

BAB II

TINJAUAN PUSTAKA

2.1 Arduino

Modul *hardware* Arduino diciptakan pertama kali di Ivrea, Italia pada tahun 2005 oleh Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino, David A. Mellis, dan Nicholas Zambetti (*Arduino, 2011 dan Banzi 2008*). Bahasa Arduino merupakan *fork* (turunan) bahasa *Wiring Platform* dan bahasa *Processing*. *Wiring Platform* diciptakan oleh Hernando Barragan pada tahun 2003 (*Wiring.org.co, 2011*) dan *Processing* dibuat oleh Casey Reas dan Benjamin Fry pada tahun 2011 (*Processing.org, 2012*).

Arduino adalah sistem punarupa elektronika (*electronic prototyping platform*) berbasis *open-source* yang fleksibel dan mudah digunakan baik dari sisi perangkat keras/*hardware* maupun perangkat lunak/*software*. Di luar itu, kekuatan utama arduino adalah jumlah pemakai yang sangat banyak sehingga tersedia pustaka kode program (*code library*) maupun modul pendukung (*hardware support modules*) dalam jumlah yang sangat banyak. Hal ini memudahkan para pemula untuk mengenal dunia mikrokontroler.

Arduino didefinisikan sebagai sebuah *platform* elektronik yang *open source*, berbasis pada *software* dan *hardware* yang fleksibel dan mudah digunakan, yang ditujukan untuk seniman, desainer, *hobbies* dan setiap orang yang tertarik dalam membuat sebuah objek atau lingkungan yang interaktif (*Artanto,2012:1*).

Arduino sebagai sebuah *platform* komputasi fisik (*Physical Computing*) yang *open source* pada *board input output* sederhana, yang dimaksud dengan *platform* komputasi fisik disini adalah sebuah sistem fisik yang interaktif dengan penggunaan *software* dan *hardware* yang dapat mendeteksi dan merespon situasi dan kondisi.

Kelebihan arduino dari *platform hardware* mikrokontroller lain adalah:

1. IDE Arduino merupakan *multiplatform*, yang dapat dijalankan di berbagai sistem operasi, seperti *Windows, Macintosh* dan *Linux*.

2. IDE Arduino dibuat berdasarkan pada IDE *Processing* sederhana sehingga mudah digunakan.
3. Pemrograman Arduino menggunakan kabel yang terhubung dengan *port* USB bukan *port* serial. Fitur ini berguna karena banyak komputer sekarang ini tidak memiliki *port* serial.
4. Arduino adalah *hardware* dan *software open source*, pembaca bisa mendownload *software* dan gambar rangkaian arduino tanpa harus membayar ke pembuat arduino.
5. Biaya *hardware* cukup murah, sehingga tidak terlalu menakutkan untuk membuat kesalahan.
6. Proyek arduino ini dikembangkan dalam lingkungan pendidikan sehingga bagi pemula akan lebih cepat dan mudah mempelajarinya.
7. Memiliki begitu banyak pengguna dan komunitas di internet dapat membantu setiap kesulitan yang dihadapi terutama oleh programmer pemula. (Artanto, 2012:2)

2.1.1 Arduino Mega 2560

Arduino mega 2560 adalah papan mikrokontroler Atmega 2560 berdasarkan *datasheet* memiliki 54 digital pin *input* atau *output* dimana 15 pin dapat digunakan sebagai *output* PWM atau (*Pulse Width Modulation*), 16 *analog input*, 4 UART (*Universal Asynchronous Receiver/Transmitter*), *osilator* kristal 16 MHz, koneksi USB, *jack* listrik, *header ICSP (In-Circuit Serial Progaming)*, dan tombol *reset*. Semuanya diperlukan untuk mendukung mikrokontroler, hanya menghubungkannya ke komputer dengan kabel USB atau *power* dengan adaptor AC (*Alternating Current*) – DC (*Direct Current*) atau baterai. Gambar 2.1 menunjukkan bentuk fisik Arduino Mega 2560. Adapun spesifikasi mengenai Arduino Mega 2560 yaitu:

- | | |
|-------------------------------------|--------------|
| 1. Mikrokontroler | : ATmega2560 |
| 2. Tegangan Operasional | : 5V |
| 3. Tegangan Masukan (direkomendasi) | : 7-12V |
| 4. Tegangan Masukan (batas) | : 6-20V |

- | | |
|---|----------|
| 5. Pin Digital I/O | : 54 |
| (14 pin untuk keluaran PWM) | |
| 6. Analog Input Pin | : 16 |
| 7. Arus DC per I/O Pin | : 40 mA |
| 8. Arus DC untuk 3.3V Pin | : 50 mA |
| 9. Memori <i>Flash</i> | : 256 KB |
| (8 KB digunakan untuk <i>bootloader</i>) | |
| 10. SRAM (<i>Static Random Access Memory</i>) | : 8 KB |
| 11. EEPROM | : 4 KB |
| 12. <i>Clock Speed</i> | : 16 MHz |



Gambar 2.1 Arduino Mega 2560

(Sumber: Arduino, 1990)

Arduino Mega 2560 berbeda dari semua papan sebelumnya dalam hal itu tidak menggunakan FTDI *chip driver* USB-to-serial. Sebaliknya, fitur Atmega 16U2 (Atmega 8U2 dalam *board* revisi 1 dan revisi 2) diprogram sebagai *converter* USB-to-serial. Revisi 2 dari Arduino Mega 2560 memiliki resistor menarik garis 8U2 HWB ke *ground*, sehingga lebih mudah untuk dimasukkan ke dalam mode DFU (*Device Firmware Update*). Revisi 3 dari arduino mega 2560 memiliki fitur-fitur baru sebagai berikut:

- a. 1,0 *pinout* tambah SDA (*Serial Data*) dan SCL (*Serial Clock*) pin yang dekat dengan pin AREF (*ADC Reference*) dan dua pin baru lainnya ditempatkan dekat dengan *reset*.

- b. Sirkuit *reset* lebih kuat.
- c. Atmega 16U2 menggantikan 8U2.

2.1.2 Power

Arduino Mega 2560 dapat diaktifkan melalui koneksi USB atau dengan catu daya eksternal. Sumber daya dipilih secara manual. Daya eksternal (non-USB) dapat berasal baik dari adaptor AC (*Alternating Current*) – DC (*Direct Current*) atau baterai. Adaptor ini dapat dihubungkan dengan steker 2.1mm pusat-positif ke sumber listrik.

Papan Arduino Mega 2560 dapat beroperasi pada tegangan 6 sampai 20 VDC. Jika menggunakan lebih dari 12V, regulator tegangan bisa panas dan merusak *board arduino*. Kisaran yang dianjurkan adalah 7V sampai 12V. Pin listrik *board arduino mega 2560* adalah sebagai berikut:

- a. VIN

Tegangan *input* ke papan arduino menggunakan sumber daya eksternal (sebagai lawan 5V dari koneksi USB atau sumber daya lainnya). Jika ingin menyediakan tegangan eksternal dapat digunakan pin ini.

- b. 5V

Pin *output* 5V ini diatur dari regulator di *board* dapat diaktifkan dengan daya baik dari colokan listrik DC (7 - 12V), konektor USB (5V), atau pin VIN dari *board* (7-12V).

- c. 3.3V

Sebuah pasokan 3,3 volt yang dihasilkan oleh *regulator on-board*. Arus maksimum adalah 50 mA.

- d. *Ground*

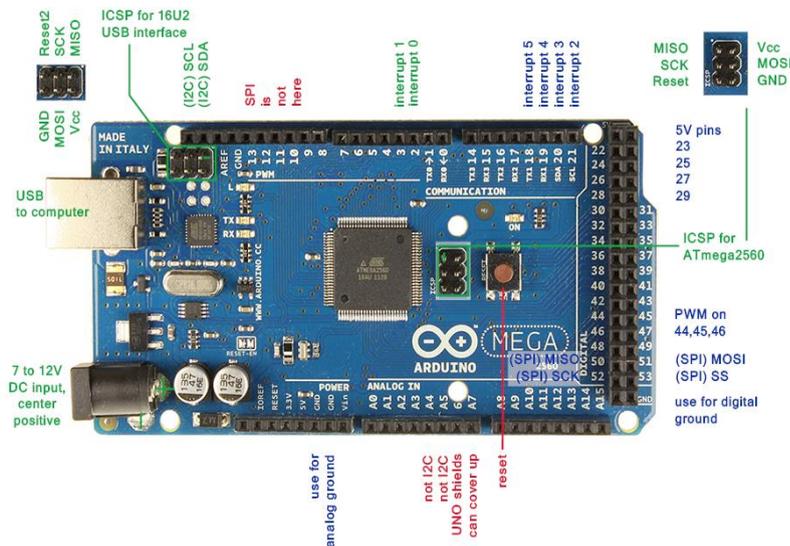
Pin *ground*.

- e. IOREF

Pin ini memberikan tegangan referensi dengan yang mikrokontroler beroperasi. Sebuah perisai dikonfigurasi dengan benar dapat membaca tegangan pin IOREF dan pilih sumber daya yang tepat atau mengaktifkan penerjemah tegangan pada output untuk bekerja dengan 5V atau 3.3V.

2.1.3 Input dan Output

Masing-masing dari 54 pin digital pada arduino mega 2560 dapat digunakan sebagai *input* atau *output*, menggunakan fungsi `pinMode ()`, `digitalWrite ()`, dan `digitalRead ()` fungsi. Mereka beroperasi di 5 volt. Setiap pin dapat memberikan atau menerima 20 mA sebagai kondisi operasi yang direkomendasikan dan memiliki resistor *pull-up internal* yang (terputus secara *default*) dari 20-50 k ohm. Maksimal 40 mA adalah nilai yang tidak boleh melebihi untuk menghindari kerusakan permanen ke mikrokontroler.



Gambar 2.2 Pin Input dan Output Arduino Mega 2560

(Sumber: Arduino, 1990)

Selain itu, beberapa pin memiliki fungsi khusus sebagai berikut:

1. Serial: 0 (RX) dan 1 (TX); Serial 1: 19 (RX) dan 18 (TX); Serial 2: 17 (RX) dan 16 (TX); Serial 3: 15 (RX) dan 14 (TX). Digunakan untuk menerima (RX) dan mengirimkan (TX) TTL data serial. Pin 0 dan 1 juga terhubung ke pin yang sesuai dari Atmega 16U2 USB- to-Serial TTL.
2. *External Interrupts*: 2 (*interrupt 0*), 3 (*interrupt 1*), 18 (*interrupt 5*), 19 (*interrupt 4*), 20 (*interrupt 3*), dan 21 (*interrupt 2*). Pin ini dapat dikonfigurasi untuk memicu interupsi pada nilai yang rendah, yang naik atau jatuh tepi, atau perubahan nilai. Lihat fungsi `attachInterrupt ()` untuk lebih rinci.

3. PWM: 2-13 dan 44 sampai 46. Memberikan 8-bit PWM output dengan fungsi `analogWrite ()`.
4. SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS). Pin ini mendukung komunikasi SPI (*Serial Peripheral Interface*) menggunakan *library* SPI. Pin SPI juga terpisah dari *header* ICSP, yang secara fisik kompatibel dengan Uno, Duemilanove dan Diecimila.
5. LED: 13. Ada *built-in* LED (*Light Emiting Diode*) terhubung ke pin digital 13. Ketika logika pin bernilai nilai tinggi atau *high*, LED akan menyala, ketika logika pin rendah atau *low*, maka LED akan mati atau *off*.

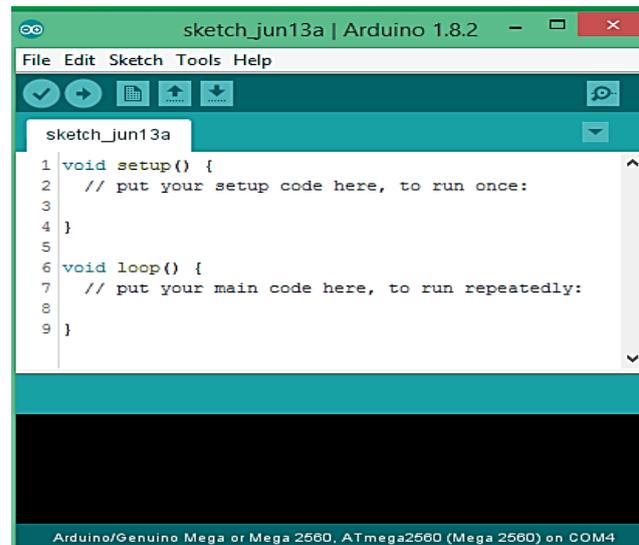
Arduino mega 2560 memiliki 16 *input* analog, yang masing-masing menyediakan 10 bit resolusi (yaitu 1024 nilai yang berbeda). Secara *default* nilai tersebut dari 0V sampai 5V, meskipun mungkin untuk mengubah nilai jangkauan atas (5V) dengan menggunakan pin AREF dan fungsi `analogReference ()` fungsi.

2.1.4 Memori

ATmega2560 memiliki 256 KB dari memori flash untuk menyimpan kode yang 8 KB digunakan untuk *bootloader*, 8 KB dari SRAM dan 4 KB EEPROM (*Electrically Erasable Programmable Read-Only Memory*) yang dapat dibaca dan ditulis dengan perpustakaan EEPROM.

2.1.5 Aplikasi Program IDE (*Integrated Development Environment*)

Ketika kita membuka program Arduino IDE (*Integrated Development Environment*), akan terlihat serupa dengan tampilan gambar 2.3 di bawah ini. Jika kita menggunakan Windows atau Linux, akan terlihat perbedaan, tetapi pada dasarnya IDE akan sama, tidak peduli Operasi Sistem apa yang digunakan.

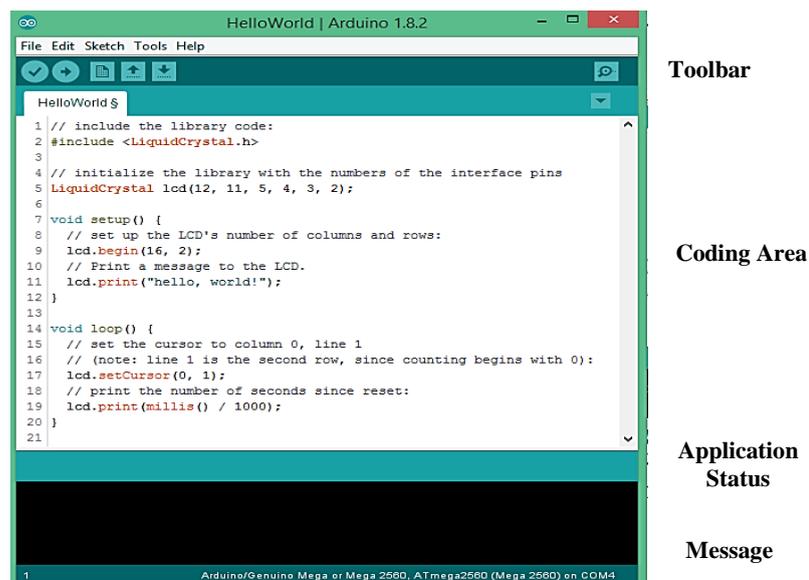


Gambar 2.3 Tampilan Program IDE

(Sumber: Arduino Comp, 2017)

2.1.6 Arduino Programming Tool

Arduino merupakan perangkat pemrograman mikrokontroler jenis AVR yang tersedia secara bebas (*open source*) untuk membuat *prototype* elektronika yang dapat berinteraksi dengan keadaan sekitarnya. Arduino dapat menerima input dari berbagai jenis sensor dan mengendalikan sensor lainnya.

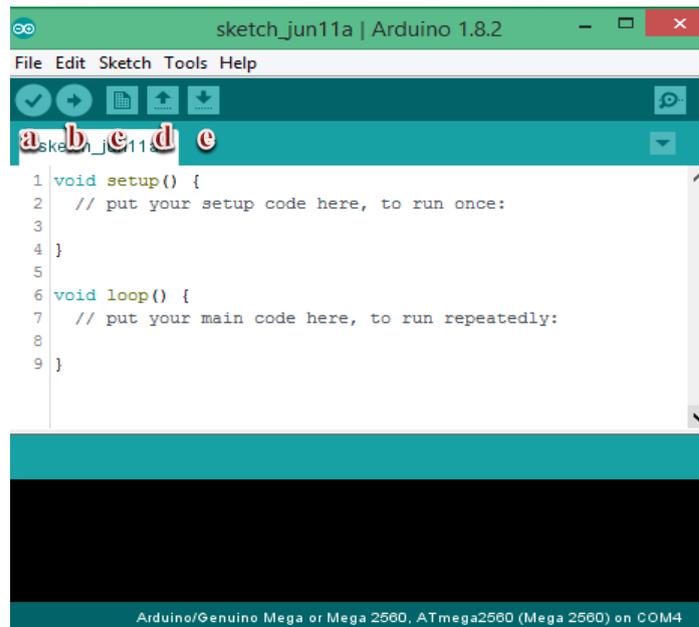


Gambar 2.4 Tampilan Utama Aplikasi Arduino

(Sumber: Arduino Comp, 2017)

1. *Toolbar*

Tombol-tombol toolbar memungkinkan Anda untuk memverifikasi dan meng-upload program, membuat, membuka, menyimpan sketsa, dan juga membuka monitor serial.



Gambar 2.5 Toolbar pada Aplikasi Arduino

(Sumber: *Arduino Comp*, 2015)

a. *Verify*

Tombol ini digunakan untuk meng-*compile* program yang telah dibuat. *Compile* berguna untuk mengetahui apakah program yang telah dibuat benar atau masih terdapat kesalahan. Apabila masih terdapat kesalahan, bagian *message* akan menampilkan letak kesalahan tersebut.

b. *Upload*

Tombol ini digunakan untuk mengirim *coding* yang sudah dikerjakan ke mikrokontroler.

c. *New*

Tombol ini digunakan untuk membuat *coding* pada layar baru.

d. *Open*

Tombol ini digunakan untuk membuka *coding* yang sudah disimpan sebelumnya.

e. Save

Tombol ini digunakan untuk menyimpan *coding* yang sedang dikerjakan.

f. Serial Monitor

Tombol ini digunakan untuk melihat aktivitas komunikasi serial dari mikrokontroler baik yang dikirim oleh *user* ke mikrokontroler maupun sebaliknya.

2. Coding Area

Bagian ini merupakan tempat penulisan *coding* dengan menggunakan bahasa pemrograman C. *Coding* di dalam Arduino memiliki dua bagian utama, yaitu :

a. void setup ()

Bagian ini merupakan inisialisasi yang diperlukan sebelum program utama dijalankan, contoh:

```
1.void setup ( ){
2.Serial.begin (9600) ; // Inisialisasi baudrate komunikasi
3.serial
4.pinMode (6, INPUT) ; // set pin 6 Arduino sebagai input
5.pinMode (7, OUTPUT) ; // set pin 7 Arduino sebagai output }
```

b. void loop ()

Bagian ini merupakan fungsi utama yang dijalankan terus menerus selama modul Arduino terhubung dengan power *supply*. Contoh :

```
1.voidloop ( )
2.{
3.digitalWrite (6, HIGH) ;
4.delay (1000) ; // menunda selama 1 detik
5.digitalWrite (6, LOW) ;
6.delay (2000) ; // menunda selama 2 detik
7.}
```

c. *Application Status*

Bagian ini memberikan informasi kepada pengguna mengenai tugas yang sedang dijalankan oleh aplikasi Arduino.

d. *Message*

Bagian ini memberikan informasi kepada pengguna mengenai besarnya ukuran file dari *coding* yang dibuat dan letak kesalahan yang terjadi pada *coding*.

2.1.7 Tipe-Tipe data Arduino

Setiap bagian dari data yang anda simpan dalam program arduino memiliki tipe datanya masing-masing. Tergantung pada kebutuhan anda, anda dapat memilih dari tipe-tipe data berikut ini :

- a. Tipe data boolean mengambil satu byte memori dan dapat bernilai benar atau salah.
- b. Tipe data *char* mengambil satu byte nomor memori dan menyimpan dari -128 sampai 127. Angka-angka ini biasanya mewakili karakter yang dikodekan dalam ASCII.
- c. Tipe data *int (integer)* membutuhkan dua *byte* memori. Anda dapat menggunakannya untuk menyimpan angka dari -32.768 ke 32.767. *unsigned int* juga menghabiskan dua *byte* memori tetapi menyimpan angka dari 0 sampai 65.535.
- d. Untuk angka yang lebih besar, digunakan tipe data *long*. Mengonsumsi empat *byte* memori dan menyimpan nilai dari -214783648 ke 2147483647. *Unsigned long* juga perlu empat *byte* tetapi menyimpan rentang nilai dari 0 sampai 4.294.967.295.
- e. Tipe data *float* dan *double* adalah tipe data yang sama. Anda dapat menggunakan jenis tipe ini untuk menyimpan angka floating-point. Keduanya menggunakan empat byte memori dan mampu menyimpan nilai-nilai dari -3.4028235E+38 untuk 3.4028235E+38.
- f. Tipe data *void* hanya untuk deklarasi fungsi. Ini menunjukkan bahwa fungsi tersebut tidak mengembalikan nilai.

- g. *Array* menyimpan nilai yang memiliki tipe data yang sama.
- h. Sebuah *string* adalah sebuah *array* nilai *char*. Arduino IDE mendukung penciptaan *string* dengan beberapa sintaksis gula semua ini deklarasi membuat *string* dengan isi yang sama.

2.1.8 Komplikasi dan Program *Uploading*

Sebelum anda mengkompilasi dan meng-*upload* program ke arduino, anda harus mengkonfigurasi dua hal dalam IDE yaitu jenis Arduino anda menggunakan dan port serial arduino anda terhubung. Ketika anda telah mengidentifikasi dengan tepat jenis arduino anda. Memilih dari menu *tools>board*. Sekarang anda harus memilih *port* serial arduino anda terhubung untuk dari >menu serial *port tools*. Pada sistem *windows*, *Device Manager*, dan mencari USB Serial *Port* dibawah *ports* (COM dan LPT) entri menu. Biasanya *port* bernama COM1, COM2, atau sesuatu yang serupa. Setelah anda telah memilih *port* serial kanan, klik tombol *verify* dan anda akan melihat *output* berukut di daerah pesan IDE (yang arduino IDE menyebut program sketsa).

Selama proses upload, TX dan RX LED akan berkedip selama beberapa detik. Ini adalah normal itu terjadi setiap kali Arduino dan komputer anda berkomunikasi melalui port serial. Ketika arduino mengirimkan informasi ternyata pada TX LED. Ketika mendapat beberapa bit, ternyata pada RX LED. (M Bangun Agung. 2014).

2.2 Baterai LiPo 12 Volt

Baterai Lithium Polimer atau biasa disebut dngan LiPo merupakan salah satu jenis baterai yang sering digunakan dalam dunia Robot. Baterai Lipo terdiri dari 3 cell yaitu Lipo1s = 3,7v - 4,2v, Lipo 2s = 7,4v - 8,4v, dan Lipo 3s = 11,1v - 12,6v. Ada tiga kelebihan utama yang ditawarkan oleh baterai berjenis LiPo ketimbang baterai jenis lain yaitu :

1. Baterai LiPo memiliki bobot yang ringan dan tersedia dalam berbagai macam bentuk dan ukuran.
2. Baterai LiPo memiliki kapasitas penyimpanan energi listrik yang besar.

3. Baterai LiPo memiliki tingkat *discharge rate* energi yang tinggi, dimana hal ini sangat berguna sekali dalam bidang RC.

Apabila kapasitas baterai sudah habis, dapat di charge sehingga kapasitas baterai terisi kembali dan dapat digunakan lagi. (Musbikhin, 2014).



Gambar 2.6 Baterai LiPo 12 Volt
(Sumber: Dokumen Penulis)

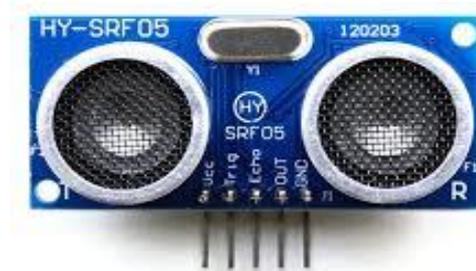
2.3 Sensor Ultrasonik

Sensor Ultrasonik merupakan sebuah sensor yang mampu mendeteksi adanya objek di depannya. Jarak deteksi bergantung terhadap jenis sensor ultrasonik yang digunakan, karena setiap ultrasonik mempunyai spesifikasi tersendiri. Sensor ultrasonik HY-SRF05 merupakan sensor pengukur jarak yang menggunakan pantulan gelombang ultrasonik. Dimana prinsip kerja sensor ultrasonik adalah pemancar (*transmitter*) mengirimkan seberkas gelombang ultrasonik, lalu penerima (*receiver*) menangkap gelombang hasil pantulan hingga datangnya pantulan dari suatu objek. Lamanya waktu ini sebanding dengan dua kali jarak sensor dengan obyek. (Hendri Suhendri, 2013).

Sensor HY-SRF05 terdiri dari PIN VCC, Trigger (*control side*), Echo (*receiving end*), dan GND (*ground*). Modul kerja sensor ini adalah *Input Output* untuk memicu *Start*, untuk sinyal tingkat tinggi setidaknya 10us, modul secara otomatis mengirim delapan gelombang persegi 40kHz, otomatis mendeteksi apakah sinyal kembali, Gelombang ini akan merambat di udara dengan kecepatan 344.424 m/detik (atau 1 cm setiap 29.034uS), mengenai objek untuk kemudian terpantul kembali. (Hadijaya dkk, 2012).

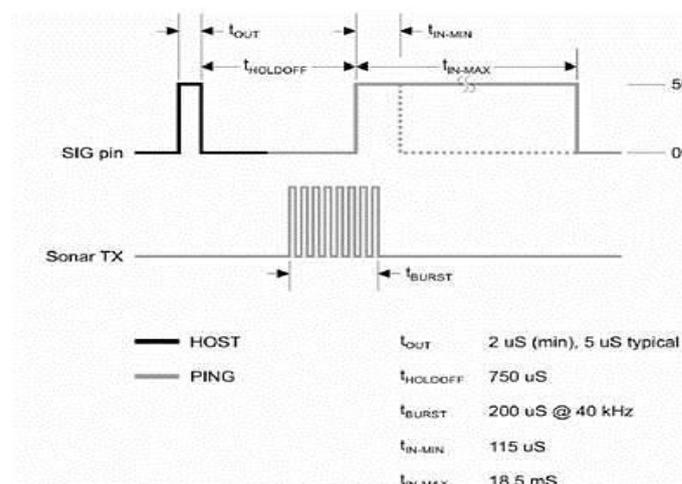
Spesifikasi:

1. Bekerja pada tegangan DC 5 Volt
2. Beban arus sebesar 30 mA – 50 mA
3. Menghasilkan gelombang dengan frekuensi 40 kHz
4. Jangkauan jarak yang dapat dideteksi 3 cm – 400 cm
5. Membutuhkan trigger input minimal sebesar 10 uS



Gambar 2.7 Sensor Ultrasonik

(Sumber: Ebay, 2017)



Gambar 2.8 Timing Diagram HY-SRF05

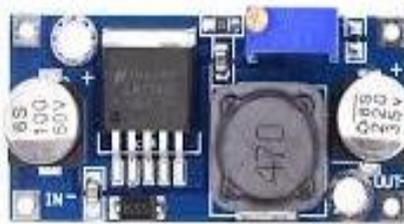
(Sumber: Ebay, 2017)

Untuk mendapatkan jarak pengukuran dengan mengubah dari hasil pengukuran ultrasonik menjadi jarak sebenarnya (cm), sebagai berikut:

$$\text{Jarak} = \frac{\text{Lebar Pulsa}}{2 \times 29.034 \mu\text{S}} \text{ Cm} \dots\dots\dots (2.1)$$

2.4 Modul LM2596

LM2596 merupakan sebuah regulator yang dapat digunakan sebagai penyetabil tegangan. LM2596 berfungsi sebagai *Step-Down DC converter* dengan *current rating* 3A. Regulator seri LM ini memiliki beberapa variasi tegangan output tetap 3,3 Volt, 5 Volt, 9 Volt, 12 Volt dan versi output yang dapat disesuaikan dengan kebutuhan. Seri LM2596 beroperasi pada frekuensi *switching* 150 KHz, sehingga memungkinkan komponen berukuran lebih kecil dari apa yang diperlukan dengan regulator beralih ke frekuensi lebih rendah.



Gambar 2.9 Modul LM2596
(Sumber: repository.umy.ac.id)

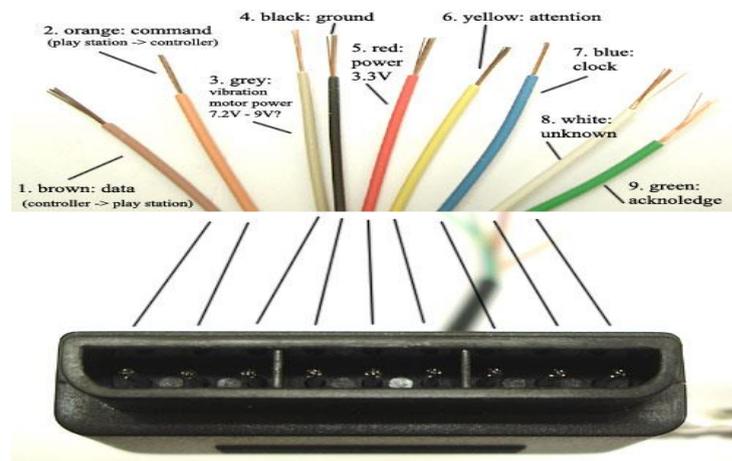
2.5 Joystik PS2

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.



Gambar 2.10 Joystik PS2
(Sumber: [famosastudio](http://famosastudio.com), 2015)

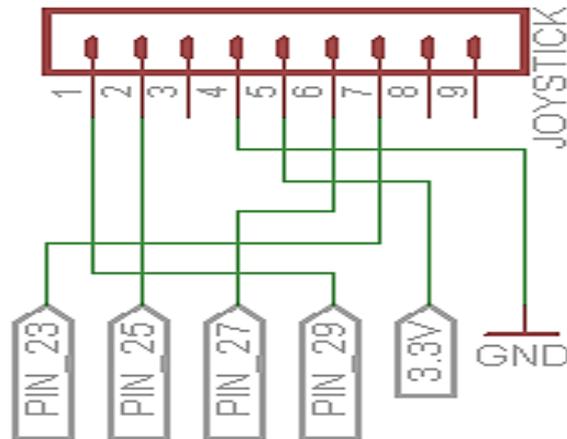
Joystik merupakan sebuah piranti pengendali tak langsung, gerakan robot dikendalikan oleh gerakan tuas pada *joystick* absolut atau dengan tekanan pada tuas. Pada *joystick* biasanya terdapat tombol yang dapat dipilih atau diaplikasikan dengan papan ketik. *Joystick* digunakan untuk mengendalikan alat pengaman mobil dalam pengoperasiannya, *joystick* tidak memerlukan tempat yang luas. Setelah tombol pada *joystick* ditekan maka data akan dikirimkan menuju arduino untuk diproses. Tiap tombol-tombol *joystick* disambungkan dengan port-port input pada arduino 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.



Gambar 2.11 Terdapat 9 Pin Konektor *Joystick* PS2

(Sumber: famosastudio, 2015)

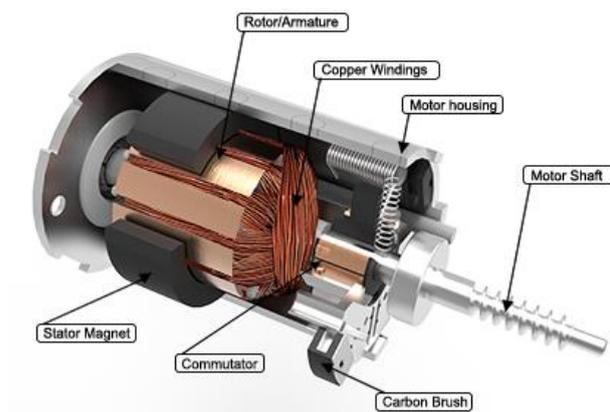
Cara kerja Joystik pada robot ini disesuaikan dengan protokol akses/komunikasi antara PS2 dan *joystick* PS2. Akses data berupa serah terima paket data (beberapa byte) antara *joystick* dan PS2 yang berisi inisiasi komunikasi sampai informasi tombol mana saja yang sedang dioperasikan (ditekan) serta 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 *joystick* nya, alat yang dibuat (menggantikan posisi PS2) akan mengakses data *joystick* PS2. Selanjutnya alat yang dibuat dilabelkan dengan uC untuk mempermudah. Jadi komunikasi antara uC ↔ JoyStick PS2. Untuk mengakses *joystick* PS2, menggunakan komunikasi SPI *Full Duplex*.



Gambar 2.12 Rangkaian Antarmuka *Joystik PS* dengan Arduino
(Sumber: Dokumen Penulis)

2.6 Motor DC

Salah satu komponen yang tidak dapat dilupakan dalam sistem pengaturan adalah aktuator. Aktuator adalah komponen yang selalu bergerak mengubah energi listrik menjadi pergerakan mekanik. Motor DC baik kecepatan, laju dan arah putarnya dapat diatur sesuai dengan keinginan. Motor DC yang kecil bahkan dapat digerakkan dengan tegangan DC yang kecil misalnya motor pada *disk drive* yang digerakkan dengan tegangan 12 Volt.



Gambar 2.13 Bagian - bagian motor DC
(Sumber: www.timotion.com)

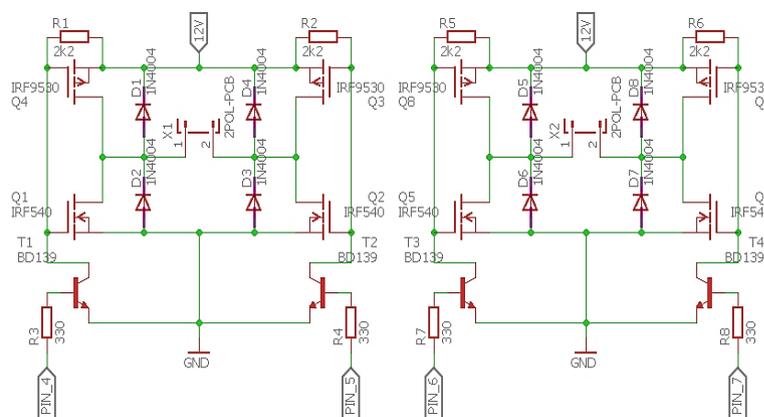
Bagian – bagian pada motor DC, sebagai berikut :

1. Stator: Kumparan medan pada motor dc bagian luar motor yang dapat menghasilkan medan magnet (bagian yang tidak berputar).
2. Rotor: Kumparan jangkar adalah bagian dalam motor yang berputar.
3. Komutator: Sepasang pelat yang melekat pada poros motor. Piring ini menyediakan dua koneksi untuk kumparan elektromagnet. Komutator digunakan untuk membalikkan polaritas arus di motor dan pada dasarnya menjaga motor berputar tanpa kehilangan torsi.
4. Sikat Carbon: Mengirimkan arus listrik dari stator ke rotor di motor.
5. Motor Shaft: Poros motor menghubungkan motor gigi ke bawah stator pada motor DC.

2.7 Driver Motor Mosfet

Driver Motor Mosfet merupakan modifikasi dari rangkaian H Bridge yang berfungsi untuk menggerakkan motor dc dengan arus yang cukup besar (lebih dari 1 Ampere) dan tegangan kerja yang cukup besar. Dapat mengubah arah putaran dan juga kecepatan putar dengan metode PWM (*Pulse Width Modulation*).

Driver Motor Mosfet digunakan untuk mengendalikan motor agar motor dapat dikendalikan melalui arduino, sehingga motor dapat berputar. H-Bridge yang paling sederhana dan mudah dipergunakan dan untuk rangkaianpun menjadi lebih simple dan sederhana.

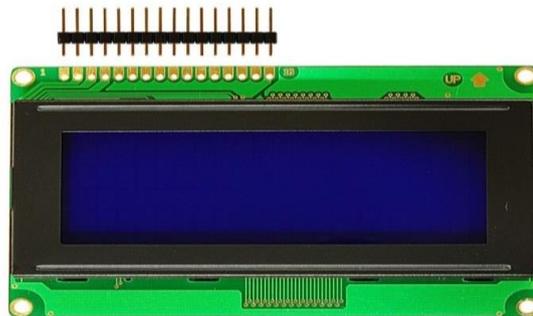


Gambar 2.14 Rangkaian *Driver Motor MOSFET*

(Sumber: Dokumen Penulis)

2.8 LCD (*Liquid Crystal Display*)

LCD adalah lapisan dari campuran organik antara lapisan kaca bening dengan elektroda transparan indium oksida dalam bentuk tampilan *seven-segment* dan lapisan elektroda pada kaca belakang. Ketika elektroda diaktifkan dengan medan listrik (tegangan), molekul organik yang panjang dan silindris menyesuaikan diri dengan elektroda dari segmen. Lapisan sandwich memiliki polarizer cahaya vertikal depan dan polarizer cahaya horizontal belakang yang diikuti dengan lapisan reflektor. Cahaya yang dipantulkan tidak dapat melewati molekul-molekul yang telah menyesuaikan diri dan segmen yang diaktifkan terlihat menjadi gelap dan membentuk karakter data yang ingin ditampilkan.



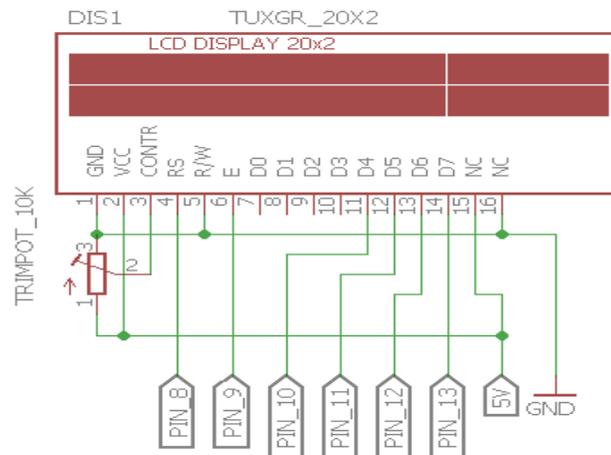
Gambar 2.15 LCD 20x4

(Sumber: www.gravitech.us)

Tabel 2.1 Konfigurasi Pin LCD

Pin No.	Symbol	Level	Description
1	VSS	0V	Ground
2	VDD	5V	Supply voltage for logic
3	VO	(Variable)	Operating voltage for LCD
4	RS	H/L	H : Data, L : Instruction code
5	R/W	H/L	H : Read, L : Write
6	E	H, H->L	Chip Enable signal
7	DB0	H/L	Data bit 0
8	DB1	H/L	Data bit 1
9	DB2	H/L	Data bit 2
10	DB3	H/L	Data bit 3
11	DB4	H/L	Data bit 4
12	DB5	H/L	Data bit 5
13	DB6	H/L	Data bit 6
14	DB7	H/L	Data bit 7
15	A	4,2 – 4,6 V	LED +
16	K	0V	LED -

Sumber: Datasheet Liquid Crystal Display



Gambar 2.16 Rangkaian *Liquid Crystal Display*
(Sumber: Dokumen Penulis)

LCD memiliki dua buah register yang aksesnya diatur menggunakan kaki RS. Pada saat RS berlogika 0, yang diakses adalah register perintah. Saat RS berlogika 1, yang diakses adalah register data.

Register yang berisi perintah-perintah dari mikrokontroler ke LCD pada saat proses penulisan data atau tempat status dari LCD dapat dibaca pada saat pembacaan data.

1. Penulisan Data ke Register Perintah

Penulisan data ke register perintah dilakukan dengan tujuan mengatur tampilan LCD, inisialisasi, dan mengatur *Address Counter* maupun *Address Data*. Kondisi RS dengan logika 0 berarti terjadi akses data ke register perintah. RW berlogika 0 berarti proses penulisan data akan dilakukan.

Pada Mode 4 bit *interface*, data dikirimkan secara terpisah. *Nible* tinggi (bit7 sampai bit4) dikirimkan terlebih dahulu dengan diawali pulsa logika 1 pada E Clock. Kemudian, *nible* rendah (bit3 sampai bit0) dikirimkan dengan diawali pulsa logika 1 pada E Clock. Pada Mode 8 bit *interface*, proses penulisan dapat langsung dilakukan sekaligus 8 bit (bit7 ... bit0) dan diawali pulsa logika 1 pada E Clock.

2. Pembacaan Data dari Register Perintah

Pembacaan data pada register perintah biasanya dilakukan untuk melihat status *busy* dari LCD atau membaca *Address Counter*. RS diberi logika 0 untuk akses ke register perintah, dan R/W diberi logika 1 agar terjadi proses pembacaan data.

2.9 Buzzer

Buzzer adalah komponen elektronika yang berfungsi untuk mengubah getaran listrik menjadi getaran suara. Pada dasarnya prinsip kerja *buzzer* hampir sama dengan *loud speaker*, jadi *buzzer* juga terdiri dari kumparan yang terpasang pada diafragma dan kemudian kumparan tersebut dialiri arus sehingga menjadi electromagnet. Kumparan tadi akan tertarik ke dalam atau keluar, tergantung dari arah arus dan polaritas magnetnya, karena kumparan dipasang pada diafragma maka setiap gerakan kumparan akan menggerakkan diafragma secara bolak-balik sehingga membuat udara bergetar yang akan menghasilkan suara.



Gambar 2.17 Buzzer

(Sumber: <http://www.creativeelectro.com>)

2.10 Self Holding

Sistem *Self Holding* dapat diartikan sebagai sistem menahan diri yang merupakan suatu sistem peralatan yang dirancang untuk mengamankan suatu peralatan yang satu terhadap lainnya. Sistem *Self Holding* dapat dilakukan dengan sistem penguncian gas pada rancang bangun alat pengaman mobil yang mengambil aksi seluruh fungsi keamanan, agar dapat dicegah adanya situasi yang membahayakan baik untuk peralatannya sendiri maupun manusia sehingga mengurangi kecelakaan saat berkendara. (Nugroho Danang. 2015).

Sistem *Self Holding* disini dapat diatur melalui *software* dari program arduino IDE. Misalnya ketika mobil mencapai jarak kurang dari 20cm terhadap objek atau benda di depannya, maka mobil akan berhenti dan melakukan penguncian gas agar terhindar dari tabrakan ketika berkendara sehingga walaupun pengemudi menekan gas maka mobil tidak akan berjalan di jarak bahaya yang terukur dibawah 20cm.

