

BAB II

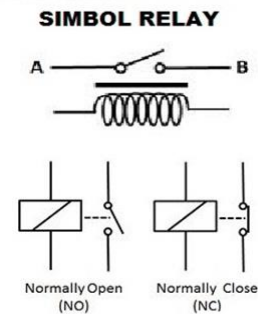
TINJAUAN PUSTAKA

2.1 Relay

Relay adalah komponen elektronika yang berupa saklar atau switch elektrik yang dioperasikan menggunakan listrik. Relay juga biasa disebut sebagai komponen electromechanical atau elektromekanikal yang terdiri dari dua bagian utama yaitu coil atau elektromagnet dan kontak saklar atau mekanikal. Komponen relay menggunakan prinsip elektromagnetik sebagai penggerak kontak saklar, sehingga dengan menggunakan arus listrik yang kecil atau low power, dapat menghantarkan arus listrik yang memiliki tegangan lebih tinggi. Berikut adalah gambar dan juga simbol dari komponen relay.



(a)



(b)

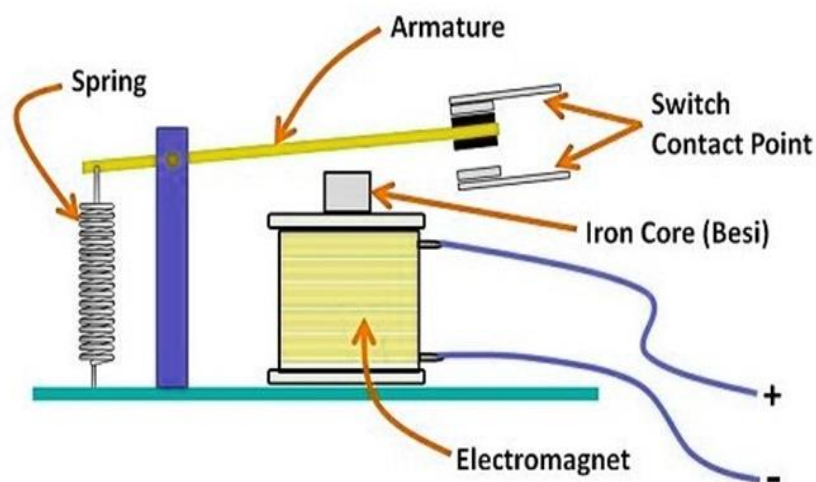
Gambar 2.1 (a) Relay, (b) Simbol Relay

Relay memiliki fungsi sebagai saklar elektrik. Namun jika diaplikasikan ke dalam rangkaian elektronika, relay memiliki beberapa fungsi yang cukup unik. Berikut adalah beberapa fungsi komponen relay saat diaplikasikan ke dalam sebuah rangkaian elektronika.

1. Mengendalikan sirkuit tegangan tinggi dengan menggunakan bantuan signal tegangan rendah.
2. Menjalankan fungsi logika alias logic function.
3. Memberikan fungsi penundaan waktu alias time delay function.
4. Melindungi motor atau komponen lainnya dari kelebihan tegangan atau korsleting.

2.1.1 Cara Kerja Relay

Setelah mengetahui pengertian dan fungsi relay, berikut adalah cara kerja atau prinsip kerja relay yang juga harus anda ketahui. Namun sebelumnya anda perlu tahu bahwa dalam sebuah relay terdapat 4 buah bagian penting yakni Electromagnet (Coil), Armature, Switch Contact Point (Saklar), dan Spring. Untuk info lebih jelasnya silahkan lihat gambar di bawah ini.



Gambar 2.2 Cara Kerja Relay

Dari gambar tersebut dapat diketahui bahwa sebuah Besi (Iron Core) yang dililit oleh kumparan Coil, berfungsi untuk mengendalikan Besi tersebut. Apabila Kumparan Coil dialiri arus listrik, maka akan muncul gaya elektromagnetik yang

dapat menarik Armature sehingga dapat berpindah dari posisi sebelumnya tertutup (NC) menjadi posisi baru yakni terbuka (NO).

Dalam posisi (NO) saklar dapat menghantarkan arus listrik. Pada saat tidak dialiri arus listrik, Armature akan kembali ke posisi awal (NC). Sedangkan Coil yang digunakan oleh relay untuk menarik Contact Poin ke posisi close hanya membutuhkan arus listrik yang relatif cukup kecil. Oh iya, buat anda yang belum tahu apa itu NO dan NC, berikut penjelasannya.

1. NC atau Normally Close adalah kondisi awal relay sebelum diaktifkan selalu berada di posisi CLOSE (tertutup).
2. NO atau Normally Open adalah kondisi awal relay sebelum diaktifkan selalu berada di posisi OPEN (terbuka).

2.2 Pompa Air

Pompa adalah suatu alat yang digunakan untuk memindahkan suatu cairan dari suatu tempat ke tempat lain dengan cara menaikkan tekanan cairan tersebut. Kenaikan tekanan cairan tersebut digunakan untuk mengatasi hambatan-hambatan pengaliran. Hambatan-hambatan pengaliran itu dapat berupa perbedaan tekanan, perbedaan ketinggian atau hambatan gesek.

Pada prinsipnya, pompa mengubah energi mekanik motor menjadi energi aliran fluida. Energi yang diterima oleh fluida akan digunakan untuk menaikkan tekanan dan mengatasi tahanan – tahanan yang terdapat pada saluran yang dilalui.

Pompa memiliki dua kegunaan utama:

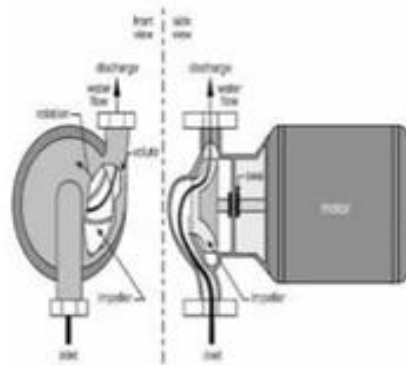
1. Memindahkan cairan dari satu tempat ke tempat lainnya (misalnya air dari akuifer bawah tanah ke tangki penyimpanan air).
2. Mensirkulasikan cairan sekitar sistim (misalnya air pendingin atau pelumas yang melewati mesin-mesin dan peralatan).



Gambar 2.3 Pompa Air

Pompa juga dapat digunakan pada proses - proses yang membutuhkan tekanan hidraulik yang besar. Hal ini bisa dijumpai antara lain pada peralatan - peralatan berat. Dalam operasi, mesin - mesin peralatan berat membutuhkan tekanan discharge yang besar dan tekanan isap yang rendah. Akibat tekanan yang rendah pada sisi isap pompa maka fluida akan naik dari kedalaman tertentu, sedangkan akibat tekanan yang tinggi pada sisi discharge akan memaksa fluida untuk naik sampai pada ketinggian yang diinginkan.

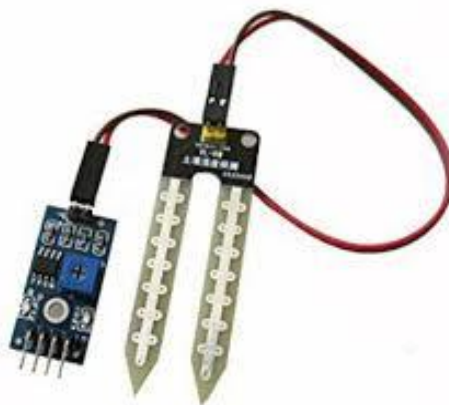
Pada prinsipnya, sebuah pompa air menyedot dan membuang air dengan menggunakan putaran impeler sehingga menimbulkan tarikan, air yang ditarik akan terus menerus menarik air dari dasar sumur untuk dialirkan menuju pipa out kemudian pada pipa out, impeler akan mendorong air untuk menuju kepenampungan atau pembuangan. Pompa air termasuk motor listrik jenis kapasitor running Motor listrik ini mempunyai kapasitor yang dihubungkan seri dengan kumparan bantu, terhubung paralel dengan kumparan utama dan terhubung langsung paralel dengan sumber listrik. Belitan utama, lilitan bantu dan kapasitor tetap terhubung pada sirkuit jala-jala saat motor listrik bekerja.



Gambar 2.4 Prinsip kerja pompa air

2.3 Sensor *Moisture*

Sensor *Moisture* adalah sensor kelembaban yang dapat mendeteksi kelembaban dalam tanah. Sensor ini sangat sederhana, tetapi ideal untuk memantau taman kota, atau tingkat air pada tanaman pekarangan anda. Sensor ini terdiri dua probe untuk melewati arus melalui tanah, kemudian membaca resistansinya untuk mendapatkan nilai tingkat kelembaban. Semakin banyak air membuat tanah lebih mudah menghantarkan listrik (resistansi kecil), sedangkan tanah yang kering sangat sulit menghantarkan listrik (resistansi besar). Sensor ini sangat membantu Anda untuk mengingatkan tingkat kelembaban pada tanaman Anda atau memantau kelembaban tanah.



Gambar 2.5 Sensor *Moisture*

2.4 Perangkat Arduino

Arduino adalah pengendali mikro *single-board* yang bersifat *open-source*, diturunkan dari *wiring platform*, dirancang untuk memudahkan penggunaan elektronik dalam berbagai bidang. *Hardware* memiliki prosesor AtmelAVR dan *software* memiliki bahasa pemrograman sendiri. Arduino juga merupakan *platform hardware* terbuka yang ditujukan kepada siapa saja yang ingin membuat purwarupa peralatan elektronik interaktif berdasarkan *hardware* dan *software* yang fleksibel dan mudah digunakan. Mikrokontroler diprogram menggunakan bahasa pemrograman arduino yang memiliki kemiripan syntax dengan bahasa pemrograman C. Karena sifatnya yang terbuka maka dapat mengunduh skema *hardware* arduino dan membangunnya dengan mudah.

Arduino menggunakan keluarga mikrokontroler ATmega yang dirilis oleh Atmel sebagai basis, namun ada individu atau perusahaan yang membuat *clone* arduino dengan menggunakan mikrokontroler lain dan tetap kompatibel dengan arduino pada level *hardware*. Untuk fleksibilitas, program dimasukkan melalui *bootloader* meskipun ada opsi untuk *bypass bootloader* dan menggunakan *downloader* untuk memprogram mikrokontroler secara langsung melalui port ISP.

2.4.1 Arduino

Semuanya berawal dari sebuah thesis yang dibuat oleh Hernando Barragan, di institute Ivrea, Italia pada tahun 2005, dikembangkan oleh Massimo Banzi dan David Cuartielles dan diberi nama Arduin of Ivrea, lalu berganti nama menjadi Arduino yang dalam bahasa Italia berarti teman yang berani. Tujuan awal dibuat Arduino adalah untuk membuat perangkat mudah dan murah, dari perangkat yang ada saat itu. Perangkat tersebut ditujukan untuk para siswa yang akan membuat perangkat desain dan interaksi.

Saat ini tim pengembangnya adalah Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino, David Mellis, dan Nicholas Zambetti. Mereka mengupayakan 4 hal dalam Arduino ini, yaitu:

1. Harga terjangkau.

2. Dapat dijalankan diberbagai sistem operasi, Windows, Linux, Max, dan sebagainya.
3. Sederhana, dengan bahasa pemograman yang mudah bisa dipelajari orang awam, bukan untuk orang teknik saja.
4. *Open Source, hardware* maupun *software*.

Sifat Arduino yang *Open Source*, membuat Arduino berkembang sangat cepat banyak bermunculan perangkat-perangkat sejenis Arduino. Seperti DFRduino atau Freeduino, dan kalau yang lokal ada namanya Cipaduino yang dibuat oleh SKIR70, terus ada Murmerduino yang dibuat oleh Robot Unyil, ada lagi AViShaduino yang salah satu pembuatnya adalah Admin Kelas Robot.

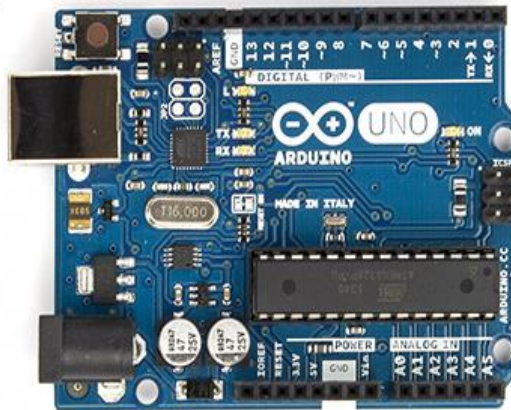
Sampai saat ini pihak resmi, sudah membuat berbagai jenis-jenis Arduino. Mulai dari yang paling mudah dicari dan paling banyak digunakan, yaitu Arduino Uno. Hingga Arduino yang sudah menggunakan ARM Cortex, berbentuk *Mini PC*. Dan sudah ada ratusan ribu Arduino yang digunakan di gunakan di dunia pada tahun 2011. Arduino juga sudah banyak dipakai oleh perusahaan besar. Contohnya Google menggunakan Arduino untuk *Accessory Development Kit*, NASA memakai Arduino untuk *prototypin*, ada lagi Large Hadron Colider memakai Arduino dalam beberapa hal untuk pengumpulan data.

2.4.2 Jenis-Jenis Arduino

Dan seperti Mikrokontroler yang banyak jenisnya, Arduino lahir dan berkembang, kemudian muncul dengan berbagai jenis, diantaranya adalah :

2.4.2.1 Arduino Uno

Arduino Jenis ini adalah yang paling banyak digunakan,terutama untuk pemula sangat disarankan untuk menggunakan Arduino Uno. Dan banyak sekali referensi yang membahas Arduino Uno. Versi yang terakhir adalah Arduino Uno R3 (Revisi 3), menggunakan ATMEGA328 sebagai Mikrokontrolernya, memiliki 14 pin I/O (*input/output*) digital dan 6 pin masukan analog. Untuk pemograman cukup menggunakan koneksi USB *type A to type B* Sama seperti yang digunakan pada USB *printer*.



Gambar 2.6 Bentuk Fisik Arduino Uno

2.4.2.2 Arduino Due

Berbeda dengan arduino yang lain, Arduino *due* tidak menggunakan ATMEGA, melainkan dengan *chip* yang lebih tinggi ARM Cortex CPU. Memiliki 54 I/O pin digital dan 12 pin masukan analog. Untuk pemrogramannya menggunakan *micro USB*, terdapat pada beberapa *handphone*.



Gambar 2.7 Bentuk Fisik Arduino Due

2.4.2.3 Arduino Mega

Mirip dengan Arduino Uno, sama-sama menggunakan USB type A to B untuk pemrogramannya. Tetapi Arduino Mega, menggunakan *chip* yang lebih tinggi ATMEGA 2560 dan tentu saja untuk Pin I/O digital dan pin masukan analognya lebih banyak dari Uno.



Gambar 2.8 Bentuk Fisik Arduino Mega

2.4.2.4 Arduino Leonardo

Bisa dibilang Leonardo adalah saudara kembar dari Uno. Mulai dari jumlah pin I/O digital dan pin masukan Analog yang sama. Pada Leonardo menggunakan *micro* USB untuk pemrogramannya.



Gambar 2.9 Bentuk Fisik Arduino Leonardo

2.4.2.5 Arduino Fio

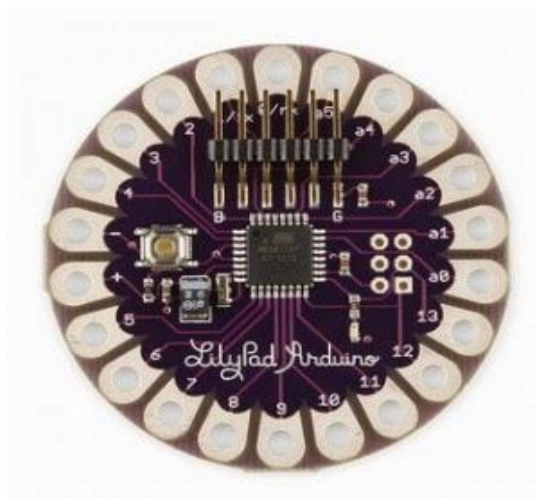
Arduino Fio walau jumlah pin I/O digital dan masukan analog sama dengan uno dan leonardo, tapi Fio memiliki Socket XBee. XBee membuat Fio dapat digunakan untuk keperluan proyek yang berhubungan dengan *wireless*.



Gambar 2.10 Bentuk Fisik Arduino Fio

2.4.2.6 Arduino Lilypad

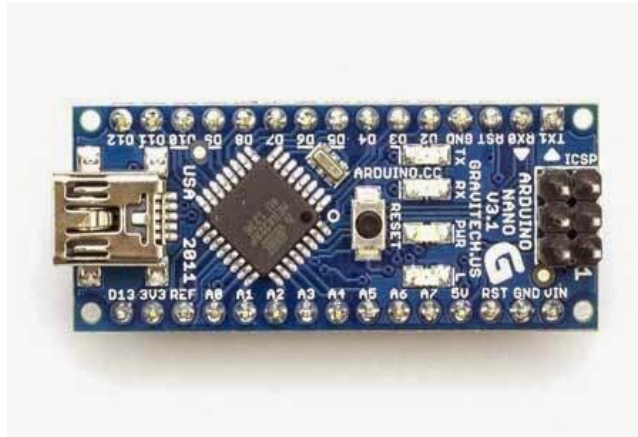
Arduino Lilypad mempunyai bentuk *board* yang melingkar. Lilypad versi lama menggunakan ATMEGA168 Dengan 14 pin I/O digital, dan 6 pin masukan analognya.



Gambar 2.11 Bentuk Fisik Arduino Lilypad

2.4.2.7 Arduino Nano

Sepertinya namanya, Nano yang berukuran kecil menyimpan banyak fasilitas dengan dilengkapi dengan FTDI untuk pemrograman lewat *micro* USB. 14 pin I/O digital, dan 8 Pin masukan Analog (lebih banyak dari Uno) Dan ada yang menggunakan ATMEGA 168, atau ATMEGA 328.



Gambar 2.12 Bentuk Fisik Arduino Nano

2.4.2.8 Arduino Mini

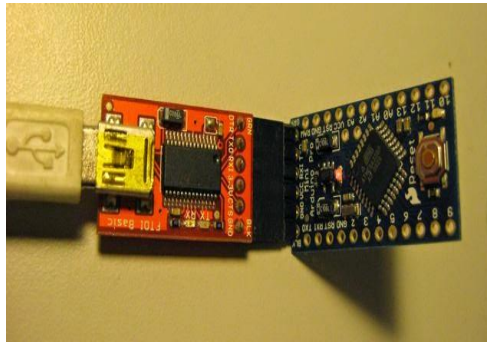
Fasilitasnya sama dengan yang dimiliki Nano yang dilengkapi dengan *micro* USB untuk pemrograman dengan ukuran hanya 30 mm x 18 mm.



Gambar 2.13 Bentuk Fisik Arduino Mini

2.4.2.9 Arduino *Micro*

Ukurannya lebih panjang dari Nano dan Mini. Karena memang fasilitasnya lebih banyak yaitu; memiliki 20 pin I/O digital dan 12 pin masukan analog.



Gambar 2.14 Bentuk Fisik Arduino *Micro*

2.4.2.10 Arduino *Ethernet*

Arduino yang sudah dilengkapi dengan fasilitas *ethernet*. Membuat Arduino dapat berhubungan melalui jaringan LAN pada komputer. Untuk fasilitas pada Pin I/O Digital dan *Input* Analognya sama dengan Uno.



Gambar 2.15 Bentuk Fisik Arduino *Ethernet*

2.4.2.11 Arduino Esplora

Arduino Esplora adalah papan mikrokontroler berasal dari Arduino Leonardo. Arduino Esplora berbeda dari semua papan Arduino sebelumnya karena Arduino Esplora sudah dilengkapi dengan *Joystick*, *button*, dan sebagainya. Arduino Esplora menggunakan mikrokontroler AVR Atmega 32U4.



Gambar 2.16 Bentuk Fisik Arduino Esplora

2.4.2.12 Arduino BT

Arduino BT mikrokontroler Arduino yang mengandung modul *Bluetooth* untuk komunikasi *wireless*.



Gambar 2.17 Bentuk Fisik Arduino BT

2.5 Modul GSM 800L

SIM800L adalah modul SIM yang digunakan pada penelitian ini. Modul SIM800L GSM/GPRS adalah bagian yang berfungsi untuk berkomunikasi antara pemantau utama dengan *Handphone*. ATCommand adalah perintah yang dapat diberikan modem GSM/CDMA seperti untuk mengirim dan menerima data berbasis GSM/GPRS, atau mengirim dan menerima SMS. SIM800L GSM/GPRS dikendalikan melalui perintah AT.

AT+Command adalah sebuah kumpulan perintah yang digabungkan dengan karakter lain setelah karakter „AT“ yang biasanya digunakan pada komunikasi serial. Dalam penelitian ini ATcommand digunakan untuk mengatur atau memberi perintah modul GSM/CDMA. Perintah ATCommand dimulai dengan karakter “AT” atau “at” dan diakhiri dengan kode (0x0d).



Gambar 2.18 Modul GSM SIM800L

Berikut ini spesifikasi dari modul gsm sim 800L :

1. Quad-Band 850/900/1800/1900 MHz.
2. Terhubung dengan jaringan GSM global menggunakan 2G SIM.

3. Voice call dengan external 8 speaker dan electret microphone.
4. Kirim dan terima SMS.
5. Kirim dan terima GPRS data (TCP/IP , HTTP).
6. GPIO ports , misalnya untuk buzzer dan vibrational motor.
7. AT command interface dengan deteksi “auto baud”.

Modul GSM merupakan bagian dari pusat kendali yang berfungsi sebagai transceiver. Modul GSM mempunyai fungsi yang sama dengan sebuah telepon seluler yaitu mampu melakukan fungsi pengiriman dan penerimaan SMS. Dengan adanya sebuah modul GSM maka aplikasi yang dirancang dapat dikendalikan dari jarak jauh dengan menggunakan jaringan GSM sebagai media akses.. Sebuah modem GSM terdiri dari beberapa bagian, di antaranya adalah lampu indikator, terminal daya, terminal kabel ke komputer, antena dan untuk meletakkan kartu SIM.

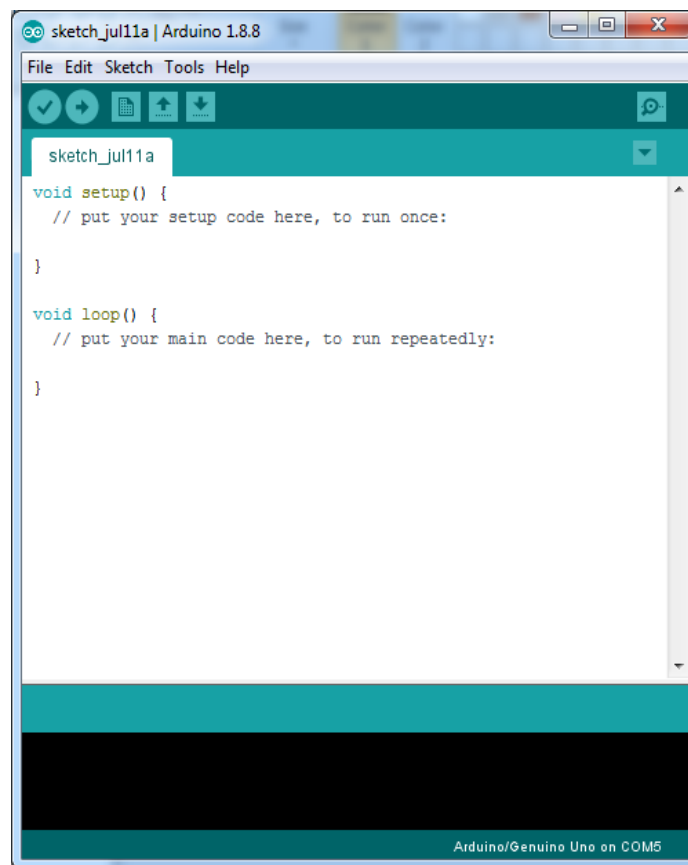


Gambar 2.19 Datasheet SIM800L

2.6 Program Arduino (IDE)

Software IDE Arduino adalah pengendali mikro single-board yang bersifat open-source, diturunkan dari platform Wiring, dirancang untuk memudahkan penggunaan elektronik dalam berbagai bidang, hardware-nya menggunakan

prosesor Atmel AVR dan software-nya memiliki bahasa pemrograman C++ yang sederhana dan fungsi-fungsinya yang lengkap, sehingga arduino mudah dipelajari oleh pemula. Gambar 2.15 memperlihatkan tampilan awal software IDE Arduino (Andrianto, 2016). Arduino IDE dibuat dari bahasa pemrograman JAVA. Arduino IDE juga dilengkapi dengan library C/C++ yang biasa disebut *Wiring* yang membuat operasi input dan output menjadi lebih mudah. Arduino IDE ini dikembangkan dari software Processing yang dirombak menjadi Arduino IDE khusus untuk pemrograman dengan Arduino.



Gambar 2.20 Program Arduino (IDE)

2.7 Bahasa Pemrograman Arduino Berbasis C

Struktur dasar dalam pemrograman arduino terdiri atas dua bagian, yaitu fungsi persiapan (*setup()*) dan fungsi utama (*loop()*). Fungsi *setup()* digunakan untuk mendefinisikan variabel-variabel yang digunakan dalam program,

sedangkan *loop()* adalah program inti/utama dari arduino yang dijalankan secara terus menerus (Andrianto, 2016).

Berikut adalah fungsi-fungsi dasar dalam bahasa pemrograman C pada arduino :

1. *setup()*

Fungsi *setup()* dipanggil ketika program dijalankan, berfungsi untuk inisialisasi mode *pin* sebagai *input* atau *output* dan inisialisasi serial. Fungsi ini harus ada meski tidak ada instruksi yang ditulis.

2. *loop()*

Program yang berada dalam fungsi *loop()* akan dieksekusi secara terus menerus.

3. *Function*

Fungsi adalah sekumpulan blok instruksi yang memiliki nama sendiri dan blok instruksi ini akan dieksekusi ketika fungsi ini dipanggil. Penulisan fungsi ini harus didahului dengan tipe fungsi setelah itu nama fungsi dan kemudian parameternya, bila tidak ada nilai yang dihasilkan dari fungsi tersebut, tipe fungsinya adalah *void()*.

4. {} (kurung kurawal)

Digunakan untuk mengawali dan mengakhiri sebuah fungsi, blok instruksi seperti *loop()*, *void()* dan instruksi *for* dan *if*.

5. ; (titik koma)

Digunakan sebagai tanda akhir instruksi.

6. /*.....*/ (blok komentar)

Digunakan pada komentar yang memiliki baris lebih dari satu. Apapun yang ditulis dalam blok komen ini tidak berpengaruh terhadap program yang dibuat dan tidak akan menghabiskan memori.

7. // (komentar baris)

Sama seperti blok komentar hanya saja digunakan untuk satu baris komentar.

8. Variabel

Adalah suatu ekspresi yang digunakan untuk mewakili suatu nilai yang

digunakan dalam program. Suatu variabel akan menampung nilai sesuai definisi yang telah dibuat. Variabel hanya perlu didefinisikan satu kali saja tetapi nilainya dapat sesuai program. Terdapat dua macam variabel. Ada variabel global yang dapat digunakan oleh semua fungsi dan instruksi dalam program. Variabel ini didefinisikan pada awal program sebelum fungsi *setup()*. Dan ada variabel lokal yang mana variabel ini didefinisikan pada suatu fungsi atau dalam fungsi *loop*. Variabel ini hanya dapat dilihat dan digunakan di dalam fungsi tersebut.

Tipe-tipe data dalam variabel :

- a. *Byte* 26
- b. *Int*
- c. *Long*
- d. *Float*

9. *Array*

Array adalah kumpulan nilai yang diakses dengan nomor indeks. Setiap nilai dalam *array* dapat dipanggil dengan memanggil nama *array* dan nomor tersebut.

10. Aritmatika

Operator aritmatika meliputi penambahan, pengurangan, perkalian, dan pembagian.

11. Operasi Gabungan

Adalah operasi matematika gabungan yang biasa digunakan dalam program.

12. Operator Perbandingan

Operator untuk membandingkan 2 konstanta atau variabel yang sering digunakan untuk menguji suatu kondisi benar atau salah.

13. Operator Logika

Operator logika, *AND*, *OR*, dan *NOT* sering digunakan dalam pernyataan *if*.

14. Konstanta

Bahasa arduino memiliki nilai-nilai yang telah ditetapkan yang disebut konstanta. Mereka digunakan untuk membuat program lebih mudah dibaca.

15. *TRUE / FALSE*

Adalah konstanta Boolean yang mendefinisikan nilai logika.

16. *HIGH / LOW*

Konstanta ini menentukan nilai pin sebagai *HIGH* atau *LOW* dan digunakan ketika membaca atau menulis ke pin digital. *HIGH* didefinisikan sebagai tingkat logika 1/ON/5 Volt, sedangkan *LOW* adalah tingkat logika 0/OFF/0 Volt.

17. *Input / Output*

Konstanta yang digunakan pada fungsi *pinMode()* untuk menentukan mode pin digital sebagai *input* atau *output*.

18. *If*

Instruksi untuk menguji apakah suatu kondisi telah tercapai, seperti membandingkan nilai variabel berada diatas jumlah tertentu, dan menjalankan setiap instruksi di dalam kurung jika pernyataan tersebut benar, jika tidak maka akan dilewati.

19. *if.....else*

Memungkinkan untuk mengeksekusi instruksi yang lain jika suatu kondisi tidak terpenuhi. *else* juga dapat digunakan lebih dari satu.

20. *for*

Pernyataan *for* digunakan untuk mengulang suatu blok instruksi dalam kurung kurawal.

21. *while*

Fungsi *while* akan menjalankan program secara terus menerus hingga suatu kondisi pada fungsi *while* bernilai salah.

22. *do.....while*

Perintah untuk melakukan sesuatu secara terus menerus hingga mencapai suatu kondisi yang tidak memenuhi kondisi yang diinginkan.

23. *pinMode(pin,Mode)*

Instruksi yang digunakan pada fungsi *setup()* untuk menginisialisasi suatu pin sebagai *input* atau *output*.

24. *digitalRead(pin)*

Instruksi yang digunakan untuk membaca *input* dari suatu pin yang hasilnya berupa logika *HIGH* atau *LOW*. Pin dapat diartikan sebagai suatu variabel atau konstanta 0-13 yang mewakili *input* dan *output* dari *board* arduino.

25. *digitalWrite(pin,value)*

Instruksi untuk memberikan nilai output *HIGH* (1) atau *LOW* (0) pada pin digital.

26. *analogRead(pin)*

Instruksi untuk membaca nilai *input* analog dengan resolusi 10 bit. Instruksi ini hanya berlaku untuk pin A0-A5 yang mampu membaca nilai analog. Karena beresolusi 10 bit maka hasil pembacaan digital adalah 0 sampai 1023.

27. *analogWrite(pin,value)*

Instruksi yang berfungsi untuk memberi nilai PWM (*pulse width modulation*) pada *output*. Pada arduino pin PWM ditandai dengan *tilde* (~), yaitu pin 3,5,6,9,10, dan 11.

28. *delay(ms)*

Instruksi untuk memberi jeda sebelum lanjut ke program selanjutnya.

29. *millis()*

Instruksi untuk mengambil nilai waktu sejak program dijalankan hingga program berhenti atau dimatikan.

30. *tone(pin,frekuensi,durasi)*

Instruksi untuk menghasilkan nada frekuensi dengan durasi tertentun dan dikirimkan ke pin yang dituju.

31. *noTone(pin)*

Instruksi untuk menghentikan frekuensi yang dihasilkan pada pin yang dituju.

32. *randomSeed(seed)*

Instruksi untuk mengambil nilai acak dengan seed nilai awal fungsi.

33. *random(max); random(min,max)*

Instruksi *random(max)* berfungsi mengambil nilai acak dengan max sebagai batas nilai maksimal fungsi *random()*. Instruksi *random(min,max)* berfungsi untuk mengambil nilai acak diantara nilai *min* dan *max*.

34. *Serial.begin(rate)*

Instruksi untuk membuka port data serial untuk komunikasi serial baik mengirim atau menerima data dari serial. Rate adalah *baud rate* yang digunakan untuk komunikasi serial (biasa digunakan 9600).

2.8 Kapasitor

Kapasitor atau disebut juga dengan Kondensator adalah Komponen Elektronika Pasif yang dapat menyimpan muatan listrik dalam waktu sementara dengan satuan kapasitansinya adalah Farad. Satuan Kapasitor tersebut diambil dari nama penemunya yaitu Michael Faraday (1791 ~ 1867) yang berasal dari Inggris. Namun Farad adalah satuan yang sangat besar, oleh karena itu pada umumnya Kapasitor yang digunakan dalam peralatan Elektronika adalah satuan Farad yang dikecilkan menjadi pikoFarad, NanoFarad dan MicroFarad.

Konversi Satuan Farad adalah sebagai berikut :

$$1 \text{ Farad} = 1.000.000 \mu\text{F} \text{ (mikro Farad)}$$

$$1 \mu\text{F} = 1.000 \text{nF} \text{ (nano Farad)}$$

$$1 \mu\text{F} = 1.000.000 \text{pF} \text{ (piko Farad)}$$

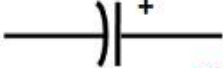
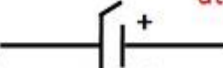
$$1 \text{nF} = 1.000 \text{pF} \text{ (piko Farad)}$$

Kapasitor merupakan Komponen Elektronika yang terdiri dari 2 pelat konduktor yang pada umumnya adalah terbuat dari logam dan sebuah Isolator diantaranya sebagai pemisah. Dalam Rangkaian Elektronika, Kapasitor disingkat dengan huruf “C”.

2.8.1 Jenis – Jenis Kapasitor

1. Kapasitor Nilai Tetap

Kapasitor Nilai Tetap atau Fixed Capacitor adalah Kapasitor yang nilainya konstan atau tidak berubah-ubah. Berikut ini adalah Jenis-jenis Kapasitor yang nilainya Tetap :

Nama Komponen	Gambar	Simbol
Kapasitor Keramik (Ceramic Capacitor)		
Kapasitor Polyester (Polyester Capacitor)		
Kapasitor Kertas (Paper Capacitor)		
Kapasitor Mika (Mica Capacitor)		
Kapasitor Elektrolit (Electrolyte Capacitor)		 atau  atau 
Kapasitor Tantalum (Tantalum Capacitor)		

Gambar 2.21 Daftar tabel komponen kapasitor tetap

2. Kapasitor Variabel

Kapasitor Variabel adalah Kapasitor yang nilai Kapasitansinya dapat diatur atau berubah-ubah. Secara fisik, Kapasitor Variabel ini terdiri dari 2 jenis yaitu :

KAPASITOR VARIABEL (VARIABLE CAPACITOR)

Nama Komponen	Gambar	Simbol
VARCO (Variable Condensator)		
Trimmer		

teknikelektronika.com

Gambar 2.22 Daftar tabel kapasitor variable