

## BAB VII

### ROBOTIK

Robotik adalah ilmu yang mematerikan kecerdasan/*inteligencia* terhadap energi artinya pengendalian secara cerdas terhadap gerakan yang terkoordinasi secara nyata.

Robot berasal dari bahasa chekoslovakia 'robota' yang artinya tenaga kerja (*Create the Artificial life Robot → Cheko is "worker"*). Robot diharapkan dapat melihat, mendengar, menganalisa lingkungannya dan dapat melakukan tindakan-tindakan yang terprogram. Sekarang ini robot digunakan untuk maksud-maksud tertentu dan yang paling banyak adalah untuk keperluan industri seperti robot *cybotech P15* yang banyak digunakan untuk pekerjaan mengecat. Robot juga banyak diperlukan untuk mengerjakan pekerjaan-pekerjaan yang berbahaya, kotor dan sulit(tiga dimensi/3D).

Robot modern pertama kali dikembangkan oleh Joseph Engelberger dan George Devoe yang kemudian mendirikan perusahaan Unimation Company. Pengaruh robot mulai terasa dalam industri ketika negara Jepang mulai menggunakan secara intensif.

#### ***Asimov's 3 laws of robotics :***

1. Tidak boleh melukai manusia
2. Harus patuh terhadap perintah manusia, kecuali yang bertentangan dengan no. 1.
3. Harus mempertahankan diri, kecuali bertentangan dengan no. 1. dan 2.

*Lovable Friends and loyal companions to humans.*

***Robot*** : motor primitif kapasitas *intelligence di computer science* dan *control theory (sensory perception, decision making, kemampuan intelligent)*.

Aplikasi robotik terdapat pada :

- ❑ Bidang Industri : Keandalan Keefektifitas Biaya
- ❑ Bidang *Scientific* :
  - ◆ *Sensory Perception*
  - ◆ *Motor Control*
  - ◆ *Intelligent Behavior*

### **Sistem Pusat Urat Syaraf :**

Sistem pusat urat syaraf, struktur paling kompleks, berisi triliun urat syaraf, yang dihubungkan sedemikian rupa sehingga dapat menghasilkan sifat atau tindak tanduk dan imajinasi.

Pembagian secara hierarki menjadi 3 tingkat:

- Tingkat paling bawah : *Spinal Cord*
- Tingkat paling tengah : *Brain Stem*
- Tingkat paling atas : *Fore Brain*

Ketiga tingkatan ini membuat model otak.

Otak merupakan organ paling sulit, banyak komputasi yang berbeda, terlaksana pada saat yang bersamaan, dilaksanakan di berbagai tempat.

### **Komputer VS Otak**

Otak masih lambat dibandingkan dengan *Digital Komputer* sekarang, tetapi operasi paralel yang banyak komputasi melebihi kecepatan atau kapasitas komputer paling cepat yang pernah ada.

Jadi robot merupakan *control system*.

APA :

- Ingatan
- Hubungan ingatan dan otak
- Pikiran
- Bagaimana mekanisme yang membangkitkan imajinasi
- Persepsi dan bagaimana hubungan dengan objek yang didapat
- Emosi dan mengapa kita mempunyainya

- Keinginan dan bagaimana kita memilih apa yang kita inginkan
- Bagaimana kita merubah keinginan menjadi kegiatan
- Bagaimana kita merencanakan dan bagaimana kita tahu apa yang diharapkan pada masa yang akan datang

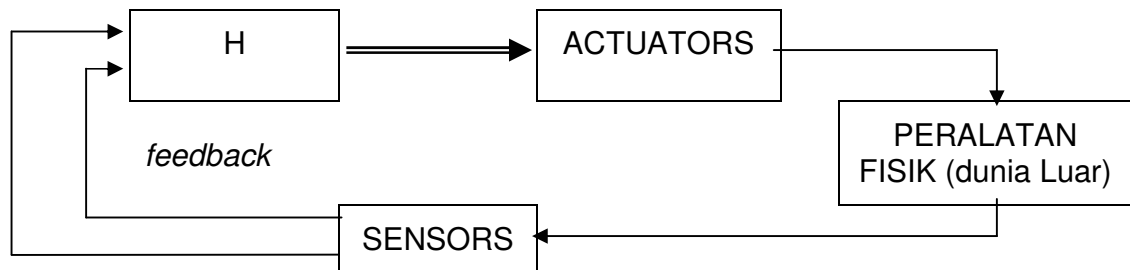
### Elemen-elemen Dasar dari Otak :

Bagaimana struktur dasar dan fungsi urat syaraf.

Struktur otak setiap individu berlainan, begitu juga urat syaraf yang satu dengan yang lainnya.

### Sensory Input

Suara, penglihatan, bau, rasa, raba + kekuatan, gerakan-gerakan otot adalah merupakan umpan balik ke pengatur mekanisme urat syaraf.



Gambar 7.1. *GOAL SEEKING CONTROL SYSTEM PADA SERUDMOTOR*

### Penjelasan Gambar 7.1.:

S = SOURCE

C : Command

F : Feedback

Fungsi H menghasilkan vector P untuk membangkitkan *ACTUATORS*(Pengerak).

## PERSEPSI

Menganalisis dan menentukan representasi object atau data Penglihatan, pendengaran, raba, bau, rasa. Tidak ada orang yang sama persis suaranya. Penglihatan merupakan masalah/*problem*? Yaitu jauh, dekatnya.

Pemecahan persoalan persepsi:

- *Distiraction*
- *Shoothing*
- *Segmentation*
- *Modeling*
- Analisis dan perlu *constraints*

Penggunaan *constraints* dalam pemecahan persoalan:

Analisis domain persoalan untuk menentukan apa saja *constraints*nya.  
Pecahkan persoalan dengan mengaplikasikan algoritma *constraints* yang mempergunakan *constraints* mulai tahap 1 untuk mengendalikan *search*.

Bila seseorang mendengar kata: '**ROBOT**', orang akan menanggapi bahwa robot adalah sebuah gambaran dari *hardware*(perangkat keras), yaitu peralatan mekanikal dan elektronikal yang membentuk fisik dari robot. Yang dapat mengerjakan tugas-tugas seperti halnya manusia.

Namun demikian sesungguhnya robot adalah: hubungan perangkat lunak(*software*) dengan perangkat keras(*hardware*), di mana perangkat lunak tersebut merupakan kecerdasan yang berada dibalik mesin itu yang mengendalikan seluruh gerak-gerik mesin tersebut dan dari kecerdasan inilah yang membedakan sebuah robot dari bentuk-bentuk otomatis lainnya.

### 7.1. Tipe Robot

Tipe robot ada 2(dua), yaitu:

1. Tipe pertama bentuknya sudah fix dan membutuhkan tempat yang tetap, misalnya robot-robot perangkat industri seperti yang digunakan untuk merakit mobil. Jenis ini dioperasikan hanya dalam lingkungan terkendali tinggi yang telah dirancang untuk hal tersebut.
2. Tipe kedua terdiri dari berbagai jenis robot otomatis/robot otonomous. Robot-robot ini dirancang untuk dioperasikan dalam dunia nyata. Robot jenis ini dibuat untuk bekerja seperti manusia. Robot jenis ini yang menarik bagi pemrogram AI.

Sebuah robot manipulator berfungsi ganda yang dapat diprogramkan kembali serta dirancang untuk menggerakkan bahan, onderdil, atau

peralatan khusus melalui gerakan-gerakan terprogram untuk melaksanakan berbagai tugas.

*Motionmate* adalah sebuah robot industri yang paling sederhana untuk melakukan proses mengambil dan meletakkan komponen-komponen di dalam proses produksi. Robot ini dapat mengangkat komponen seberat 5 pound.

## 7.2. Gerakan-Gerakan Robot, Industri Robot

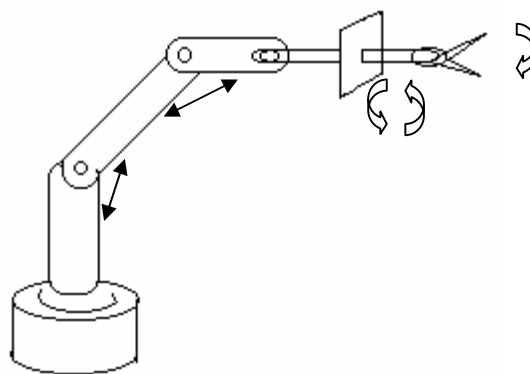
### a. Tangan Robot:

Pada dasarnya menampilkan manipulasi peralatan dari sebuah robot baik untuk tipe industri maupun otomatis.

Contoh: Mengambil sebuah gelas berisi air. Kelihatan gerakan tersebut tidak membutuhkan dukungan dan tanpa pemikiran, kenyataannya hal tersebut merupakan sebuah proses yang komplikasi di mana membutuhkan koordinasi dari beberapa otot.

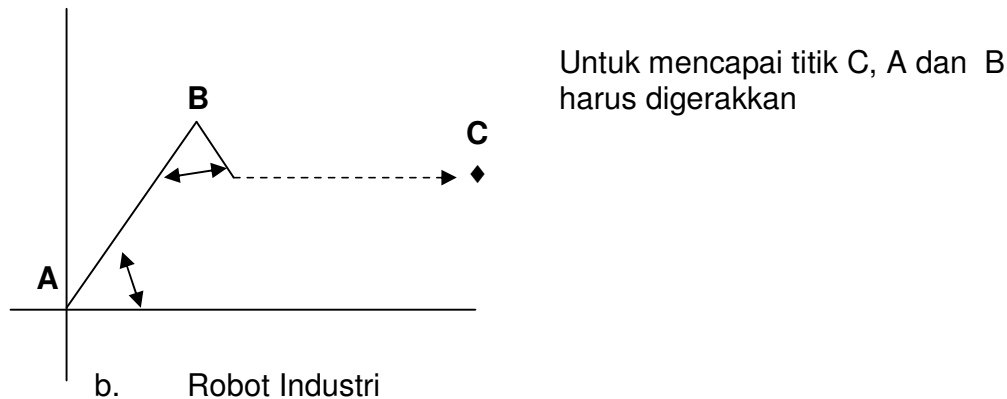
( Ingat! Bayi membutuhkan beberapa bulan untuk mempelajari hal tersebut ).

Tangan robot umumnya dibentuk pada tangan manusia, kebanyakan tangan robot adalah *six-axis arm*, karena diperbolehkan untuk mendapatkan gerakan bebas.



Gambar 7.2. Tangan Robot

Kesulitan umum dari pengendalian tangan jenis robot ini bukan pada ketepatan pergerakan, tetapi masalah koordinasi dari 6 titik.



Lapangan robot-robot jenis ini pada umumnya diterapkan untuk membentuk dan mempekerjakan robot-robot perakitan industri, karena robot-robot ini digunakan dalam lingkungan terkendali, di mana robot-robot ini dipertimbangkan kurang pandai dibandingkan robot-robot otomatis.

Robot-robot industri hanya dapat menunjukkan proses-proses di mana secara eksplisit telah diprogramkan untuk melakukan hal-hal tersebut.

Terdapat dua cara dalam mengajarkan robot, di mana robot-robot berpikir:

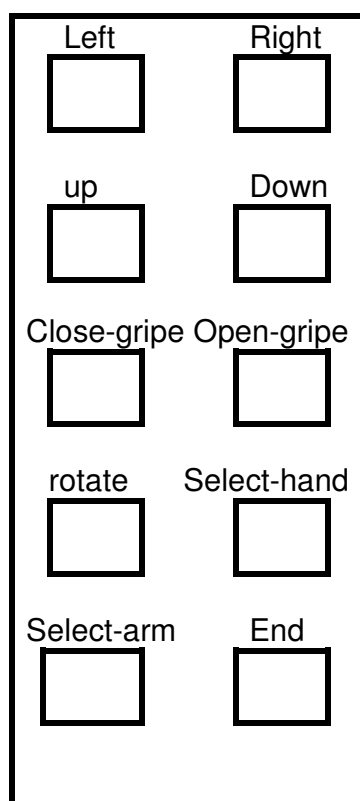
1. Dengan menggunakan *Teach Pendant*.
2. Diprogramkan dengan menggunakan bahasa kendali robot *robotic control language*.

#### 1) *Teach Pendant*

*Teach pendant* merupakan metode yang paling banyak digunakan untuk mengajarkan robot hal-hal baru. Bentuk dan fungsi *teach pendant* mirip dengan remote control mobil-mobilan. *Teach pendant* digunakan untuk mengontrol setiap titik dari tangan robot. Misalkan kita ingin agar robot menutup gripnya, maka yang perlu kita lakukan adalah menekan tombol yang telah difungsikan untuk

menutup grip tangan robot tersebut. Walaupun setiap pembuat robot tidak akan sama persis bentuk *teach pendant*nya, namun prinsip kerja semua *teach pendant* adalah sama. Metoda umum dari pemrograman sebuah robot untuk menunjukkan proses baru melalui kegunaan *Teach Pendant*.

*Teach pendant* adalah tangan diletakkan pada kotak kendali yang memperbolehkan seorang operator untuk memudahkan berbagai hubungan dengan robot.



*Teach pendant* tidak terhubung secara langsung ke robot, tetapi terhubung secara keseluruhan ke komputer kendali robot utama.

Jika diinginkan mengajari robot sebuah proses dapat digunakan *Teach Pendant* untuk membimbing robot ke rangkaian pergerakan yang membentuk proses tersebut.

Setiap pergerakan, komputer merekam setiap posisi. Setelah pengajarannya selesai, robot dapat menunjukkan pekerjaannya sendiri tanpa membutuhkan pembantu.

Misalnya robot diajarkan mengambil benda di sebelah kiri robot maka robot digerakkan ke kiri dengan menempel panel kiri kemudian menutup gripnya. Setelah rangkaian pekerjaan tadi selesai, komputer merekam semua posisi yang dilalui robot agar robot dapat melakukan gerakan yang sama pada kesempatan berikutnya tanpa dipandu lagi

Gambar 7.3. *Teach pendant*

## 2) *Robotic Control Language*

*Teach Pendant* adalah sebuah metoda yang sangat baik untuk mengajari robot pekerjaan yang sederhana seperti mengelas dan mengecat. Namun ketika pekerjaan yang harus dilakukan semakin hari semakin kompleks dan *event* sinkronisasi eksternal semakin hari semakin penting atau ketika robot perlu untuk mengenali dan

menanggapi(merespon) berbagai kemungkinan situasi yang berbeda, maka sistem *teach pendant* lama kelamaan menjadi terbebani dan tidak sanggup lagi untuk menangani pekerjaan tersebut, sehingga *teach pendant* tidak berguna.

Hal ini menjadi alasan di mana bahasa kendali robotik (*Robotic Control Language*) dikembangkan. Bahasa kendali robotik adalah bahasa komputer yang secara khusus dirancang untuk mengendalikan sebuah robot..

Dalam *Robotic Control Language* tambahan terdapat perintah-perintah baru, seperti kendali LOOP(*Loop Control*) dan statement kondisional(*conditional statement*). Bahasa kendali robotik juga memasukkan perintah-perintah yang mengendalikan pergerakan robot. Bahasa ini mengendalikan pergerakan di mana menentukan bahasa kendali robotik yang merupakan bagian dari yang lain, seperti bahasa pemrograman untuk tujuan umum. Sebuah robotik dikendalikan oleh bahasa yang berisi sebuah basis data yang telah tersedia(*built in database*) di mana digunakan untuk menyimpan informasi tentang setiap gerakan yang akan dilakukan oleh robot. Bahasa kendali robotik bukan diciptakan untuk menggantikan *teach pendant* melainkan menyempurnakan *teach pendant*. Satu hal yang perlu dipahami bahwa bahasa kendali robotik dirancang untuk meletakkan kembali *teach pendant*, tetapi untuk mendukung bahasa tersebut. Oleh karena itu bahasa kendali robotik harus menyediakan *interface* untuk *teach pendant*.

Dalam metoda ini mengajarkan robot untuk menggunakan informasi dengan menggunakan arahan dan kemudian menggunakan bahasa kendali robotik untuk menjelaskan bagaimana robot seharusnya menggunakan informasi tersebut. Umumnya setiap lokasi tertentu diberikan sebuah nama simbolik di mana program dapat mengacu ke nama tersebut. Bahasa kendali robotik memiliki syntax yang serupa dengan bahasa BASIC dan paling banyak



digunakan adalah VAL dikembangkan oleh UNIMATION Corporation. Untuk lebih mengerti tentang program VAL, pelajari program berikut yang digunakan untuk mesin *conveyor belt* agar robot dapat memindahkan kotak pada roda ban berjalan.

contoh :

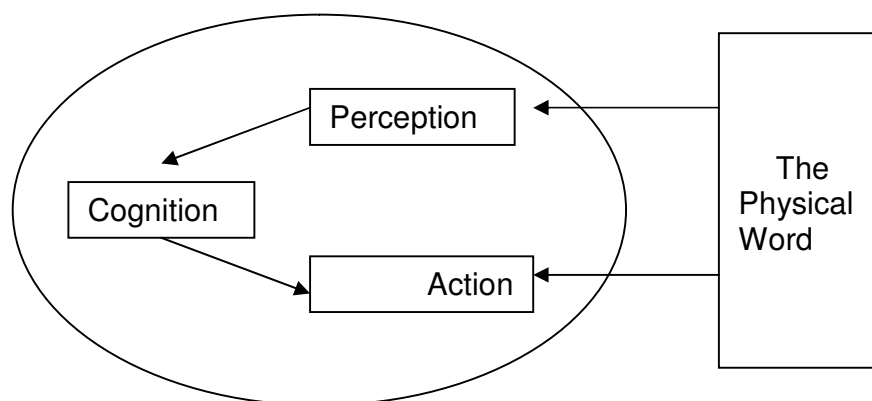
```
REMARK WAIT FOR OBJECT
10          WAIT 2
REMARK OBJECT PRESENT, REMOVE IT
MOVE POS1
MOVE POS2
CLOSE1 REMARK CLOSE THE GRIP
MOVE BOX1
OPEN1 REMARK DROP THE OBJECT
GOTO 10
```

Dengan menggunakan program ini, robot akan menunggu sampai masukan sinyal 2 aktif: artinya:robot tidak akan melakukan apa-apa sampai sinyal nomor 2 meninggi(aktif). Sinyal aktif ini menandakan atau memberitahukan robot bahwa ada objek(kotak) pada mesin ban berjalan. Kemudian robot akan mendekat ke mesin ban berjalan(POS1), setelah itu robot akan berada di atas objek(POS2), lalu menutup grip untuk memegang objek dan memindahkannya dari ban berjalan. Akhirnya program akan loop dan menunggu objek berikutnya. VAL hanya mendukung huruf kapital. Ketiga posisi: POS1, POS2, dan BOX1 adalah nama simbolik dari lokasi dimana robot telah diajarkan oleh *teach pendant* (merekam lokasi tersebut).

### 7.3. ***Autonomous Robot***

*Autonomous Robot* jauh lebih kompleks dari industrial robot, karena robot ini harus jauh lebih pandai. Jika *Autonomous Robot* berhasil beroperasi dalam lingkungan tak terkendali dari dunia nyata(kehidupan sebenarnya), maka robot ini akan membutuhkan berbagai keahlian di mana industrial robot tidak membutuhkan.

Misalkan saja diperlukan sensor agar robot dapat melihat dan mendengar berarti robot tersebut harus mengerti *natural language* dan arti dari bahasa tersebut. Memberikan robot dua kemampuan itu bukanlah pekerjaan yang mudah. Selain itu robot juga harus mampu menyelesaikan masalah yang merupakan pekerjaan programming yang paling sulit diantara pekerjaan lainnya. Hal ini penting agar robot dapat beradaptasi dengan banyak situasi sebab jelas kita tidak dapat memprogram robot untuk menyelesaikan semua masalah secara terperinci, yang dapat terjadi melainkan hanya agar robot dapat beradaptasi untuk kemudian memilih alternatif.



Gambar 7.4. Sebuah Rancangan untuk sebuah Autonomous Robot

*Kita mendukung sebuah definisi masalah dasar AI secara dauni yang ditangani, umumnya agar komputer mempunyai kemampuan seperti manusia.*

Perception meliputi interpretasi penglihatan (*sights*), suara (*sounds*), *smell* (rasa) dan menyuruh / maraba (*tauch*).

Action meliputi kemampuan mengendalikan / navigasi alam atau dunia dan memanipulasi objek.

Salah satu pemecakannya kita membuat robot, yang .... Ia harus mengerti proses ini

#### 7.4. Pembuatan Simulator Robot(**ROBOT SIMULATOR**)

Robot Simulator terdiri dari sebuah lingkungan yang diisi dengan 4(empat) obyek: segiempat, dua segitiga, dan robot dan sisa dari screen adalah *blank*.

Terdapat dua cara untuk memprogram robot:

1. Menggunakan *robotic-control language*, terdiri dari *command-command* :
  - a. moveto
  - b. move
  - c. findt
  - d. finds
  - e. label
  - f. goto
  - g. ifsense/then

2. Mengajarkan robot

#### **moveto**

Menyuruh robot pergi ke lokasi baris, kolom tertentu pada screen. Sudut kiri atas screen adalah lokasi 0,0. Hal ini berarti baris-baris dinomori dari 0 sampai dengan 24 dan kolom dinomori dari 0 sampai dengan 79

contoh: moveto 12 60

#### **move**

Menyebabkan robot memindahkan satu posisi dalam arah yang ditentukan, di mana harus antara lain: kiri, kanan, atas atau bawah.

contoh :        move up  
                  move left

#### **findt dan finds**

Memerintahkan robot untuk menemukan sebuah segitiga atau sebuah segiempat. robot hanya dapat menemukan sebuah obyek jika obyek tersebut dilokasikan di atas obyek. Setelah robot menemukan obyek, robot akan ditempatkan satu kolom ke kiri dari sudut kanan paling atas obyek.

#### **label**

Sebuah label dapat berupa sebuah rangkaian karakter.

contoh:        label one  
                  label box38

**goto**

*Statement* ini menyebabkan eksekusi program dipindahkan ke label tertentu.

contoh : goto five

**ifsense**

*ifsense* menentukan apakah terdapat sebuah obyek baik di sisi kanan, kiri, atas atau bawah robot. Jika terdapat obyek, maka *statement* yang mengikuti 'then' dieksekusi, jika tidak terdapat obyek, eksekusi melanjutkan ke baris berikut dari program arah 'sense' berupa up, down, left, atau right.

contoh :

```
label one
    move down
    ifsense right then goto one
```

a. Mengajarkan Robot

Perintah(*command*) '*teach*' dapat digunakan setiap saat dalam program. Perintah ini menyebabkan layar editor akan diletakkan kembali di layar. Robot ditempatkan pada posisi asalnya 0,0. Pada saat ini kita dapat menggunakan *keypad* angka untuk memindahkan robot dengan menggunakan tombol-tombol panah. Untuk mengembalikan posisi ke 0,0 dapat menggunakan tombol:

***Home***

Setiap gerakan direkam dan sesungguhnya menjadi bagian program. Untuk menghentikan mengajarkan robot maka menekan tombol: **End**.

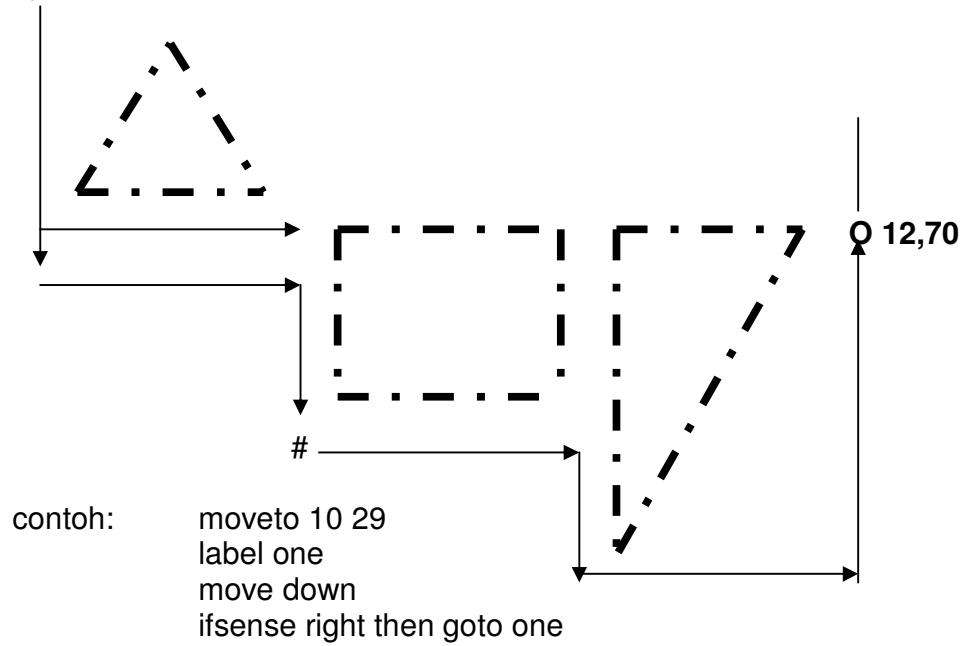
b. Penggunaan Simulator

Bila mengeksekusi program simulator, pertama kali diinstruksikan untuk memasukkan sebuah program. Hal ini dapat dilakukan dengan memasukkan sebuah garis pada setiap saat.

contoh :      moveto      12    70  
                   moveto      0     0  
                   finds

Lintasan Robot

0,0



Lintasan Robot

0,0

