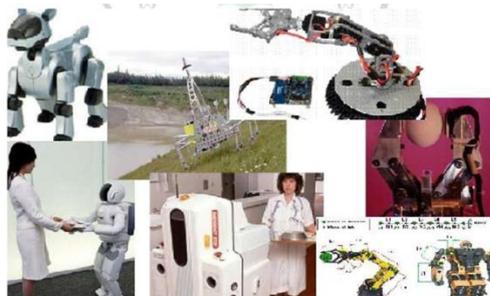


## BAB II TINJAUAN PUSTAKA

### 2.1 Pengertian Robot

Robot berasal dari bahasa *czech* (ceko) yaitu “*robota*” yang berarti pekerja. Robot dapat diartikan sebagai sebuah mesin yang dapat bekerja secara terus menerus baik secara otomatis maupun manual. Robot digunakan untuk membantu tugas-tugas manusia mengerjakan hal yang kadang sulit atau tidak bisa dilakukan manusia secara langsung. Misalnya untuk menangani material radio aktif, merakit mobil dalam industri perakitan mobil, menjelajahi planet mars, sebagai media pertahanan atau perang, serta banyak lagi fungsi lainnya.

Pada dasarnya robot jika ditinjau dari jenisnya terdiri dari dua jenis, yaitu *non mobile* dan *mobile robot*. *Robot non mobile* adalah robot yang melaksanakan aksinya tidak berpindah tempat, sebagai contoh robot Manipulator Lengan yaitu robot yang hanya mempunyai satu tangan yang berfungsi untuk memegang dan memindahkan barang dengan jarak yang relatif dekat, biasanya pengaplikasian robot ini pada industri perakitan elektronik dan perakitan otomotif, sedangkan *robot mobile* adalah robot yang melaksanakan aktivitasnya dengan bergerak dan berpindah dari suatu tempat ke tempat lain, sebagai contoh adalah robot pengikut garis (*line follower*) dan robot pengikut dinding (*wall follower*). Saat ini robot selain untuk membantu pekerjaan manusia juga digunakan sebagai hiburan untuk tujuan yang bermacam-macam seperti kompetisi robot yang ditunjukkan untuk pengembangan inovasi teknologi robot. Berikut adalah beberapa contoh gambar dari berbagai jenis robot.

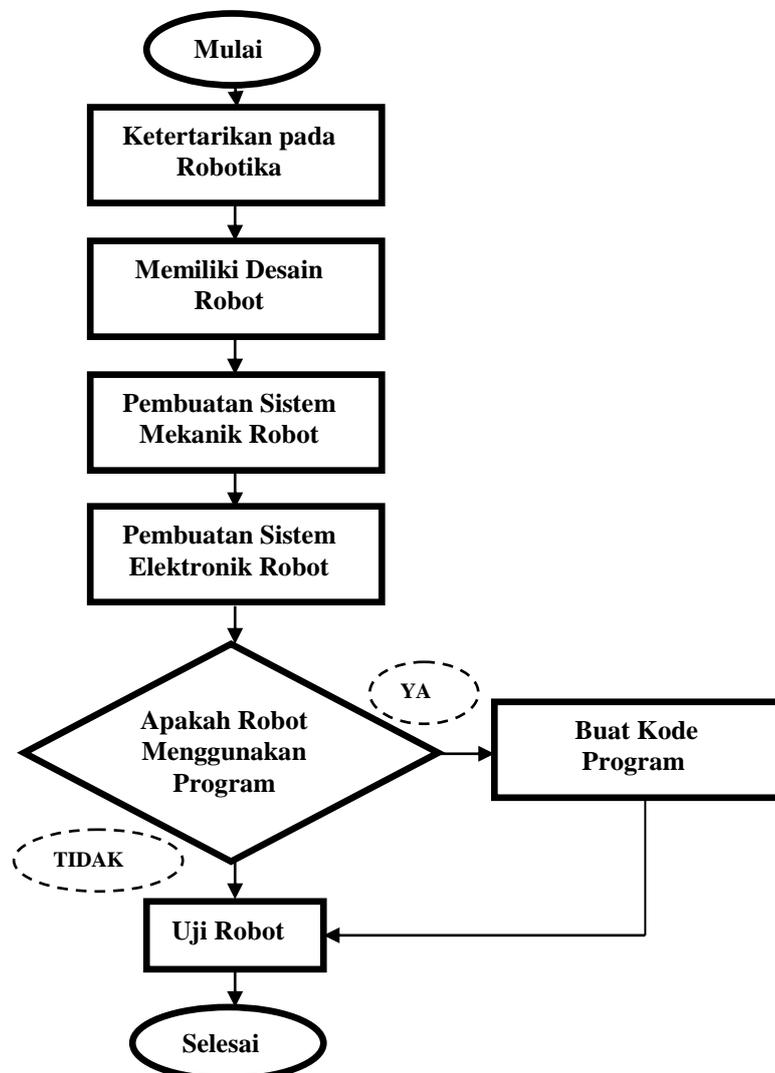


**Gambar 2.1** Contoh Bentuk Robot  
(Sumber : <http://en.wikipedia.org/wiki/Robot>)

Beberapa hal yang penting dalam pembangunan sebuah robot:

1. Sistem mekanik robot
2. Sistem elektronik robot
3. Sistem kendali robot
4. Catu daya (*power supply*)
5. Sensor

Proses Pembangunan Sebuah Robot:



**Gambar 2.2** Tahap-tahap Pembangunan Sebuah Robot

(Sumber : <http://en.wikipedia.org/wiki/Robot>)

## 2.2 Sistem Kendali Robot

Sistem Kendali adalah bagian yang amat penting dalam robot, tanpa sistem kendali hanya akan menjadi benda mekatronik yang mati. Dalam suatu sistem robot, sistem kendali merupakan elemen penting yang tidak dapat terpisahkan. Dalam hal ini sistem kendali bertugas mengkolaborasikan sistem elektronik dan mekanik dengan baik agar mencapai fungsi seperti yang dikehendaki.

Sistem kendali sendiri memiliki mekanisme kerja seperti yang di ilustrasikan berikut ini yaitu, menunggu perintah dari operator atau indikasi dari sensor, kemudian data perintah dari operator atau sensor akan dikirim dan diproses oleh pengendali seperti mikrokontroler sesuai dengan program yang ditanamkan didalamnya, setelah itu mikrokontroler akan mengirim sinyal perintah *aktuasi* ke *actuator* atau *driver* untuk memberikan aksi seperti gerakan motor, *pneumatic*, dan lain sebagainya. Terdapat 2 macam cara pengendalian robot yaitu sistem kendali otomatis dan sistem kendali manual. Pada robot kali ini menggunakan sistem kendali manual yang menggunakan perangkat joystick sebagai pengendalinya.

### 2.2.1 Joystik PS 2 Wireless

Joystik adalah alat inputan yang berwujud tuas dan dapat bergerak ke segala arah. *Joystick* pada umumnya digunakan sebagai pelengkap untuk memainkan permainan video yang dilengkapi lebih dari satu tombol. Berikut adalah tampilan joystick PS2 yang akan di gunakan.



**Gambar 2.3** Joystik PS2 Wireless  
(Sumber : [www.famosastudio.com](http://www.famosastudio.com))

Joystik merupakan piranti pengendali tak langsung, gerakan robot dikendalikan oleh gerakan tuas pada *joystick* absolut atau dengan tekanan Pada tuas. Pada *joystick* biasanya terdapat tombol yang dapat dipilih atau diaplikasikan dengan papan ketik. *Joystick* digunakan untuk mengendalikan robot manual pengangkat dan pemindah barang agar bergerak dan dapat memindahkan barang, dalam pengoperasiannya, *joystick* tidak memerlukan tempat yang luas. Setelah tombol pada *joystick* ditekan maka data akan dikirimkan menuju mikrokontroler untuk diproses. Tiap tombol-tombol *joystick* disambungkan dengan port-port input pada mikrokontroler dan tiap port-port output yang telah disambungkan dengan beban, memberikan perintah dari input *joystick* setelah diproses oleh mikrokontroler agar beban yang berupa motor dc dapat bergerak.

Joystik menggunakan saklar Push Button di setiap tombolnya, Push Button disini mempunyai dua masukan yakni untuk pemberi input dan kommon (pada perancangan kommon dihubungkan ke ground). Dengan disetnya kommon dengan ground, apabila menekan tombol otomatis ketiga masukan terhubung, dengan kata lain kolom dan baris berlogika '0' perubahan logika inilah yang diproses oleh mikrokontroler.

Ada 9 pin di dalam konektor PS2 Controller, yaitu:

1. Pin Data
2. Pin Command
3. Pin Vibration Motor Power
4. Pin Ground
5. Pin Power 3.3V
6. Pin Attention
7. Pin Clock
8. Pin Unknown
9. Pin Acknowledge

Cara kerja Joystik pada robot ini disesuaikan dengan protokol akses/komunikasi antara PS2 dan Joy Stick PS2. Akses data berupa serah terima paket data (beberapa byte) antara JoyStick dan PS2 yang berisi inisiasi

komunikasi sampai informasi tombol mana saja yang sedang dioperasikan (ditekan) serta untuk mode analog juga berisi informasi pembacaan data analog-analog tersebut (tiap analog tersusun atas dua buah potensiometer). Dengan memanfaatkan sistem yang sama antara PS2 dan Joy Stick nya, alat yang dibuat (menggantikan posisi PS2) akan mengakses data Joy Stick PS2. Selanjutnya alat yang dibuat dilabelkan dengan uC untuk mempermudah. Jadi komunikasi antara uC ↔ JoyStick PS2. Untuk mengakses Joy Stick PS2, menggunakan komunikasi SPI *Full Duplex*.

Pertama-tama, uC harus menginisiasi percakapan dengan JoyStick dengan mengirim data 0x01 (sebagai Start-Up). Kedua uC harus mengirim data 0x42 yang mengartikan “request for data” dan pada saat yang bersamaan JoyStick akan mengirimkan nomor model / model number / ID si JoyStick tersebut. Ketiga, uC mengirim data apapun / idle yang biasanya hanya dengan mengirim 0x00, maka JoyStick akan mengirim data 0x5A yang mengartikan data-data tombol (dan analog jika diaktifkan) akan ia berikan pada cycle selanjutnya. Keempat, uC kembali idle – cukup memberi data 0x00, dan JoyStick akan memberikan seluruh data-data penekanan tombol (dan analog jika diaktifkan). Untuk tombol, jika ditekan akan bernilai ‘0’ – default ‘1’. Untuk Joy Stick ber-analog, jika analog non aktif, maka default operating nya adalah mode Digital, sedangkan jika diaktifkan, tergantung tipe analognya – nilai pembacaan jika tepat berada ditengah adalah 128 [DEC]. berikut adalah tabel pertukaran data joystick digital dan analog.

**Tabel 2.1** Pertukaran Data Untuk Joystick Digital  
(Sumber : <https://hamdi88.files.wordpress.com/2010/12/digital>)

Byte	PsxCmd	PsxData	Keterangan							
01	0x01	-								
02	0x42	0x41								
03	-	0x5A	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
04	-	data digital 1	←	↓	→	↑	Start			Select
05	-	data digital 2	□	X	O	Δ	R1	L1	R2	L2

Contoh, pada Joy Stick Digital, penekanan tombol X data yang akan diterima uC bernilai '0' pada bit ke-6, sehingga data yang diterima tersebut adalah 0b10111111 [BIN] atau 0xBF [HEX].

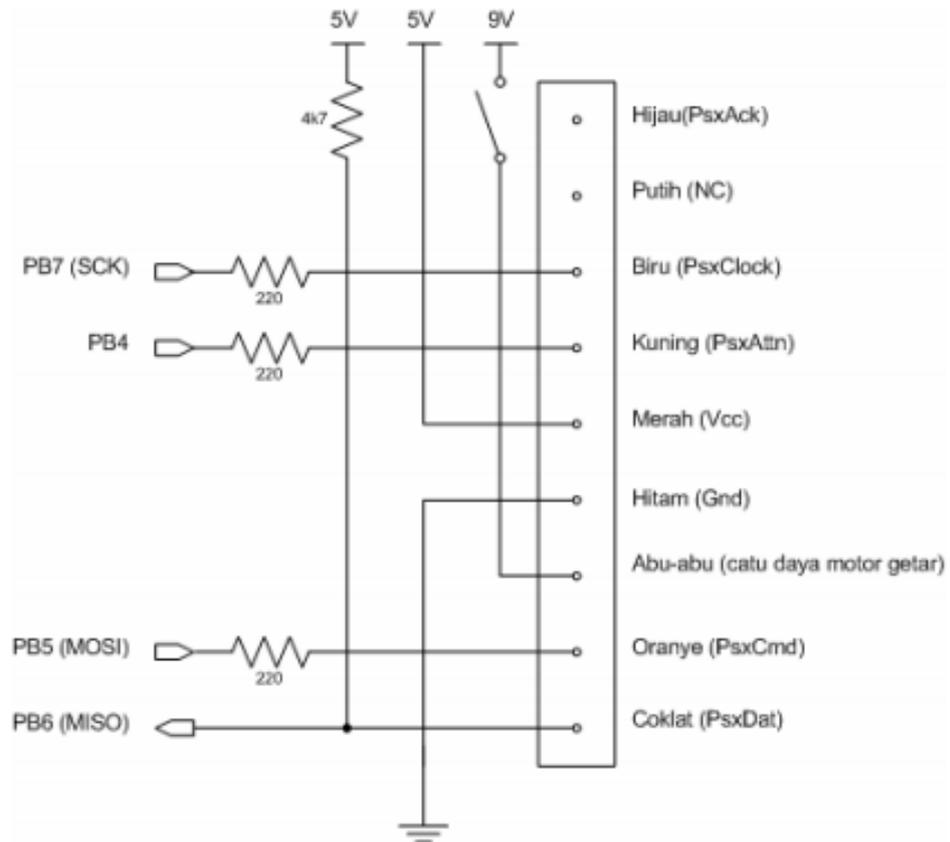
Pada Joy Stick PS2 terdapat dua buah tombol analog, sehingga data yang diberikan Joy Stick kepada uC akan bertambah 4 byte, yaitu masing-masing dua byte pada tiap tombol analog, karena tombol analog memiliki dua buah potensiometer yang masing-masing menyimpan 1 byte data. Pada keadaan idle, nilai analog adalah 128, tetapi jika digerakkan ke suatu arah akan naik sampai batas 255 dan untuk arah kebalikannya akan turun sampai nol. Berikut adalah tabel data untuk joystick analog.

**Tabel 2.2** Pertukaran Data Untuk Joystick Analog  
(Sumber : <https://hamdi88.files.wordpress.com/2010/12/analog>)

Byte	PsxCmd	PsxData	Keterangan							
1	0x01	-								
2	0x42	0x73								
3	-	0x5A	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
4	-	data digital 1	←	↓	→	↑	Start	Joy-R	Joy-L	Select
5	-	data digital 2	□	X	O	Δ	R1	L1	R2	L2
6	-	data analog 1	Joystick Analog Kanan – Sumbu X 0x00 – kiri, 0xff – kanan							
7	-	data analog 2	Joystick Analog Kanan – Sumbu Y 0x00 – atas, 0xff – bawah							
8	-	data analog 3	Joystick Analog Kiri – Sumbu X 0x00 – kiri, 0xff – kanan							
9	-	data analog 4	Joystick Analog Kiri – Sumbu Y 0x00 – atas, 0xff – bawah							

Gambar diatas adalah letak dan urutan data-data informasi / status tombol yang dikirim oleh JoyStick PS2 ke uC – dari atas kebawah. Dapat diamati bahwa terdapat tambahan empat byte data dibandingkan Joy Stick PS digital. Keempat byte tersebut menyimpan informasi status tombol analog yang sedang digunakan.

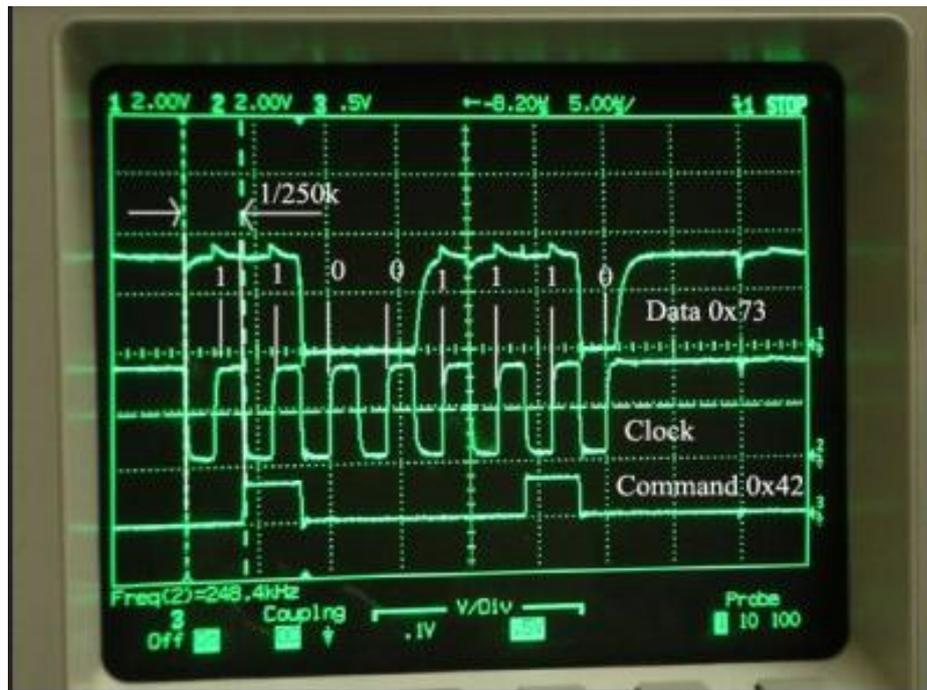
Berikut gambar rangkaian antarmuka joystick PS dengan mikrokontroler.



**Gambar 2.4** Rangkaian Antarmuka Joystick PS dengan Mikrokontroler

(Sumber : <https://hamdi88.files.wordpress.com>)

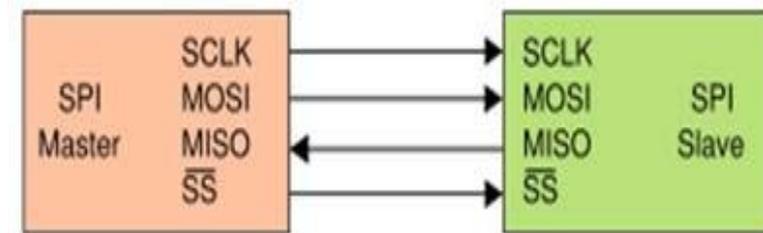
Untuk memeriksa kesalahan dalam komunikasi uC ↔ Joy Stick, dapat menggunakan osiloskop. Satu cycle dimulai dari proses inisiasi dengan memberi data 0x01 oleh uC kepada JoyStick melalui SPI (seperti yang sudah dibahas pada bagian cara kerja) sampai dengan akhir pemberian dua byte data (pada controler digital) atau enam byte data (pada controler analog). Hal ini juga dapat digunakan untuk keperluan troubleshoot data2 per-byte yang dikirimkan dalam jalur MISO dan MOSi dengan melihat lebih detail terhadap data yang dikirim pada komunikasi antara uC ↔ Joy Stick, dengan memperkecil Time/Div display osiloskop sampai terlihat tiap bit yang ditransfer / diterim uC dan JoyStick. Pemakaian osiloskop untuk keperluan tersebut seperti dibawah.



**Gambar 2.5** Tampilan Osiloskop pada Pertukaran data Joystik  
(Sumber : <https://hamdi88.files.wordpress.com>)

### 2.2.1.1 Komunikasi Serial Pheripheral Interface (SPI)

Menurut Barnet (2003, p146) SPI adalah komunikasi data serial secara sinkron yang telah distandarkan oleh Motorola yang beroperasi secara mode full-duplex atau dengan kata lain dapat berkomunikasi dua arah secara bersamaan. Dalam SPI terjadi pertukaran data perangkat komunikasi antara master dan slave, di mana master berfungsi sebagai pengatur jalannya komunikasi data dan slave sebagai perangkat yang diajak berkomunikasi. SPI memungkinkan berkomunikasi dengan banyak Slave dengan memanfaatkan pin chip select. SPI memiliki 4 jalur komunikasi utama yaitu MISO (Master In Slave Out), MOSI (Master Out Slave In), SCK (Serial Clock), SS/CS (Slave/Chip Select) dan sering disebut dengan “four wire communication”.

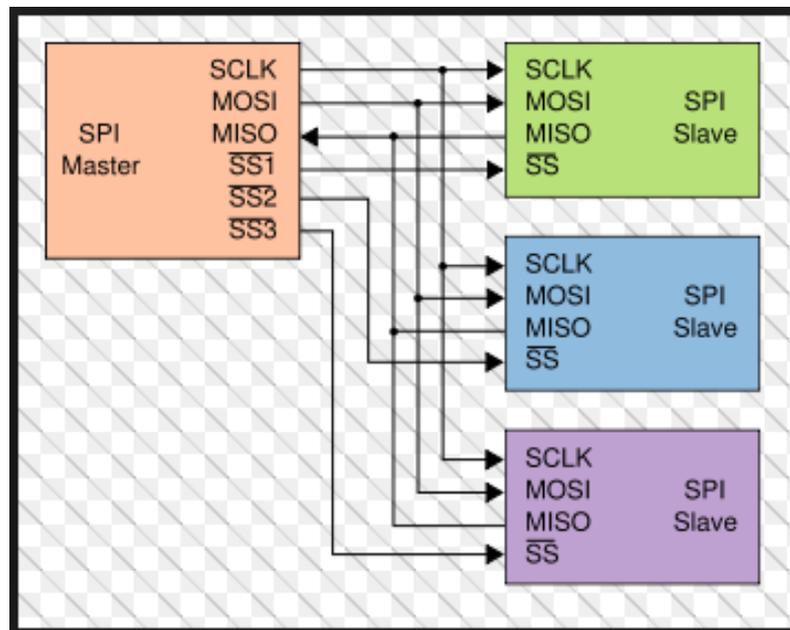


**Gambar 2.6** Komunikasi Serial Peripheral Interface (SPI)  
(Sumber : <https://mutohar.files.wordpress.com>)

Alternatif penamaan yang sering dipakai:

- SCLK, CLK sama dengan SCK (Serial Clock) berfungsi sebagai Clock.
- SDI (Serial Data In), DI (Data In), SI (Serial In) sama dengan MISO berfungsi sebagai jalur data dari *master* dan masuk kedalam *Slave*.
- SDO (Serial Data Out), DO (Data Out), SO (Serial Out) sama dengan MOSI berfungsi sebagai jalur data keluar dari *slave* dan masuk kedalam *master*.
- NCS, NSS, STE sama dengan SS/CS untuk memilih Slave (aktif LOW, output dari Master) berfungsi untuk mengaktifkan *Slave*.

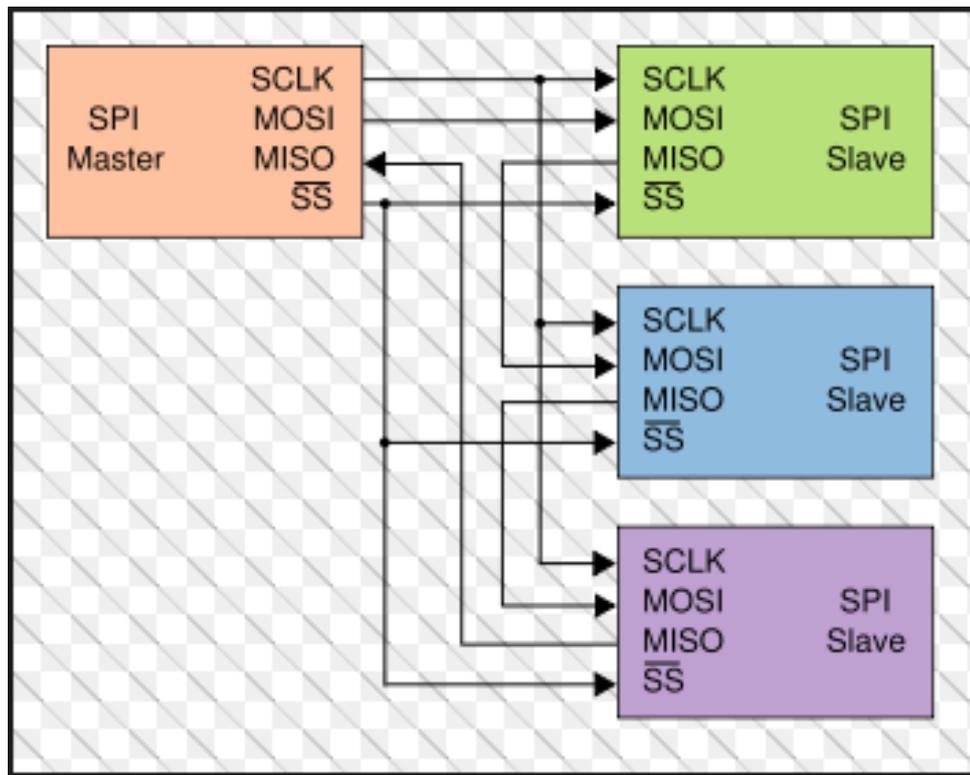
Komunikasi SPI dapat beroperasi pada satu perangkat master dengan satu atau lebih slave. Berikut konfigurasi SPI *Slave Independent*.



**Gambar 2.7** Konfigurasi SPI *Slave Independent*  
(Sumber : <https://mutohar.files.wordpress.com>)

Pada konfigurasi ini, pin SS berbeda dihubungkan ke masing-masing Slave. Walaupun MISO dan MOSI dihubungkan bersamaan, data tidak akan salah karena SS menentukan perangkat mana slave yang aktif.

Konfigurasi SPI *Daisy Chain*.



**Gambar 2.8** Konfigurasi SPI *Daisy Chain*

(Sumber : <https://mutohar.files.wordpress.com>)

Pada konfigurasi ini, pengiriman data dari master dilakukan secara berantai pada beberapa *Slave*. Jika hanya ada satu *slave* maka pin SS di mungkinkan untuk low. Ada beberapa perangkat yang memerlukan falling-edge dari pin SS, seperti IC MAX124.

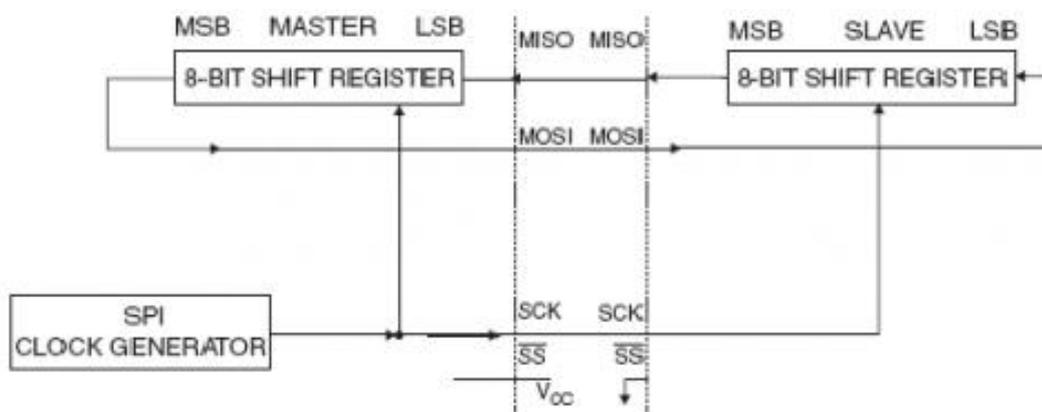
Kebanyakan perangkat slave memiliki output tri-state sehingga pin MISO menjadi memiliki impedansi tinggi apabila slave tidak dipilih. Perangkat tanpa output tri-state SPI tidak dapat berbagi segmen bus dengan perangkat lain sehingga hanya satu slave yang dapat berkomunikasi dengan master.

Untuk memulai komunikasi transmisi data dan pengaturan clock menggunakan *master magnerete clock* yang berfungsi untuk kecepatan

komunikasi. Setiap *slave* memiliki batas frekuensi masing-masing, pada umumnya frekuensi clock yang digunakan 1-70MHz. Master memberikan LOW logic pada SS untuk dapat berkomunikasi.

Pada setiap siklus clock, terjadi komunikasi full-duplex:

- Master mengirim satu bit data pada jalur MOSI, slave membaca data pada jalur Data In (jalur yang sama).
- *Slave* mengirimkan data satu bit dari data out, *master* membaca data pada MISO (jalur yang sama)



**Gambar 2.9** Transmisi data SPI

(Sumber : <https://mutohar.files.wordpress.com>)

## 2.3 Perangkat Keras (*Hardware*)

Perangkat keras atau hardware merupakan alat-alat yang secara fisik dapat dilihat dengan kasat mata dan dapat disentuh oleh tangan manusia. Fungsi utama hardware adalah menjalankan sekumpulan perintah atau instruksi yang diberikan, dan mengeluarkannya dalam bentuk informasi. Perintah-perintah tersebut terdapat pada perangkat lunak (*Software*).

### 2.3.1 Arduino Uno

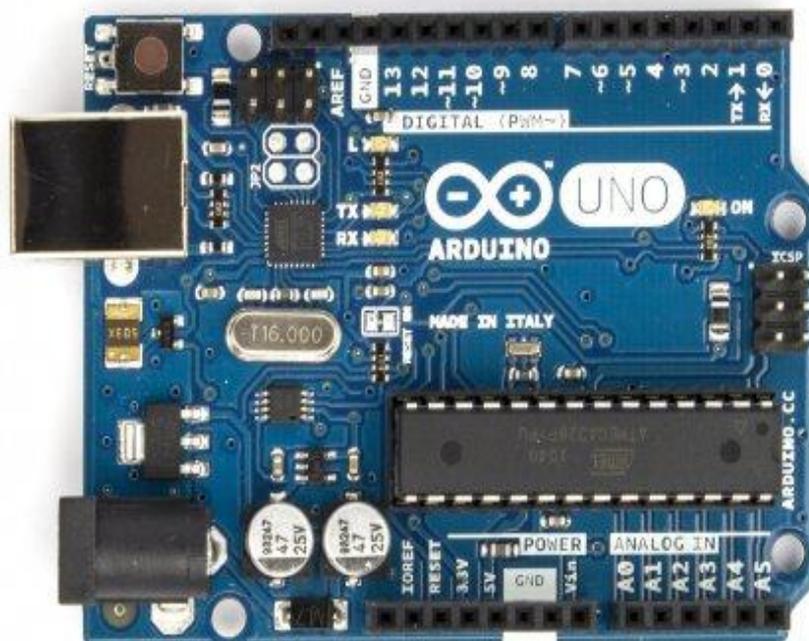
Arduino Uno adalah board berbasis mikrokontroler ATmega328. Board ini memiliki 14 digital input / output pin (dimana 6 pin dapat digunakan sebagai output PWM), 6 input analog, 16 MHz osilator kristal, koneksi USB, jack listrik tombol reset. Pin-pin ini berisi semua yang diperlukan untuk

mendukung mikrokontroler, hanya terhubung ke komputer dengan kabel USB atau sumber tegangan bisa didapat dari adaptor AC-DC atau baterai untuk menggunakannya.

Board Arduino Uno memiliki fitur-fitur baru sebagai berikut :

- Pin out 1 dan 0 ditambahkan SDA dan SCL yang dekat ke pin aref dan dua pin baru lainnya ditempatkan dekat ke pin RESET, dengan IO REF yang memungkinkan sebagai buffer untuk beradaptasi dengan tegangan yang disediakan dari board sistem. Pengembangannya, sistem akan lebih kompatibel dengan Prosesor yang menggunakan AVR, yang beroperasi dengan 5V dan dengan Arduino Karena yang beroperasi dengan 3.3V. Yang kedua adalah pin tidak terhubung, yang disediakan untuk tujuan pengembangannya.
- Sirkuit reset
- ATmega 16U2 ganti 8U yang digunakan sebagai konverter *USB-to-serial*.

Berikut adalah tampilan arduino Uno.



**Gambar 2.10** Tampilan Arduino Uno  
(Sumber : <http://arduino.ua/ru/hardware/Uno>)

Berikut adalah tabel spesifikasi Arduino Uno.

**Tabel 2.3** Spesifikasi Arduino Uno

<b>Mikrokontroler</b>	<b>ATmega328</b>
Operating Voltage	5 V
Input Voltage (recommended)	7-12 V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 pin (6 pin untuk PWM)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB
SRAM	2 KB
EEPROM	1 KB
Clock Speed	16 MHz

### 2.3.1.1 Catu Daya

Arduino uno dapat diaktifkan melalui koneksi USB atau dengan catu daya eksternal. Sumber listrik dipilih secara otomatis. Eksternal (non USB) daya dapat datang baik dari AC-DC adaptor atau baterai. Adaptor ini dapat dihubungkan dengan cara menghubungkannya plug pusat-positif 2.1 mm ke dalam board colokan listrik. Lead dari baterai dapat dimasukkan ke dalam header pin Gnd dan Vin dari konektor Power.

Board dapat beroperasi pada pasokan daya dari 6 - 20 volt. Jika diberikan dengan kurang dari 7V, bagaimanapun, pin 5V dapat menyuplai kurang dari 5 volt dan board mungkin tidak stabil. Jika menggunakan lebih dari 12V, regulator tegangan bisa panas dan merusak board. Rentang yang dianjurkan adalah 7-12 volt.

Pin catu daya adalah sebagai berikut:

- **VIN**. Tegangan input ke board Arduino ketika menggunakan sumber daya eksternal (sebagai lawan dari 5 volt dari koneksi USB atau sumber daya

- diatur). Anda dapat menyediakan tegangan melalui pin ini, atau, jika memasok tegangan melalui colokan listrik, mengaksesnya melalui pin ini.
- **5V**. Catu daya diatur digunakan untuk daya mikrokontroler dan komponen lainnya di board. Hal ini dapat terjadi baik dari VIN melalui regulator onboard, atau diberikan oleh USB .
  - **3,3V**. Pasokan yang dihasilkan oleh regulator on-board. Menarik arus maksimum adalah 50 mA.
  - **GND**.

### 2.3.1.2 Memory

ATmega328 ini memiliki 32 KB dengan 0,5 KB digunakan untuk loading file. Ia juga memiliki 2 KB dari SRAM dan 1 KB dari EEPROM (yang dapat dibaca dan ditulis (RW/read and written) dengan EEPROM library). 2KB RAM pada memory kerja bersifat *volatile* (hilang saat daya dimatikan), digunakan oleh variable-variabel di dalam program. 32KB RAM flash memory bersifat *non-volatile*, digunakan untuk menyimpan program yang dimuat dari komputer. Selain program, flash memory juga menyimpan *bootloader*.

*Bootloader* adalah program inisiasi yang ukurannya kecil, dijalankan oleh CPU saat daya dihidupkan. Setelah *bootloader* selesai dijalankan, berikutnya program di dalam RAM akan dieksekusi. 1KB EEPROM bersifat *non-volatile*, digunakan untuk menyimpan data yang tidak boleh hilang saat daya dimatikan. Tidak digunakan pada papan Arduino (*red*: namun bisa diakses/diprogram oleh pemakai dan digunakan sesuai kebutuhan).

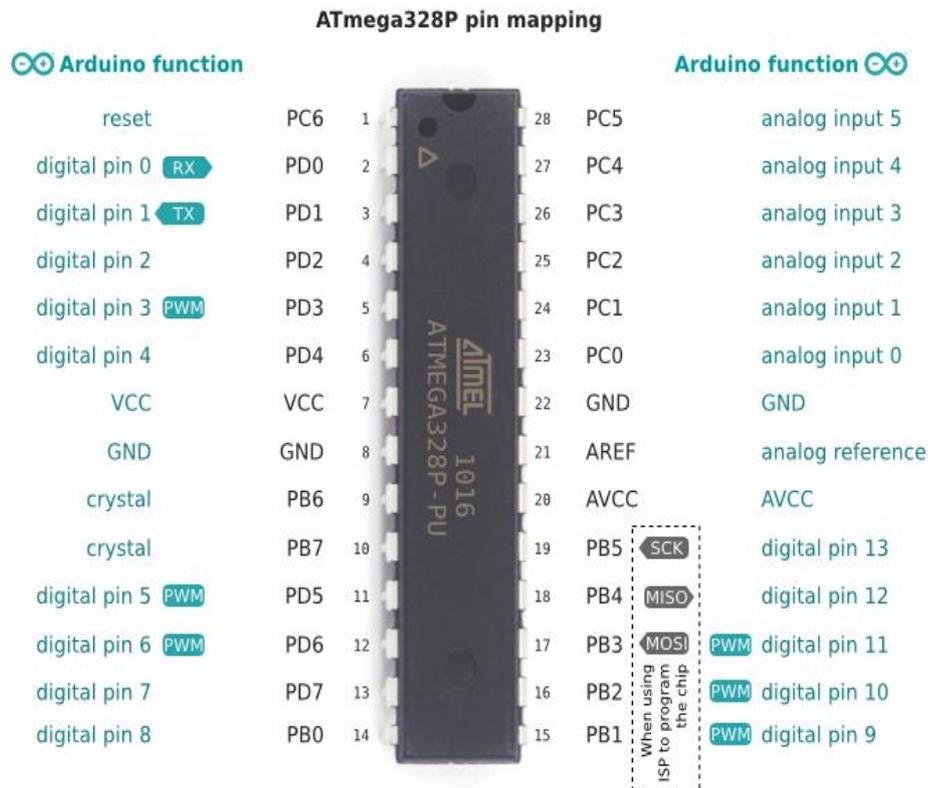
### 2.3.1.3 Input & Output

Masing-masing dari 14 pin digital pada Uno dapat digunakan sebagai input atau output, menggunakan fungsi `pinMode()`, `digitalWrite()`, dan `digitalRead()`. Mereka beroperasi di 5 V. Setiap pin dapat memberikan

atau menerima maksimum 40 mA dan memiliki resistor pull-up internal dari 20-50 K $\Omega$ . Selain itu, beberapa pin memiliki fungsi khusus:

- **Serial:** 0 (RX) dan 1 (TX). Digunakan untuk menerima (RX) dan mengirimkan (TX) data TTL serial. Pin ini terhubung ke pin yang sesuai dari chip ATmega8U2 USB-to-Serial TTL.
- **Eksternal Interupsi:** 2 dan 3. Pin ini dapat dikonfigurasi untuk memicu interupsi pada nilai yang rendah, tepi naik atau jatuh, atau perubahan nilai.
- **PWM:** 3, 5, 6, 9, 10, dan 11. Menyediakan 8-bit output PWM dengan analogWrite () fungsi.
- **SPI:** 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Pin ini mendukung komunikasi SPI menggunakan perpustakaan SPI.
- **LED:** 13. Ada built-in LED terhubung ke pin digital 13. Ketika pin adalah nilai TINGGI, LED menyala, ketika pin adalah RENDAH, itu off.
- **TWI:** A4 atau SDA pin dan A5 atau SCL pin. Mendukung komunikasi TWI
- **Aref.** Referensi tegangan untuk input analog. Digunakan dengan analog Reference().
- **Reset.**

Arduino Uno juga memiliki 6 input analog, diberi label A0 melalui A5, masing-masing menyediakan 10 bit resolusi yaitu 1024 nilai yang berbeda. Secara default sistem mengukur dari tanah sampai 5 volt. Berikut adalah pemetaan pin ATmega 328.



**Gambar 2.11** Pemetaan Pin ATmega 328

(Sumber : <http://arduino.ua/ru/hardware/Uno>)

#### 2.3.1.4 Komunikasi

Arduino Uno memiliki sejumlah fasilitas untuk berkomunikasi dengan komputer, Arduino lain, atau mikrokontroler lain. ATmega328 ini menyediakan UART TTL (5V) komunikasi serial, yang tersedia pada pin digital 0 (RX) dan 1 (TX). Sebuah ATmega16U2 pada saluran board ini komunikasi serial melalui USB dan muncul sebagai com port virtual untuk perangkat lunak pada komputer. Firmware Arduino menggunakan USB driver standar COM, dan tidak ada driver eksternal yang dibutuhkan. Namun pada Windows file Ini diperlukan. Perangkat lunak Arduino termasuk monitor serial yang memungkinkan data sederhana yang akan dikirim ke board Arduino. RX dan TX LED di board akan berkedip ketika data sedang dikirim melalui chip USB-to-serial dan koneksi USB ke

komputer. ATmega328 ini juga mendukung komunikasi I2C (TWI) dan SPI. Fungsi ini digunakan untuk melakukan komunikasi interface pada sistem.

### **2.3.1.5 Programming**

Arduino Uno dapat diprogram dengan perangkat lunak Arduino. Pilih Arduino Uno dari Tool lalu sesuaikan dengan mikrokontroler yang digunakan.

ATmega328 pada Arduino Uno memiliki bootloader yang memungkinkan Anda untuk meng-upload program baru untuk itu tanpa menggunakan programmer hardware eksternal. Ini berkomunikasi menggunakan protokol dari bahasa C.

Sistem dapat menggunakan perangkat lunak FLIP Atmel (Windows) atau programmer DFU (Mac OS X dan Linux) untuk memuat firmware baru. Atau dapat menggunakan header ISP dengan programmer eksternal. Berikut adalah penjelasan dalam penulisan bahasa program Arduino.

#### **2.3.1.5.1 Struktur**

Setiap program Arduino (biasa disebut sketch) mempunyai dua buah fungsi yang harus ada.

- **void setup() { }**

Semua kode didalam kurung kurawal akan dijalankan hanya satu kali ketika program Arduino dijalankan untuk pertama kalinya.

- **void loop() { }**

Fungsi ini akan dijalankan setelah setup (fungsi void setup) selesai. Setelah dijalankan satu kali fungsi ini akan dijalankan lagi, dan lagi secara terus menerus sampai catu daya (power) dilepaskan.

#### **2.3.1.5.2 Syntax**

Berikut ini adalah elemen bahasa C yang dibutuhkan untuk format penulisan.

- **//(komentar satu baris)**

Kadang diperlukan untuk memberi catatan pada diri sendiri apa arti dari kode-kode yang dituliskan. Cukup menuliskan dua buah garis miring dan apapun yang kita ketikkan dibelakangnya akan diabaikan oleh program.

- **/\* \*/(komentar banyak baris)**

Jika anda punya banyak catatan, maka hal itu dapat dituliskan pada beberapa baris sebagai komentar. Semua hal yang terletak di antara dua simbol tersebut akan diabaikan oleh program.

- **{ }(kurung kurawal)**

Digunakan untuk mendefinisikan kapan blok program mulai dan berakhir (digunakan juga pada fungsi dan pengulangan).

- **;(titik koma)**

Setiap baris kode harus diakhiri dengan tanda titik koma (jika ada titik koma yang hilang maka program tidak akan bisa dijalankan).

### 2.3.1.5.3 Variabel

Sebuah program secara garis besar dapat didefinisikan sebagai instruksi untuk memindahkan angka dengan cara yang cerdas. Variabel inilah yang digunakan untuk memindahkannya.

- **int (integer)**

Digunakan untuk menyimpan angka dalam 2 byte (16 bit). Tidak mempunyai angka desimal dan menyimpan nilai dari -32,768 dan 32,767.

- **long (long)**

Digunakan ketika integer tidak mencukupi lagi. Memakai 4 byte (32 bit) dari memori (RAM) dan mempunyai rentang dari -2,147,483,648 dan 2,147,483,647.

- **boolean (boolean)**

Variabel sederhana yang digunakan untuk menyimpan nilai TRUE (benar) atau FALSE (salah). Sangat berguna karena hanya menggunakan 1 bit dari RAM.

- **float (float)**

Digunakan untuk angka desimal (floating point). Memakai 4 byte (32 bit) dari RAM dan mempunyai rentang dari  $-3.4028235E+38$  dan  $3.4028235E+38$ .

- **char (character)**

Menyimpan 1 karakter menggunakan kode ASCII (misalnya 'A' = 65). Hanya memakai 1 byte (8 bit) dari RAM.

#### 2.3.1.5.4 Operator Matematika

Operator yang digunakan untuk memanipulasi angka (bekerja seperti matematika yang sederhana).

- = : Membuat sesuatu menjadi sama dengan nilai yang lain (misalnya:  $x = 10 * 2$ , x sekarang sama dengan 20).
- % : Menghasilkan sisa dari hasil pembagian suatu angka dengan angka yang lain (misalnya:  $12 \% 10$ , ini akan menghasilkan angka 2).
- + : Penjumlahan
- - : Pengurangan
- \* : Perkalian
- / : Pembagian

#### 2.3.1.5.5 Operator Pembandingan

Digunakan untuk membandingkan nilai logika.

- == : Sama dengan (misalnya:  $12 == 10$  adalah FALSE (salah) atau  $12 == 12$  adalah TRUE (benar))
- != : Tidak sama dengan (misalnya:  $12 != 10$  adalah TRUE (benar) atau  $12 != 12$  adalah FALSE (salah))
- < : Lebih kecil dari (misalnya:  $12 < 10$  adalah FALSE (salah) atau  $12 < 12$  adalah FALSE (salah) atau  $12 < 14$  adalah TRUE (benar))
- > : Lebih besar dari (misalnya:  $12 > 10$  adalah TRUE (benar) atau  $12 > 12$  adalah FALSE (salah) atau  $12 > 14$  adalah FALSE (salah))

### 2.3.1.5.6 Struktur Pengaturan

Program sangat tergantung pada pengaturan apa yang akan dijalankan berikutnya, berikut ini adalah elemen dasar pengaturan (banyak lagi yang lain dan bisa dicari di internet).

1. **if..else**, dengan format seperti berikut ini:

```
if (kondisi) { }
else if (kondisi) { }
else { }
```

Dengan struktur seperti diatas program akan menjalankan kode yang ada di dalam kurung kurawal jika kondisinya TRUE, dan jika tidak (FALSE) maka akan diperiksa apakah kondisi pada else if dan jika kondisinya FALSE maka kode pada else yang akan dijalankan.

2. **for**, dengan format seperti berikut ini:

```
for (int i = 0; i < #pengulangan; i++) { }
```

Digunakan bila anda ingin melakukan pengulangan kode di dalam kurung kurawal beberapa kali, ganti #pengulangan dengan jumlah pengulangan yang diinginkan. Melakukan penghitungan ke atas dengan i++ atau ke bawah dengan i--.

### 2.3.1.5.7 Digital

#### 1. **pinMode(pin, mode)**

Digunakan untuk menetapkan mode dari suatu pin, pin adalah nomor pin yang akan digunakan dari 0-19 (pin analog 0-5 adalah 14-19). Mode yang bisa digunakan adalah INPUT atau OUTPUT.

#### 2. **digitalWrite(pin, value)**

Ketika sebuah pin ditetapkan sebagai OUTPUT, pin tersebut dapat dijadikan HIGH (ditarik menjadi 5 volts) atau LOW (diturunkan menjadi ground).

### **3. digitalRead(pin)**

Ketika sebuah pin ditetapkan sebagai INPUT maka anda dapat menggunakan kode ini untuk mendapatkan nilai pin tersebut apakah HIGH (ditarik menjadi 5 volts) atau LOW (diturunkan menjadi ground).

#### **2.3.1.5.8 Analog**

##### **1. analogWrite(pin, value)**

Beberapa pin pada Arduino mendukung PWM (pulse width modulation) yaitu pin 3, 5, 6, 9, 10, 11. Ini dapat merubah pin hidup (on) atau mati (off) dengan sangat cepat sehingga membuatnya dapat berfungsi layaknya keluaran analog. Value (nilai) pada format kode tersebut adalah angka antara 0 (0% duty cycle ~ 0V) dan 255 (100% duty cycle ~ 5V).

##### **2. analogRead(pin)**

Ketika pin analog ditetapkan sebagai INPUT anda dapat membaca keluaran voltase-nya. Keluarannya berupa angka antara 0 (untuk 0 volts) dan 1024 (untuk 5 volts).

## **2.4 Perangkat Lunak (Software)**

Perangkat lunak (Software) berfungsi sebagai sarana interaksi antara pengguna dan perangkat keras. Perangkat lunak dapat juga dikatakan sebagai ‘penterjemah’ perintah-perintah yang dijalankan pengguna komputer untuk diteruskan ke atau diproses oleh perangkat keras. Perangkat lunak ini dibagi menjadi 3 :

- Sistem Operasi ( operating system )
- Program aplikasi ( application program )
- Bahasa pemrograman

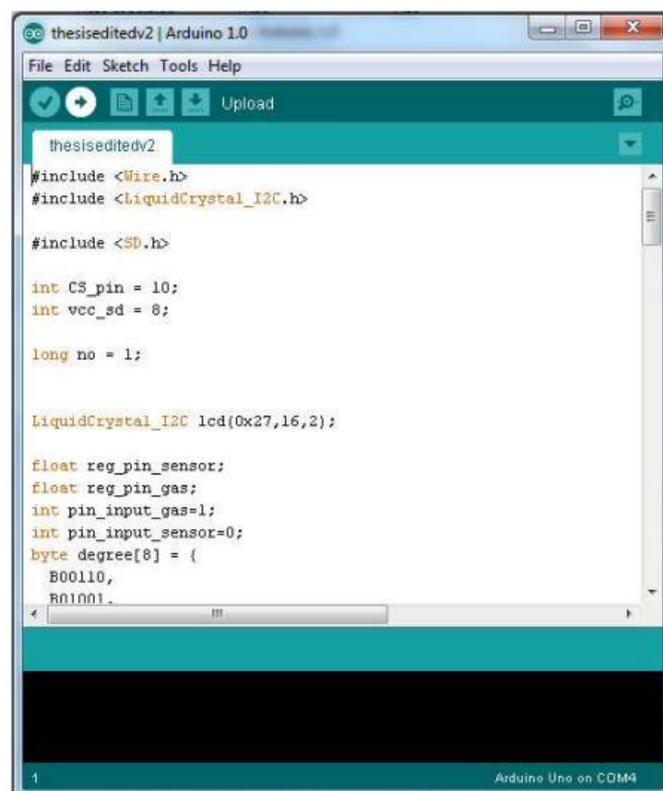
### **2.4.1 Arduino IDE 1.0.6**

Arduino IDE adalah sebuah software yang sangat berperan untuk menulis program, meng-compile menjadi kode biner dan meng-upload ke

dalam memory microcontroller. Software Ini berjalan pada Windows, Mac OS X, dan Linux. Arduino IDE terdiri dari:

- Editor program, sebuah window yang memungkinkan pengguna menulis dan mengedit program dalam bahasa Processing.
- Compiler, sebuah modul yang mengubah kode program (bahasa Processing) menjadi kode biner. Bagaimanapun sebuah mikrokontroler tidak akan bisa memahami bahasa Processing. Yang bisa dipahami oleh mikrokontroler adalah kode biner.
- Uploader, sebuah modul yang memuat kode biner dari komputer ke dalam memory didalam papan Arduino.

Berikut adalah tampilan software Arduino IDE.

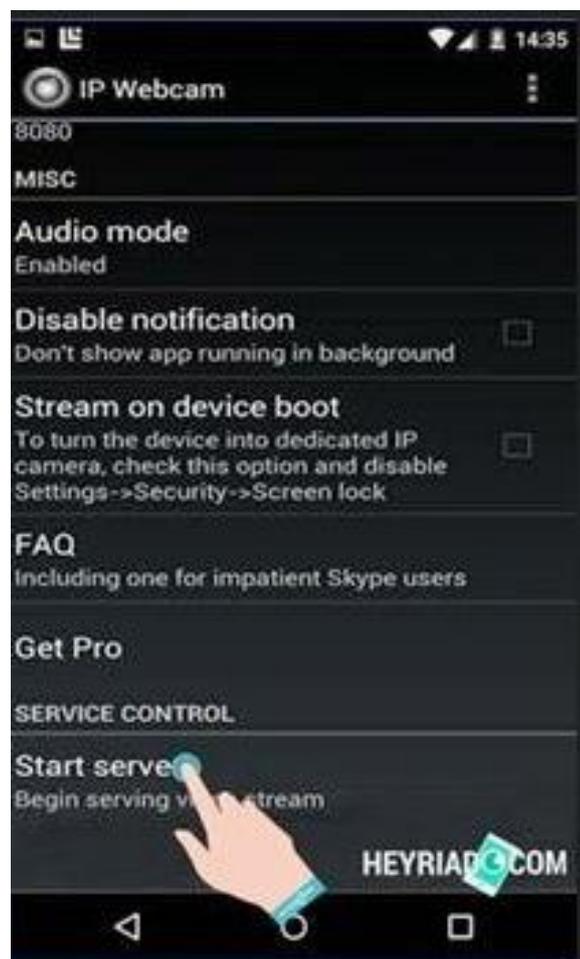


**Gambar 2.12** Tampilan Software Arduino IDE

#### 2.4.2 IP WebCam APK

IP Webcam APK adalah Aplikasi Media & Video untuk Android. Aplikasi IP Webcam ini digunakan untuk mengubah ponsel Android menjadi

kamera CCTV dengan menggunakan akses jaringan internet. CCTV adalah suatu sistem yang menggunakan video camera untuk menampilkan dan merekam gambar pada waktu dan tempat tertentu dimana perangkat ini terpasang dengan signal yang bersifat tertutup. Untuk dapat mengakses tampilan dari kamera CCTV tersebut bisa menggunakan VLC player atau browser web yaitu dengan cara memasukan alamat URL yang terletak pada HP Android. Berikut adalah tampilan Aplikasi IP Webcam pada HP Android.



**Gambar 2.13** Tampilan Aplikasi IP Webcam

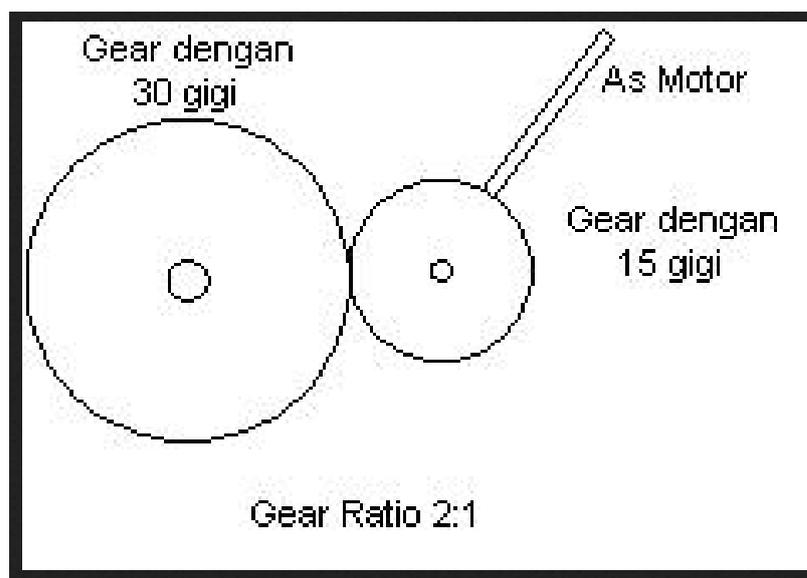
(Sumber : <http://www.appbrain.com/app/ip-webcam/com.pas.webcam#screenshots761>)

## 2.5 Penggerak Roda Robot

Penggerak roda robot yang diaplikasikan pada robot beroda adalah menggunakan 2 buah motor DC. Bagian ini biasanya membutuhkan torsi yang

cukup besar. Oleh karena itu motor DC yang digunakan dilengkapi dengan gigi atau disebut motor gearbox.

Peningkatan kekuatan pada motor gearbox dilakukan dengan menambahkan gigi dengan perbandingan yang cukup besar seperti pada gambar berikut.



**Gambar 2.14** Perbandingan Gear Ratio Motor DC

(Sumber : <http://delta-electronic.com/article/2012/01/an-0180-sistem-mekanik-robot/>)

Tampak pada gambar di atas rasio dihitung dari perbandingan jumlah gigi. Dengan rasio 2:1 maka torsi yang dihasilkan akan meningkat 2 kali lipat namun kecepatan akan menurun dua kali juga.

### 2.5.1 Motor DC

Motor DC merupakan peralatan 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.

Motor DC tersusun dari dua bagian yaitu bagian diam (stator) dan bagian bergerak (rotor). Stator motor arus searah adalah badan motor atau kutub magnet (sikat-sikat), sedangkan yang termasuk rotor adalah jangkar lilitanya. Pada motor, kawat penghantar listrik yang bergerak tersebut pada

dasarnya merupakan lilitan yang berbentuk persegi panjang yang disebut kumparan. Berikut adalah gambar tampilan bentuk fisik motor DC.



**Gambar 2.15** Motor DC dan Gearbox Motor

(Sumber : [http://www.lionballmotor.com/DC-Automobile-Motor/DC-Automobile-Motor\\_1.html](http://www.lionballmotor.com/DC-Automobile-Motor/DC-Automobile-Motor_1.html))

Motor DC memiliki tiga komponen utama:

- Kutub medan. Secara sederhana digambarkan bahwa interaksi dua kutub magnet akan menyebabkan perputaran pada motor DC. Motor DC memiliki kutub medan yang stasioner dan dinamo yang menggerakkan bearing pada ruang diantara kutub medan. Motor DC sederhana memiliki dua kutub medan: kutub utara dan kutub selatan. Garis magnetik energi membesar melintasi bukaan diantara kutub-kutub dari utara ke selatan. Untuk motor yang lebih besar atau lebih kompleks terdapat satu atau lebih elektromagnet. Elektromagnet menerima listrik dari sumber daya dari luar sebagai penyedia struktur medan.
- Dinamo. Bila arus masuk menuju dinamo, maka arus ini akan menjadi elektromagnet. Dinamo yang berbentuk silinder, dihubungkan ke as penggerak untuk menggerakkan beban. Untuk kasus motor DC yang kecil, dynamo berputar dalam medan magnet yang dibentuk oleh kutub-kutub, sampai kutub utara dan selatan magnet berganti lokasi. Jika hal ini terjadi, arusnya berbalik untuk merubah kutub-kutub utara dan selatan dinamo.
- Commutator. Komponen ini terutama ditemukan dalam motor DC. Kegunaannya adalah untuk membalikan arah arus listrik

dalam dinamo. Commutator juga membantu dalam transmisi arus antara dinamo dan sumber daya.

Keuntungan utama motor DC adalah sebagai pengendali kecepatan, yang tidak mempengaruhi kualitas pasokan daya. Motor ini dapat dikendalikan dengan mengatur:

- Tegangan dinamo – meningkatkan tegangan dinamo akan meningkatkan kecepatan
- Arus medan – menurunkan arus medan akan meningkatkan kecepatan.

Hubungan antara kecepatan, flux medan dan tegangan dinamo ditunjukkan dalam persamaan berikut:

Gaya Elektromagnetik (E)

$$E = K\Phi N$$

Torque (T) :

$$T = K\Phi I_a$$

Dimana:

E = gaya elektromagnetik yang dikembangkan pada terminal dinamo (volt)

$\Phi$  = flux medan yang berbanding lurus dengan arus medan

N = kecepatan dalam RPM (putaran per menit)

T = torque electromagnetic

$I_a$  = arus dynamo

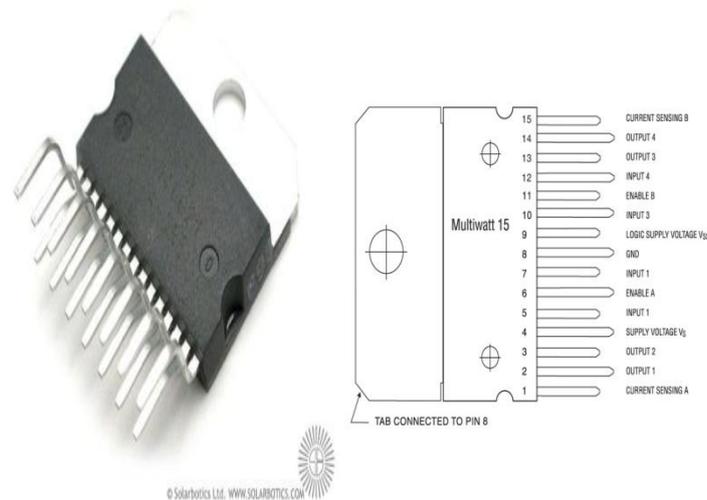
K = konstanta persamaan

Prinsip Kerja Motor DC yaitu memerlukan suplai tegangan yang searah pada kumparan medan untuk diubah menjadi energi mekanik. Catu tegangan DC dari baterai menuju ke lilitan melalui sikat yang menyentuh komutator, dua segmen yang terhubung dengan dua ujung lilitan. Kumparan satu lilitan disebut angker dinamo. Angker dinamo adalah sebutan untuk komponen yang berputar di antara medan magnet.

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.

### 2.5.2 L298

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 H-bridge nya. Kedua H bridge di dalam IC ini bisa di parallel untuk meningkatkan kemampuan menopang arus mencapai 4 A, serta mendukung control PWM dengan frekuensi mencapai 20 KHZ. Berikut ini adalah bentuk dan konfigurasi pin IC L298 yang digunakan sebagai driver motor.



**Gambar 2.16** IC L298

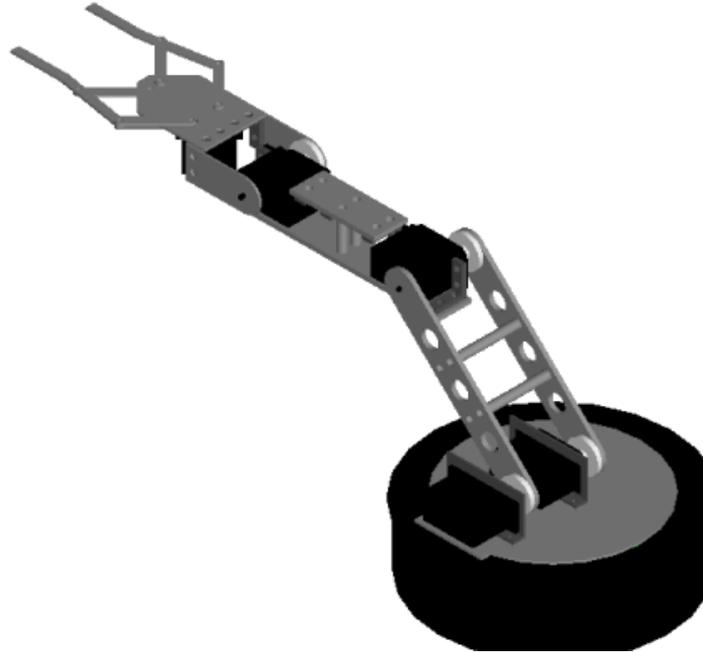
(Sumber : <http://microhobby.net/14-05-2011/projects/l298-breakout-module>)

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). PWM yang dikirimkan menentukan kecepatan putar motor DC.

## 2.6 Penggerak Lengan Robot

Penggerak lengan robot dengan sebuah penjepit (gripper) sebagai end-effector. Lengan robot yang dirancang mempunyai dua buah persendian

sehingga mempunyai 2 DOF. Pada penggerak lengan robot ini terdiri dari dua buah motor servo. Berikut adalah contoh tampilan lengan robot.



**Gambar 2.17** Lengan Robot  
(Sumber : <http://wawands.blogspot.com/>)

### 2.6.1 Motor Servo

Motor servo adalah sebuah perangkat atau aktuator putar (motor) yang dirancang dengan sistem kontrol umpan balik loop tertutup (servo), sehingga dapat di set-up atau di atur untuk menentukan dan memastikan posisi sudut dari poros output motor. motor servo merupakan perangkat yang terdiri dari motor DC, serangkaian gear, rangkaian kontrol dan potensiometer. Serangkaian gear yang melekat pada poros motor DC akan memperlambat putaran poros dan meningkatkan torsi motor servo, sedangkan potensiometer dengan perubahan resistansinya saat motor berputar berfungsi sebagai penentu batas posisi putaran poros motor servo. Berikut adalah contoh bentuk fisik motor servo.



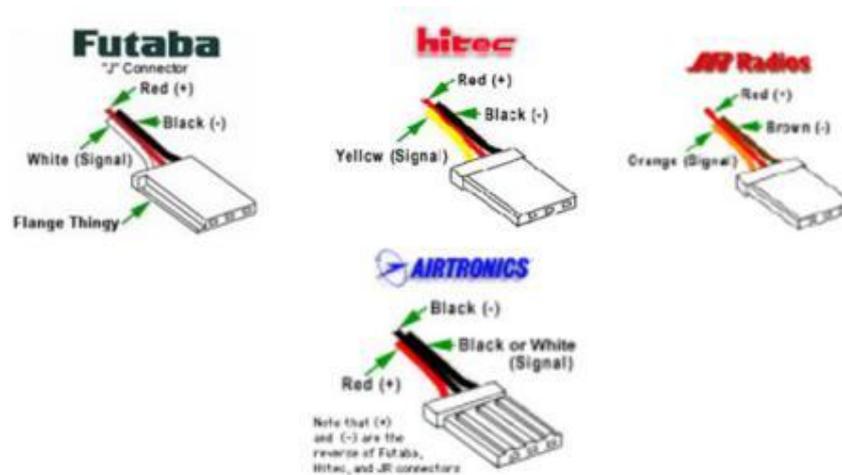
**Gambar 2.18** Motor Servo

(Sumber : [www.elektronika-dasar.web.id/teori-elektronika/motor-servo](http://www.elektronika-dasar.web.id/teori-elektronika/motor-servo))

Ada dua jenis motor servo, yaitu motor servo AC dan DC. Motor servo AC lebih dapat menangani arus yang tinggi atau beban berat, sehingga sering diaplikasikan pada mesin-mesin industri. Sedangkan motor servo DC biasanya lebih cocok untuk digunakan pada aplikasi-aplikasi yang lebih kecil. Dan bila dibedakan menurut rotasinya, umumnya terdapat dua jenis motor servo yang dan terdapat di pasaran, yaitu **motor servo standar rotation 180°** dan **motor servo rotation continuous**.

1. Motor servo standard (servo rotation 180°) adalah jenis yang paling umum dari motor servo, dimana putaran poros outputnya terbatas hanya 90° kearah kanan dan 90° kearah kiri. Dengan kata lain total putarannya hanya setengah lingkaran atau 180°.
2. Motor servo rotation continuous merupakan jenis motor servo yang sebenarnya sama dengan jenis servo standard, hanya saja perputaran porosnya tanpa batasan atau dengan kata lain dapat berputar terus, baik ke arah kanan maupun kiri.

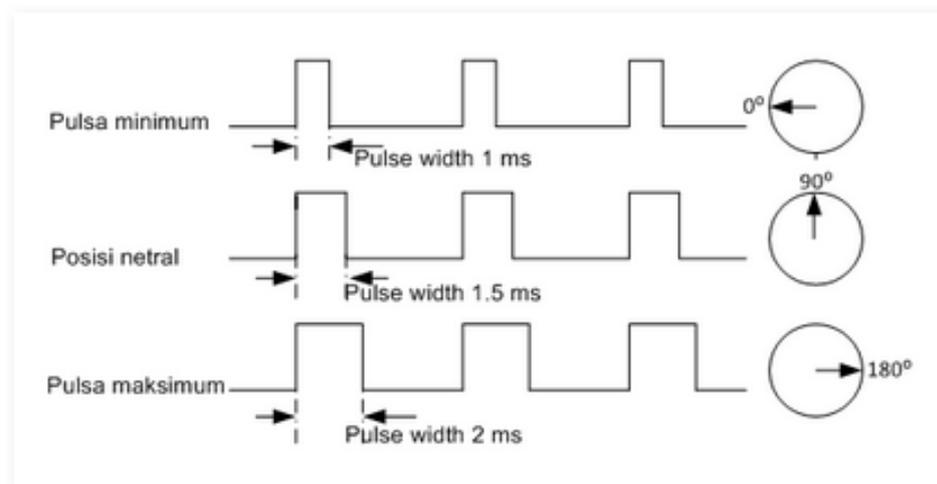
Motor servo mempunyai 3 buah pin yang terdiri dari **VCC, GND dan Signal/kontrol**. Berikut adalah konfigurasi pin motor servo berdasarkan merk.



**Gambar 2.19** Konfigurasi Pin Motor Servo

(Sumber : [www.elektronika-dasar.web.id/teori-elektronika/motor-servo](http://www.elektronika-dasar.web.id/teori-elektronika/motor-servo))

Operasional motor servo dikendalikan oleh sebuah pulsa selebar  $\pm 20$  ms, dimana lebar pulsa antara 0.5 ms dan 2 ms menyatakan akhir dari range sudut maksimum. Apabila motor servo diberikan pulsa dengan besar 1.5 ms mencapai gerakan  $90^\circ$ , maka bila kita berikan pulsa kurang dari 1.5 ms maka posisi mendekati  $0^\circ$  dan bila kita berikan pulsa lebih dari 1.5 ms maka posisi mendekati  $180^\circ$ . Berikut adalah gambar pulsa kendali motor servo.



**Gambar 2.20** Kendali Pulsa Motor Servo

(Sumber : [www.elektronika-dasar.web.id/teori-elektronika/motor-servo](http://www.elektronika-dasar.web.id/teori-elektronika/motor-servo))

Motor Servo akan bekerja secara baik jika pada bagian pin kontrolnya diberikan sinyal PWM dengan frekuensi 50 Hz. Dimana pada saat sinyal dengan frekuensi 50 Hz tersebut dicapai pada kondisi Ton duty cycle 1.5 ms, maka rotor dari motor akan berhenti tepat di tengah-tengah (sudut  $0^\circ$ / netral).

Pada saat Ton duty cycle dari sinyal yang diberikan kurang dari 1.5 ms, maka rotor akan berputar ke berlawanan arah jarum jam (Counter Clock wise, CCW) dengan membentuk sudut yang besarnya linier terhadap besarnya Ton duty cycle, dan akan bertahan diposisi tersebut. Dan sebaliknya, jika Ton duty cycle dari sinyal yang diberikan lebih dari 1.5 ms, maka rotor akan berputar searah jarum jam (Clock Wise, CW) dengan membentuk sudut yang linier pula terhadap besarnya Ton duty cycle, dan bertahan diposisi tersebut

## 2.7 Sistem Pengereman Otomatis

Sistem pengereman otomatis digunakan sebagai sistem keamanan pada robot untuk menghindari robot dari tabrakan. Pada sistem pengereman otomatis robot menggunakan gelombang ultrasonik. Sensor ultrasonik digunakan sebagai detektor jarak antara robot dengan obyek di depannya.

### 2.7.1 Sensor HC-SR04

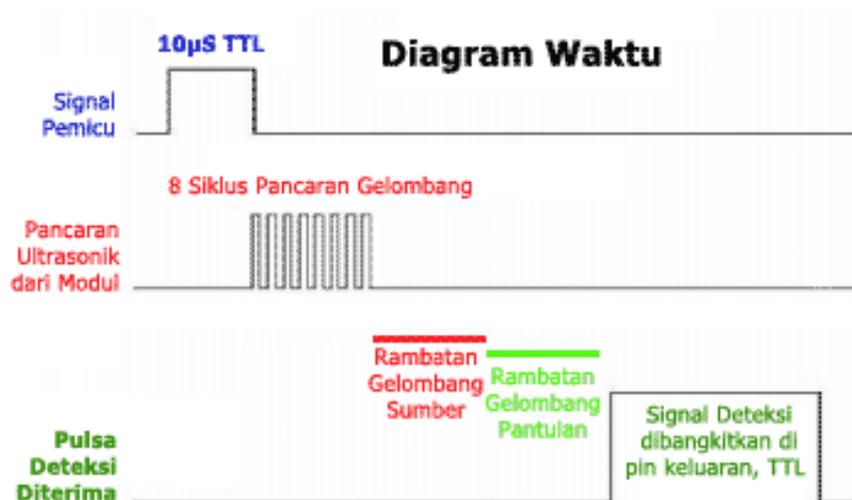
Sensor HC-SR04 adalah sensor pengukur jarak berbasis gelombang ultrasonik. Prinsip kerja sensor ini mirip dengan radar ultrasonik. Gelombang ultrasonik di pancarkan kemudian di terima balik oleh receiver ultrasonik. Jarak antara waktu pancar dan waktu terima adalah representasi dari jarak objek. Sensor ini cocok untuk aplikasi elektronik yang memerlukan deteksi jarak termasuk untuk sensor pada robot. Berikut adalah gambar tampilan sensor HC-SR04.



**Gambar 2.21** Sensor HC-SR04

(Sumber : <https://www.aimagin.com/hc-sr04-ultrasonic-sensor.html>)

Sensor HC-SR04 adalah versi low cost dari sensor ultrasonic PING buatan parallax. Perbedaannya terletak pada pin yang digunakan. HC-SR04 menggunakan 4 pin sedangkan PING buatan parallax menggunakan 3 pin. Pada Sensor HC-SR04 pin trigger dan output diletakkan terpisah. Sedangkan jika menggunakan PING dari Parallax pin trigger dan output telah diset default menjadi satu jalur. Tidak ada perbedaan signifikan dalam pengimplementasiannya. Jangkauan karak sensor lebih jauh dari PING buatan parllax, dimana jika ping buatan parllax hanya mempunyai jarak jangkauan maksimal 350 cm sedangkan sensor HC-SR04 mempunyai kisaran jangkauan maksimal 400-500cm. Berikut diagram waktu dari penggunaan sensor ini:



**Gambar 2.22** Timing Diagram Sensor HC-SR04

(Sumber : <http://www.vcc2gnd.com/2014/02/hc-sr04-ultrasonic-range-sensor.html>)

Bangkitkan pulsa pendek sepanjang  $10\mu\text{s}$  sebagai sinyal pemicu ke pin picu masukan dari modul ini untuk mulai pendeteksian (catat waktu saat ini), HC-SR04 akan memancarkan 8 siklus gelombang ultrasonik pada frekuensi 40 kHz. Saat gelombang suara ini menabrak objek (benda atau dinding di depannya), gelombang akan dipantulkan balik dan diterima oleh detektor yang kemudian membangkitkan sinyal deteksi di pin keluaran modul. Lama selang waktu antara pengiriman signal hingga pendeteksian sinyal pantulan adalah waktu yang ditempuh gelombang suara, yaitu sepanjang dua kali jarak antara sensor dan objek yang terdeteksi karena signal berjalan pulang-pergi. Dengan mengetahui selang waktu ini dan kecepatan rambat suara di udara

(340 meter/detik pada udara kering, atau  $3,4 \times 10^8 \mu\text{s}$ ), Untuk menghitung nilai jarak, dengan menggunakan rumus, yaitu :

$$\text{jarak} = (\text{lebar pulsa} / 29.1) / 2$$

Berikut adalah tabel spesifikasi sensor HC-SR04.

**Tabel 2.4** Spesifikasi Sensor Ultrasonik

Sensor	HC-SR04
Working Voltage	5 VDC
Static current	< 2mA
Output signal:	Electric frequency signal, high level 5V, low level 0V
Sensor angle	< 15 degrees
Detection distance (claimed)	2cm-450cm
precision	~3 mm
Input trigger signal	10us TTL impulse
Echo signal	output TTL PWL signal
Pins	1. VCC 2. trig(T) 3. echo(R) 4. GND