MoveJ Target_100, v100, fine, tool0;
2065     MoveL Target_10, v100, fine, tool0;
2066     MoveL Target_30, v100, z100, tool0;
2067     MoveL Target_60, v100, z100, tool0;
2068     MoveL Target_40, v100, fine, tool0;
2069     MoveL Target_60, v100, z100, tool0;
2070     MoveL Target_90, v100, z100, tool0;
2071     MoveL Target_70, v100, fine, tool0;
2072     MoveL Target_10, v100, fine, tool0;
2073     MoveL Target_100, v100, fine, tool0;
2074 ENDPROC

```

Gambar 3.11 Cuplikan Kode untuk *Routine* B

Untuk memanggil *routine-routine* yang telah dibuat, diperlukan program utama pada robot, di mana ketika ada input, *routine* tertentu akan dipanggil. Input yang digunakan untuk memanggil *routine* merupakan output dari PLC Mitsubishi, tapi untuk simulasi awal ini input dapat diaktifkan secara *software*. Selain input, robot juga ingin dibuat untuk mengeluarkan sebuah sinyal output ketika sedang bergerak, sinyal output ini digunakan untuk menyalakan sebuah LED pada HMI yang

menandakan robot sedang bergerak. Semua pengkodean robot dilakukan dalam RAPID yang merupakan bahasa pemrograman RobotStudio. Setelah program RAPID untuk membaca input dan mengeluarkan output dibuat, dilakukan simulasi.

```

1730 IF CI_14 = 1 THEN
1731   Set CO_11;
1732   Kevin_A;
1733   Reset CO_11;
1734   next := TRUE;
1735   counterChar := counterChar + 1;
1736 ELSEIF CI_15 = 1 THEN
1737   Set CO_11;
1738   Kevin_B;
1739   Reset CO_11;
1740   next := TRUE;
1741   counterChar := counterChar + 1;

```

Gambar 3.12 Cuplikan Kode Utama untuk Melihat Apakah Ada Input A atau B

CI merupakan sinyal input robot dari PLC, dan CO merupakan sinyal output robot menuju PLC. CI\_14 digunakan untuk membaca input dari *keyboard* huruf A pada HMI, CI\_15 untuk *keyboard* huruf B, CI\_16 untuk *keyboard* huruf C, dan seterusnya. Dari cuplikan kode, terlihat bahwa robot akan mengeluarkan output CO\_11 untuk menyalakan LED HMI ketika ada input dari *keyboard* apapun. Robot kemudian memanggil *routine* yang dinamakan Kevin\_(huruf), untuk menggerakkan robot membentuk huruf yang dipilih pada papan. Setelah selesai menulis, output CO\_11 robot akan mati, sehingga LED penanda “Robot Sedang Bergerak” juga akan mati pada HMI.



Gambar 3.13 PLC Mitsubishi Q06H yang Digunakan

Karena kesepuluh titik posisi robot tidak bergerak, robot hanya akan menulis di satu tempat dan menimpa huruf sebelumnya. Oleh sebab itu ada kode berupa *offset* titik posisi setiap kali robot telah selesai menulis sebuah huruf. Kesepuluh titik posisi robot akan bergeser ke kanan dengan jarak yang telah ditentukan penulis secara otomatis. Ditambahkan juga *counter* yang menghitung sudah seberapa banyak titik posisi bergeser. Karena batas ukuran papan tulis, dibuat jumlah maksimal karakter yang dapat ditulis adalah 14.

Ketika *counter* karakter telah mencapai 14 atau pengguna menekan tombol “Enter”, titik-titik posisi akan di-*offset* kembali ke kiri sampai lokasi awal kemudian akan di-*offset* ke bawah agar robot mulai menulis di baris selanjutnya. *Offset* ke bawah juga memiliki *counter* yang berbeda dan dibatasi menjadi 5 baris. Bila *counter* baris telah mencapai batas, titik-titik posisi akan di-*offset* ke atas sampai kembali ke baris pertama dan juga di-*offset* kembali ke kiri bila robot belum berada di lokasi awal. Selain kembali ke awal ketika telah mencapai batas *counter*, terdapat juga tombol “Done” yang berfungsi untuk mengembalikan robot ke posisi awal di mana pun robot sedang berada.

Setelah keseluruhan sistem diuji, program dipindahkan ke sistem robot yang sebenarnya. Robot ABB IRB 6700 di area *training* dihubungkan dengan laptop, kemudian program yang telah dibuat ditambahkan ke program yang sudah ada sebelumnya pada robot. Ini bertujuan untuk mencegah terjadinya *error* dari penggantian program total, sekaligus tidak menghilangkan program-program yang dibuat sebelumnya. Setelah program ditambahkan dan dipastikan tidak ada *error*, titik-titik posisi simulasi diubah nilai posisi dan orientasinya menjadi nilai posisi dan orientasi baru karena perlu menyesuaikan dengan lokasi papan tulis di area *training*.

Titik posisi baru ini kemudian dipastikan telah benar dan tidak menabrak apa-apa dengan menggerakkan robot ke setiap titik menggunakan perintah *jogging*. Untuk membuat robot dapat menerima input dan mengeluarkan output, pertama ditambahkan *address* input untuk setiap *button* yang akan ada pada HMI (CI\_14 untuk *button* A, CI\_15 untuk *button* B, CI\_16 untuk *button* C, dan seterusnya) dan

juga tambahkan *address* output (CO\_11 untuk LED) pada RobotStudio. Alamat input dan output pada RobotStudio kemudian dicocokkan dengan alamat input dan output yang ada di PLC. *Address* input PLC dari HMI ditentukan sendiri mulai dari M1000, sedangkan *address* output PLC menyesuaikan dengan input robot yang masih kosong melalui CC-Link.

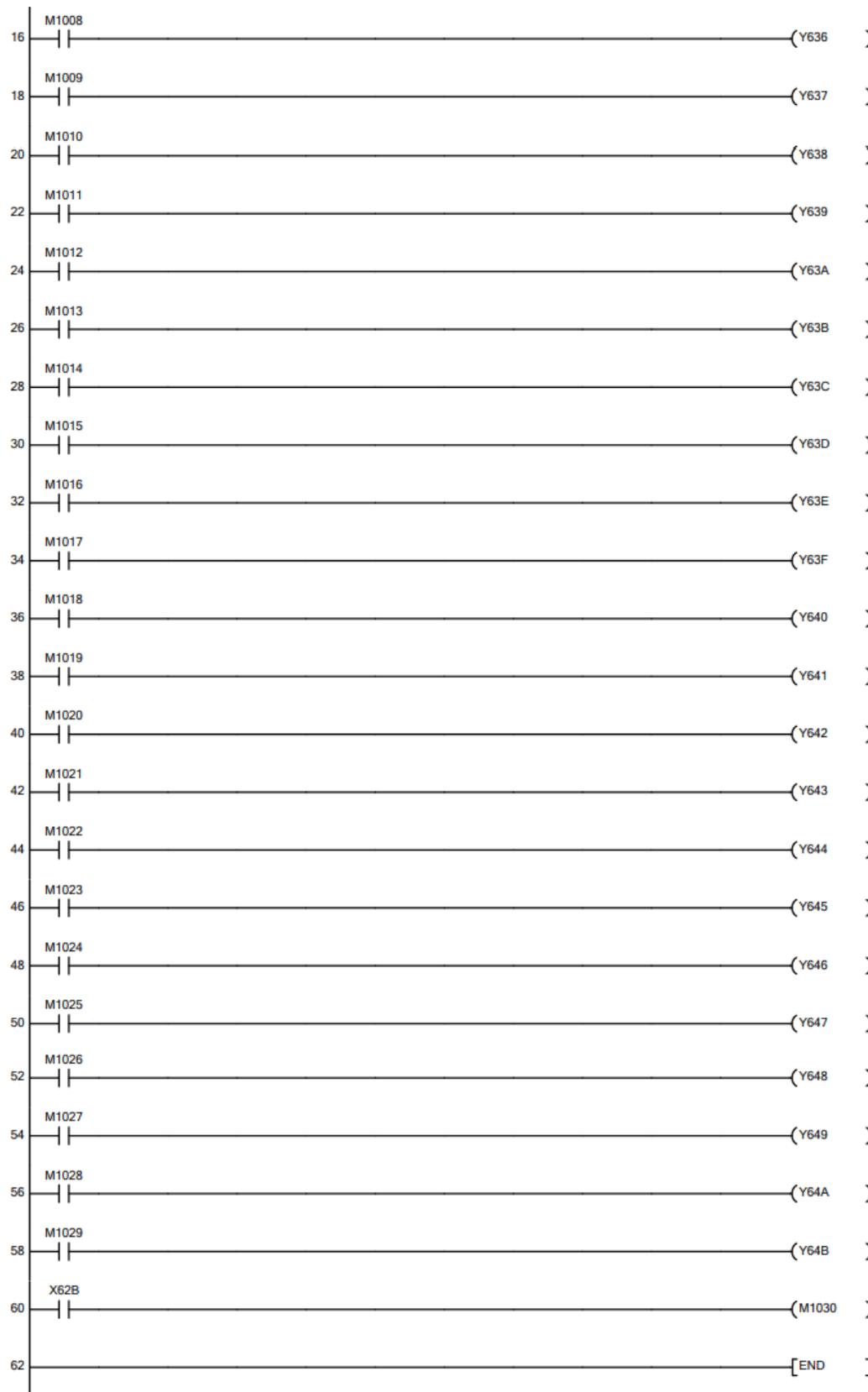
Type	Name	Type of Signal	Assigned to Device	Signal Identification Label	Device Mapping	Category	Access Level	Default Value	Fi
Access Level	CI_14	Digital Input	d378B	huruf A	14	cc_link	Default	0	0
Cross Connection	CI_15	Digital Input	d378B	Huruf B	15	cc_link	Default	0	0
Device Trust Level	CI_16	Digital Input	d378B	Huruf C	16	cc_link	Default	0	0
DeviceNet Command	CI_17	Digital Input	d378B	Huruf D	17	cc_link	Default	0	0
DeviceNet Device	CI_18	Digital Input	d378B	Huruf E	18	cc_link	Default	0	0
DeviceNet Internal Device	CI_19	Digital Input	d378B	Huruf F	19	cclink	Default	0	0
EtherNet/IP Command	CI_20	Digital Input	d378B	Huruf G	20	cclink	Default	0	0
EtherNet/IP Device	CI_21	Digital Input	d378B	Huruf H	21	cclink	Default	0	0
Industrial Network	CI_22	Digital Input	d378B	Huruf I	22	cclink	Default	0	0
Route	CI_23	Digital Input	d378B	Huruf J	23	cclink	Default	0	0
Signal	CI_24	Digital Input	d378B	Huruf K	24	cclink	Default	0	0
Signal Safe Level	CI_25	Digital Input	d378B	Huruf L	25	cclink	Default	0	0
System Input	CI_26	Digital Input	d378B	Huruf M	26	cclink	Default	0	0
System Output	CI_27	Digital Input	d378B	Huruf N	27	cclink	Default	0	0
	CI_28	Digital Input	d378B	Huruf O	28	cclink	Default	0	0
	CI_29	Digital Input	d378B	Huruf P	29	cclink	Default	0	0
	CI_30	Digital Input	d378B	Huruf Q	30	cclink	Default	0	0
	CI_31	Digital Input	d378B	Huruf R	31	cclink	Default	0	0
	CI_32	Digital Input	d378B	Huruf S	32	cclink	Default	0	0
	CI_33	Digital Input	d378B	Huruf T	33	cclink	Default	0	0
	CI_34	Digital Input	d378B	Huruf U	34	cclink	Default	0	0
	CI_35	Digital Input	d378B	Huruf V	35	cclink	Default	0	0
	CI_36	Digital Input	d378B	Huruf W	36	cclink	Default	0	0
	CI_37	Digital Input	d378B	Huruf X	37	cclink	Default	0	0
	CI_38	Digital Input	d378B	Huruf Y	38	cclink	Default	0	0
	CI_39	Digital Input	d378B	Huruf Z	39	cclink	Default	0	0
	CI_40	Digital Input	d378B	Spasi	40	cclink	Default	0	0
	CI_41	Digital Input	d378B	Dash	41	cclink	Default	0	0

Gambar 3.14. Cuplikan *Address* Input dan Output yang Dipakai Sistem

Lalu dibuat *ladder diagram* yang menerima input tombol dari HMI dan mengeluarkan output untuk robot, serta menerima input dari robot dan mengeluarkan output LED untuk HMI.

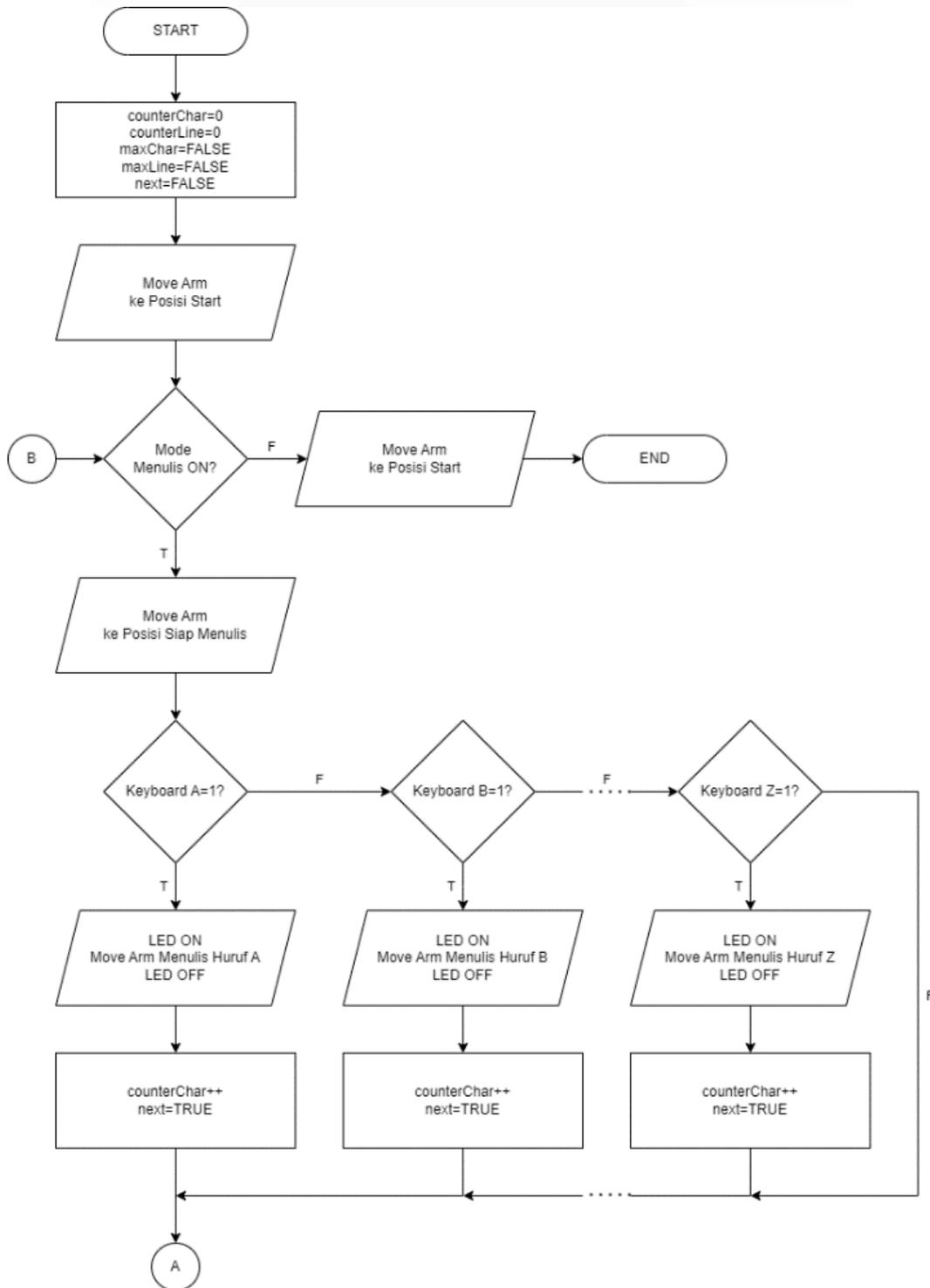


Gambar 3.15. *Ladder Diagram* Robot Penulis (Bagian 1)

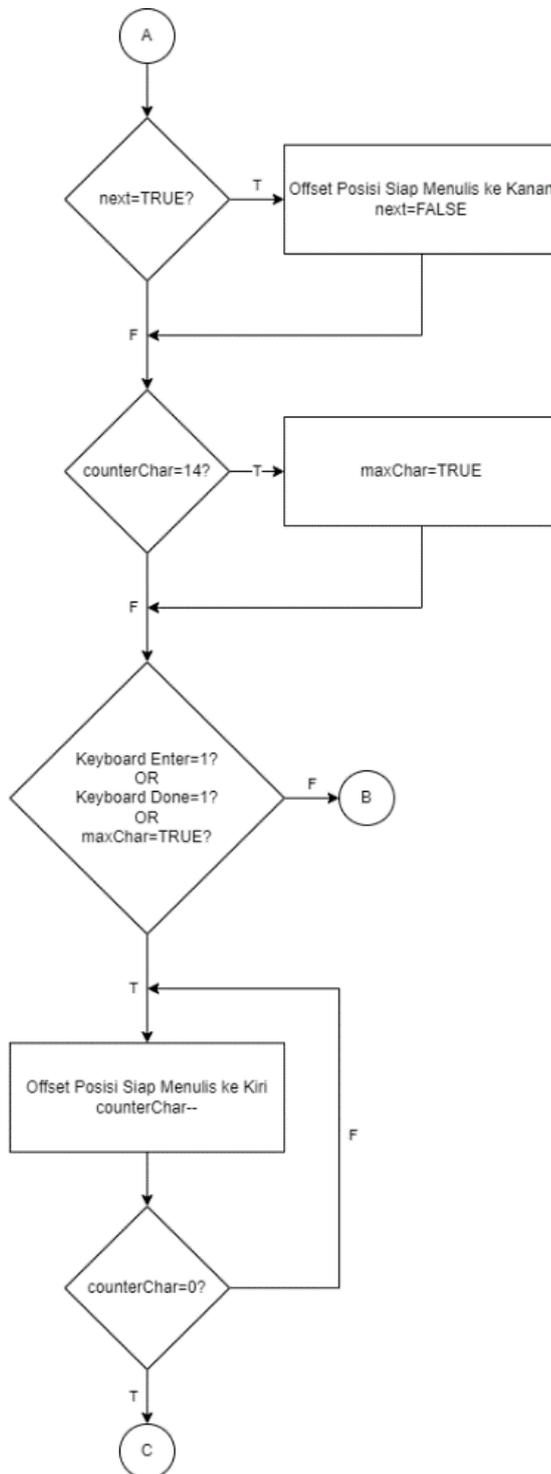


Gambar 3.16 Ladder Diagram Robot Penulis (Bagian 2)

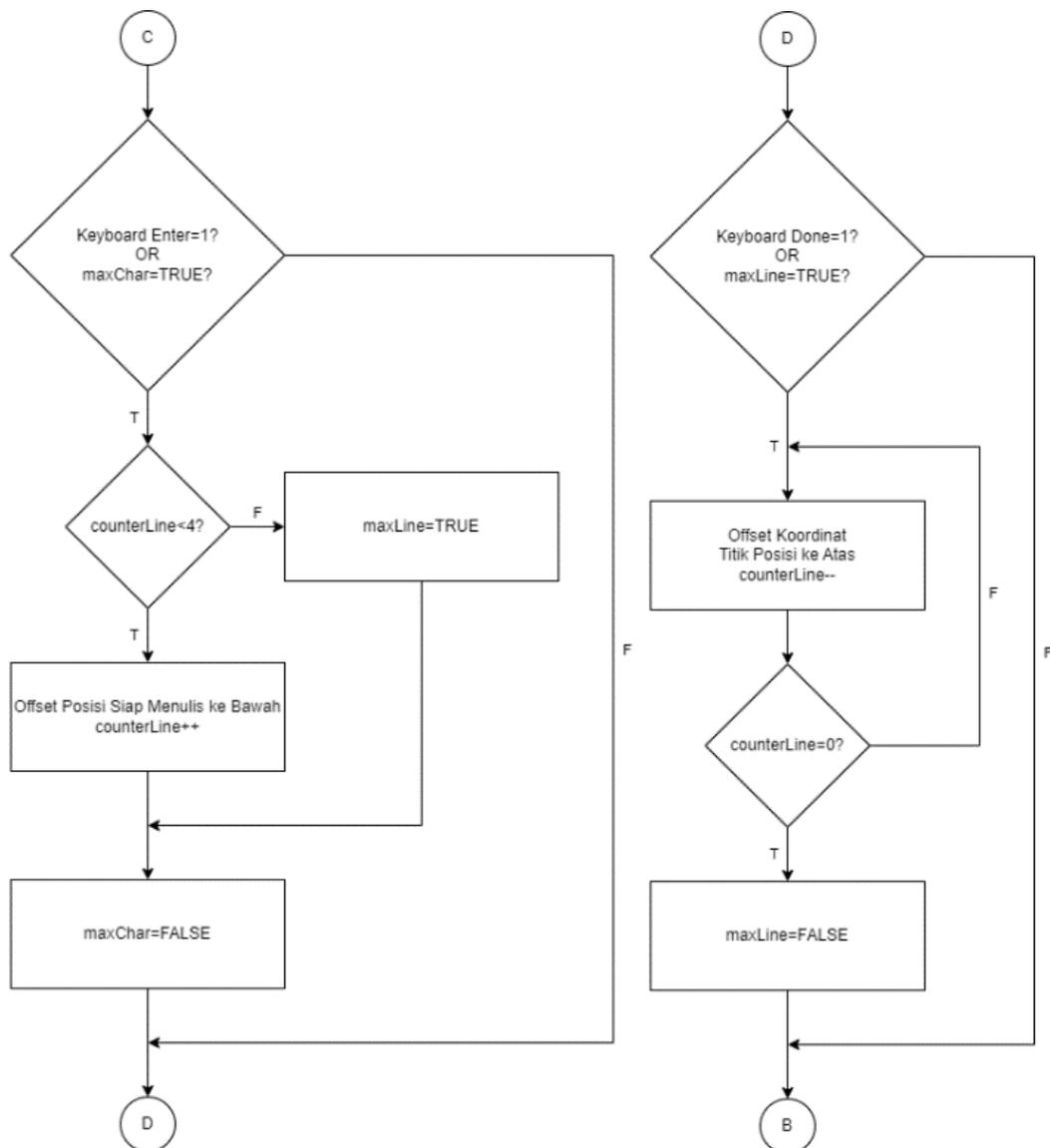
Setelah HMI selesai dibuat oleh penanggung jawab *interface* dan koneksi input output telah berhasil, robot penulis bisa langsung didemokan. Kinerja program robot dapat direpresentasikan dalam bentuk *flowchart* kerja, digambarkan sebagai:



Gambar 3.17 *Flowchart* Alur Program (Bagian 1)

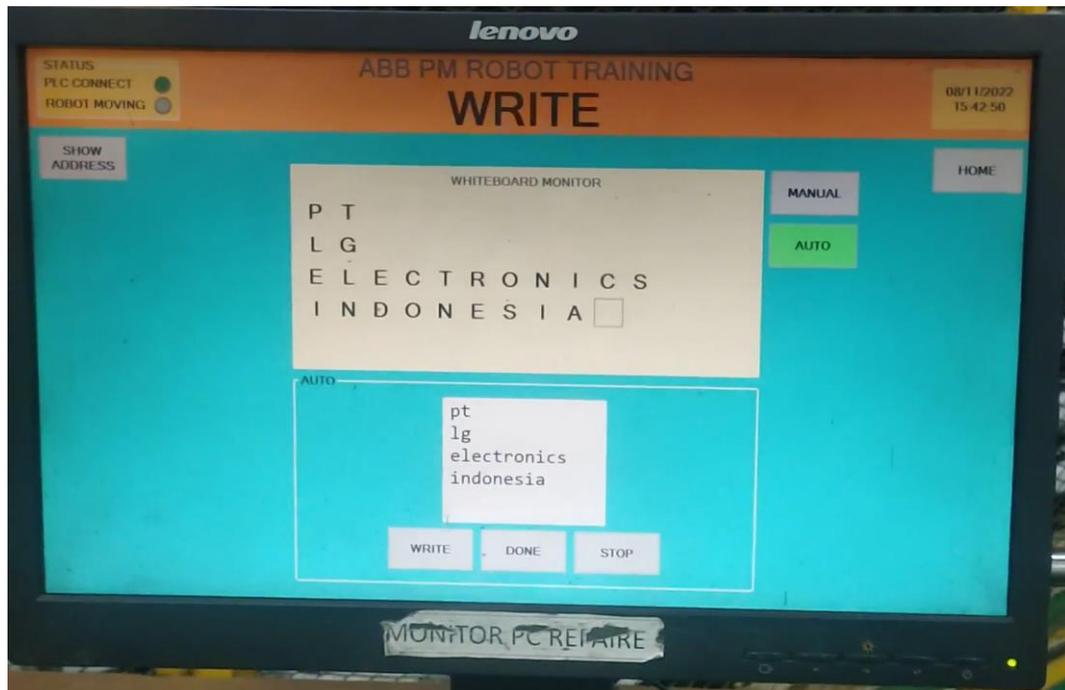


Gambar 3.18 *Flowchart* Alur Program (Bagian 2)

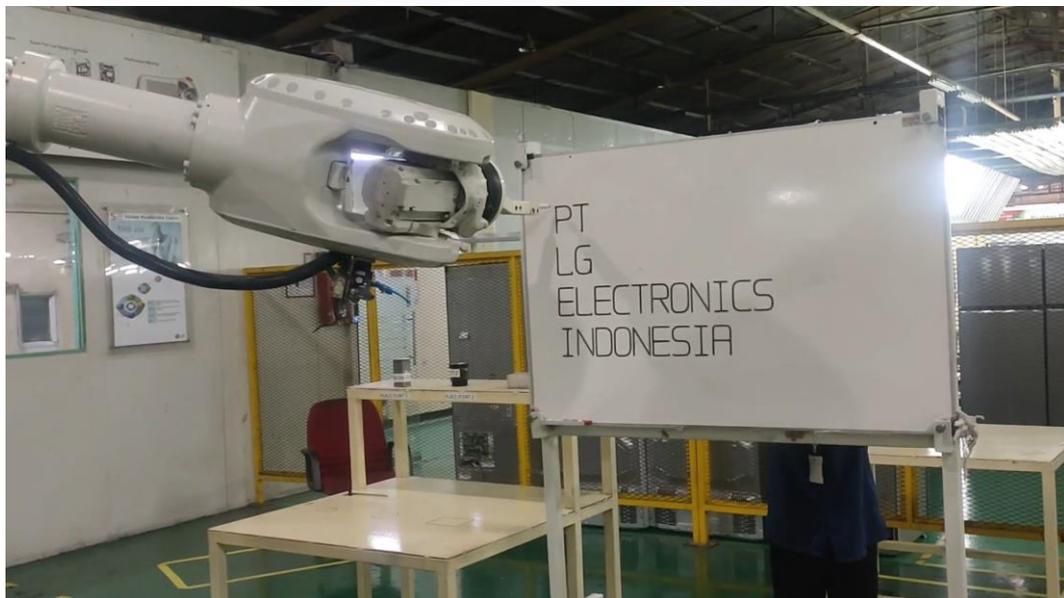


Gambar 3.19 Flowchart Alur Program (Bagian 3)

Hasil demo menunjukkan bahwa bahwa Robot ABB IRB 6700 berhasil menuliskan huruf yang diinginkan di posisi yang seharusnya pada papan. HMI yang telah dibuat oleh penanggung jawab *interface* berhasil mengirimkan output ke PLC. PLC kemudian memakai input dari HMI untuk mengirimkan output ke robot dan membuat robot dapat bergerak sesuai program. Fitur robot untuk dapat bergerak ke samping setelah menulis sebuah huruf, dan bergerak ke bawah ketika diminta atau mencapai batas juga berhasil. Setelah robot selesai menulis, robot juga dapat diminta untuk kembali ke lokasi awal dengan menekan tombol “Done” pada HMI.



Gambar 3.20 Tampilan HMI Robot Penulis



Gambar 3.21 Tampilan Hasil Tulisan Robot pada Papan Tulis

Program RAPID robot sepenuhnya dapat dilihat pada lampiran A.