

BAB II TINJAUAN PUSTAKA

2.1 Mikrokontroler AVR ATmega328P - PU

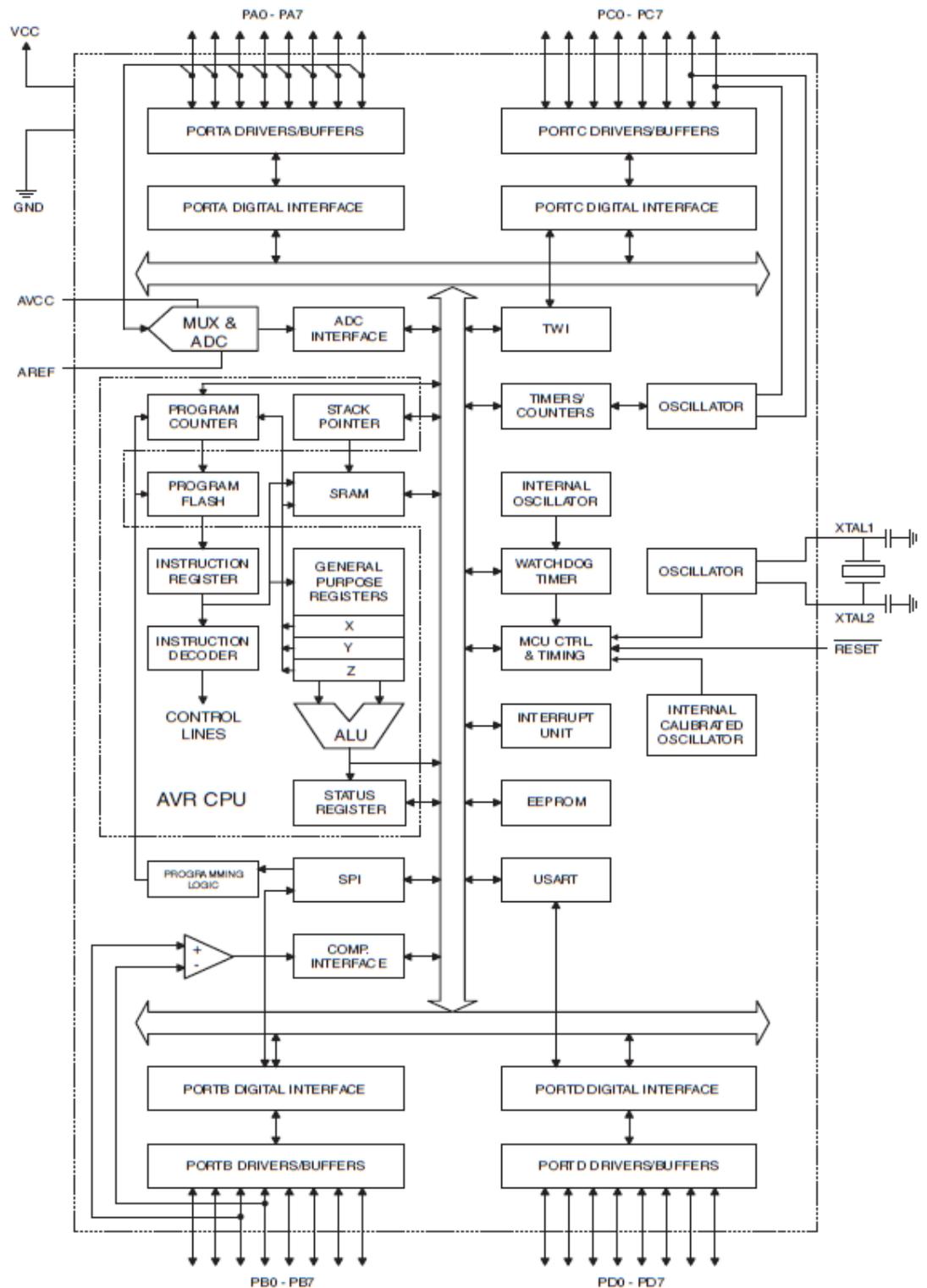
Mikrokontroler AVR ATmega32 merupakan CMOS dengan konsumsi daya rendah, mempunyai 8-bit proses data (CPU) berdasarkan arsitektur AVR RISC. Dengan mengeksekusi instruksi dalam satu (siklus) clock tunggal, ATmega32 memiliki kecepatan data rata-rata (throughputs) mendekati 1 MIPS per MHz, yang memungkinkan perancang sistem dapat mengoptimalkan konsumsi daya dan kecepatan pemrosesan.



Gambar 2.1 ATmega32

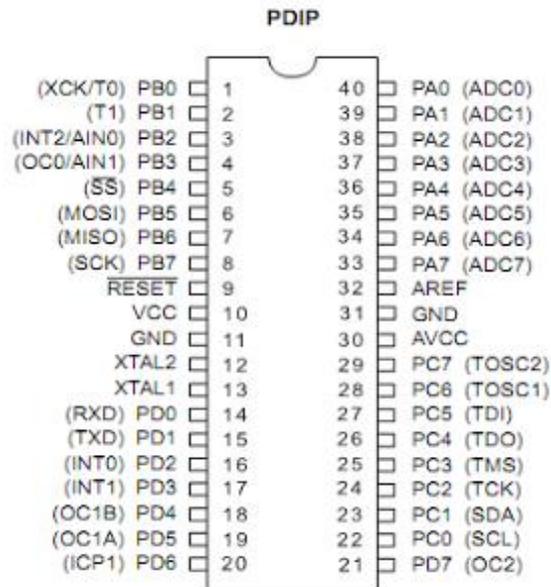
(Sumber: atmel datasheet, 2011)

Pada diagram blok ATmega32 digambarkan 32 *general purpose Working register* yang dihubungkan secara langsung dengan *Arithmetic Logic Unit (ALU)*. Sehingga memungkinkan dua *register* yang berbeda dapat diakses dalam satu siklus *clock*.



Gambar 2.2 Blok diagram AVR ATmega328P-PU

(Sumber: atmel datasheet, 2011)



Gambar 2.3 Konfigurasi Pin-pin ATmega328P-PU

(Sumber: *amel datasheet, 2011*)

2.2 Arduino

Arduino adalah pengendali mikro single-board yang bersifat *open-source*, diturunkan dari *Wiring platform*, dirancang untuk memudahkan penggunaan elektronik dalam berbagai bidang. Hardwarenya memiliki prosesor Atmel AVR dan softwarenya 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 siapa saja dapat mengunduh skema hardware arduino dan membangunnya.

Arduino menggunakan keluarga mikrokontroler ATmega yang dirilis oleh Atmel sebagai basis, namun ada individu/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 membypass bootloader dan menggunakan downloader untuk memprogram mikrokontroler secara langsung melalui port ISP.

2.3 Arduino Uno (R3)



Gambar 2.4 Papan Arduino Uno

(Sumber: *ArduinoBoardUno*, 2014)

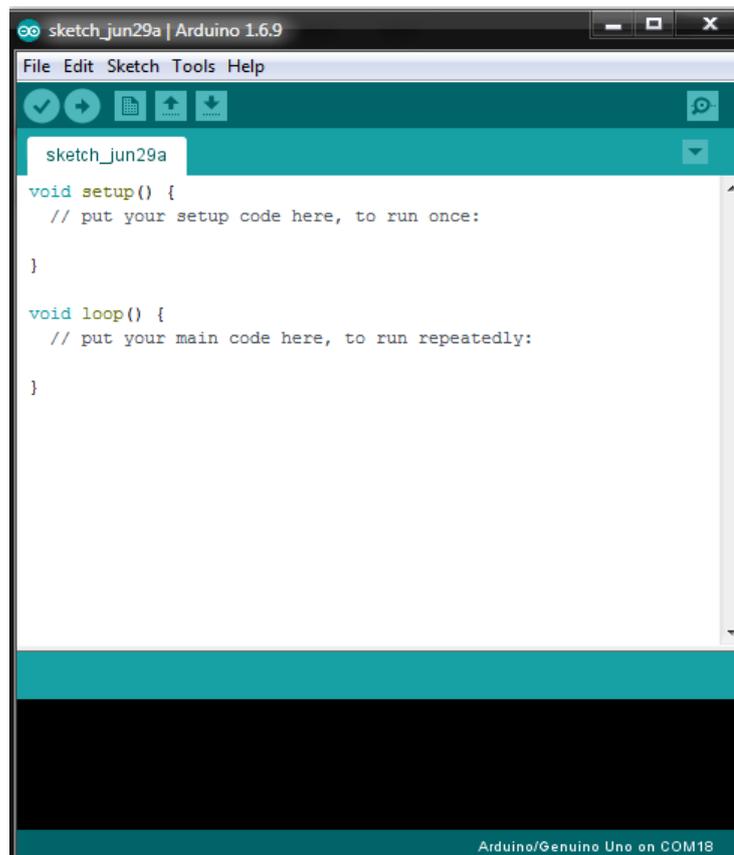
Arduino Uno adalah pilihan yang tepat bagi mereka yang baru pertama kali ingin mempelajari Arduino. Karena Uno merupakan paket lengkap untuk memulai belajar Arduino. Memiliki 14 pin input/output digital (dimana 6 pin dapat digunakan sebagai output PWM), 6 pin input analog, koneksi USB, jack daya, tombol reset, dan masih banyak lagi. Ini semua sudah cukup untuk keperluan belajar mikrokontroler. Hanya dengan menghubungkan papan Arduino ke komputer melalui kabel USB, atau menggunakan adaptor AC-DC, atau menggunakan baterai untuk mengaktifkan papan Arduino.

Spesifikasi Arduino Uno:

- Mikrokontroler ATmega328
- Catu Daya 5V
- Tegangan Input (rekomenadasi) 7-12V
- Tegangan Input (batasan) 6-20V
- Pin I/O Digital 14 (of which 6 provide PWM output) dengan keluaran normal 5v saat on/high dan 0v saat off/low
- Pin Input Analog 6, secara default mengukur dari ground sampai tegangan 5v
- Arus DC per Pin I/O 40 mA

2.4 Aplikasi Program Arduino IDE

Ketika kita membuka program Arduino IDE (*Integrated Development Enviroment*), akan terlihat serupa dengan tampilan gambar 2.6 dibawah ini. Jika kita menggunakan Windows atau Linux, akan terlihat perbedaan, tetapi pada dasarnya IDE (*Integrated Development Enviroment*) akan sama tidak peduli Operasi Sistemnya apa yang digunakan.



Gambar 2.6 Tampilan Program IDE (*Integrated Development Enviroment*)

(Sumber: arduinosoftware, 2016)

IDE terpisah dari toolbar, *The code* ada ditengah dan *The Serial Output* ada dibawah terdiri dari tujuh tombol diantaranya :

a. Verify / compile

Digunakan untuk mengecek atau memeriksa apakah kode sudah benar sebelum di kirim ke papan Arduino.

b. Stop

Berfungsi untuk memberhentikan Serial Monitor dari pengoprasian.

c. New

Berfungsi untuk membuat tampilan lembar kerja atau sketch baru untuk memasukan kode.

d. Open

Menampilkan list lembar kerja yang telah di simpan.

e. Save

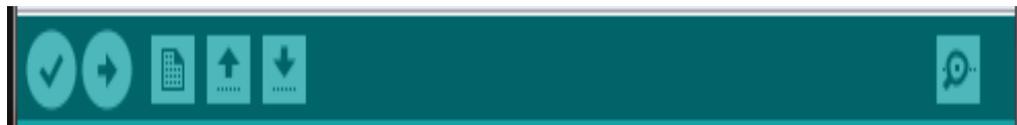
Menyimpan lembar kerja atau sketch.

f. Upload

Mengirim lembar kerja kedalam papan Arduino.

g. Serial Monitor

Menampilkan hasil data-data yang telah dikirim dari Arduino.



Gambar 2.7 Tampilan ToolBar Program IDE

(Sumber: arduinosoftware, 2016)

Untuk memulai Serial Monitor, tekan tombol Serial Monitor dan untuk menghentikan tekan tombol Stop. Pada Linux, Arduino akan me-reset sendiri ketika meng-klik tombol Serial Monitor. Untuk mengoprasikan atau menggabungkan Arduino pada *PC* (*Personal Computer*), kita dapat menggunakan program – program seperti *Processing*, *Flash*, *MaxMSP*, *Visual Basic*, dan lain – lain.

2.4.1 Menu Software Arduino

Kita dapat melihat beberapa menu untuk mengakses software arduino diantaranya adalah:

a. Menu Help

Pada menu ini dapat membantu kita menemukan informasi lebih lagi tentang IDE (*Integrated Development Enviroment*).

- * Getting Started
- * Find in reference
- * Enviroment
- * About Arduino
- * Troubleshooting
- * Visit Arduino.cc
- * Reference
- * Frequently Asked Question

b. Menu Tools

Pada menu ini terdapat peralatan untuk mengisi lembar kerja dari mulai mengedit kode-kode agar terlihat lebih rapih, memilih tipe papan Arduino, memilih serial port dan lain-lain diantaranya :

- * Auto format
- * Burn Bootloader
- * *Archive Sketch*
- * *Serial Port*
- * *Fix Encoding dan Reload*
- * *Board*
- * Serial Monitor

c. Menu Sketch

Pada menu ini kita dapat menambah file kerja, mengeksekusi program yang kita buat, mengimport Library didalam menu ini antara lain :

- * Verify / Compile
- * Import Library
- * Add File
- * Show Sketch Folder
- * Stop

d. Menu Edit

Pada menu ini kita dapat melakukan pengeditan pada lembar kerja, menu Edit ini berisi diantaranya :

- | | |
|-------------------|--------------------------|
| * Undo addition | * Find nex |
| * Cut | * Find |
| * Copy | * Decrease Indent |
| * Copy from forum | * Increase Indent |
| * Copy as HTML | * Comment /
uncomment |

e. Menu File

Pada menu ini kita dapat mengakses atau membuat lembar kerja baru, menyimpan atau mengirim program ke papan Aduino didalam menu File terdapat.

- | | |
|---------------|--------------------------|
| * New | * Quit |
| * Open | * Preferences |
| * Sketch book | * Print |
| * Example | * Page setup |
| * Close | * Upload to I/O
Board |
| * Save | * Save as |

2.5 Program Bahasa C Arduino

C adalah bahasa yang standar, artinya suatu program yang ditulis dengan versi bahasa C tertentu akan dapat dikompilasi dengan versi bahasa C yang lain dengan sedikit modifikasi. Standar bahasa C yang asli adalah standar dari UNIX.

Sistem operasi, kompiler C dan seluruh program aplikasi *UNIX* yang esensial ditulis dalam bahasa C. Patokan dari standar *UNIX* ini diambilkan dari buku yang ditulis oleh Brian Kernighan dan Dennis Ritchie berjudul "The C Programming Language", diterbitkan oleh Prentice-Hall tahun 1978. Deskripsi C dari Kernighan dan Ritchie ini kemudian dikenal secara umum sebagai "K&R C".

Kepopuleran bahasa C membuat versi-versi dari bahasa ini banyak dibuat untuk komputer mikro. Untuk membuat versi-versi tersebut menjadi standar, *ANSI (American National Standards Institute)* membentuk suatu komite (*ANSI committee X3J11*) pada tahun 1983 yang kemudian menetapkan standar *ANSI* untuk bahasa C. Standar *ANSI* ini didasarkan kepada standar *UNIX* yang diperluas. Standar *ANSI* menetapkan sebanyak 32 buah kata-kata kunci (*keywords*) standar. Versi-versi bahasa C yang menyediakan paling tidak 32 kata-kata kunci ini dengan sintaks yang sesuai dengan yang ditentukan oleh standar, maka dapat dikatakan mengikuti standar *ANSI*. Buku ajar ini didasarkan pada bahasa C dari standar *ANSI*.

a. 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.

b. 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).

b. 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.

c. 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

d. Operator Pembandingan

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: $12 \neq 10$ adalah TRUE (benar) atau $12 \neq 12$ 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))

e. Struktur Pengaturan

Program sangat tergantung pada pengaturan apa yang akan dijalankan berikutnya, berikut ini adalah elemen dasar pengaturan.

- **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.

- **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--*.

g. Digital

- **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*.

- **digitalWrite(pin, value)**

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

h. 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).

i. Analog

Arduino adalah mesin digital tetapi mempunyai kemampuan untuk beroperasi di dalam alam analog (menggunakan trik). Berikut ini cara untuk menghadapi hal yang bukan digital.

- **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).

- **analogRead(pin)**

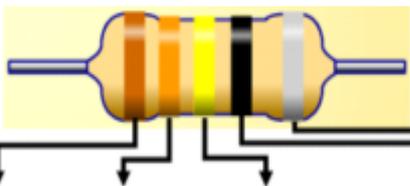
Ketika pin analog ditetapkan sebagai *INPUT* anda dapat membaca keluaran voltase-nya. Keluarannya berupa angka antara 0 (untuk 0 volts) dan 1024 (untuk 5 volts).

2.6 Resistor

Resistor adalah komponen dasar elektronika yang digunakan untuk membatasi jumlah arus yang mengalir dalam satu rangkaian. Sesuai dengan namanya resistor bersifat resistif dan umumnya terbuat dari bahan karbon. Dari hukum Ohms diketahui, resistansi berbanding terbalik dengan jumlah arus yang mengalir melaluinya. Satuan resistansi dari suatu resistor disebut. Tipe resistor yang umum adalah berbentuk tabung dengan dua kaki tembaga di kiri dan kanan. Pada badannya terdapat lingkaran membentuk gelang kode warna untuk memudahkan pemakai mengenali besar resistansi tanpa mengukur besarnya dengan Ohmmeter. Hubungan antara hambatan, tegangan, arus, dapat disimpulkan melalui hukum berikut ini, yang dikenal sebagai Hukum Ohm:

$$R = \frac{V}{I}$$

Dimana V adalah beda beda potensial antara kedua ujung benda penghambat, I adalah besar arus yang melalui benda penghambat, dan R adalah besarnya hambatan benda penghambat tersebut. Kode warna tersebut adalah standar manufaktur yang dikeluarkan oleh EIA (Electronic Industries Association) seperti yang ditunjukkan pada tabel berikut :



Warna	Gelang 1	Gelang 2	Gelang 3	Multiplier Gelang 4	Toleransi Gelang 5
	Hitam		0	0	1 Ohm
Coklat	1	1	1	10 Ohm	± 1 %
Merah	2	2	2	100 Ohm	± 2 %
Orange	3	3	3	1 K Ohm	
Kuning	4	4	4	10 K Ohm	
Hijau	5	5	5	100 K Ohm	± 0,5 %
Biru	6	6	6	1 M Ohm	± 0,25 %
Ungu	7	7	7	10 M Ohm	± 0,10 %
Abu-abu	8	8	8		± 0,05 %
Putih	9	9	9		
Emas				0,1 Ohm	± 5 %
Perak				0,01 Ohm	± 10 %

Gambar 2.8 Tabel Kode Warna Resistor

(Sumber: *Elektronika Dasar*, 2015)

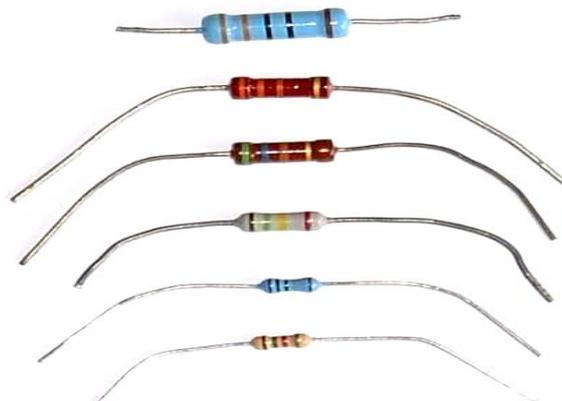
Fungsi dari resistor ini sendiri adalah sebagai pengatur kuat arus ataupun pengatur dan pembagi tegangan (beda potensial).



Gambar 2.9 Simbol Resistor

(Sumber: *Elektronika Dasar*, 2015)

)



Gambar 2.10 Contoh Resistor

(Sumber: *Elektronika Dasar*, 2015)

Jenis resistor sendiri dibedakan menjadi dua macam, yakni komponen Axial atau biasa disebut Radial dan Chip. Pada resistor Radial, perhitungan dilakukan berdasarkan warna, sedangkan untuk Chip, perhitungan resistor ini berdasarkan kode tertentu. Dalam cara menghitung resistor ini, standar dunia menggunakan ukuran satuan Ohm. Pada setiap resistor sendiri biasanya terdapat 4 hingga 5 kabel penghubung.

Resistansi dibaca dari warna gelang yang paling depan ke arah gelang toleransi berwarna coklat, merah, emas atau perak. Biasanya warna gelang toleransi ini berada pada badan resistor yang paling pojok atau juga dengan lebar yang lebih menonjol, sedangkan warna gelang yang pertama agak sedikit ke dalam. Dengan demikian pemakai sudah langsung mengetahui berapa toleransi dari resistor tersebut.

Biasanya resistor dengan toleransi 5%, 10% atau 20% memiliki 3 gelang (tidak termasuk gelang toleransi). Tetapi resistor dengan toleransi 1% atau 2% (toleransi kecil) memiliki 4 gelang (tidak termasuk gelang toleransi). Gelang pertama dan seterusnya berturut-turut menunjukkan besar nilai satuan, dan gelang terakhir adalah faktor pengalinya.

2.7 Switch /Saklar

Switch (Saklar) adalah sebuah perangkat elektronik yang digunakan untuk memutuskan jaringan listrik atau untuk menghubungkannya. Jadi saklar pada dasarnya adalah alat penyambung atau pemutus aliran listrik. Selain untuk jaringan listrik arus kuat, saklar berbentuk kecil juga dipakai untuk alat komponen elektronika arus lemah.

- **Saklar Toggle**



Gambar 2.11 Saklar Toggle

(Sumber: *Teknik Elektronika*, 2015)

Saklar Toggle adalah bentuk saklar yang paling sederhana, dioperasikan oleh sebuah tuas toggle yang dapat ditekan ke atas atau ke bawah. Menurut konvensinya, posisi ke bawah mengindikasikan keadaan ‘hidup’, atau ‘menutup’ atau ‘disambungkan’. Saklar toggle yang diperlihatkan di dalam foto memiliki tuas dengan posisi ke atas. Di belakang tuas terdapat sebuah alur sekrup (*dolly*) yang dilengkapi dengan sebuah mur besar. Alur dan mur ini digunakan untuk memasang saklar disebuah panel. Di bagian belakang saklar terdapat dua buah terminal, tempat dimana kawat-kawat listrik disambung dan disolder. Saklar beban besar (*heavy duty*), memiliki kemampuan untuk menyambungkan arus hingga sebesar 10 A AC. Saklar-saklar toggle beban-besar seringkali digunakan untuk mensaklarkan pasokan listrik dari sumber PLN ke berbagai peralatan dan perangkat listrik. Akan tetapi, saklar-saklar jenis ini juga dapat digunakan untuk menyambungkan

arus listrik yang lebih kecil. Saklar toggle berukuran kecil (miniatur) disebelah ini cocok untuk digunakan pada sebuah panel kontrol. Saklar-saklar toggle yang lebih besar memiliki dua buah tag terminal, yang mengindikasikan bahwa saklar ini memiliki kontak-kontak jenis single-pole, single-throw (satu-kutub, satu arah-SPST). Simbol untuk saklar-saklar ini memperlihatkan bagaimana cara kerjanya. Saklar hanya menyambungkan sebuah rangkaian listrik tunggal dan berada dalam keadaan menutup atau membuka. Saklar toggle yang berukuran lebih kecil memiliki kontak-kontak jenis single-pole, double-throw (satu-kutub, dua-arah-SPDT) Tag terminal yang berada di tengah adalah jalur arus bersama dan dapat membentuk sambungan (kontak) dengan salah satu dari kedua tag lainnya. Kontak-kontak semacam ini disebut sebagai kontak-kontak ganti (*changeover contacts*).

- **Saklar Push Button**



Gambar 2.12 Push Button

(Sumber: *Teknik Elektronika*, 2015)

Pada umumnya saklar push button adalah tipe saklar yang hanya kontak sesaat saja saat ditekan dan setelah dilepas maka akan kembali lagi menjadi *NO*, biasanya saklar tipe *NO* ini memiliki rangkaian penguncinya yang dihubungkan dengan kontaktor dan tipe *NO* digunakan untuk tombol on. Push

button ada juga yang bertipe *NC*, biasanya digunakan untuk tombol *off*. Terdapat 4 konfigurasi saklar push button:

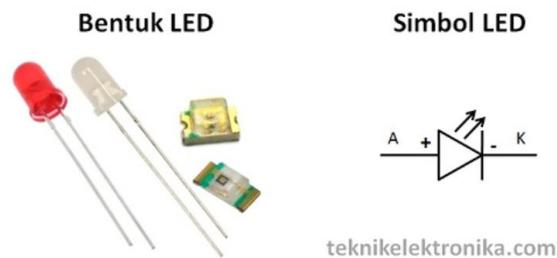
- a. Tanpa-pengunci (*no guard*),
- b. Pengunci-penuh (*full guard*),
- c. *Extended guard*, dan
- d. *Mushroom button*.

Alat ini berfungsi sebagai pemberi sinyal masukan pada rangkaian listrik, ketika / selama bagian knopnya ditekan maka alat ini akan bekerja sehingga kontak-kontaknya akan terhubung untuk jenis normally open dan akan terlepas untuk jenis normally close, dan sebaliknya ketika knopnya dilepas kembali maka kebalikan dari sebelumnya, untuk membuktikannya pada terminalnya bisa digunakan alat ukur tester / ohm meter. pada umumnya pemakaian terminal jenis *NO* digunakan untuk menghidupkan rangkaian dan terminal jenis *NC* digunakan untuk mematikan rangkaian, namun semuanya tergantung dari kebutuhan

2.8 Light Emitting Diode

Light Emitting Diode atau sering disingkat dengan *LED* adalah komponen elektronika yang dapat memancarkan cahaya monokromatik ketika diberikan tegangan maju. *LED* merupakan keluarga Dioda yang terbuat dari bahan semikonduktor. Warna-warna Cahaya yang dipancarkan oleh *LED* tergantung pada jenis bahan semikonduktor yang dipergunakannya. *LED* juga dapat memancarkan sinar inframerah yang tidak tampak oleh mata seperti yang sering kita jumpai pada *Remote Control TV* ataupun *Remote Control* perangkat elektronik lainnya.

Bentuk *LED* mirip dengan sebuah bohlam (bola lampu) yang kecil dan dapat dipasangkan dengan mudah ke dalam berbagai perangkat elektronika. Berbeda dengan Lampu Pijar, *LED* tidak memerlukan pembakaran filamen sehingga tidak menimbulkan panas dalam menghasilkan cahaya.



Gambar 2.13 LED

(Sumber: *Teknikelektronika, 2015*)

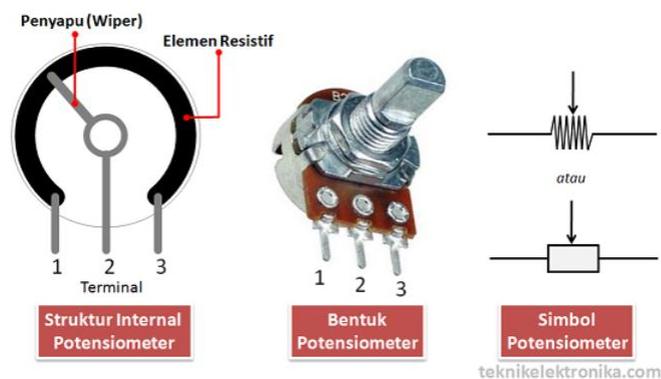
2.9 Potensiometer

Potensiometer adalah resistor tiga terminal dengan sambungan geser yang membentuk pembagi tegangan dapat disetel. Jika hanya dua terminal yang digunakan (salah satu terminal tetap dan terminal geser), potensiometer berperan sebagai resistor variabel atau Rheostat. Potensiometer biasanya digunakan untuk mengendalikan peranti elektronik seperti pengendali suara pada penguat. Potensiometer yang dioperasikan oleh suatu mekanisme dapat digunakan sebagai transduser, misalnya sebagai sensor joystick.

1. Elemen resistif
2. Badan
3. Penyapu (wiper)
4. Sumbu
5. Sambungan tetap pertama
6. Sambungan penyapu
7. Cincin
8. Baut
9. Sambungan tetap kedua

Potensiometer jarang digunakan untuk mengendalikan daya tinggi (lebih dari 1 Watt) secara langsung. Potensiometer digunakan untuk menyetel taraf isyarat analog (misalnya pengendali suara pada peranti audio), dan sebagai pengendali masukan untuk sirkuit elektronik. Sebagai contoh, sebuah peredup

lampu menggunakan potensiometer untuk menendalikan pensakelaran sebuah TRIAC, jadi secara tidak langsung mengendalikan kecerahan lampu.



Gambar 2.14 Potensiometer

(Sumber: *Teknikelektronika*, 2015)

2.10 Buzzer

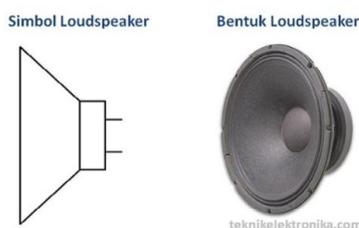
Buzzer adalah sebuah komponen elektronika yang berfungsi untuk mengubah getaran listrik menjadi getaran suara. Pada dasarnya prinsip kerja buzzer hampir sama dengan loud speaker, jadi *buzzer* juga terdiri dari kumparan yang terpasang pada diafragma dan kemudian kumparan tersebut dialiri arus sehingga menjadi elektromagnet, 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. *Buzzer* biasa digunakan sebagai indikator bahwa proses telah selesai atau terjadi suatu kesalahan pada sebuah alat (alarm).



Gambar 2.15 Buzzer

(Sumber: TDK datasheet, 2013)

2.11 LoudSpeaker



Gambar 2.16 LoudSpeaker

(Sumber: Teknikelektronika, 2015)

Loudspeaker yang dalam bahasa Indonesia disebut dengan Pengeras Suara. *Loudspeaker* atau lebih sering disingkat dengan *Speaker* adalah Transduser yang dapat mengubah sinyal listrik menjadi Frekuensi Audio (sinyal suara) yang dapat didengar oleh telinga manusia dengan cara mengetarkan komponen membran pada *Speaker* tersebut sehingga terjadilah gelombang suara.

Suara adalah Frekuensi yang dapat didengar oleh Telinga Manusia yaitu Frekuensi yang berkisar di antara 20Hz – 20.000Hz. Timbulnya suara dikarenakan adanya fluktuasi tekanan udara yang disebabkan oleh gerakan atau getaran suatu obyek tertentu. Ketika Obyek tersebut bergerak atau bergetar, Obyek tersebut akan mengirimkan Energi Kinetik untuk partikel udara disekitarnya. Hal ini dapat

di-analogi-kan seperti terjadinya gelombang pada air. Sedangkan yang dimaksud dengan Frekuensi adalah jumlah getaran yang terjadi dalam kurun waktu satu detik. Frekuensi dipengaruhi oleh kecepatan getaran pada obyek yang menimbulkan suara, semakin cepat getarannya makin tinggi pula frekuensinya.

Dalam rangka menterjemahkan sinyal listrik menjadi suara yang dapat didengar, Speaker memiliki komponen Elektromagnetik yang terdiri dari Kumputan yang disebut dengan Voice Coil untuk membangkitkan medan magnet dan berinteraksi dengan Magnet Permanen sehingga menggerakkan Cone Speaker maju dan mundur. Voice Coil adalah bagian yang bergerak sedangkan Magnet Permanen adalah bagian Speaker yang tetap pada posisinya. Sinyal listrik yang melewati Voice Coil akan menyebabkan arah medan magnet berubah secara cepat sehingga terjadi gerakan “tarik” dan “tolak” dengan Magnet Permanen. Dengan demikian, terjadilah getaran yang maju dan mundur pada Cone Speaker.

Cone adalah komponen utama Speaker yang bergerak. Pada prinsipnya, semakin besarnya Cone semakin besar pula permukaan yang dapat menggerakkan udara sehingga suara yang dihasilkan Speaker juga akan semakin besar.

Suspension yang terdapat dalam Speaker berfungsi untuk menarik Cone ke posisi semula setelah bergerak maju dan mundur. Suspension juga berfungsi sebagai pemegang Cone dan Voice Coil. Kekakuan (rigidity), komposisi dan desain Suspension sangat mempengaruhi kualitas suara Speaker itu sendiri. Berdasarkan Frekuensi yang dihasilkan, Speaker dapat dibagi menjadi :

1. Speaker Tweeter, yaitu speaker yang menghasilkan Frekuensi Tinggi (sekitar 2kHz – 20kHz)
2. Speaker Mid-range, yaitu speaker yang menghasilkan Frekuensi Menengah (sekitar 300Hz – 5kHz)
3. Speaker Woofer, yaitu speaker yang menghasilkan Frekuensi Rendah (sekitar 40Hz – 1kHz)
4. Speaker Sub-woofer, yaitu speaker yang menghasilkan Frekuensi sangat rendah yaitu sekitar 20Hz – 200Hz.

5. Speaker Full Range, yaitu speaker yang dapat menghasilkan Frekuensi Rendah hingga Frekuensi Tinggi.

Berdasarkan Fungsi dan bentuknya, Speaker juga dapat dibedakan menjadi :

1. Speaker Corong
2. Speaker Hi-fi
3. Speaker Handphone
4. Headphone
5. Earphone
6. Speaker Televisi
7. Speaker Sound System (Home Theater)
8. Speaker Laptop

Speaker yang digunakan untuk Sound System Entertainment pada umumnya dapat dibedakan menjadi 2 kategori, yaitu Speaker Pasif dan Speaker Aktif.

1. Speaker Pasif (*Passive Speaker*)

Speaker Pasif adalah Speaker yang tidak memiliki Amplifier (penguat suara) di dalamnya. Jadi Speaker Pasif memerlukan Amplifier tambahan untuk dapat menggerakannya. Level sinyal harus dikuatkan terlebih dahulu agar dapat menggerakkan Speaker Pasif. Sebagian besar Speaker yang kita temui adalah Speaker Pasif.

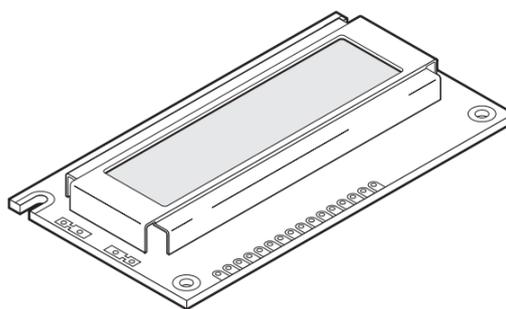
2. Speaker Aktif (*Active Speaker*)

Speaker Aktif adalah Speaker yang memiliki Amplifier (penguat suara) di dalamnya. Speaker Aktif memerlukan kabel listrik tambahan untuk menghidupkan Amplifier yang terdapat didalamnya.

2.12 LCD (*Liquid Cristal Display*)

Layar *LCD* merupakan layar penampil data yang sangat efektif dalam suatu sistem elektronika. Agar sebuah pesan atau gambar dapat tampil pada layar LCD, diperlukan sebuah rangkaian pengatur *scanning* dan pembangkit tegangan

sinus. LCD matrik konfigurasi 16 karakter dan 2 baris setiap karakternya dibentuk oleh 8 baris *pixel* dan 5 kolom *pixel*. Pada modul LCD telah terdapat suatu *driver* yang berfungsi untuk mengendalikan tampilan pada layar LCD. Modul LCD dilengkapi terminal keluaran yang digunakan sebagai jalur komunikasi dengan mikrokontroler. LCD mengirim data penerima data 4 bit atau 8 bit dari perangkat prosesor kemudian data tersebut diproses dan ditampilkan berupa titik-titik yang membentuk karakter atau huruf. Adapun modul LCD dapat dilihat pada gambar di bawah ini.

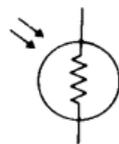


Gambar 2.17 Modul LCD Karakter

(Sumber: *sparkfun datasheet, 2008*)

2.13 LDR

LDR atau *Light Dependent Resistor* adalah sebuah komponen elektronika yang termasuk ke dalam jenis resistor yang nilai resistansinya (nilai tahanannya) akan berubah apabila intensitas cahaya yang diserap juga berubah. Dengan demikian *LDR* juga merupakan resistor yang mempunyai koefisien temperature negative, dimana resistansinya dipengaruhi oleh intrensitas cahaya. *LDR* terbuat dari Cadium Sulfida, bahan ini dihasilkan dari serbuk keramik. Biasanya Cadium Sulfida disebut juga bahan photoconductive, apabila konduktivitas atau resistansi dari Cadium Sulfida bervariasi terhadap intensitas cahaya. Jika intensitas cahaya yang diterima rendah maka hambatan juga akan tinggi yang mengakibatkan tahanan yang kelua juga akan tinggi begitu juga sebaliknya disinilah mekanisme proses perubahan cahaya menjadi listrik terjadi. Berikut adalah simbol LDR:



Gambar 2.18 Simbol *Light Dependent Resistor*

(Sumber: *RS component datasheet, 1997*)

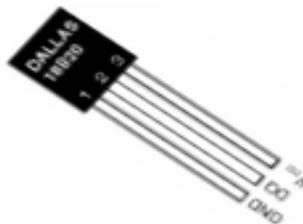
Pada dasarnya *LDR* terbuat dari sebuah cakram semikonduktor yang mempunyai dua buah elektroda pada permukaannya. Pada saat gelap atau intensitas cahaya rendah, bahan tersebut menghasilkan elektron bebas dengan jumlah yang relatif kecil. Sehingga hanya sedikit elektron yang dihasilkan untuk mengangkut muatan elektrik. Hal ini berarti, pada saat keadaan gelap atau intensitas cahaya rendah, maka *LDR* akan menjadi konduktor yang buruk, sehingga *LDR* memiliki resistansi yang besar pada saat gelap atau intensitas cahaya rendah.

Pada saat terang atau intensitas cahaya tinggi, bahan tersebut lebih banyak menghasilkan elektron yang lepas dari atom. Sehingga akan lebih banyak elektron yang dihasilkan untuk mengangkut muatan elektrik. Hal ini berarti, pada saat terang atau intensitas cahaya tinggi, maka *LDR* menjadi konduktor yang baik, sehingga *LDR* memiliki resistansi yang kecil pada saat terang atau intensitas cahaya tinggi.

2.14 Sensor suhu DS18B20

Sensor temperatur *DS18B20* dikeluarkan oleh Dallas Semiconductor. *DS18B20* telah memiliki keluaran digital sehingga tidak diperlukan rangkaian ADC, serta akurasi nilai suhu dan kecepatan pengukuran memiliki kestabilan yang jauh lebih kecil dibandingkan dengan sensor *LM35DZ*. Untuk pembacaan suhu, sensor menggunakan protokol 1-wire communication. *DS18B20* memiliki 3 pin yang terdiri dari +5V, Ground dan data Input/Output. *DS18B20* merupakan sensor yang sangat praktis karena hanya membutuhkan 1 pin I/O saja untuk bias bekerja sama

dengan mikrokontroler. Sensor *DS18B20* memiliki kemampuan untuk mengukur suhu pada kisaran -55°C sampai 125°C dan bekerja secara akurat dengan kesalahan $\pm 0,5^{\circ}\text{C}$ pada kisaran -10°C sampai 85°C . Selain itu daya yang digunakan bias langsung didapat dari data line (“parasite power”), sehingga tidak perlu lagi listrik eksternal. Kecepatan mengukur suhu dalam 750-800ms.



Gambar 2.19 DS18B20

(Sumber : Dallas Semiconductor datasheet, 2014)

2.15 Sensor Ultrasonik 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.



Gambar 2.20 Gambar Sensor Ultrasonik HC-SR04

(Sumber: electroschematics datasheet, 2014)