BAB II

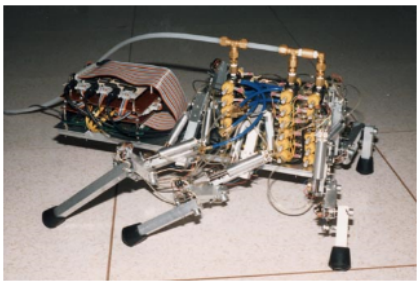
DASAR TEORI

2.1. Sejarah Robot [12]

Robot adalah sebuah alat mekanik yang dapat melakukan tugas fisik, baik menggunakan pengawasan dan kontrol manusia, atau pun 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 lebih 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. Berkembanglah robot dengan konsep heksapoda (konsep kaki enam) dan beberapa *platform* kaki banyak lainya. Pada pada masa itu, serangga menjadi inpirasi dalam pembuatan robot.



Gambar 2.1 Contoh Robot Heksapoda

2.2. Perkembangan Robot [12]

Adapun contoh perkembangan hebat robot belakangan ini yaitu robot medis dengan dua perusahaan khusus yaitu Computer Motion dan Intuitive Surgical, yang menerima pengesahan pengaturan di Amerika Utara, Eropa dan Asia atas robot-robotnya untuk digunakan dalam prosedur pembedahan minimal. Otomasi laboratorium juga merupakan area yang berkembang. Di sini, robot *benchtop* digunakan untuk memindahkan sampel biologis atau kimiawi antar perangkat seperti inkubator, berupa pemegang dan pembaca cairan. Tempat lain dimana robot disukai untuk menggantikan pekerjaan manusia adalah dalam eksplorasi laut dalam dan eksplorasi antariksa. Untuk tugas-tugas ini, bentuk tubuh artropoda umumnya disukai. Mark W. Tilden dahulunya spesialis Laboratorium Nasional Los Alamos membuat robot murah dengan kaki bengkok tetapi tidak menyambung, sementara orang lain mencoba membuat kaki kepiting yang dapat bergerak dan tersambung penuh. Berdasarkan perkembangannya, saat ini robot memiliki berbagai macam konstruksi. Diantaranya adalah:

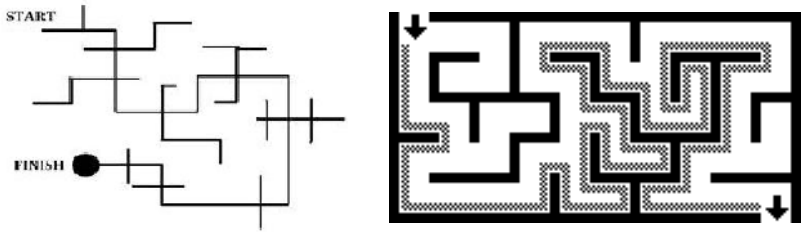
1. Robot Mobile (bergerak)
2. Robot Manipulator (tangan)
3. Robot Humanoid
4. Flying Robot
5. Robot Berkaki
6. Robot jaringan
7. Robot Animalia

2.3. Robot Maze Solving

Robot *maze solving* adalah *mobile robot* yang dapat menelusuri jalur yang rumit, mengingat jalur tersebut dan mengambil jalur terpendek. Jalur yang dilewati dapat berupa garis (*line*) atau dinding (*wall*).

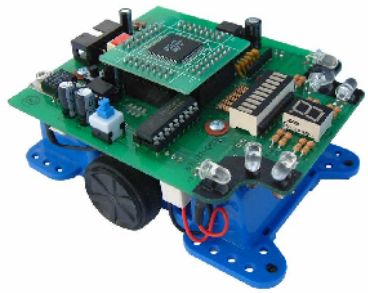
Robot *maze solving* yang pertama dibuat oleh seorang yang bernama Claude Elwood Shannon pada tahun 1950. Pada saat itu robot *maze solving* dikendalikan oleh suatu rangkaian relay, robot tersebut menyerupai tikus yang berukuran sama dengan tikus yang sesungguhnya. Robot tersebut mengembara pada lorong (

dinding) rumit dengan 25 penyiku. Lorong tersebut bisa di ubah sesuka hati dan robot tersebut kemudian memeriksa secara menyeluruh jalan yang harus ditempuh untuk menemukan titik tujuan. Setelah melalui lorong tersebut, robot bisa ditempatkan di manapun dan akan bergerak secara langsung sampai tujuan. Hal ini nampak sebagai alat yang bisa belajar untuk pertama kalinya.



Gambar 2.2 Jalur Berupa Garis dan Jalur Berupa Dinding

Robot *maze solving* pada saat ini tidak hanya dikenal di kalangan ilmuan tetapi telah di kenal di kalangan siswa SD, SMP, sampai universitas bahkan saat ini sering dilakukan perlombaan robot *maze solving*. Contoh *moment* perlombaan yaitu ROBOTIC COMPETITION STMIK LIKMI tingkat SMA pada tanggal 15 April 2010. Berikut contoh robot *maze solving*:



Gambar 2.3 Robot Maze Solving

2.4. Perangkat Keras (Hardware)

2.5.1. Mikrokontroler (ATmega32) [1]

Mikrokontroler adalah piranti elektronik berupa IC (*Integrated Circuit*) yang memiliki kemampuan manipulasi data (informasi) berdasarkan suatu urutan instruksi (program). Dalam sebuah struktur

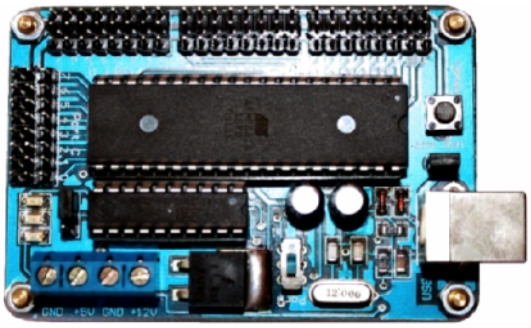
mikrokontroller akan kita temukan juga komponen-komponen seperti: *processor*, *memory*, *clock*, dll.

Salah satu arsitektur mikrokontroler yang terdapat di pasaran adalah jenis AVR (*Advanced Virtual RISC*). Arsitektur mikrokontroler jenis AVR ini pertama kali dikembangkan pada tahun 1996 oleh dua orang mahasiswa *Norwegian Institute of Technology* yaitu Alf-Egil Bogen dan Vegard Wollan.

Dalam perkembangannya, AVR dibagi menjadi beberapa varian yaitu AT90Sxx, ATmega, dan AT86RFxx. Pada dasarnya yang membedakan masing-masing varian adalah kapasitas memori dan beberapa fitur tambahan saja.

Pemrograman mikrokontroler AVR dapat menggunakan *low level language* (*assembly*) dan high *level language* (C, Basic, Pascal, JAVA, dll) tergantung *compiler* yang digunakan.

Salah satu yang banyak dijumpai di pasaran adalah AVR tipe ATmega, yang tediri dari beberapa versi, yaitu ATmega8535, ATmega16, ATmega162, ATmega32, ATmega324P, ATmega644, ATmega644P dan ATmega128. Mikrokontroler yang digunakan adalah AVR ATmega32. Berikut ini adalah tampilan ATmega32.



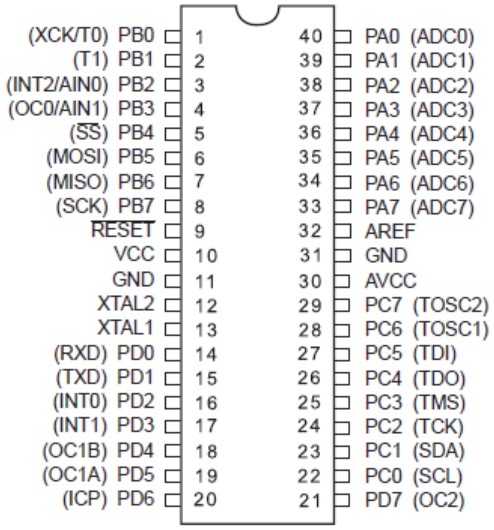
Gambar 2.4 Modul AVR ATmega32

Fitur-fitur yang dimiliki ATmega32 sebagai berikut:

1. Frekuensi clock maksimum 16 MHz
2. Jalur I/O 32 buah, yang terbagi dalam PortA, PortB, PortC dan PortD

3. Analog to Digital Converter 10 bit sebanyak 8 input, 4 chanel PWM
4. Timer/Counter sebanyak 3 buah
5. CPU 8 bit yang terdiri dari 32 register
6. Watchdog Timer dengan osilator internal
7. SRAM sebesar 2K Byte
8. Memori Flash sebesar 32K Byte dengan kemampuan *read while write*
9. Interrupt internal maupun eksternal
10. Port komunikasi SPI
11. EEPROM sebesar 512 byte yang dapat diprogram saat operasi
12. Analog Comparator
13. Komunikasi serial standar USART dengan kecepatan maksimal 2,5 Mbps

2.5.2. Konfigurasi Pin AVR ATmega32 [1]



Gambar 2.5 Konfigurasi Pin ATmega32

Konfigurasi *pin* pada mikrokontroler ATmega32 dapat dilihat pada gambar 2.5. Dari gambar tersebut dapat terlihat jumlah *pin* ATmega32 adalah 40 *pin* yang memiliki fungsi yang berbeda-beda yaitu:

1. Vcc merupakan *pin* yang berfungsi sebagai masukan catu daya.
2. GND merupakan *pin* ground.

3. *Port* A(PA0-PA7) merupakan *pin input/output* dua arah dan *pin* masukan ADC.
4. *Port* B(PB0-PB7) merupakan *pin input/output* dua arah dan *pin* fungsi khusus, seperti yang dapat dilihat pada tabel 2.1.

Tabel 2.1 Fungsi Khusus Port B

Pin	**Fungsi Khusus**
PB0	T0 T1 (*Timer/Counter0 External Counter input)* XCK (USART *External Clock Input/Output*)
PB1	T1 (*Timer/Counter0 External Counter Input*)
PB2	AIN0 (*Analog Comparator Positive Input*) INT2 (*External Interrupt 2 Input*)
PB3	AIN0 (*Analog Comparator Positive Input*) OC0 (*Timer/Counter0 Output Compare Match Output*)
PB4	SS (SPI *Slave Select Input*)
PB5	MOSI (SPI *Bus Master Output/Slave Input*)
PB6	MISO (SPI *Bus Master Input/Slave Output*)
PB7	SCK (SPI *Bus Serial Clock*)

5. *Port* B(PB0-PB7) merupakan *pin input/output* dua arah dan *pin* fungsi khusus, seperti yang dapat dilihat pada tabel 2.2.

Tabel 2.2 Fungsi Khusus Port C

Pin	**Fungsi Khusus**
PC0	SCL (*Two-wire Serial Bus Clock Line*)
PC1	SDA (*Two-wire Serial Bus Data Input/Output Line*)
PC2	TCK (JTAG *Test Clock*)
PC3	TMS (JTAG *Test Mode Select*)
PC4	TDO (JTAG *Test Data Out*)
PC5	TDI (JTAG *Test Data In*)
PC6	TOSC1 (*Timer Oscillator* PIN1)
PC7	TOSC2 (*Timer Oscillator* PIN2)

6. *Port* B(PB0-PB7) merupakan *pin input/output* dua arah dan *pin* fungsi khusus, seperti yang dapat dilihat pada tabel 2.3.

Tabel 2.3 Fungsi Khusus Port D

Pin	Fungsi Khusus
PD0	RXD (USART *Input Pin*)
PD1	TXD (USART *Output Pin*)
PD2	INT0 (*External / Interupt* 0 Input)
PD3	INT1 (*External / Interupt* 1 Input)
PD4	OC1B OC1A (*Timer / Counter2 Output Compare B Match Output*)
PD5	OC1A (*Timer / Counter2 Output Compare A Match Output*)
PD6	ICP (*Timer / Counter1 Input Capture Pin*)
PD7	OC2 (*Timer / Counter2 Output Compare Match Output*)

7. RESET merupakan *pin* yang digunakan untuk me-*reset* mikrokontroler.
8. XTAL1 dan XTAL2 merupakan *pin* masukan *clock* eksternal.
9. AVCC merupakan *pin* masukan tegangan ADC.
10. AREF merupakan *pin* masukan referensi ADC.

2.5.3. Sensor Garis

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

Sensor garis terdiri dari komponen yang berfungsi sebagai pemancar cahaya yaitu infrared, led superbright, dll. 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.

2.5.2.1. Phototransistor [3]

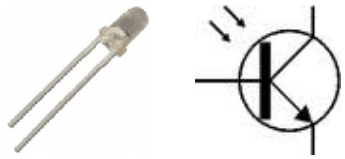
Phototransistor adalah sebuah benda padat pendeteksi cahaya yang memiliki gain internal. Hal ini yang membuat *phototransistor* memiliki sensitivitas yang lebih tinggi dibandingkan *photodioda*,

dalam ukuran yang sama. Cara kerja phototransistor yaitu ketika basis menangkap cahaya maka *collector* akan terhubung dengan emitter dalam hal ini transistor bekerja.

Phototransistor memiliki karakteristik:

1. Pendeteksi jarak dekat Infra merah.
2. Bisa dikuatkan sampai 100 sampai 1500.
3. Respon waktu cukup cepat.
4. Bisa digunakan dalam jarak lebar.
5. Bisa dipasangkan dengan (hampir) semua penghasil cahaya atau cahaya yang dekat dengan inframerah, seperti Neon, Fluorescent, lampu bohlam, cahaya laser dan api.
6. Mempunyai karakteristik seperti transistor, kecuali bagian basis digantikan oleh besar cahaya yang diterima.

Bentuk fisik dan simbol phototransistor dapat dilihat pada gambar 2.6.



Gambar 2.6 Bentuk Fisik dan Simbol Phototransistor

2.5.4. Aktuator (Penggerak Robot)

2.5.2.1. Motor DC

Motor DC merupakan perangkat yang berfungsi merubah besaran listrik menjadi besaran mekanik. Prinsip kerja motor didasarkan pada gaya elektromagnetik. Motor DC bekerja bila mendapatkan tegangan searah yang cukup pada kedua kutubnya. Tegangan ini akan menimbulkan induksi elektromagnetik yang menyebabkan motor berputar. 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 diberkan juga dirubah. Bentuk fisik motor DC dapat dilihat pada gambar 2.7.



Gambar 2.7 Motor DC

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

2.5.2.2. Driver Motor

Driver motor adalah sirkuit elektronika yang memungkinkan tegangan dan arus mengalir ke arah beban atau motor DC secara benar artinya dapat mengatur arah putaran motor DC sesuai dengan keinginan. Di dalam IC driver motor berisi empat buah driver-H yang berfungsi sebagai pengatur arus listrik secara dua arah karena dalam pembuatan robot membutuhkan dua motor DC. Berikut jenis IC yang berfungsi sebagai *driver* motor.

1. IC L298D
2. IC L293D

Pada gambar 2.8 dapat dilihat bentuk fisik IC L293D dan IC L298D.

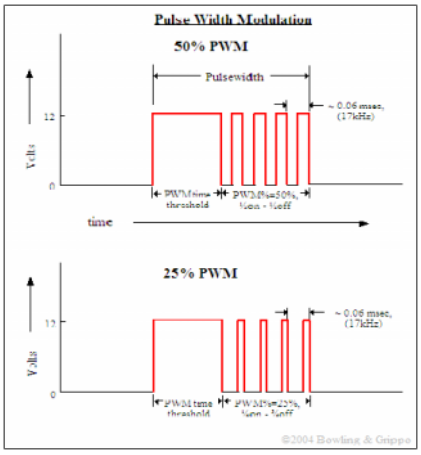


Gambar 2.8 IC L293D dan IC L298D

2.5.2.3. PWM (Pulse Width Modulation)

Salah satu cara untuk mengirimkan informasi analog adalah menggunakan pulsa-pulsa arus. Dengan modulasi pulsa, pembawa informasi terdiri dari pulsa-pulsa persegi yang berulang-ulang. Salah satu teknik modulasi pulsa yang digunakan adalah teknik modulasi durasi atau lebar dari waktu tunda negatif pulsa-pulsa persegi tersebut. Metode tersebut dikenal dengan nama *Pulse Width Modulation* (PWM).

Untuk membangkitkan sinyap PWM, digunakan komparator untuk membandingkan dua buah masukan yaitu generator sinyal dan sinyal referensi. Hasil dari keluaran dari komparator dalah sinyal PWM yang berupa pulsa-pulsa persegi yang berulang-ulang. Durasi atau lebar pulsa dapat dimodulasi dengan cara mengubah sinyal referensi.



Gambar 2.9 Ilustrasi Prosentase PWM

Metode PWM digunakan untuk mengatur kecepatan putaran motor, informasi yang dibawa oleh pulsa-pulsa persegi merupakan tegangan rata-rata. Besarnya tegangan rata-rata tersebut dapat diperoleh dari: Vout = (Vref * duty cycle) / periode. Semakin lebar durasi waktu tunda positif pulsa dari sinyal PWM yang dihasilkan, maka perputaran motor akan semakin cepat, demikian sebaliknya.

2.5.5. 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 *Lithium polymer*. Hal ini karena jenis baterai ini merupakan jenis baterai yang dapat diisi ulang (*rechargeable*). Baterai ini memiliki tegangan kerja 7,4 Volt. Gambar 2.10 memperlihatkan bentuk baterai *Lithium polymer* 1300 mAh.



Gambar 2.10 Lithium polymer 1300mA

Untuk pengisian baterai ini dapat dilakukan kapan saja, selain jenis baterai *Lithium polymer*, 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.

2.5. Perangkat Lunak (Software)

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 AVR (ATmega32) dapat menggunakan bahasa pemrograman *basic, assembly, pascal, c++*. Beberapa contoh *Software editor* dan *compiler* yang dapat digunakan adalah *CodeVisionAVR*, BASCOM-*AVR*, *Programmers Notepad [AVR]*, *mikroBasic PRO for AVR*, *Arduino*. *Software* ini memungkinkan penggunanya memprogram AVR dengan bahasa tingkat tinggi yang relatif ringan dibandingkan bahasa pemrograman lainnya. *Software downloader* yang dapat digunakan adalah *AVRprog versi 1.4*0.

2.5.1. Pengenalan Arduino IDE

Arduino IDE adalah sebuah editor yang digunakan untuk menulis program, meng*compile* ke mikrokontroler keluarga *AVR*. Program ini memungkinkan penggunanya memprogram *AVR* dengan bahasa *C/C++* yang relatif lebih familiar dibandingkan bahasa pemrograman lainnya. Dalam penggunaan, arduino hanya perlu mendefinisikan dua fungsi untuk membuat program *runable*, yaitu:

1. Setup () : fungsi dijalankan sekali pada awal program yang dapat menginisialisasi pengaturan.
2. Loop (): fungsi yang disebut berulang-ulang sampai mikrokontoler off.

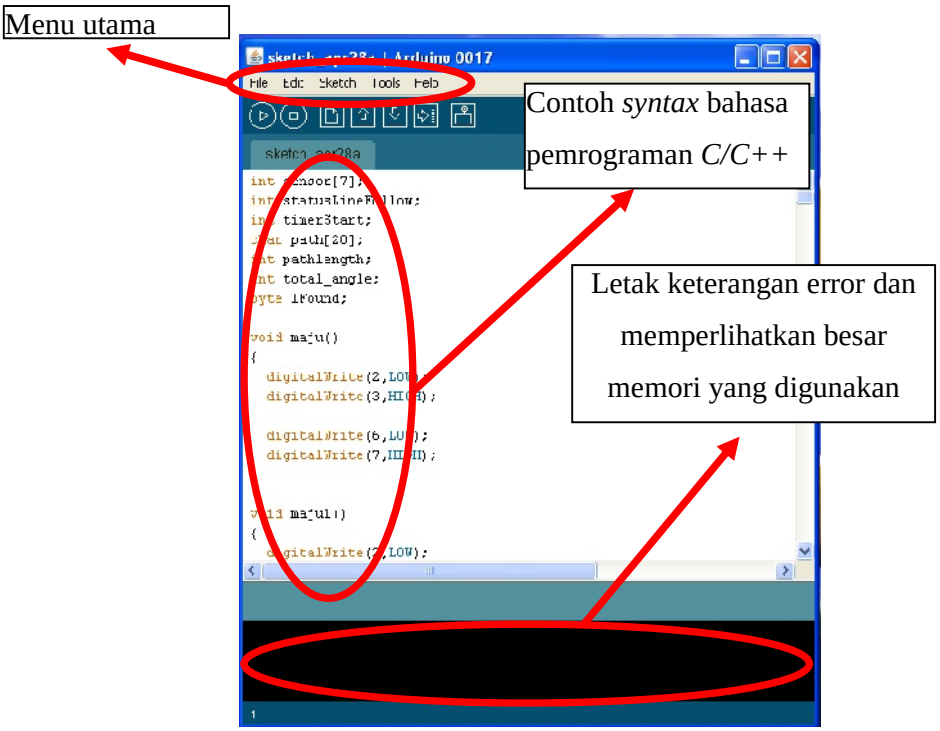
Arduino IDE menggunakan GNU toolchain dan AVR libc untuk mengkompilasi program-program, dan menggunakan avrdude untuk meng-*upload* program. [9]

Berikut dapat dilihat pada tabel 2.4 beberapa instruksi-instruksi dasar yang dapat digunakan pada mikrokontroler ATmega32.

Tabel 2.4 Beberapa instruksi dasar Arduino IDE

Instruksi	Keterangan
do...while	Perulangan
pinMode()	Mengatur, apakah sebuah pin berfungsi sebagai input atau output.
If()...else	Percabangan
for()...	Perulangan
delay()	Waktu tunda milidetik
digitalWrire()	Digunakan untuk memberikan nilai digital (High atau Low) pada pin output
digitalRead()	Digunakan untuk memberikan nilai digital (High atau Low) pada pin input
analogWrite()	Digunakan untuk memberikan nilai analog pada pin output tertentu
analogRead()	Digunakan untuk memberikan nilai analog pada pin inputtertentu

Pada gambar 2.11 memperlihatkan bentuk tampilan Arduino IDE.



Gambar 2.11 Tampilan Arduino

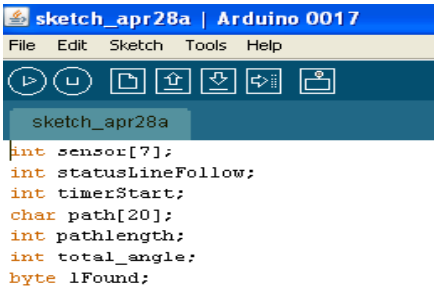
2.5.2. Membuat program pada Arduino IDE

Dalam pembuatan program pada Arduino IDE ada beberapa langkah yang harus dilakukan yaitu:

1. Mendeklarasikan Variable, konstanta.
2. Mendefinisikan beberapa fungsi yang akan digunakan pada program utama.
3. Mendefinisikan fungsi setup.
4. Mendefinisikan fungsi loop.

2.5.2.1. Mendeklarasikan Variabel, Konstanta

Pada langkah ini, bertujuan mendeklarasikan variabel, konstanta dan tipe data yang akan digunakan pada program yang akan dibuat. Beberapa tipe data yang digunakan yaitu *int (integer), char (character), byte.* Pada gambar 2.12 memperlihatkan tampilan pendeklarasian pada arduino IDE.



Gambar 2.12 Tampilan Deklarasi Variabel, Konstanta

2.5.2.2. Mendefinisikan Beberapa Fungsi

Pada langkah ini, bertujuan untuk pendefinisian beberapa fungsi yang nantinya akan dipanggil atau digunakan pada bagian program utama. Seperti halnya pada *mobile robot* ini dibutuhkan fungsi mundur, belok kiri, belok kanan, berhenti dan lain-lain. Pada gambar 2.13 memperlihatkan tampilan pendefinisian fungsi pada arduino IDE.

```
void maju()
{
  digitalWrite(2,LOW);
  digitalWrite(3,HIGH);

  digitalWrite(6,LOW);
  digitalWrite(7,HIGH);
}

void majul()
{
  digitalWrite(2,LOW);
  digitalWrite(3,HIGH);

  digitalWrite(6,LOW);
  digitalWrite(7,HIGH);

  kecepatan(40,40);

}
```

Gambar 2.13 Tampilan pendefinisian fungsi

2.5.2.3. Mendefinisikan fungsi setup

Langkah ini merupakan salah satu langkah terpenting dalam pembuatan program pada Arduino IDE karena fungsi ini yang akan membuat program *runable*, jadi fungsi ini jangan sampai dilewatkan. Pada fungsi ini terdapat beberapa instruksi pengaturan komunikasi serial, pin input/output dan lain-lain. Pada gambar 2.14 memperlihatkan tampilan pendefinisian fungsi setup pada arduino IDE.



Gambar 2.14 Pendefinisian fungsi setup

2.5.2.4. Mendefinisikan fungsi loop

Pada langkah ini bertujuan mendefinisikan fungsi loop yang berisi program utama yang akan dieksekusu terus menerus. Fungsi ini juga merupakan fungsi yang terpenting dan jangan sampai dilewatkan. Pada gambar 2.15 memperlihatkan tampilan pendefinisian fungsi *loop* pada arduino IDE.

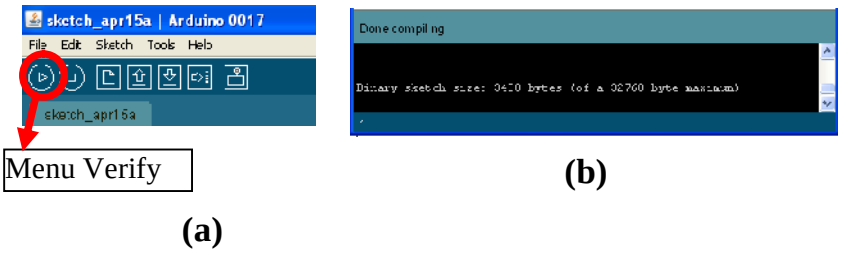


Gambar 2.15 Tampilan pendefinisian fungsi loop

2.5.3. Memeriksa Sintaks dan Mengcompile Program

Sangat penting untuk memeriksa sintaks program, hal ini kita lakukan untuk memastikan semua sintaks sudah benar. Untuk memeriksa *sintaks* sekaligus meng*compile* bisa pilih menu *Verify* yang iconnya mirip tombol *play*.

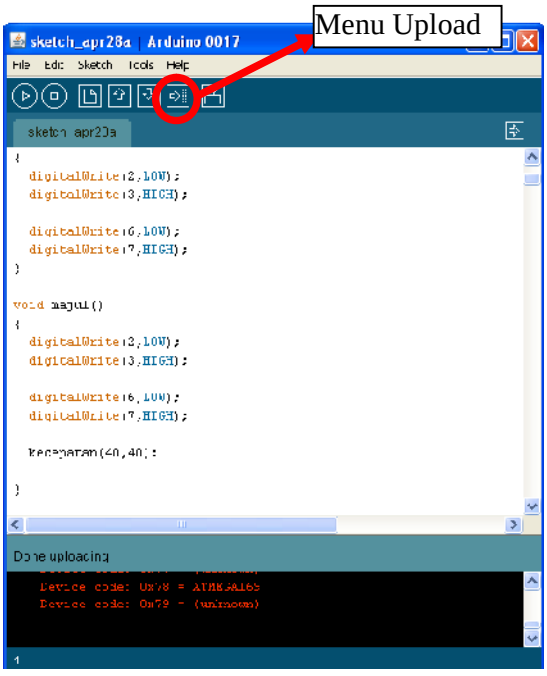
Tampilan *sintaks* program yang sudah benar dapat terlihat pada gambar 2.16.



Gambar 2.16 (a)Tampilan Menu untuk check sintaks dan untuk mengkompile
(b)Tampilan sintaks program yang sudah benar dan proses compile berhasil

2.5.4. Upload Program

Setelah program di*compile* menggunakan, selanjutnya program di *upload* ke mikrokontroler dengan cara men*click* menu *upload*. Tetapi sebelumnya kabel data harus dalam keadaan terhubung ke PC (*Personal Computer*). Pada gambar 2.17 memperlihatkan tampilan mengupload program ke mikrokontroler.



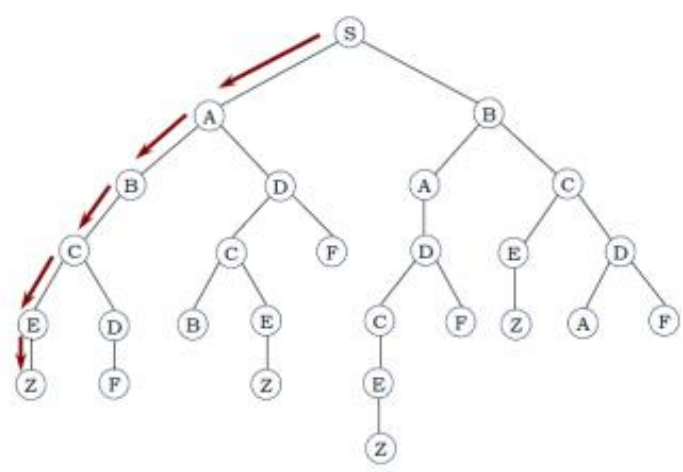
Gambar 2.17 Tampilan Upload program ke Mikrokontroler

2.6. Depth-First Search (Pencarian Kedalam Pertama) [2]

Depth-first search adalah salah satu metode atau algoritma pencarian.. Metode pencarian dikatakan penting untuk menyelesaikan permasalahan karena setiap *state* (keadaan) menggambarkan langkah - langkah untuk menyelesa ikan permasalahan dan setiap state juga akan menggambarkan kemungkinan posisi setiap saat.

Jadi, metode pencarian adalah bagian dari kesimpulan, dimana setiap *state* menggambarkan hipotesis dalam sebuah rangkaian deduktif.

Proses pencarian *Depth-First Search* dilakukan ke node -node yang selevel. Pencarian dimulai dari node akar ke level yang lebih tinggi. Proses ini diulangi terus hingga ditemukannya solusi. Berikut adalah *tree* dari *depth-first search* :



Gambar 2.18 Bentuk tree Depth - first Search

Berikut algoritmanya:

1. Buat sebuah antrian atau stack, inisialisasi node pertama dengan *root* dari *tree*.
2. Bila node pertama, jika bukan *finish*, node dihapus diganti dengan anak-anaknya dengan urutan langkah baru.
3. Bila node pertama merupakan *finish*, selesai.

Keuntungan dari algoritma *Dept First Seacrh* :

1. Membutuhkan memori yang *relative* kecil, karena hanya node-node pada lintasan yang aktif saja.
2. Secara kebetulan, metode *depth-first search* akan menemukan solusi tanpa harus menguji lebih banyak lagi keadaan.

Kelemahan dari algoritma *Depth First Seacrh* :

1. Memungkinkan tidak ditemukannya tujuan yang diharapkan.
2. Hanya akan menemukan satu solusi pada setiap pencarian.