BAB II
DASAR TEORI

2.1 Sejarah Robot

Robot adalah sebuah alat mekanik yang dapat melakukan tugas fisik, baik menggunakan pengawasan dan kontrol manusia, ataupun menggunakan program yang telah didefinisikan terlebih dulu (kecerdasan buatan).

Kata robot sendiri, diperkenalkan oleh seorang penulis sains-fiksi berkebangsaan Ceko (Czech), Karel Capek (9 Januari 1890-25 Desember 1938) pada tahun 1920an. Robot sendiri, menurut Karel Capek, bukanlah pengertian "robot" yang seperti kita kenal sekarang ini. Robot menurut dia adalah sesuatu yang mengarah kepada "monoton", dan "pekerja". Karel Capek memperkenalkan istilah robot (robota, bahasa Ceko-Ensiklopedia Indonesia) dalam sebuah karya sandiwaranya, Rossum's Universal Robot.

Penciptaan robot yang sesunguhnya (bukan robot dalam sandiwara Karel Capek), bermula dari keinginan manusia untuk membuat tiruan binatang atau manusia itu sendiri. Namun fasilitas pengetahuan pada masa itu nampaknya menjadi kendala terbesar dalam terwujudnya proyek tersebut. Akhirnya, dibuatlah robot sederhana beroda yang digunakan untuk keperluan navigasi, pengamatan tingkah laku, sampai dengan perencanaan jalur.

2.2 Perangkat Keras (*Hardware*)

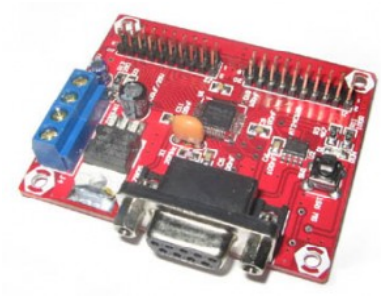
2.2.1 Mikrokontroler (Basic Stamp 2p40)

Basic stamp adalah mikrokontroler yang dikembangkan oleh *Parallax Inc* yang diprogram menggunakan *format* bahasa pemrograman *basic*. Program yang telah dibuat akan diunduh melalui *port* serial. Mikrokontroler *basic stamp* membutuhkan *power supply* saat mengunduh program dan program yang sudah diunduh tidak akan hilang meskipun baterai atau *power supply* dilepas.

Kode pemograman *basic* disimpan di dalam EEPROM serial pada *board basic stamp*. EEPROM menyediakan penyimpanan yang sulit diubah dan menjaga

memory saat kehilangan *power*. EEPROM digunakan dalam *basic stamp* yang dijamin menyimpan data 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 versi, yaitu *basic stamp* 1, *basic stamp* 2, *basic stamp* 1e, *basic stamp* 2P, *basic stamp* 2Pe dan *basic stamp* 2sx. 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 dipakai adalah *basic stamp* 2P40 yang mempunyai 32 pin I/O. Berikut ini adalah tampilan *basic stamp* 2P40.

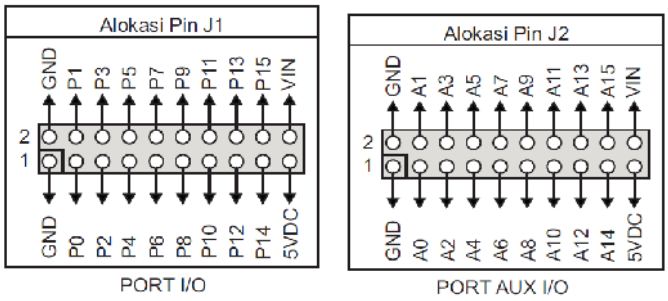


Gambar 2.1 Modul basic stamp 2p40 (BS2P40)

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* 2P40.



Gambar 2.2 Alokasi pin basic stamp

2.2.2 Sensor Garis

Sensor garis dalam seuah robot line follower ibarat mata yang dapat menentukan arah geraknya. Sensor garis adalah sensor yang berfungsi mendateksi garis yang berwarna hitam atau putih.

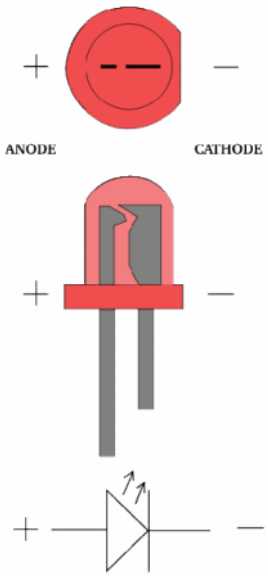
Sensor garis terdiri dari komponen yang berfungsi sebagai pemancar cahaya yaitu *infrared*, LED *superbright*, dll dan komponen yang berfungsi penerima cahaya yaitu *phototransistor*, *photodiode*, LDR (*light defendant Resistor*). Dengan memanfatkan fitur ADC (*Analog to Digital Converter*) yang terdapat pada mikrokontroler sebagai pembanding tegangan, yang nantinya akan menghasilkan *output* digital dari sensor.

2.2.2.1 Light Emitting Diode (LED)

LED (*Light Emitting Diode*) merupakan komponen yang mampu mengeluarkan emisi cahaya. LED merupakan produk temuan lain setelah *diode*. Strukturnya juga sama dengan *diode*, tetapi belakangan ditemukan bahwa elektron yang menerjang sambungan P-N juga melepaskan energi berupa energy panas dan energy cahaya. LED dibuat agar lebih efisien jika mengeluarkan cahaya. Untuk mendapatkan emisi cahaya pada semikonduktor, doping yang dipakai adalah

gallium, arsenic dan phosphorus. Jenis doping yang berbeda menghasilkan warna cahaya yang berbeda pula.

Saat ini, warna-warna cahaya LED yang banyak tersedia adalah warna merah, kuning dan hijau. LED berwarna biru sangat langka. Pada dasarnya, semua warna bisa dihasilkan, tetapi itu akan menjadi sangat mahal dan tidak efisien. Dalam memilih LED, selain warna, perlu juga diperhatikan tegangan kerja, asrus maksimum dan dispasi daya-nya. Rumah (*chasing*) LED dan bentuknya juga bermacam-macam, ada yang persegi empat, bulat dan lonjong. Berikut ini adalah tampilan LED.

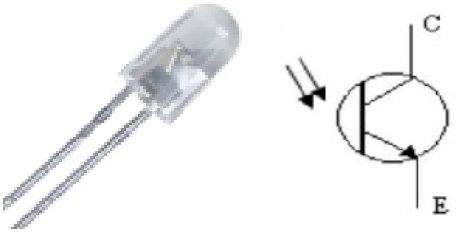


Gambar 2.3 Bentuk fisik LED dan simbol LED

2.2.2.2 Phototransistor

Phototansistor merupakan suatu jenis *transistor* yang sangat peka terhadap 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*.

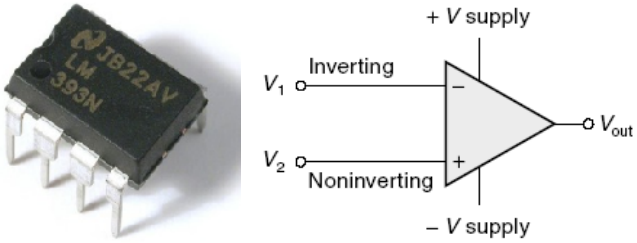
Cara kerja *phototransistor* yaitu ketika *basis* menangkap cahaya maka *collector* akan terhubung dengan *emitter* dalam hal ini *transistor* bekerja. Berikut ini adalah simbol *phototransistor*.



Gambar 2.4 Bentuk fisik phototransistor dan simbol phototransistor

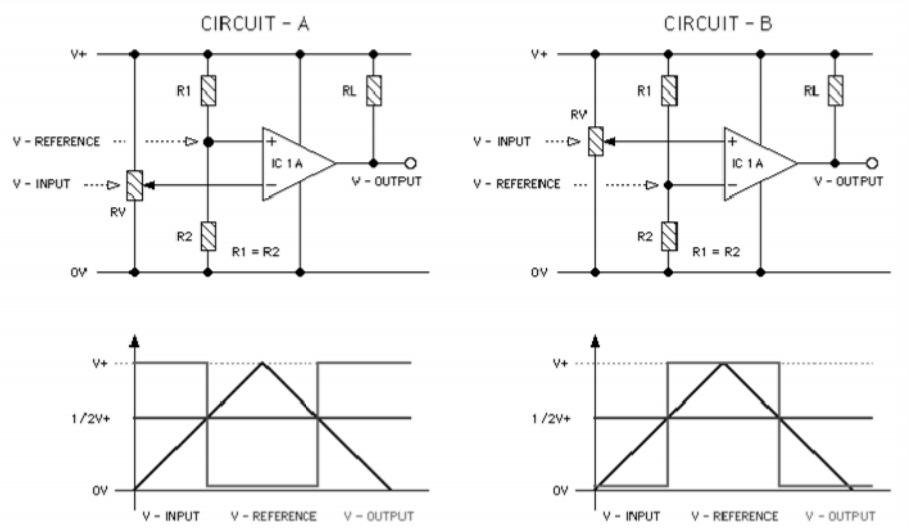
2.2.3 Komparator

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. Berikut ini adalah tampilan IC LM 393.



Gambar 2.5 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 2.6 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 2.6 bagian bawah kiri dan bawah kanan.

2.2.4 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. Bentuk fisik motor DC dapat dilihat pada gambar 2.7.



Gambar 2.7 Bentuk fisik Motor DC

Motor 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.

2.2.5 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 2.8 Bentuk fisik IC driver motor L298

2.2.6 Double Gearbox

Gearbox berfungsi sebagai pereduksi putaran atau *rotate per minutes* (rpm) dari motor dan juga berfungsi menghasilkan kekuatan putar (*torsi*) yang lebih besar. Diketahui bahwa sebuah robot tidak hanya memerlukan kecepatan motor yang tinggi, tetapi juga torsi yang besar. Untuk model robot tertentu, kecepatan motor bahkan tidak diperlukan. Sebagai contoh robot yang akan di buat robot pengangkut *box* berdasarkan perbedaan warna yang membutuhkan *torsi* yang besar untuk mengangkut *box* dengan beban ± 150 gram. Berikut ini adalah tampilan *double gearbox.*



Gambar 2.9 Bentuk fisik double gearbox

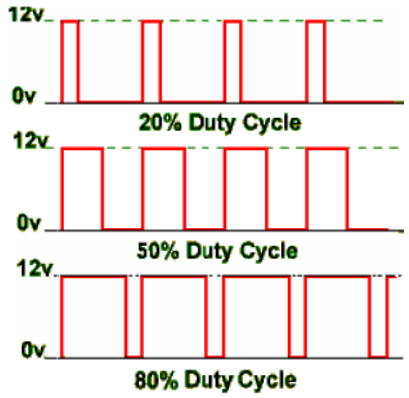
2.2.7 Pulse Width Modulation (PWM)

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*). Ilustrasi sebuah prosentase PWM seperti terdapat pada gambar berikut.



Gambar 2.10 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.

2.2.8 Motor Servo

Sebuah motor *servo* adalah alat yang dapat mengendalikan posisi, dapat membelokkan dan menjaga suatu posisi berdasar penerimaan pada suatu signal elektronik itu. Pada perancangan gripper digunakan 2 buah motor *servo* pada bagian kanan dan kiri untuk mencapit sebuah *box*. Pada saat robot dijalankan motor *servo* akan membuka bagian kiri dan bagian kanan, ketika tidak ada *box* maka *servo* akan menutup bagian kiri dan bagian kanan sehingga mencapit *box*. Berikut ini adalah tampilan motor *servo.*

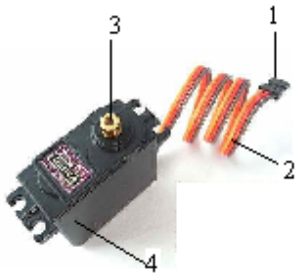


Gambar 2.11 Bentuk fisik motor servo HiTec HS-5645MG

HS-5645MG adalah motor *servo* yang dibuat oleh *HiTec*. Berikut merupakan spesifikasi dari motor *servo* tersebut :

1. Jenis motor adalah 3 *pole ferrite*
2. Jenis b*earing* adalah *dual ball bearing*
3. Torsi 10.3 kg/ cm di 4,8 V
4. Kecepatan 0,23 detik/ 60º di 4,8 V
5. Ukuran 40.6 X 19.8 X 37.8 mm
6. Berat 60 g

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 *servo* merupakan alat untuk mengubah energi listrik menjadi energi mekanik, maka magnet *permanent* motor *servo* yang mengubah energi listrik ke dalam energi mekanik melalui interaksi dari dua medan magnet. Berikut ini adalah tampilan motor *servo.*



Gambar 2.12 Bentuk fisik motor 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, 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 atau posisi *servo*.

2.2.9 Sensor Ultrasonik

Sensor ultrasonik (sensor jarak) adalah salah satu elemen penting pada sebuah robot yang sedang bergerak. Hal ini dibutuhkan oleh robot untuk mengetahui posisi robot terhadap objek-objek tertentu seperti *box*. Dalam jarak antara 3 cm hingga 3 meter, ultrasonik adalah media yang sesuai dengan perancangan robot ini. Contoh penerapannya di simpan di depan robot, agar robot mengetahui apakah di depannya terdapat *box* atau tidak ada *box*. Berikut ini adalah tampilan fisik modul sensor ultrasonik.

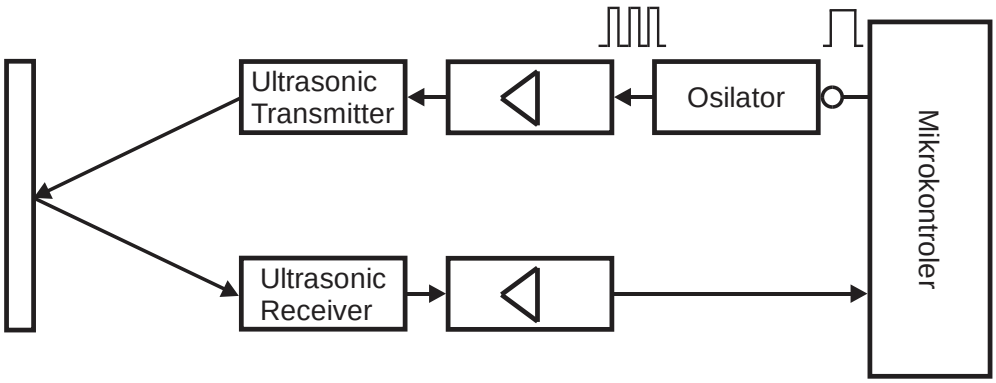


Gambar 2.13 Bentuk fisik modul sensor ultrasonik

Proses pengukuran dilakukan dengan menembakkan sinyal ultrasonic dan menghitung kapan sinyal tersebut diterima kembali oleh sensor. Sensor ultrasonik

yang sering digunakan dipasaran adalah sensor yang memiliki respon frekuensi 40kHz. Oleh karena itu, untuk memancarkan sinyal dengan respon maksimum, dibutuhkan gelombang dengan frekuensi kHz yang dibangkitkan dengan osilator.

Pada Gambar 2.14, tampak bahwa osilator dibangkitkan oleh trigger dari mikrokontroler dan dikuatkan oleh bagian penguat sebelum dipancarkan oleh pemancar ultrasonik. Sinyal ultrasonik akan terpancar setelah getaran ke-8 dari osilator dilakukan. Berikut ini adalah tampilan sistem pengukuran jarak dengan ultrasonik.



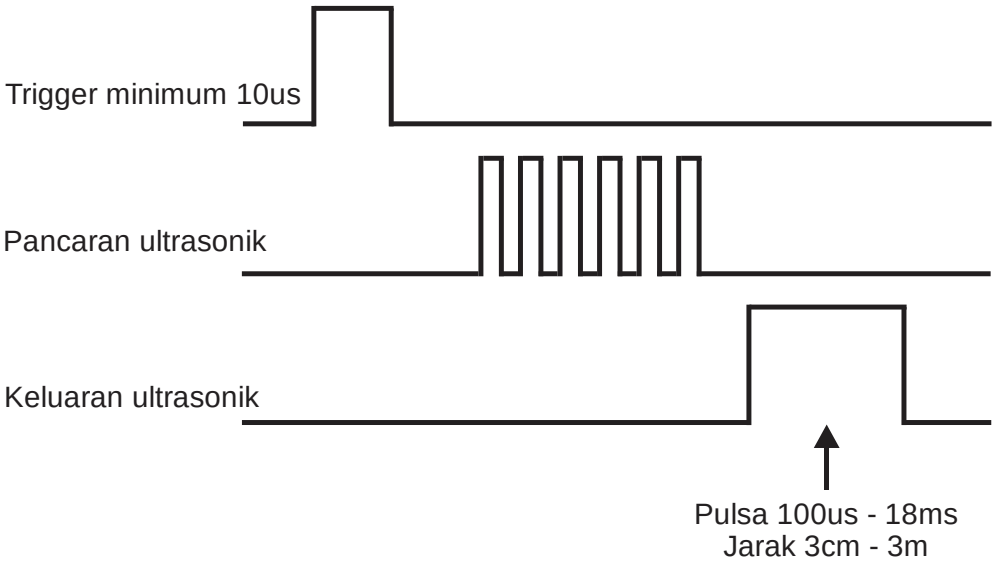
Gambar 2.14 Sistem pengukuran jarak dengan ultrasonik

Penerima ultrasonik akan menerima pantulan dari objek dan mengubahnya menjadi getaran-getaran listrik. Namun, getaran tersebut masih terlalu lemah sehingga perlu dikuatkan oleh sebuah penguat sebelum masuk ke mikrokontroler.

Mikrokontroler akan berhenti melakukan perhitungan saat sinyal ultrasonik diterima kembali. Perhitungan waktu dari saat sinyal ultrasonik pertama kali dipancarkan hingga diterima telah diperoleh. Jarak yang ditempuh oleh rambatan gelombang ultrasonik mulai dari dipancarkan hingga memantul pada objek dan kembali diterima akan diperoleh dengan mengalirkan kecepatan rambatan suara dengan waktu yang diperoleh.

Sedangkan jarak antara sensor dengan objek adalah $^1/_2$ kali jarak rambatan gelombang ultrasonik. Untuk mempermudah kinerja otak dari robot dalam mengukur jarak, maka proses perhitungan waktu dan konversi ke dalam jarak tadi

dilakukan oleh mikrokontroler tersendiri yang dikemas dalam suatu modul yang di sebut sensor ultrasonik. Berikut ini adalah timing diagram sensor ultrasonik.



Gambar 2.15 Timing diagram sensor ultrasonik

Dengan modul sensor ultrasonik, mikrokontroler yang menjadi otak robot hanya perlu mengirimkan sebuah trigger dan mikrokontroler pada sensor ultrasonik yang akan membangkitkan sinyal ultrasonik serta melakukan perhitungan seperti yang telah dijelaskan sebelumnya.

Hasil perhitungan akan dikonversi menjadi sebuah pulsa, di mana lebar pulsa akan menentukan jarak yang diperoleh. Otak robot hanya perlu menghitung lebar pulsa tersebut untuk memperoleh nilai jarak antara sensor dengan objek.

2.2.10 Sensor Warna

Salah satu identifikasi objek yang sering digunakan pada aplikasi robotik adalah pengenalan warna. Dengan teknologi infrared, proses pengenalan warna dapat dilakukan dengan memperhitungkan kuat lemahnya pantulan cahaya.

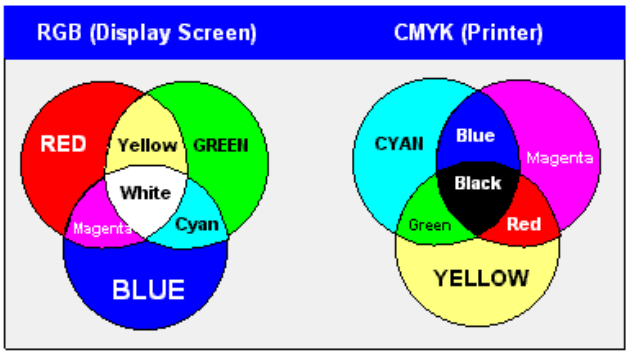
Sensor warna TCS 3200 adalah sensor warna buatan TAOS Parralax. TCS 3200 adalah produk penyempurnaan dari produk sebelumnya yaitu TCS 230. Perbedaan antara TCS 3200 dan TCS 230 adalah konsumsi arusnya.

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 2.16 Modul sensor warna TCS3200

RGB (*Red Green Blue*) adalah ruang warna asli yang digunakan pada system grafik computer yang merupakan system warna untuk menagkap gambar. RGB digunakan karena mata manusia sensitif terhadap warna merah, hijau dan biru. Gambar berikut menampilkan perbedaan antara warna RGB dan CMYK.



Gambar 2.17 Perbandingan warna RGB dan CMYK

Dua buah warna disebut komplementer, jika campuran keduanya dengan perbandingan yang tepat menghasilkan warna putih. Misalnya magenta jika dicampur dengan perbandingan yang tepat dengan hijau menghasilkan putih. Oleh karena itu magenta disebut merupakan komplementer dari hijau.

Tabel 2.1 Nilai warna RGB refrensi dari custom color pada paint O/S Windows

No	Warna	R	G	B
1	Hitam	0	0	0
2	Merah	255	0	0
3	Hijau	0	255	0
4	Biru	0	0	255
5	Ungu/ Magenta	128	0	128
6	Biru Muda/ Cyan	0	128	128
7	Kuning	255	255	0
8	Putih	255	255	255

2.2.11 Catu Daya

Catu daya memegang peranan yang sangat penting dalam hal perancangan sebuah robot. Pemilihan catu daya yang tepat akan menghasilkan robot yang 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: 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 perancangan robot ini berjenis *lithium polymer* (LiPo). Baterai ini dapat diisi ulang (*rechargeable*). Baterai yang digunakan memiliki tegangan 7,4 Volt dan arus sebesar 2200 mAh dengan 2 *cell* di dalamnya. *Cell* merupakan teknologi konversi energi elektrokimia yang mampu mengubah senyawa hidrogen dan oksigen menjadi air, dan dalam prosesnya menghasilkan listrik. Pemakaian baterai jenis ini harus dihentikan atau dilepas

jika tegangan baterai turun mendekati batas tegangan 7,4 Volt, sehingga harus diisi ulang agar melebihi tegangan 7,4 Volt. Berikut ini adalah contoh sebuah baterai *lithium polymer*.



Gambar 2.18 Baterai lithium polymer

Selain jenis baterai *lithium polymer* (LiPo), masih banyak lagi jenis baterai yang tersedia di pasaran dengan spesifikasi yang beragam dan dapat digunakan untuk catu daya. Diantaranya baterai Ni-Cd, *Alkaline*, *Lead Acid* dan sebagainya, yang masing-masing mempunyai kelebihan dan kekurangannya.

2.3 Perangkat Lunak (Software)

2.3.1 Pengenalan Basic Stamp Editor

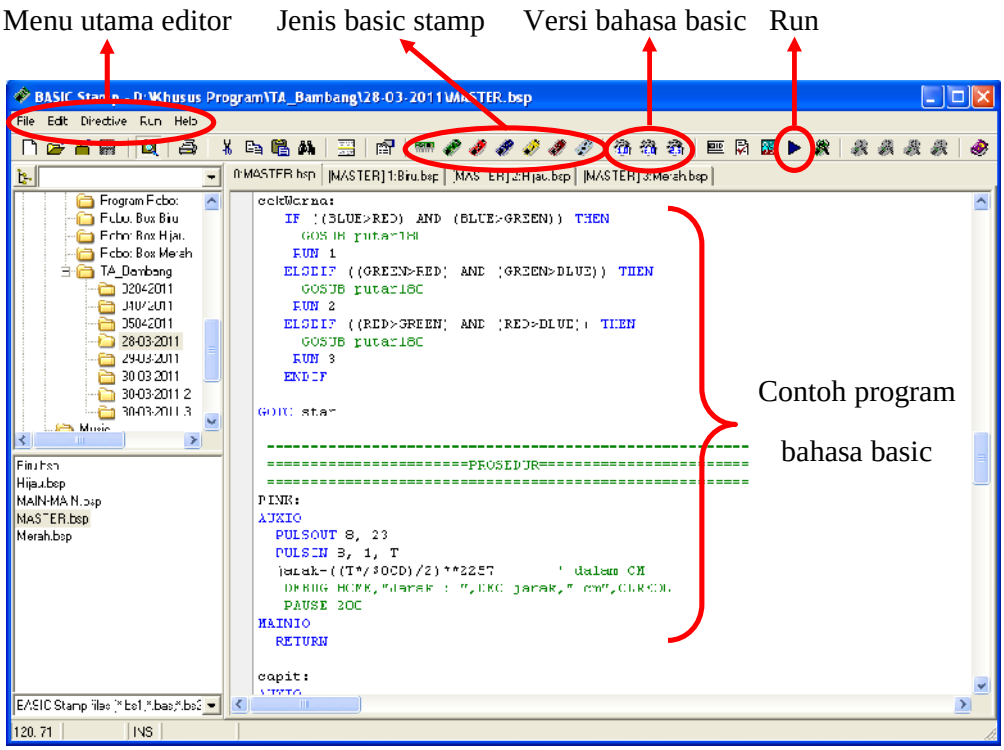
Perangkat lunak merupakan faktor penting dalam tahap perancangan robot. Perangkat lunak ini merupakan algoritma gerak dan tugas robot dalam bentuk listing program yang ditanamkan kedalam mikrokontroler. Program dapat bermacam - macam bentuk versi dan bahasa pemrogramannya, sesuai dengan spesifikasi dari 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 2.2 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



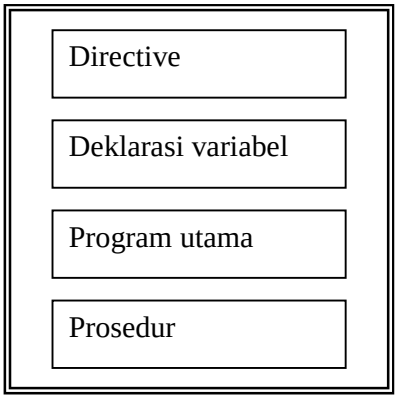
Gambar 2.19 Tampilan basic stamp editor

2.3.2 Memulai Pemrograman di Basic Stamp Editor

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

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

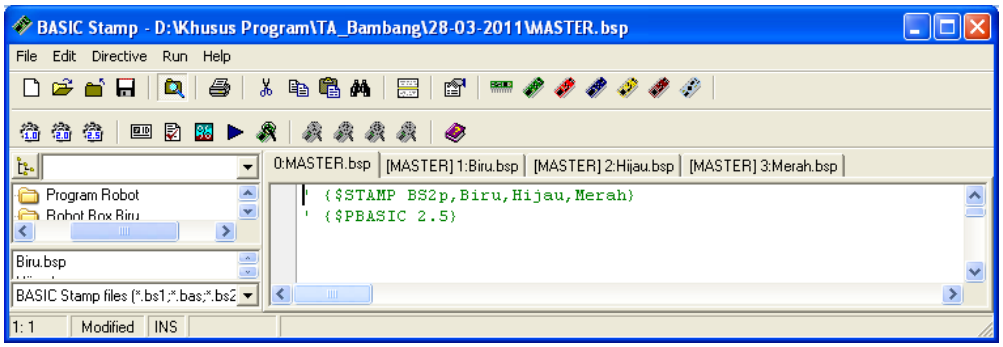
Pemrograman dalam basic stamp editor, secara blok dibagi menjadi empat bagian penting.



Gambar 2.20 Urutan bagian dari program dalam basic stamp

2.3.3 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 :



Gambar 2.21 Tampilan bagian directive

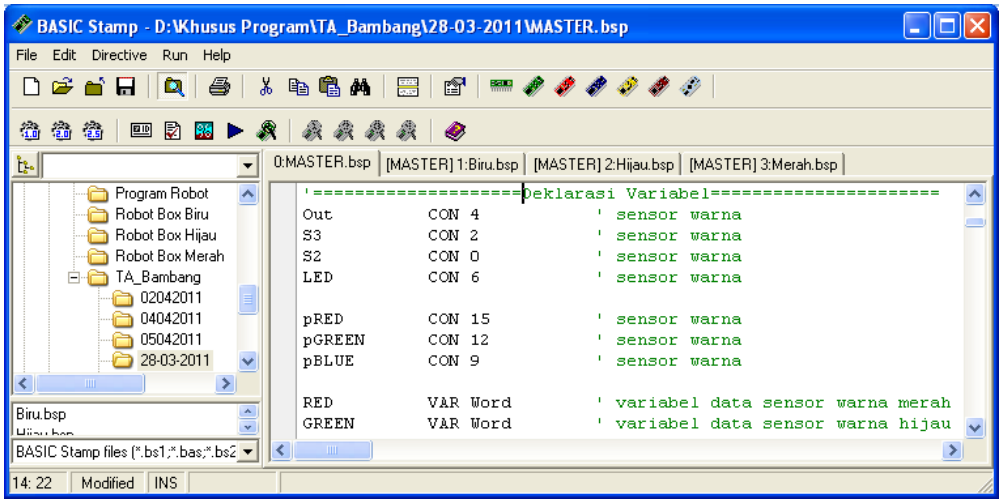
2.3.4 Menentukan variabel

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.

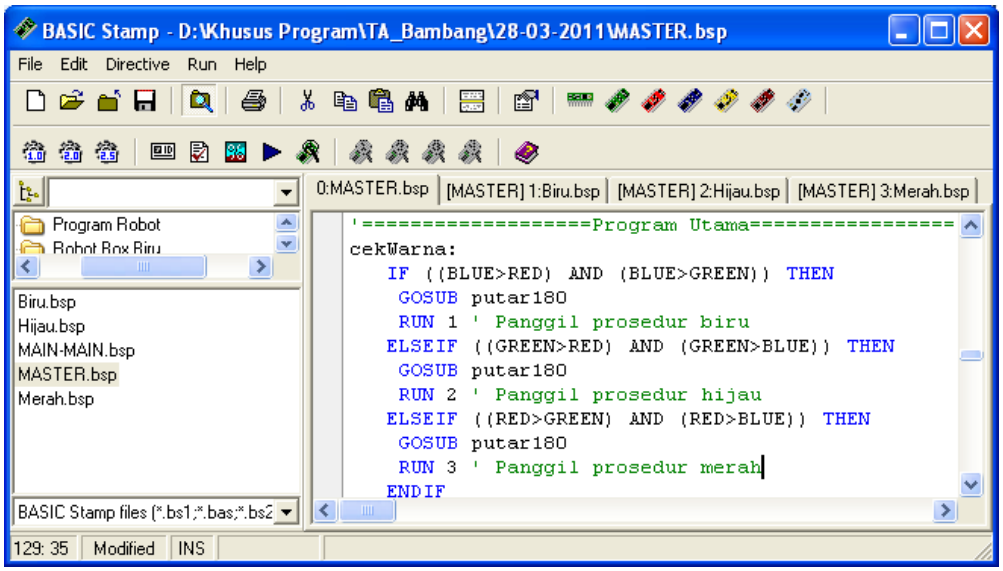


Gambar 2.22 Tampilan bagian deklarasi variable

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.

2.3.5 Bagian Program Utama

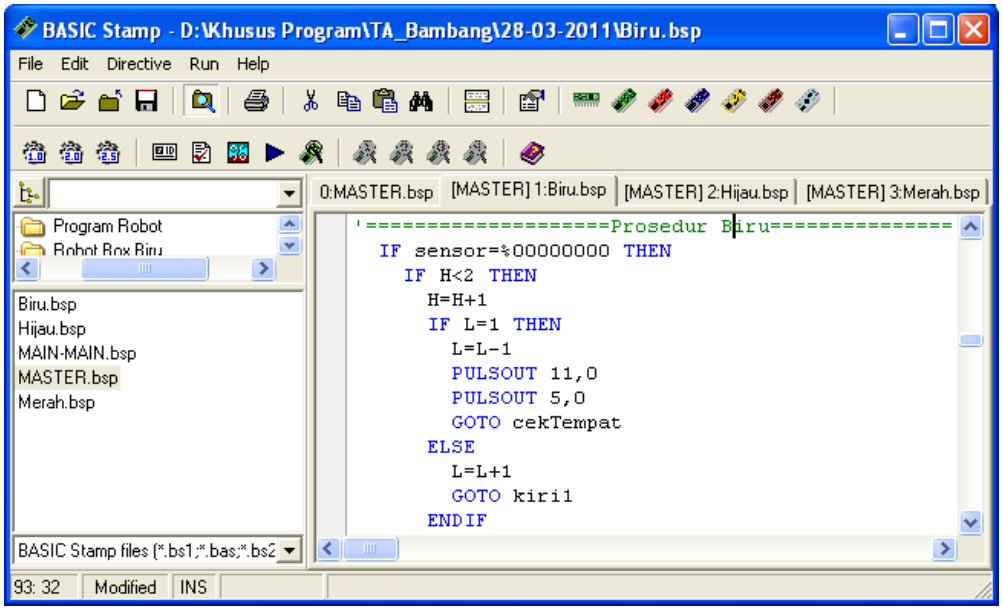
Berikut ini contoh program utama yang memanggil prosedur warna biru, warna hijau dan warna merah. Listing programnya dapat dilihat pada gambar berikut.



Gambar 2.23 Tampilan bagian program utama yang memanggil prosedur

2.3.6 Bagian Prosedur

Berikut ini adalah contoh blok prosedur warna biru yang dipanggil oleh program utama, dapat dilihat pada gambar berikut.



Gambar 2.24 Tampilan bagian prosedur biru

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.

2.3.7 Memeriksa Kebenaran Sintaks Program

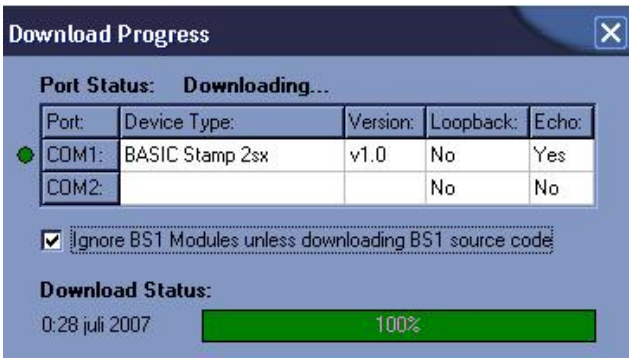
Sangat penting untuk memeriksa sintaks program, hal ini kita lakukan untuk memastikan semua sintaks sudah benar. Untuk memeriksa sintaks ini bisa pilih menu RUN, Cek Syntax atau kombinasi tombol CTRL+T. Berikut ini adalah tampilan jika listing program yang kita buat sudah benar.



Gambar 2.25 Hasil pemeriksaan sintaks yang sukses (tokenize successful)

2.3.8 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 2.26 Tampilan jika pendownloadan program sukses