BAB II
LANDASAN TEORI

Robot telah banyak dikembangkan, karena robot berguna untuk membantu kerja manusia misalnya, untuk pekerjaan dengan resiko bahaya ataupun melakukan pekerjaan yang membutuhkan tenaga besar seperti di bidang industri. Jenis robot yang dipakai merupakan *mobile* robot, yaitu robot yang ciri khasnya adalah mempunyai aktuator berupa roda untuk menggerakkan keseluruhan badan robot tersebut, sehingga robot tersebut dapat melakukan perpindahan posisi dari satu titik ke titik yang lain. Dalam bab ini dijelaskan mengenai teori - teori tentang perancangan robot pengangkut *box* berdasarkan perbedaan warna. Berikut ini teori perangkat keras dan perangkat lunak yang digunakan.

2.1 Perangkat Keras (Hardware)

2.1.1 Mikrokontroler Basic Stamp (BS2P40)

Basic stamp adalah mikrokontroler yang dikembangkan oleh *Parallax Inc* yang diprogram menggunakan *format* bahasa pemrograman *basic*. Program yang dibuat di-*downloa*d melalui *port* serial dan dapat menggunakan konverter USB to Serial untuk komputer yang tidak memiliki *port* serial. Mikrokontroler *basic stamp* membutuhkan *power supply* saat men*download* pemrograman dan tidak kehilangan program yang sudah di *download* saat baterai atau *power supply* dicabut.

Kode PBasic disimpan di dalam EEPROM serial pada *board basic stamp*. EEPROM menyediakan penyimpanan yang sulit diubah, yaitu menjaga memory saat kehilangan *power*. EEPROM digunakan dalam *basic stamp* 1 dan 2 yang dijamin menyimpan selama 40 tahun ke depan dan mampu ditulisi ulang 10.000.000 kali per lokasi memori.

Mikrokontroler *basic stamp* memiliki versi yang berbeda - beda. Basic stamp memiliki tujuh versi, yaitu *basic stamp* 1, *basic stamp* 2, *basic stamp* 1e, *basic stamp* 2P, *basic stamp* 2Pe dan *basic stamp* 2sx. *Basic*

stamp jalan pada tegangan DC 5 sampai 15 volt. Pada modul *basic stamp* terdapat IC *regulator* LM7805 dengan *output* 5 volt yang mengubah input 6 hingga 15 volt (pada pin V_{IN}) turun menjadi 5 volt yang dibutuhkan komponen. *Basic stamp* yang di pakai adalah *basic stamp* BS2P40 yang mempunyai 40 pin I/O. Pemilihan basic stamp ini karena membutuhkan banyak *input* atau *output* yang dipakai. Berikut ini adalah tampilan *basic stamp* BS2P40.

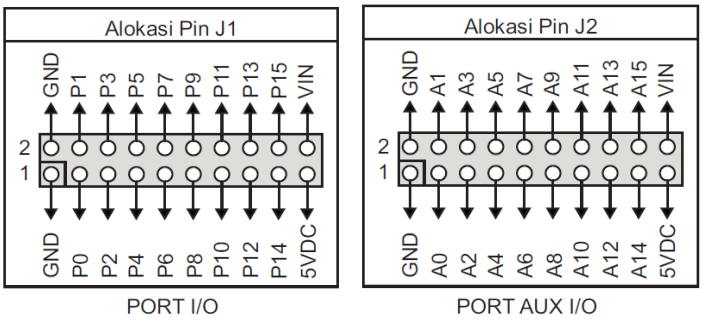


Gambar II.1. Modul basic stamp (BS2P40)

Basic stamp ini mempunyai spesifikasi *hardware* sebagai berikut:

1. Mikrokontroler *basic stamp* 2P40 *Interpreter Chip* (PBASIC48W/P40)
2. 8 x 2Kbyte EEPROM yang mampu menampung hingga 4000 instruksi.
3. Kecepatan prosesor 20MHz Turbo dengan kecepatan eksekusi program hingga 12000 instruksi per detik.
4. RAM sebesar 38*byte* (12 I/O, 26 variabel) dengan *Scratch Pad* sebesar 128 *byte*.
5. Jalur *input* / *output* sebanyak 32 pin.
6. Tersedia jalur komunikasi serial UART RS-232 dengan konektor DB9.
7. Tegangan input 9 – 12 V_{DC} dengan tegangan output 5 V_{DC}.

Berikut ini adalah alokasi pin yang terdapat pada mikrokontroler *basic stamp* BS2P40.



Gambar II.2. Alokasi pin basic stamp

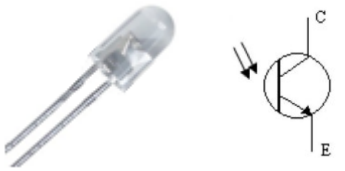
2.1.2 Sensor Garis

Sensor garis adalah jenis sensor yang yang berfungsi mendeteksi warna garis hitam atau putih. Sensor ini sangatlah penting karena sensor menentukan arah dan gerakan robot.

Sensor pendeteksi garis yang digunakan dalam robot adalah mendasarkan pada prinsip pemantulan cahaya dan *phototransistor* sebagai penerima cahaya. Dengan memanfaatkan IC komparator sebagai pembanding tegangan, yang nantinya akan menghasilkan *output digital* dari sensor.

2.1.2.1 Phototransistor

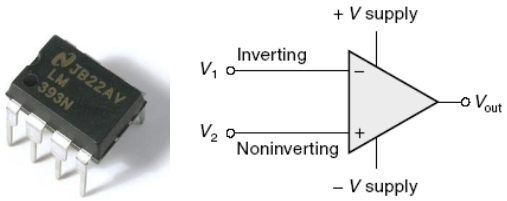
Phototansistor merupakan suatu jenis transistor yang sangat peka trahadap cahaya yang ada disekitarnya. Ketika basis menangkap cahaya maka *collector* akan terhubung dengan *emitter* dalam hal ini transistor bekerja. Prinsip kerja *phototransistor* sama seperti transistor pada umumnya dengan kata lain *phototransistor* akan bekerja seperti saklar dengan parameter cahaya untuk mendapatkan kondisi *on* dan *off*. Berikut ini adalah simbol *phototransistor*. [5]



Gambar II.3. Bentuk fisik phototransistor dan simbol phototransistor

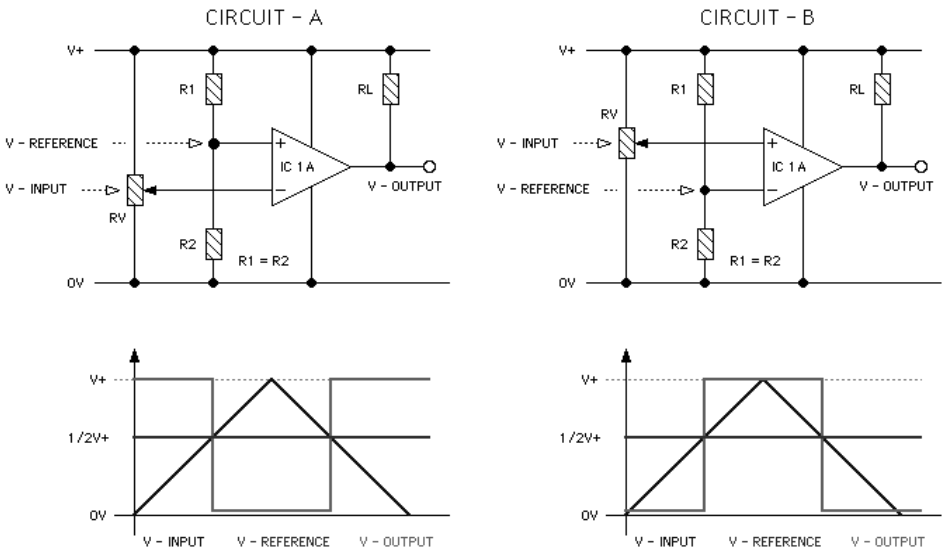
2.1.2.2 Komparator

Komparator adalah sebuah rangkaian elektronik yang berfungsi untuk membandingkan sebuah sinyal masukan dengan tegangan referensi (V_{Ref}). Asumsi sebuah garis bilangan, dimana nol dianggap sebagai tegangan referensi atau *threshold* atau pembatas, jika ada bilangan yang lebih dari nol maka bilangan itu disebut bilangan positif tetapi sebaliknya, jika ada bilangan di bawah nol maka disebut bilangan negatif. [4]



Gambar II.4. Bentuk fisik IC LM 393 dan simbol komparator

Pada komparator, *threshold* berfungsi membandingkan sebuah sinyal *input*, sedangkan *output*nya akan memiliki dua kondisi berbeda yaitu *low* atau *high*, tergantung rancangan dan konfigurasi dari rangkaian *op-amp* yang digunakan. Ilustrasi sebuah rangkaian komparator seperti terdapat pada gambar berikut:



Gambar II.5. Rangkaian komparator

Dari dua gambar di atas, komparator terbagi ke dalam dua konfigurasi, yaitu *mode inverting* atau *mode non-inverting*. Pada rangkaian *inverting* (gambar sebelah kiri), sinyal input masuk ke pin + dari *op-amp*, dan tegangan referensi masuk ke pin -. Jika sinyal *input* melewati / di atas *threshold*, maka *output* akan berlogika *low*, dan jika sinyal *input* di bawah *threshold*, maka *output* berlogika *high*.

Untuk konfigurasi *non-inverting* ada di (gambar sebelah kanan), jika sinyal *input* di bawah *threshold*, maka *output* akan berlogika *low*, dan jika sinyal *input* di atas *threshold* maka *output* akan berlogika *high*. Detil ilustrasi sinyal *input*, *threshold* dan *output* ada di gambar II.5. bagian bawah kiri dan bawah kanan. [4]

2.1.3 Penggerak Mobil Robot

2.1.3.1 Motor DC

Roda digerakkan menggunakan dua buah motor DC yang dipasang pada roda sebelah kiri dan kanan. Motor DC merupakan peralatan elektromekanik dasar yang berfungsi untuk mengubah

tenaga listrik menjadi tenaga mekanik. Secara umum, kecepatan putaran poros motor DC akan meningkat seiring dengan meningkatnya tegangan yang diberikan. Dengan demikian, putaran motor DC akan berbalik arah jika polaritas tegangan yang diberikan juga dirubah.



Gambar II.6. Motor DC

Motor gear DC tidak dapat dikendalikan langsung oleh mikrokontroler, karena kebutuhan arus yang besar sedangkan keluaran arus dari mikrokontroler sangat kecil. Motor *driver* merupakan alternatif yang dapat digunakan untuk menggerakkan motor DC.

2.1.3.2 IC Motor Driver

L298 adalah IC yang dapat digunakan sebagai *driver* motor DC. IC ini menggunakan prinsip kerja *H-Bridge*. Tiap *H-Bridge* dikontrol menggunakan level tegangan TTL yang berasal dari *output* mikrokontroler. L298 dapat mengontrol 2 buah motor DC. Tegangan yang dapat digunakan untuk mengendalikan robot bisa mencapai tegangan 46 V_{DC} dan arus 2 A untuk setiap kanalnya. Berikut ini bentuk IC L298 yang digunakan sebagai motor *driver*.



Gambar II.7. IC driver motor L298

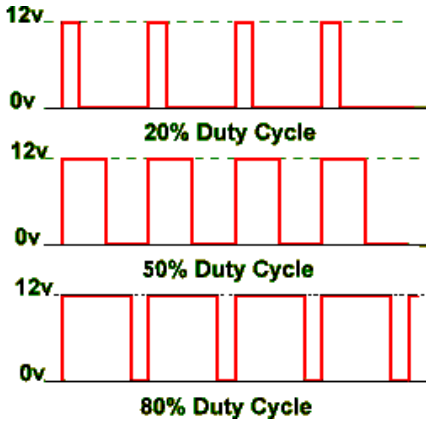
Pengaturan kecepatan kedua motor dilakukan dengan cara pengontrolan lama pulsa aktif (mode PWM – *Pulse width Modulation*) yang dikirimkan ke rangkaian *driver* motor oleh pengendali (mikrokontroler *basic stamp*). Duty *cycle* PWM yang dikirimkan menentukan kecepatan putar motor DC.

2.1.3.3 PWM (Pulse Width Modulation)

PWM *(Pulse width Modulation*), adalah sebuah metode untuk pengaturan kecepatan perputaran, dalam hal ini adalah motor DC untuk gerak robot. PWM dapat dihasilkan oleh empat metode, sebagai berikut :

1. Metode analog
2. Metode digital
3. IC diskrit
4. Mikrokontroler

Pada robot ini, metode PWM dikerjakan oleh mikrokontroler. Metode PWM ini akan mengatur lebar atau sempitnya periode pulsa aktif yang dikirimkan oleh mikrokontroler ke driver motor. Pada pengaturan kecepatan robot, nilai PWM mulai dari 0-255. Secara *analog* besaran PWM dihitung dalam prosentase, nilai ini didapat dari perbandingan: T *high* / (T *high* + T *low*) * 100%. Dimana T adalah periode atau waktu tempuh untuk sebuah pulsa, yang terbagi menjadi bagian puncak positif (T *high*) dan puncak negatif (T *low*).



Gambar II.8. Ilustrasi prosentase PWM

Semakin rapat periode antar pulsa, maka frekuensi yang dihasilkan akan semakin tinggi, ini berarti kecepatan akan bertambah. Semakin lebar jarak antar pulsa, maka frekuensi semakin rendah ini berarti kecepatan berkurang atau menurun. Kondisi pemberian kecepatan harus disesuaikan dengan kondisi *track* yang akan dilewati oleh robot, misal pada saat jalan lurus, naik atau turun harus mendapatkan nilai PWM yang tepat. [1]

2.1.4 Catu Daya

Catu daya memegang peranan yang sangat penting dalam hal perancangan sebuah robot. Tanpa bagian ini robot tidak akan berfungsi. Begitu juga bila pemilihan catu daya tidak tepat, maka robot tidak akan bekerja dengan baik.

Penentuan sistem catu daya yang akan digunakan ditentukan oleh banyak faktor, diantaranya :

1. Tegangan

   Setiap aktuator tidak memiliki tegangan yang sama. Hal ini akan berpengaruh terhadap disain catu daya. Tegangan tertinggi dari salah satu aktuator akan menentukan nilai tegangan catu daya.

2. Arus

   Arus memiliki satuan Ah (*Ampere-hour*). Semakin besar Ah, semakin

lama daya tahan baterai bila digunakan pada beban yang sama.

3. Teknologi Baterai

Baterai isi ulang ada yang dapat diisi hanya apabila benar-benar kosong, dan ada pula yang dapat diisi ulang kapan saja tanpa harus menunggu baterai tersebut benar-benar kosong.

Baterai yang digunakan pada *mobile* robot ini adalah baterai berjenis *Nickel Metal Hydride*. Hal ini karena jenis baterai ini merupakan jenis baterai yang dapat diisi ulang (*rechargeable*). Baterai ini memiliki tegangan kerja 1,2 Volt. Berikut ini adalah contoh sebuah baterai Ni-MH 2700 mAh.



Gambar II.9. Baterai Ni-MH 2700 mAh

Untuk pengisian baterai ini dapat dilakukan kapan saja, selain jenis baterai Ni-MH, masih banyak lagi jenis baterai yang tersedia di pasaran dengan spesifikasi yang beragam dan dapat digunakan untuk catu daya pada sebuah robot. Diantaranya baterai Ni-cd, *Alkaline*, *Lithium*, *Lead Acid* dan sebagainya, yang masing-masing mempunyai kelebihan dan kekurangannya. [8]

2.1.5 Penggerak Gripper

Gripper / penjepit digunakan untuk alat pengangkat *box*, *gripper* digerakkan dengan menggunakan 2 buah mini servo, dimana untuk menjepit dan mengangkat *box*. Berikut adalah mini servo yang dipakai:



Gambar II.10. GWS servo

Sebuah motor servo *standard* adalah alat yang dapat mengendalikan posisi, dapat membelokkan dan menjaga suatu posisi berdasar penerimaan pada suatu signal elektronik itu. Karena motor DC servo merupakan alat untuk mengubah energi listrik menjadi energi mekanik, maka magnit permanent motor DC servo yang mengubah energi listrik ke dalam energi mekanik melalui interaksi dari dua medan magnit.



Gambar II.11. Servo standar

Bagian - bagian dari sebuah motor servo *standard* adalah sebagai berikut:

1. Konektor yang digunakan untuk menghubungkan motor servo dengan Vcc, *Ground* dan signal *input* yang dihubungkan ke *Basic Stamp*.
2. Kabel menghubungkan Vcc, *Ground* dan signal *input* dari konektor ke motor servo.
3. Tuas menjadi bagian dari motor *servo* yang kelihatan seperti suatu bintang *four-pointed*. Ketika motor servo berputar, tuas motor servo akan bergerak ke bagian yang dikendalikan sesuai dengan program.
4. *Cassing* berisi bagian untuk mengendalikan kerja motor servo yang pada dasarnya berupa motor DC dan *gear*. Bagian ini bekerja untuk menerima instruksi dari *basic stamp* dan mengkonversi ke dalam sebuah pulsa untuk menentukan arah / posisi servo.

2.1.6 Sensor Warna

Sensor warna berfungsi sebagai pembeda warna *box*. Warna di pakai adalah warna dasar yaitu *Red*, *Green* dan *Blue* (RGB) untuk nantinya akan dikelompokkan berdasarkan warna yang sama di tempat penyimpanan. Berikut ini adalah sensor warna seri TCS3200 yang dipakai.



Gambar II.12. Modul sensor warna TCS3200

2.2 Perangkat Lunak (Software)

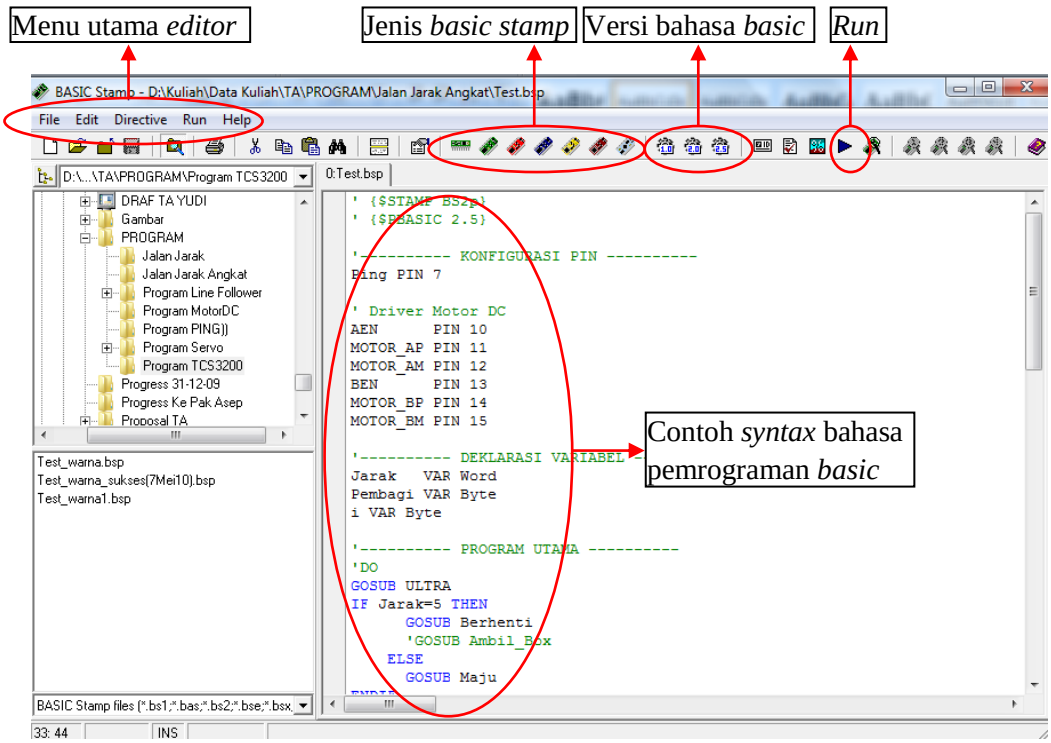
2.2.1 Pengenalan Basic Stamp Editor

Perangkat lunak merupakan faktor penting dalam tahap perancangan robot. Perangkat lunak ini merupakan algoritma gerak dan tugas robot dalam bentuk listing program yang ditanamkan kedalam mikrokontroler. Program dapat bermacam - macam bentuk versi dan bahasa pemrogramannya, sesuai dengan spesifikasi dari mikrokontroler yang digunakan.

Mikrokontroler *basic stamp* (BS2P40) menggunakan bahasa pemrograman *basic*. *Software* yang digunakan adalah *basic stamp editor*. *Basic stamp editor* adalah sebuah editor yang dibuat oleh *Paralax Inc* untuk menulis program, meng*kompile* dan men*download*nya ke mikrokontroler keluarga *basic stamp*. Program ini memungkinkan penggunanya memprogram *basic stamp* dengan bahasa *basic* yang relatif ringan dibandingkan bahasa pemrograman lainnya. Berikut ini beberapa instruksi-instruksi dasar yang dapat digunakan pada mikrokontroler *basic stamp*.

Tabel II.1. Beberapa instruksi dasar basic stamp

Instruksi	Keterangan
DO...LOOP	Perulangan
GOSUB	Memanggil prosedur
IF..THEN	Percabangan
FOR...NEXT	Perulangan
PAUSE	Waktu tunda milidetik
IF...THEN	Perbandingan
PULSOUT	Pembangkit pulsa
PULSIN	Menerima pulsa
GOTO	Loncat ke alamat memori tertentu
HIGH	Menset pin I/O menjadi 1
LOW	Menset pin I/O menjadi 0
PWM	Konversi suatu nilai digital ke keluaran analog lewat pulse width modulasi



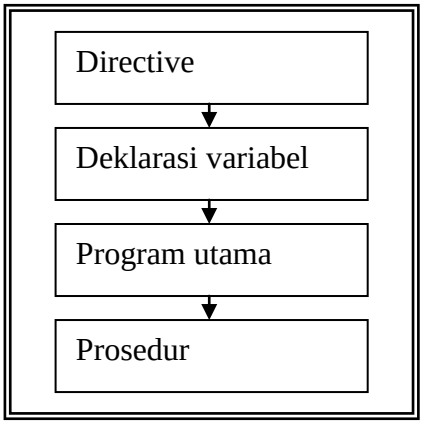
Gambar II.13. Tampilan basic stamp editor

2.2.2 Memprogram Basic Stamp

Dalam pemrograman, sebuah program lengkap secara umum dapat dibagi menjadi empat bagian penting, yaitu :

1. Header
2. Variabel
3. Program utama
4. Prosedur

Pemrograman dalam basic stamp editor, secara blok dibagi menjadi empat bagian penting.



Gambar II.14. Urutan bagian dari program dalam basic stamp

2.2.2.1 Directive

Directive ditulis paling awal program yang dibuat. Bagian ini menentukan *tipe prosesor* yang digunakan dan *versi dari compiler PBASIC* yang digunakan untuk mengkompile *bahasa basic* menjadi bahasa mesin. Tampilannya adalah seperti gambar berikut :

```
' {$STAMP BS2sx}     ' pemilihan tipe prosesor
' {$PBASIC 2.5}      ' pemilihan versi kompiler PBASIC
```

Gambar II.15. Tampilan bagian directive

2.2.2.2 Menentukan Variabel

Menentukan PIN mikrokontroler yang digunakan serta membuat variabel. Ada beberapa ketentuan untuk mendeklarasikan variabel yaitu :

1. PIN : PIN dari mikrokontroler (0-15)
2. VAR : Variabel
3. CON : Konstanta

PIN yang digunakan sudah ditentukan sesuai dengan konfigurasi *hardware* / *mainboard* yang digunakan adalah BS2P40. Selain itu dapat membuat variabel bebas yang nantinya dapat digunakan untuk keperluan perulangan atau yang lainnya.

```
'---- D E K L A R A S I   V A R I A B E L ----
AEN PIN 13          ' pwm motor kanan
MOTOR_A PIN 12      ' arah motor kanan
BEN PIN 11          ' pwm motor kiri
MOTOR_B PIN 10      ' arah motor kiri
i VAR Byte          ' variabel bebas
A VAR Byte          ' variabel bebas
```

Gambar II.16. Tampilan bagian deklarasi variabel

Setelah menentukan variabal dan PIN yang digunakan, selanjutnya membuat program utama. Pada bagian program utama bisa melakukan dua mode, yaitu mode pengetikan langsung atau mode pemanggilan prosedur. Mode pengetikan langsung akan efektif jika program tidak terlalu banyak dan hanya untuk menangani kasus yang sederhana. Tetapi jika program sudah mulai banyak, rumit dan lebih dari satu slot, maka sebaiknya program utama memanggil prosedur. Pemanggilan prosedur akan mempermudah urutan/alur program, lebih terkendali dan mudah dalam pemeriksaan.

2.2.2.3 Bagian Program Utama

Berikut ini contoh program utama yang memanggil prosedur MAJU. Listing programnya dapat dilihat pada gambar berikut.

```
'------ P R O G R A M    U T A M A -------
DO                      ' awal program
GOSUB MAJU              ' memanggil prosedur
LOOP                    ' akhir perulangan
END                     ' akhir program utama
```

Gambar II.17. Tampilan bagian program utama yang memanggil prosedur

2.2.2.4 Bagian Prosedur

Berikut adalah blok prosedur MAJU yang dipanggil oleh program utama.

```
'---- P R O S E D U R  -  P R O S E D U R ---
MAJU:                   ' nama prosedur maju
HIGH MOTOR_A            ' menset motor kanan MAJU
HIGH MOTOR_B            ' menset motor kiri MAJU
FOR i = 1 TO 50         ' perulangan
PWM AEN,100,40          ' pengaturan kecepatan dengan PWM untuk motor kanan
PWM BEN,100,40          ' pengaturan kecepatan dengan PWM untuk motor kiri
NEXT                    ' lanjutkan perulangan
PAUSE 20                ' berhenti 20 mili detik
RETURN                  ' kembali dari prosedur
```

Gambar II.18. Tampilan bagian prosedur MAJU

Sebuah prosedur harus mempunyai nama prosedur yang disimpan dibagian paling atas prosedur itu sendiri, serta harus diakhiri dengan Return supaya kembali lagi ke program utama dan melanjutkan kembali urutan program berikutnya. Sebuah prosedur mempunyai fungsi khusus, misal untuk prosedur maju ini hanya bertugas untuk menjalankan robot dengan arah maju.

2.2.2.5 Memeriksa Sintaks Program

Sangat penting untuk memeriksa sintaks program, hal ini kita lakukan untuk memastikan semua sintaks sudah benar. Untuk

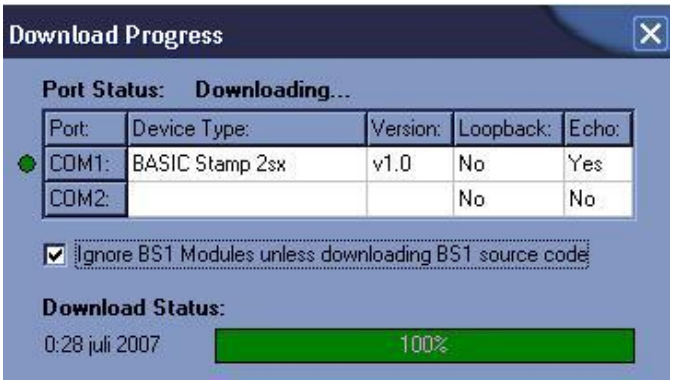
memeriksa sintaks ini bisa pilih menu RUN, Cek Syntax atau kombinasi tombol CTRL+T. Berikut ini adalah tampilan jika listing program yang kita buat sudah benar.



Gambar II.19. Hasil pemeriksaan sintaks yang sukses (tokenize successful)

2.2.3 Menjalankan Program

Setelah program selesai, program siap di download ke modul basic stamp. Cara untuk menjalankan program dapat memilih menu RUN atau kombinasi tombol CTR+R. Berikut adalah tampilan jika pen*downlod*an program sukses.



Gambar II.20. Tampilan jika pendownloadan program sukses