

# **TUTORIAL PEMBUATAN APLIKASI PREDIKSI JUMLAH KEBUTUHAN PERUSAHAAN**



---

# TUTORIAL PEMBUATAN APLIKASI PREDIKSI JUMLAH KEBUTUHAN PERUSAHAAN

---

**Roni Habibi, S.Kom., M.T.**  
**Alwan Suryansah**  
Informatics Research Center



**Kreatif Industri Nusantara**

***Penulis:***

Mohamad Nurkamal Fauzan, S.T., M.T.  
Lalita Chandiany Adiputri

ISBN : xxx-xxx-xxxx-x-2

***Editor:***

M. Yusril Helmi Setyawan

***Penyunting:***

Syafrial Fachrie Pane  
Khaera Tunnisa  
Diana Asri Wijayanti

***Desain sampul dan Tata letak:***

Deza Martha Akbar

***Penerbit:***

Kreatif Industri Nusantara

***Redaksi:***

Jl. Ligar Nyawang No. 2  
Bandung 40191  
Tel. 022 2045-8529  
Email : awangga@kreatif.co.id

***Distributor:***

Informatics Research Center  
Jl. Sariasih No. 54  
Bandung 40151  
Email : irc@poltekpos.ac.id

Cetakan Pertama, 2019

Hak cipta dilindungi undang-undang  
Dilarang memperbanyak karya tulis ini dalam bentuk dan dengan cara  
apapun tanpa ijin tertulis dari penerbit

*‘Jika Kamu tidak dapat  
menahan lelahnya  
belajar, Maka kamu harus  
sanggup menahan  
perihnya Kebodohan.’  
Imam Syafi’i*

# CONTRIBUTORS

---

RONI HABIBI, S.KOM., M.T., ALWAN SURYANSAH,  
Informatics Research Center., Politeknik Pos Indonesia, Bandung, Indonesia



# CONTENTS IN BRIEF

---

<b>1</b>	<b>Pembuatan Aplikasi</b>	<b>1</b>
<b>2</b>	<b>Sekilas Tentang Aplikasi Prediksi</b>	<b>3</b>
<b>3</b>	<b>Judul Bagian Kedua</b>	<b>59</b>
<b>4</b>	<b>Fungsi dan Kelas</b>	<b>65</b>





# DAFTAR ISI

---

Daftar Gambar	xi
Daftar Tabel	xv
Foreword	xix
Kata Pengantar	xxi
Acknowledgments	xxiii
Acronyms	xxv
Glossary	xxvii
List of Symbols	xxix
Introduction	xxx
<i>Roni Habibi, S.Kom., M.T., Alwan Suryansah.</i>	
<b>1 Pembuatan Aplikasi</b>	<b>1</b>
<b>2 Sekilas Tentang Aplikasi Prediksi</b>	<b>3</b>
2.1 Apa itu Prediksi ?	3
2.2 Tujuan <i>Prototype</i>	5
	<b>ix</b>

2.3	Bahasa Pemograman C	5
2.3.1	Sejarah Bahasa C	5
2.3.2	Pengertian Bahasa Pemograman C	6
2.3.3	Kelebihan dan Kekurangan Bahasa C	6
2.4	Penggunaan Bahasa C menggunakan Arduino IDE	6
2.4.1	Arduino IDE	6
2.4.2	Bagian-Bagian Arduino IDE	7
2.4.3	<i>Sketch</i> Arduino	9
2.5	Komponen yang Digunakan Untuk Pembuatan <i>Prototype</i> PKA	15
2.6	Pembuatan <i>Prototype</i>	46
2.7	Merakit <i>Kerangka Prototype</i>	52
2.8	Perakitan dan Memprogram Komponen <i>Prototype</i>	54
2.8.1	Perakitan Komponen	54
<b>3</b>	<b>Judul Bagian Kedua</b>	<b>59</b>
3.1	Variabel	59
3.2	Input dan Output	60
3.3	Operasi Aritmatika	60
3.4	Perulangan	60
3.4.1	For	61
3.4.2	While	61
3.5	Kondisi	61
3.6	Error	63
3.7	Try Except	64
<b>4</b>	<b>Fungsi dan Kelas</b>	<b>65</b>
4.1	Teori	65
4.1.1	Fungsi	65
4.2	Package	66
4.3	Class, Object, Atribut, and Method	66
4.4	Pemanggilan Class	67
4.5	Pemakaian Package Fungsi Apabila File Didalam Folder	67
4.6	Pemakaian Package Kelas Apabila File didalam Folder	67
	Daftar Pustaka	69

# DAFTAR GAMBAR

---

2.1	Arduino IDE	7
2.2	Tampilan Arduino IDE	8
2.3	Contoh Void Setup	10
2.4	Gambar Contoh Fungsi Setup dan Loop	11
2.5	NodeMCU	16
2.6	Spesifikasi NodeMCU	17
2.7	Skema Posisi Pin NodeMCU	18
2.8	Sensor HC-SR04	19
2.9	Prinsip Pengukuran Jarak Sensor HC SR04	20
2.10	Rumus Perhitungan Jarak	21
2.11	Konfigurasi Sensor Ultrasonic HC-SR04	21
2.12	Gambar Sensor Ultrasonic US 100	23
2.13	Prinsip Kerja Sensor Ultrasonic	24

2.14	Rumus perhitungan Jarak Sensor Ultrasonic	24
2.15	Pin Konfigurasi	25
2.16	Skema diagram sensor ultrasonic US 100	26
2.17	Buzzer	27
2.18	Konfigurasi Pin Buzzer	28
2.19	Keterangan Konfigurasi Pin Buzzer	29
2.20	Diagram model 2D Buzzer	30
2.21	Led	30
2.22	Simbol Led	31
2.23	Cara Kerja Led	32
2.24	Cara Melihat Polaritas	33
2.25	Logo Antares	33
2.26	Tampilan Antares Antares	34
2.27	Reistor	34
2.28	Macam-Macam Resistor Sesuai Warna	36
2.29	Resistor Yang Digunakan	37
2.30	Logo Telegram	38
2.31	Bentuk Akrilik	39
2.32	Kabel Jumper	40
2.33	Kabel Jumper Female	41
2.34	Kabel Jumper Male	42
2.35	Kabel Jumper Male-Female	43
2.36	Breadboard Mini	44
2.37	Breadboard Medium	45
2.38	Breadboard Large	46
2.39	Contoh Akrilik yang digunakan	47
2.40	Contoh Lem Korea	48
2.41	Contoh Lem Akrilik	49

2.42	Contoh Sticky Note	50
2.43	Contoh Double TApe	51
2.44	Contoh Styrofoam Papan	52
2.45	Kerangka Prototipe	53
2.46	Kerangka Prototipe Tempan Penyimpanan Komponen	54
2.47	Rangkaian Alat	56
2.48	Rangkaian Alat	57



# DAFTAR TABEL

---





# Listings

---

src/void.c	11
src/boolean.c	12
src/char.c	12
src/unchar.c	12
src/byte.c	12
src/int.c	12
src/unint.c	12
src/long.c	13
src/unlong.c	13
src/short.c	13
src/float.c	13
src/double.c	13
src/variabel1.c	14
src/variabel2.c	14



# FOREWORD

---

Sepatah kata dari Kaprodi, Kabag Kemahasiswaan dan Mahasiswa



# KATA PENGANTAR

---

Buku ini diciptakan bagi yang ingin membuat sebuah prototipe dengan menggunakan *microcontroller* NodeMCU, monitoring melalui android serta adanya notifikasi melalui telegram.

RONI HABIBI , ALWAN SURYANSAH

*Bandung, Jawa Barat  
Januari, 2020*



# ACKNOWLEDGMENTS

---

Terima kasih atas semua masukan dari para mahasiswa agar bisa membuat buku ini lebih baik dan lebih mudah dimengerti.

Terima kasih ini juga ditujukan khusus untuk team IRC yang telah fokus untuk belajar dan memahami bagaimana buku ini mendampingi proses Intership.

R. M. A.





# ACRONYMS

---

ACGIH	American Conference of Governmental Industrial Hygienists
AEC	Atomic Energy Commission
OSHA	Occupational Health and Safety Commission
SAMA	Scientific Apparatus Makers Association



# GLOSSARY

---

<i>Prototype</i>	Merupakan purwarupa model kerja dasar dari pengembangan sebuah program (software) atau perangkat lunak.
Arduino IDE	Merupakan ( <i>Integrated Development Environment</i> ) software yang di gunakan untuk memprogram di arduino.



# SYMBOLS

---

- $A$  Amplitude
- $\&$  Propositional logic symbol
- $a$  Filter Coefficient
  
- $\mathcal{B}$  Number of Beats



# INTRODUCTION

---

RONI HABIBI, S.KOM., M.T., ALWAN SURYANSAH.

Informatics Research Center  
Bandung, Jawa Barat, Indonesia

Indonesia sekarang sudah memasuki era industri 4.0 atau revolusi 4.0 dimana otomatisasi sistem produksi dengan memanfaatkan teknologi dan big data. Di dalam pabrik mulai menggunakan teknologi baru seperti IoT (internet of things).





# BAB 1

---

## PEMBUATAN APLIKASI

---



## BAB 2

---

# SEKILAS TENTANG APLIKASI PREDIKSI

---

### 2.1 Apa itu Prediksi ?

Prediksi adalah suatu proses memperkirakan secara sistematis tentang sesuatu yang paling mungkin terjadi di masa depan berdasarkan informasi masa lalu dan sekarang yang dimiliki, agar kesalahannya (selisih antara sesuatu yang terjadi dengan hasil perkiraan) dapat diperkecil. Prediksi tidak harus memberikan jawaban secara pasti kejadian yang akan terjadi, melainkan berusaha untuk mencari jawaban sedekat mungkin yang akan terjadi (Herdianto, 2013 : 8) .

Pengertian Prediksi sama dengan ramalan atau perkiraan. Menurut kamus besar bahasa Indonesia, prediksi adalah hasil dari kegiatan memprediksi atau meramal atau memperkirakan nilai pada masa yang akan datang dengan menggunakan data masa lalu. Prediksi menunjukkan apa yang akan terjadi pada suatu keadaan tertentu dan merupakan input bagi proses perencanaan dan pengambilan keputusan.

Prediksi bisa berdasarkan metode ilmiah ataupun subjektif belaka. Ambil contoh, prediksi cuaca selalu berdasarkan data dan informasi terbaru yang didasarkan pengamatan termasuk oleh satelit. Begitupun prediksi gempa, gunung meletus ataupun

bencana secara umum. Namun, prediksi seperti pertandingan sepakbola, olahraga, dll umumnya berdasarkan pandangan subjektif dengan sudut pandang sendiri yang memprediksinya.

Permulaan awal, walaupun pengkajian yang mendalam mengenai alternatif masa depan adalah suatu disiplin baru, barangkali orang telah menaruh perhatian besar tentang apa yang akan terjadi kemudian semenjak manusia mulai mengetahui sesuatu. Populasi tukang ramal dan tukang nujum pada zaman kuno dan abad pertengahan merupakan satu manifestasi dari keinginan tahu orang tentang masa depannya. Perhatian tentang masa depan ini berlangsung terus bahkan berkembang menjadi kolom astrologi yang disindikatkan pada tahun 1973.

Secara Eksplisit, pembahasan mengenai teori peramalan kebijakan sangatlah sedikit. Namun, secara implisit, peramalan kebijakan terkait menjadi satu dengan proses analisa kebijakan. Karena didalam menganalisa kebijakan, untuk menformulasikan sebuah rekomendasi kebijakan baru, maka diperlukan adanya peramalan-peramalan atau prediksi mengenai kebijakan yang akan diberlakukan dimasa yang akan datang. Namun, satu dari sekian banyak prosedur yang ditawarkan oleh para pakar Dunn, masih memberikan pembahasan tersendiri mengenai peramalan kebijakan. Menurut Dunn, Peramalan Kebijakan ( policy forecasting) merupakan suatu prosedur untuk membuat informasi factual tentang situasi social masa depan atas dasar informasi yang telah ada tentang masalah kebijakan.

Peramalan (forecasting) adalah suatu prosedur untuk membuat informasi factual tentang situasi sosial masa depan atas dasar informasi yang telah ada tentang masalah kebijakan. Ramalan mempunyai tiga bentuk utama: proyeksi, prediksi, dan perkiraan.

1. Suatu proyeksi adalah ramalan yang didasarkan pada ekstrapolasi atas kecenderungan masa lalu maupun masa kini ke masa depan. Proyeksi membuat pertanyaan yang tegas berdasarkan argument yang diperoleh dari metode tertentu dan kasus yang paralel.
2. Sebuah prediksi adalah ramalan yang didasarkan pada asumsi teoritik yang tegas. Asumsi ini dapat berbentuk hukum teoretis (misalnya hukum berkurangnya nilai uang), proposisi teoritis (misalnya proposisi bahwa pecahnya masyarakat sipil diakibatkan oleh kesenjangan antara harapan dan kemampuan), atau analogi (misalnya analogi antara pertumbuhan organisasi pemerintah dengan pertumbuhan organisme biologis).
3. Suatu perkiraan (conjecture) adalah ramalan yang didasarkan pada penilaian yang informative atau penilaian pakar tentang situasi masyarakat masa depan.

Tujuan dari pada diadakannya peramalan kebijakan adalah untuk memperoleh informasi mengenai perubahan dimasa yang akan datang yang akan mempengaruhi terhadap implementasi kebijakan serta konsekuensinya. Oleh karenanya, sebelum

rekomendasi diformulasikan perlu adanya peramalan kebijakan sehingga akan diperoleh hasil rekomendasi yang benar-benar akurat untuk diberlakukan pada masa yang akan. Didalam memprediksi kebutuhan yang akan datang dengan berpijak pada masa lalu, dibutuhkan seseorang yang memiliki daya sensitifitas tinggi dan mampu membaca kemungkinan-kemungkinan dimasa yang akan datang. Permalan kebijakan juga diperlukan untuk mengontrol, dalam artian, berusaha merencanakan dan menetapkan kebijakan sehingga dapat memberikan alternatif-alternatif tindakan yang terbaik yang dapat dipilih diantara berbagai kemungkinan yang ditawarkan oleh masa depan. Masa depan juga terkadang banyak dipengaruhi oleh masa lalu. Dengan mengacu pada masa depan analisis kebijakan harus mampu menaksir nilai apa yang bisa atau harus membimbing tindakan di masa depan.

alwan

## 2.2 Tujuan *Prototype*

Prototipe bertujuan untuk contoh atau model awal yang dibangun untuk menguji sebuah konsep atau proses atau aksi sebagai sesuatu yang digandakan atau dipelajarinya. Pengertian prototipe tidak selalu merujuk pada ukuran, artinya prototipe tidak selalu harus berukuran sama dengan produk yang akan dibuat. Prototipe bisa berukuran lebih kecil atau lebih besar dibanding dengan produk yang akan dibuat asalkan aksi atau proses yang akan terjadi sebenarnya .

Prototype dapat memberikan ide bagi pembuat dan pemakai potensial tentang cara sistem berfungsi dalam bentuk lengkapnya. Tujuannya adalah mengembangkan model menjadi sistem final. Artinya sistem akan dikembangkan lebih cepat daripada metode tradisional dan biayanya menjadi lebih murah. Selain hal tersebut pembuatan prototipe untuk perbaikan atau penyempurnaan rancangan .

## 2.3 Bahasa Pemograman C

### 2.3.1 Sejarah Bahasa C

Bahasa C merupakan perkembangan dari bahasa BCPL yang dikembangkan oleh Martin Richards pada tahun 1967. Selanjutnya bahasa ini memberikan ide kepada Ken Thompson yang kemudian mengembangkan bahasa yang disebut bahasa B pada tahun 1970. Perkembangan selanjutnya dari bahasa B adalah bahasa C oleh Dennis Ritchie sekitar tahun 1970-an di Bell Telephone Laboratories Inc. (sekarang adalah ATT Bell Laboratories). Bahasa C pertama kali digunakan di computer Digital Equipment Corporation PDP-11 yang menggunakan system operasi UNIX. Hingga saat ini penggunaan bahasa C telah merata di seluruh dunia. Hampir semua perguruan tinggi di dunia menjadikan bahasa C sebagai salah satu mata kuliah wajib. Selain itu, banyak bahasa pemrograman populer seperti PHP dan Java menggunakan sintaks dasar yang mirip bahasa C. Oleh karena itu, kita juga sangat perlu mempelajarinya .

### 2.3.2 Pengertian Bahasa Pemrograman C

C merupakan perkembangan dari bahasa pemrograman c yang di ciptakan oleh Brian W. Kernighan dan Dennis M. Ritchie lalu di kembangkan oleh Bjarne Stroustrup dari Laboratorium Bell, ATT, pada tahun 1983. C cukup kompatibel dengan bahasa pendahulunya C. Pada mulanya C disebut a better C . Nama C sendiri diberikan oleh Rick Mascitti pada tahun 1983, yang berasal dari operator increment pada bahasa C. Keistimewaan yang sangat berarti dari C ini adalah karena bahasa ini mendukung Pemrograman Berorientasi Objek (OOP /Object Oriented Programming) .

Bahasa pemrograman C ini dapat digunakan untuk memprogram sebuah robot, Untuk memprogram menggunakan bahasa C ini dapat menggunakan IDE (*Integrated Development Environment*) salah satu contohnya yaitu IDE Arduino.

### 2.3.3 Kelebihan dan Kekurangan Bahasa C

Dalam bahasa pemrograman C ini memiliki kelebihan yaitu sebagai berikut :

1. Bahasa C tersedia hampir di semua jenis computer.
2. Kode bahasa C sifatnya adalah portable dan fleksibel untuk semua jenis computer.
3. Bahasa C hanya menyediakan sedikit kata-kata kunci, hanya terdapat 32 kata kunci.
4. Proses executable program bahasa C lebih cepat
5. Dukungan pustaka yang banyak.
6. C adalah bahasa yang terstruktur
7. Bahasa C termasuk bahasa tingkat menengah

Selain memiliki kelebihan bahasa pemrograman C juga mempunyai kekurangannya yaitu :

1. Banyaknya Operator serta fleksibilitas penulisan program kadang-kadang membingungkan pemakai.
2. Bagi pemula pada umumnya akan kesulitan menggunakan pointer.

## 2.4 Penggunaan Bahasa C menggunakan Arduino IDE

### 2.4.1 Arduino IDE

Arduino IDE (*Integrated Development Environment*) adalah software yang di gunakan untuk memprogram di arduino, dengan kata lain Arduino IDE sebagai media

untuk memprogram board Arduino. Arduino IDE bisa di download secara gratis di website resmi Arduino IDE.

Arduino IDE ini berguna sebagai text editor untuk membuat, mengedit, dan juga mevalidasi kode program. bisa juga digunakan untuk meng-upload ke board Arduino. Kode program yang digunakan pada Arduino disebut dengan istilah Arduino sketch atau disebut juga source code arduino, dengan ekstensi file source code .ino

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.

Dalam *Software* Arduino IDE ini dapat memprogram kode tidak hanya *microcontroller* arduino saja, tetapi pada *software* ini dapat digunakan untuk memprogram *microcontroller* yang lain salah satu contohnya yaitu Nodemcu.

## 2.4.2 Bagian-Bagian Arduino IDE

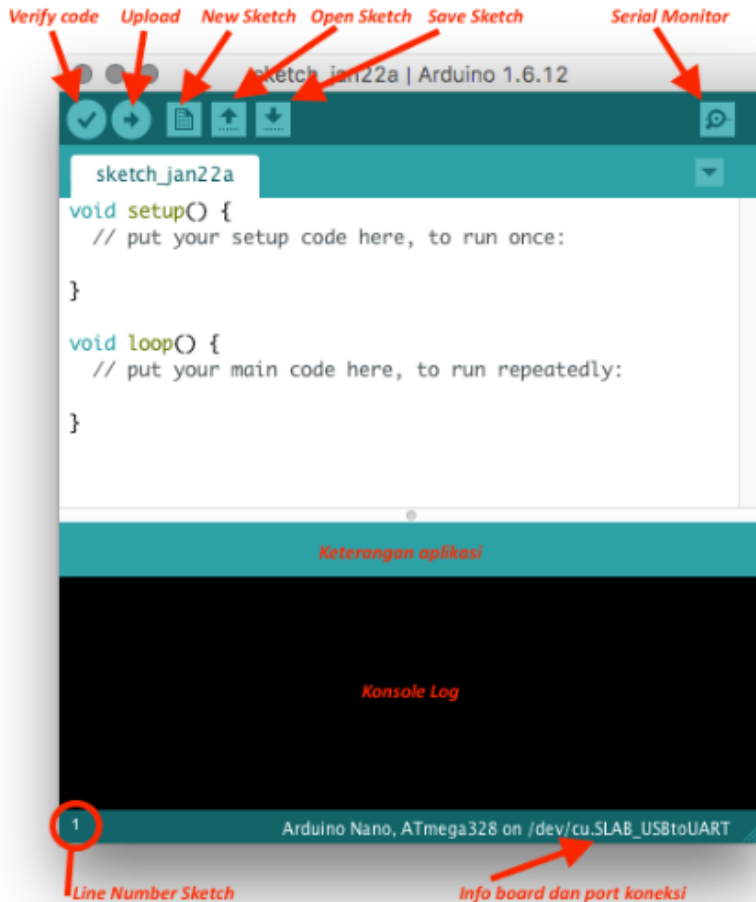


**Gambar 2.1** Arduino IDE

Editor *Programming* pada umumnya memiliki fitur untuk *cut / paste* dan untuk *find / replace teks*, demikian juga pada Arduino IDE. Pada bagian keterangan aplikasi memberikan pesan balik saat menyimpan dan mengekspor serta sebagai tempat menampilkan kesalahan. Konsol log menampilkan teks log dari aktifitas Arduino IDE, termasuk pesan kesalahan yang lengkap dan informasi lainnya. Pojok kanan bawah menampilkan *port* serial yang di gunakan. Tombol *toolbar* terdapat



ikon tombol pintas untuk memverifikasi dan meng-upload program, membuat, membuka, dan menyimpan *sketch*, dan membuka monitor serial. Untuk lebih jelasnya perhatikan gambar dibawah ini :



**Gambar 2.2** Tampilan Arduino IDE

1. **Verify** pada versi sebelumnya dikenal dengan istilah Compile. Sebelum aplikasi di-upload ke board Arduino, biasanya untuk memverifikasi terlebih dahulu sketch yang dibuat. Jika ada kesalahan pada sketch, nanti akan muncul error. Proses Verify / Compile mengubah sketch ke binary code untuk di-upload ke mikrokontroler.
2. **Upload** tombol ini berfungsi untuk mengupload sketch ke board Arduino. Walaupun kita tidak mengklik tombol verify, maka sketch akan di-compile, kemudian

langsung diupload ke board. Berbeda dengan tombol verify yang hanya berfungsi untuk memverifikasi source code saja.

3. **New Sketch** Membuka window dan membuat sketch baru
4. **Open Sketch** Membuka sketch yang sudah pernah dibuat. Sketch yang dibuat dengan IDE Arduino akan disimpan dengan ekstensi file .ino
5. **Save Sketch** menyimpan sketch, tapi tidak disertai dengan mengkompile.
6. **Serial Monitor** Membuka interface untuk komunikasi serial, nanti akan kita diskusikan lebih lanjut pada bagian selanjutnya.
7. **Keterangan Aplikasi** pesan-pesan yang dilakukan aplikasi akan muncul di sini, misal *Compiling* dan *Done Uploading* ketika kita *mengcompile* dan mengupload sketch ke board Arduino
8. **Konsol log** Pesan-pesan yang dikerjakan aplikasi dan pesan-pesan tentang sketch akan muncul pada bagian ini. Misal, ketika aplikasi mengcompile atau ketika ada kesalahan pada sketch yang kita buat, maka informasi error dan baris akan diinformasikan di bagian ini.
9. **Baris Sketch** bagian ini akan menunjukkan posisi baris kursor yang sedang aktif pada sketch.
10. **Informasi Board dan Port** Bagian ini menginformasikan port yang dipakai oleh board Arduino.

### 2.4.3 Sketch Arduino

Pada arduino bahasa pemrograman yang digunakan adalah bahasa C/C++. Program pada Arduino terbagi menjadi tiga bagian utama yaitu Structure, Values (berisi variabel dan konstanta) dan yang terakhir function. Struktur kode pada arduino yaitu berisi fungsi setup() dan loop() sebagai berikut :

#### 1. Setup()

fungsi ini dipanggil pertama kali ketika menjalankan *sketch*. digunakan sebagai tempat *inisialisasi variabel, pin mode, penggunaan library* dan lainnya. fungsi ini dijalankan sekali ketika *board* dinyalakan atau di *reset* . Berikut contoh dari void setup :

#### 2. loop()

Setelah membuat fungsi setup() sebagai tempat *inisialisasi variabel* dan menetapkan nilai maka selanjutnya fungsi loop() seperti namanya fungsi ini akan melakukan perulangan berturut-turut, memungkinkan program untuk mengubah dan menanggapi. digunakan untuk mengontrol *board* Arduino. Berikut contoh dari void loop :

```
int buttonPin = 3;

void setup()
{
  Serial.begin(9600);
  pinMode(buttonPin, INPUT);
}

void loop()
{
  // ...
}
```

**Gambar 2.3** Contoh Void Setup



**Gambar 2.4** Gambar Contoh Fungsi Setup dan Loop

Pada gambar 2.4 merupakan suatu contoh fungsi void setup dan void loop pada pemrograman bahasa C menggunakan Arduino IDE.

### 3. Values

Berisi variable atau konstanta sesuai dengan type data yang didukung oleh Arduino.

### 4. Function

Segmentasi kode ke fungsi memungkinkan programmer untuk membuat potongan-potongan modular kode yang melakukan tugas yang terdefinisi dan kemudian kembali ke asal kode dari mana fungsi itu dipanggil. Umumnya menggunakan fungsi adalah ketika salah satu kebutuhan untuk melakukan tindakan yang sama beberapa kali dalam sebuah program.

### 5. Tipe Data Pada Arduino

Tipe data merupakan kelompok data berdasarkan jenis-jenis tertentu. Tipe data banyak dijumpai dalam berbagai bahasa pemrograman. Begitu juga dalam pemrograman Arduino. Tipe-tipe data yang digunakan dalam pemrograman Arduino antara lain adalah Void, Boolean, Char, Unsigned Char, Byte, Int, Unsigned Int, Word, Long, Unsigned Long, Short, Float, dan Double.

#### (a) Void

Kata kunci void digunakan dalam deklarasi fungsi. Kata kunci ini menandakan bahwa fungsi tersebut tidak mengembalikan informasi ke fungsi yang dipanggil. Seperti sourcode dibawah ini :

```

1 Void Loop ( ) {
2 // kode program
3 }

```

**(b) Boolean**

Boolean menangani satu dari dua nilai yaitu, benar (true) atau salah (false). Setiap variabel Boolean menempati satu byte memori. Seperti sourcode dibawah ini :

```
1 boolean val = false ;
2 boolean state = true ;
```

**(c) Char**

Tipe data Char adalah tipe data yang mengambil satu byte memori yang menyimpan suatu nilai karakter. Karakter harfiah ditulis dalam kutip tunggal, misalnya A dan untuk multi karakter digunakan tanda kutip ganda, seperti ABC.

Namun demikian, karakter-karakter tersebut disimpan sebagai angka. Hal ini dapat dilihat dengan jelas pada Tabel ASCII. Hal ini berarti bahwa melakukan operasi aritmatika pada karakter, di mana nilai karakter ASCII digunakan. Sebagai contoh, A + 1 memiliki nilai 66, karena nilai ASCII dari huruf kapital A adalah 65.

```
1 Char chr_r = r
2 Char chr_e = 97 ;
```

**(d) Unsigned char**

Unsigned char merupakan tipe data tak bertanda yang mencakup satu byte memori. Tipe data ini meng-encode bilangan dari 0 sampai 255.

```
1 Unsigned Char chr_y = 231 ;
```

**(e) Byte**

Byte menyimpan bilangan tak bertanda 8 bit, dari 0 sampai 255.

```
1 byte a = 75 ;
```

**(f) INT**

Integer adalah tipe data utama untuk penyimpanan bilangan. int menyimpan nilai 16-bit (2-byte). Rentang nilainya adalah kisaran -32.768 hingga 32.767. Ukuran int berbeda-beda setiap papan arduino. Sebagai contoh untuk Arduino Due, int menyimpan nilai 32-bit (4-byte). Rentang nilainya dari -2,147,483,648 sampai 2,147,483,647.

```
1 int counter = 92 ;
```

**(g) Unsigned int**

Unsigned int sama dengan int dalam cara mereka menyimpan nilai 2 byte. Unsigned int hanya menyimpan nilai positif, menghasilkan rentang dari 0 hingga 65.535. Arduino Due menyimpan nilai 4 byte (32-bit), mulai dari 0 hingga 4.294.967.295.

```
1 Unsigned int counter = 90 ;
```

**(h) Long**

Variabel dengan tipe Long merupakan penyimpanan bilangan dengan ukuran diperluas, dan menyimpan 32 bit (4 byte), dari -2,147,483,648 sampai 2,147,483,647.

```
1 Long velocity = 172346 ;
```

**(i) Unsigned long**

Unsigned Long merupakan tipe Long tak bertanda yang diperluas untuk penyimpanan bilangan dan menyimpan 32 bit (4 byte). Tidak seperti long standar, unsigned long tidak akan menyimpan angka negatif, rentangnya dari 0 hingga 4.294.967.295.

```
1 Unsigned Long velocity = 121016 ;
```

**(j) Short**

Short adalah tipe data 16-bit. Pada semua Arduino (berbasis ATmega dan ARM), short menyimpan nilai 16-bit (2-byte). Rentang nilainya antara 32.768 hingga 32.767.

```
1 short val = 19 ;
```

**(k) Float**

Tipe data untuk bilangan floating-point adalah bilangan yang memiliki titik desimal. Angka floating-point sering digunakan untuk memperkirakan nilai analog dan kontinu karena memiliki resolusi lebih besar daripada bilangan bulat.

Bilangan floating-point dapat memiliki nilai maksimal  $3,4028235E + 38$  dan nilai minimal  $-3,4028235E + 38$ . Float disimpan sebagai informasi 32 bit (4 byte).

```
1 float num = 1.523 ;
```

**(l) Double**

Pada Arduino Uno dan arduino berbasis ATMEGA lainnya, bilangan floating-point presisi ganda menempati empat byte. Artinya, implementasi ganda persis sama dengan float, tanpa perolehan presisi. Pada Arduino Due, ganda memiliki presisi 8-byte (64 bit).

```
1 double num = 45.253 ;s
```

**6. Variabel**

Variabel dalam bahasa pemrograman C, dimana bahasa C ini digunakan dalam Arduino, memiliki suatu properti yang disebut dengan cakupan (*scope*). Suatu cakupan merupakan wilayah dari program dan ada tiga tempat dimana variabel dapat dideklarasikan. Ketiga tempat tersebut adalah sebagai berikut :

- (a) Di dalam fungsi atau blok, variabel ini disebut dengan variabel lokal (*local variable*).

- (b) Di dalam definisi parameter fungsi, yang disebut dengan parameter formal (*formal parameters*).
- (c) Di luar semua fungsi, variabel ini disebut dengan variabel global (*global variable*).

## 7. Variabel Local (*Local Variable*)

Sebagaimana disebutkan di atas, bahwa variabel lokal merupakan variabel yang dideklarasikan di dalam suatu fungsi atau blok. Variabel lokal ini hanya dapat digunakan oleh pernyataan (statement) yang berada di dalam fungsi atau blok kode. Berikut ini adalah contoh penggunaan dari variabel lokal.

```

1 Void setup () {
2
3 }
4
5 Void loop () {
6     int a , b ;
7     int c ; Local variable declaration
8     a = 0;
9     b = 0; actual initialization
10    c = 10;
11 }
```

## 8. Variabel Global (*Global Variable*)

Variabel global (*Global variable*) didefinisikan di luar semua fungsi, biasanya pada bagian atas dari program. Variabel global akan menyimpan nilainya sepanjang program dijalankan (*life-time*).

```

1 Int A , R ;
2 float j = 0 ; Global variable declaration
3
4 Void setup () {
5
6 }
7
8 Void loop () {
9     int x , y ;
10    int z ; Local variable declaration
11    x = 0;
12    y = 0; actual initialization
13    z = 10;
14 }
```

Variabel global dapat diakses oleh semua fungsi apapun. Artinya variabel global akan tersedia terus untuk digunakan di seluruh program setelah dideklarasikan. Berikut ini adalah contoh penggunaan dari variabel global dalam pemrograman Arduino.

## 2.5 Komponen yang Digunakan Untuk Pembuatan *Prototype* PKA

Komponen yang digunakan untuk membuat sebuah *prototype* prediksi ketinggian air (PKA) untuk pendeteksi banjir peringatan dini yaitu sebagai berikut :

### 1. NodeMCU

NodeMCU adalah ESP8266 (khususnya seri ESP-12, termasuk ESP-12E) maka fitur fitur yang dimiliki NodeMCU akan kurang lebih sama ESP-12 (juga ESP-12E untuk NodeMCU v.2 dan v.3) kecuali NodeMCU telah dibungkus oleh API sendiri yang dibangun berdasarkan bahasa pemrograman eLua, yang kurang lebih cukup mirip dengan javascript. Selain dapat diprogram menggunakan bahasa LUA dapat juga diprogram menggunakan bahasa C menggunakan arduino IDE. Beberapa fitur didalamnya antara lain :

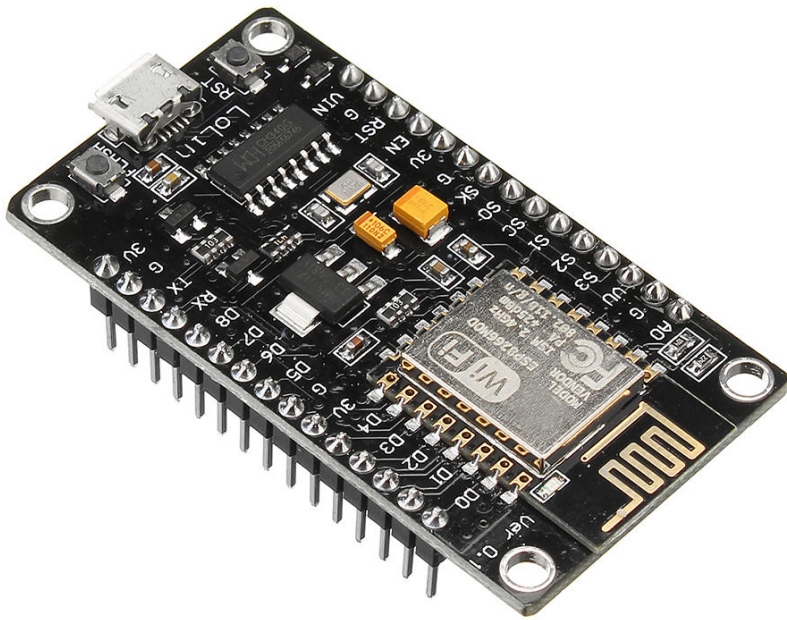
- (a) 0 Port GPIO dari D0 D10
- (b) Fungsionalitas PWM
- (c) Antar muka I2C dan SPI
- (d) Antar muka 1 Wire
- (e) ADC

NodeMCU pada dasarnya pengembangan dari ESP 8266 dengan firmware berbasis e-Lua. Pada NodeMcu dilengkapi dengan micro usb port yang berfungsi untuk pemrograman maupun power supply. Selain itu juga pada NodeMCU di lengkapi dengan tombol push button yaitu tombol reset dan flash. NodeMCU menggunakan bahasa pemrograman Lua yang merupakan package dari esp8266. Bahasa Lua memiliki logika dan susunan pemrograman yang sama dengan c hanya berbeda syntax. Jika menggunakan bahasa Lua maka dapat menggunakan tool Lua loader maupun Lua uploader.

Selain dengan bahasa Lua NodeMCU juga support dengan software Arduino IDE dengan melakukan sedikit perubahan board manager pada Arduino IDE.

Sebelum digunakan Board ini harus di Flash terlebih dahulu agar support terhadap tool yang akan digunakan. Jika menggunakan Arduino IDE menggunakan firmware yang cocok yaitu firmware keluaran dari AiThinker yang support AT Command. Untuk penggunaan tool loader Firmware yang di gunakan adalah firmware NodeMCU.





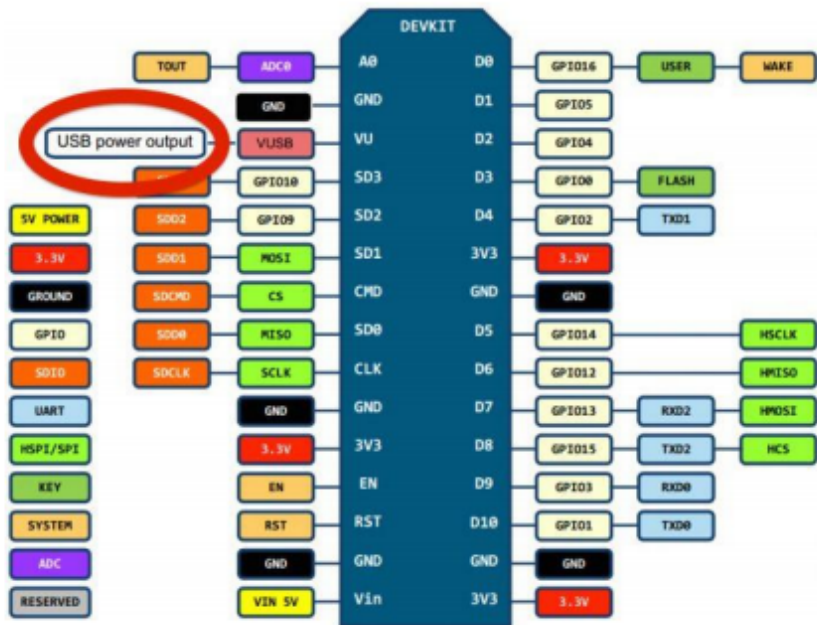
**Gambar 2.5** NodeMCU

Pada gambar 2.5 merupakan sebuah NodeMCU yang akan digunakan sebagai *microcontroller* untuk pembuatan *prototype* prediksi ketinggian air (PKA) untuk pendeteksi banjir peringatan dini dengan notifikasi melalui bot telegram .

## 2. Spesifikasi *Datasheet* NodeMCU

<b>SPESIFIKASI</b>	<b>NODEMCU V3</b>
<b>Mikrokontroler</b>	ESP8266
<b>Ukuran Board</b>	57 mmx 30 mm
<b>Tegangan Input</b>	3.3 ~ 5V
<b>GPIO</b>	13 PIN
<b>Kanal PWM</b>	10 Kanal
<b>10 bit ADC Pin</b>	1 Pin
<b>Flash Memory</b>	4 MB
<b>Clock Speed</b>	40/26/24 MHz
<b>WiFi</b>	IEEE 802.11 b/g/n
<b>Frekuensi</b>	2.4 GHz – 22.5 Ghz
<b>USB Port</b>	Micro USB
<b>Card Reader</b>	Tidak Ada
<b>USB to Serial Converter</b>	CH340G

**Gambar 2.6** Spesifikasi NodeMCU



**Gambar 2.7** Skema Posisi Pin NodeMCU

### 3. Protokol Hypertext Transfer Protoco (HTTP)

*Hypertext Transfer Protocol* (HTTP) adalah sebuah protokol jaringan lapisan aplikasi yang digunakan untuk sistem informasi terdistribusi, kolaboratif, dan menggunakan hipermedia.

Protokol HTTP didefinisikan oleh Tim Berners-Lee dalam RFC 1945 versi 1.0 dan digunakan sejak tahun 1990. Penyempurnaan protokol HTTP menjadi versi 1.1 yang dispesifikasikan oleh IETF dengan RFC 2616. HTTP bersifat *request response*, yaitu *HTTP client* (user agen misalnya) mengirimkan permintaan (*request*) ke HTTP server dan server merespon sesuai request tersebut. User agen sebagai contoh adalah Mozilla, Netscape, Google Chrome, atau browser berbasis teks contohnya Lynx atau links dan sebagainya.

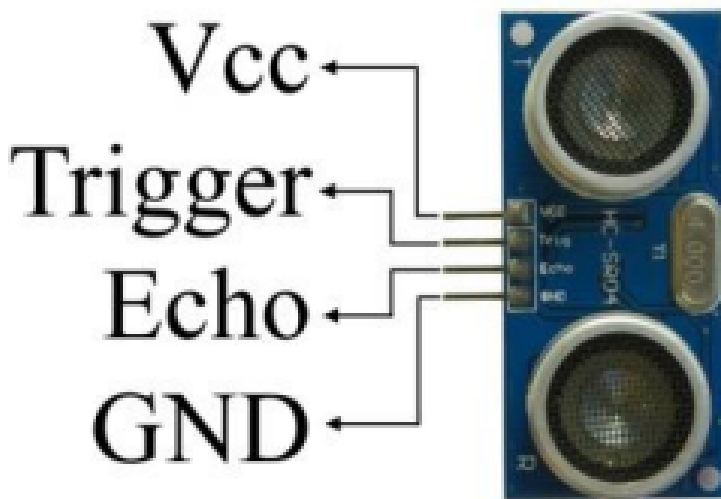
Pada protokol HTTP terdapat 3 jenis hubungan dengan perantara proxy, gateway, dan tunnel. Proxy bertindak sebagai agent penerus, menerima request dalam bentuk Uniform Resource Identifier (URI) absolut, mengubah format request dan mengirimkan request ke server yang ditunjukan oleh URI. 4 Gateway bertindak sebagai agen penerima dan menterjemahkan request ke protokol

server yang dilayaninya. Tunnel bertindak sebagai titik Relay antara dua hubungan HTTP tanpa mengubah request dan response HTTP. Tunnel digunakan jika komunikasi perlu melalui sebuah perantara dan perantara tersebut tidak mengetahui isi pesan dalam hubungan tersebut.

Perbedaan mendasar antara HTTP/1.1 dengan HTTP/1.0 adalah penggunaan hubungan persistent. HTTP/1.0 membuka satu koneksi untuk tiap permintaan satu URI, sedangkan HTTP/1.1 dapat menggunakan sebuah koneksi TCP untuk beberapa permintaan URI (persistent) (*header Connection : keepAlive*), kecuali jika *client* menyatakan tidak hendak menggunakan hubungan *persistent* (*header Connection : close*). HTTP *port* TCP default adalah 80, namun itu bisa diganti dengan nomor TCP lain diantara 1023 65535.

#### 4. Sensor Ultrasonic HC-SR04

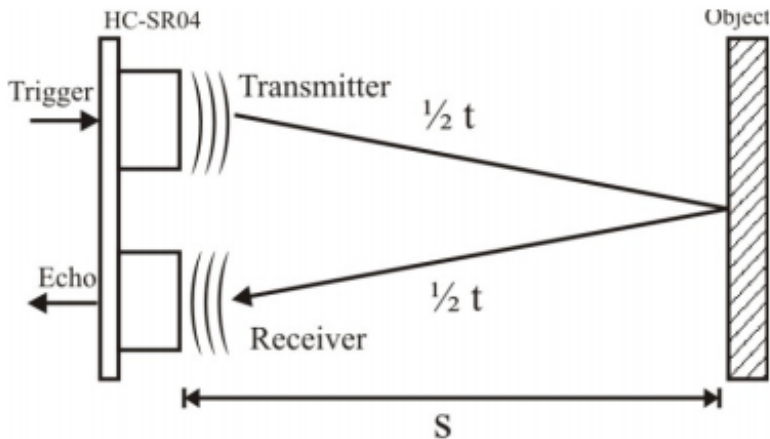
Sensor jarak ultrasonic HC-SR04 adalah sensor 40 KHz. HC-SR04 merupakan sensor ultrasonik yang dapat digunakan untuk mengukur jarak antara penghalang dan sensor. Konfigurasi pin dan tampilan sensor HC-SR04 diperlihatkan pada gambar dibawah :



**Gambar 2.8** Sensor HC-SR04

HC-SR04 memiliki 2 komponen utama sebagai penyusunnya yaitu ultrasonic transmitter dan ultrasonic receiver. Fungsi dari ultrasonic transmitter adalah memancarkan gelombang ultrasonik dengan frekuensi 40 KHz kemudian ultrasonic receiver menangkap hasil pantulan gelombang ultrasonik yang mengenai

suatu objek. Waktu tempuh gelombang ultrasonik dari pemancar hingga sampai ke penerima sebanding dengan 2 kali jarak antara sensor dan bidang pantul seperti yang diperlihatkan pada gambar :



**Gambar 2.9** Prinsip Pengukuran Jarak Sensor HC SR04

Prinsip pengukuran jarak menggunakan sensor ultrasonik HC-SR04 adalah, ketika pulsa trigger diberikan pada sensor, transmitter akan mulai memancarkan gelombang ultrasonik, pada saat yang sama sensor akan menghasilkan output TTL transisi naik menandakan sensor mulai menghitung waktu pengukuran, setelah receiver menerima pantulan yang dihasilkan oleh suatu objek maka pengukuran waktu akan dihentikan dengan menghasilkan output TTL transisi turun. Jika waktu pengukuran adalah  $t$  dan kecepatan suara adalah  $340 \text{ m/s}$ , maka jarak antara sensor dengan objek dapat dihitung dengan menggunakan Persamaan berikut :

$$s = t x \frac{340 \text{ m/s}}{2}$$

**Gambar 2.10** Rumus Perhitungan Jarak

Spesifikasi Sensor HC-SR04 :

- (a) Tegangan Kerja: DC 5V
- (b) *Operating Current*: 15mA
- (c) Frekuensi Kerja: 40Hz
- (d) Jarak Pengukuran Maks: 4m
- (e) Jarak Pengukuran Mins: 2cm
- (f) Mengukur Sudut: 15 derajat
- (g) Sinyal Input Pemicu: 10S TTL pulsa
- (h) inyal Output Echo Input sinyal tuas TTL dan kisaran proporsionals
- (i) Dimensi 45 \* 20 \* 15mm

Pin Number	Pin Name	Description
1	Vcc	The Vcc pin powers the sensor, typically with +5V
2	Trigger	Trigger pin is an Input pin. This pin has to be kept high for 10us to initialize measurement by sending US wave.
3	Echo	Echo pin is an Output pin. This pin goes high for a period of time which will be equal to the time taken for the US wave to return back to the sensor.
4	Ground	This pin is connected to the Ground of the system.

**Gambar 2.11** Konfigurasi Sensor Ultrasonic HC-SR04

HC-SR04 mempunyai modul rentang ultrasonik yang menyediakan fungsi pengukuran non-kontak 2 cm hingga 400 cm. Akurasi mulai dapat mencapai 3mm dan sudut efektif adalah  $\pm 15^\circ$ . Itu dapat didukung dari catu daya 5V.

Power Sensor menggunakan + 5V melalui Vcc pin Ground dari sensor. Saat Sensor kurang dari 15mA dan karenanya dapat langsung didukung oleh papan 5V pin (Jika tersedia). Trigger dan pin Echo keduanya / O pin dan karenanya mereka dapat terhubung ke I / pin O dari mikrokontroler. Untuk memulai pengukuran, pemicu pin harus dibuat tinggi untuk 10us dan kemudian dimatikan. Tindakan ini akan memicu gelombang ultrasonik pada frekuensi 40Hz dari pemancar dan penerima akan menunggu untuk gelombang untuk kembali. Setelah gelombang dikembalikan setelah dipantulkan oleh objek apa pun, pin Echo menjadi tinggi untuk jumlah waktu tertentu yang akan sama dengan waktu yang dibutuhkan gelombang untuk kembali ke sensor.

## 5. Sensor Ultrasonic US 100

Sensor US-100 adalah versi peningkatan dari US-020 Ultrasonic Sensor pada kelasnya HC-SR04 memiliki performa lebih bagus dibanding US-020 yang sudah dilengkapi dengan fitur kompensasi temperatur. Di kelasnya (*ultrasonic range sensor with temperature compensation*) US-100 merupakan modul sensor jarak terbaik.

Modul jarak ultrasonik bekerja dengan sistem sonar seperti yang digunakan pada kapal selam, yaitu dengan melepaskan sinyal dalam bentuk gelombang ultrasonik (gelombang suara dengan frekuensi sangat tinggi di luar jangkauan pendengaran telinga manusia) dan mengukur waktu hingga gelombang tersebut dipantulkan. Dengan mengetahui kecepatan suara di udara, kita dapat mengubah besaran waktu ini menjadi jarak dengan rumus:

jarak = ( selisih waktu \* kecepatan suara di udara ) / 2.

Fitur kompensasi suhu ini sangatlah penting untuk meningkatkan akurasi sensor sejenis ini terhubung kecepatan rambat suara di udara sangat terpengaruh oleh suhu / temperatur. Suara adalah sejenis energi kinetis. Molekul pada suhu tinggi memiliki tingkat energi lebih tinggi yang membuat mereka bergetar (vibrate) lebih cepat. Karena molekul ini bergetar lebih cepat, gelombang suara yang melewatinya dapat merambat dengan kecepatan lebih tinggi. Kecepatan rambat suara di udara pada suhu ruang (25C) sekitar 346 meter per detik, sementara pada suhu beku (0C), kecepatannya menurun menjadi 331 meter per detik. Untuk setiap derajat celcius kenaikan suhu, kecepatan rambatannya bertambah 60 cm per detik.

*Distance* Sensor US-100 ini mengukur suhu lingkungan (ambience temperature) dengan sensor suhu terpadu sehingga dapat mengkompensasi perbedaan suhu, menghasilkan pengukuran jarak yang sangat akurat. Berikut Gambar Sensor US 100 :



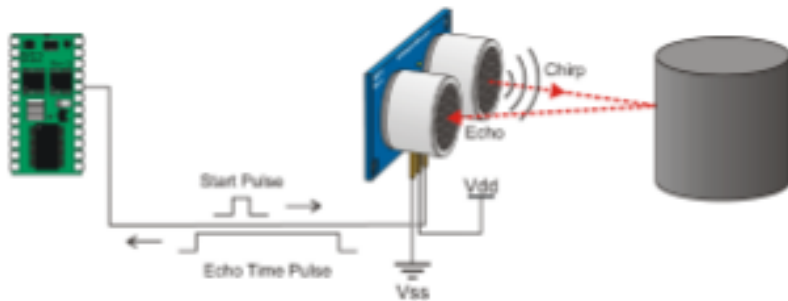
**Gambar 2.12** Gambar Sensor Ultrasonic US 100

Sensor ultrasonic menggunakan daya + 5V yang diatur melalui pin Vcc ad Ground dari sensor. Arus yang dikonsumsi oleh sensor kurang dari 15mA dan karenanya dapat langsung ditenagai oleh pin 5V on board (Jika tersedia). Trigger dan Echo pin keduanya adalah pin I / O dan karenanya mereka dapat dihubungkan ke pin I / O dari mikrokontroler. Untuk memulai pengukuran, pin pemicu harus dibuat tinggi untuk 10uS dan kemudian dimatikan. Tindakan ini akan memicu gelombang ultrasonik pada frekuensi 40Hz dari pemancar dan penerima akan menunggu gelombang kembali. Setelah gelombang dikembalikan setelah dipantulkan oleh objek apa pun, pin Echo menjadi tinggi untuk jumlah waktu tertentu yang akan sama dengan waktu yang dibutuhkan gelombang untuk kembali ke sensor .

Jumlah waktu selama pin Echo tetap tinggi diukur oleh MCU / MPU karena memberikan informasi tentang waktu yang dibutuhkan untuk gelombang untuk kembali ke Sensor.

## 6. Prinsip kerja Sensor ultrasonic





**Gambar 2.13** Prinsip Kerja Sensor Ultrasonic

Seperti yang ditunjukkan di atas, sensor adalah modul 4 pin, yang pin namanya masing-masing adalah Vcc, Trigger, Echo dan Ground. Sensor ini adalah sensor yang sangat populer digunakan dalam banyak aplikasi di mana mengukur jarak atau objek penginderaan diperlukan. Modul sensor Ultrasonik adalah cara mudah untuk mengukur jarak dari benda. Modul ini memiliki banyak aplikasi seperti sensor parkir, hambatan dan sistem pemantauan medan, pengukuran jarak industri, dll. Sistem ini memiliki stabilitas kinerja dan akurasi tinggi mulai dari 2cm hingga 450cm. Modul ini memiliki dua mata seperti proyek di bagian depan yang membentuk pemancar dan Penerima Ultrasonik. Sensornya bekerja dengan rumus :

$$\text{Jarak} = \text{Kecepatan} \times \text{Waktu}$$

**Gambar 2.14** Rumus perhitungan Jarak Sensor Ultrasonic

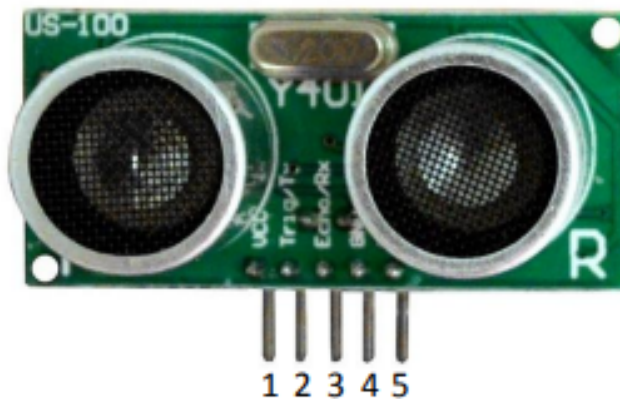
Pemancar ultrasonik mentransmisikan gelombang ultrasonik, gelombang ini bergerak di udara dan ketika ia keberatan dengan bahan apa pun itu dipantulkan kembali ke sensor gelombang. Untuk menghitung jarak menggunakan rumus di atas, harus mengetahui kecepatan dan waktu. Karena menggunakan gelombang Ultrasonik, kecepatan universal gelombang AS pada kondisi ruangan yang 330m / s. Sirkuit inbuilt pada modul akan menghitung waktu yang dibutuhkan untuk gelombang US untuk kembali dan menyalakan pin gema tinggi untuk jumlah waktu yang sama, dengan cara ini kita juga dapat mengetahui waktu yang dibutuhkan. Sekarang cukup hitung jaraknya menggunakan mikrokontroler atau mikroprosesor.

## 7. Spesifikasi(*DataSheet*)Sensor Ultrasonik US 100

Spesifikasi atau *Datasheet* dari sensor ultrasonic US 100 sebagai berikut :

- (a) Tegangan input: 5V DC
- (b) *Quiescent current*: kurang dari 2mA
- (c) output: 5V tinggi
- (d) Level output: pada akhir 0V
- (e) Sudut induksi: tidak lebih dari 15 derajat
- (f) Jarak deteksi: 2cm-450cm
- (g) Presisi: hingga 1mm
- (h) Dimensi: 4.4cm x 2.6cm x 1.4cm
- (i) Berat: 43g

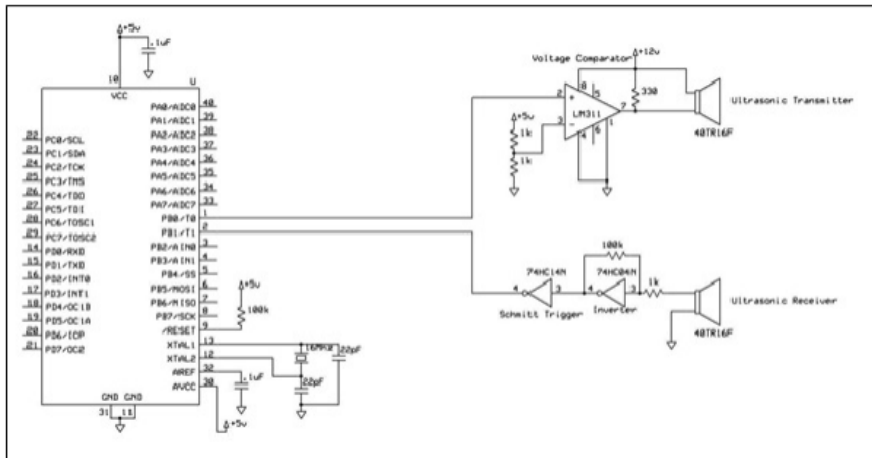
Adapun pin konfigurasi dari sensor ultrasonic Us 100 yaitu :



**Gambar 2.15** Pin Konfigurasi

- (a) VCC: 5V DC
- (b) Trig: trigger input
- (c) Echo: pulse output
- (d) GND: ground
- (e) GND: ground

Skema diagram sensor ultrasonic US 100 yaitu :



**Gambar 2.16** Skema diagram sensor ultrasonic US 100

#### 8. Cara *Test Sensor Ultrasonic US 100*

Komponen yang akan digunakan adalah:

- (a) Mikrokontroler (arduino yang kompatibel)
- (b) Modul sensor Ultrasonik US-100
- (c) Konektor pin
- (d) readboard Menggunakan kabel USB, sambungkan porta dari mikrokontroler ke komputer.
- (e) Kabel USB

Setelah hal diatas sudah terpenuhi maka ikuti langkah verikut ini :

- (a) Hubungkan komponen menggunakan konektor pin. Pin VCC terhubung ke catu daya 5V, pin GND terhubung ke GND dan pin Trig dan Echo terhubung ke pin I / O digital. Nomor pin akan didasarkan pada kode program yang sebenarnya.
- (b) Setelah koneksi perangkat keras, masukkan sketsa sampel ke dalam Arduino IDE.
- (c) Menggunakan kabel USB, sambungkan porta dari mikrokontroler ke komputer.
- (d) *Upload* program.
- (e) Lihat hasilnya di monitor serial.

#### 9. Sensor Yang Digunakan

Pada pembuatan *prototype* prediksi ketinggian air (PKA) untuk mendeteksi banjir peringatan dini ini menggunakan sensor ultra sonic US 100. Karena sensor ultrasonic US 100 ini akan membaca jarak lebih akurat dibandingkan dengan sensor ultrasonic HC-SR04.

Akan tetapi jika teman-teman ingin menggunakan sensor ultrasonic HC-SR04 tidak ada masalah, karena fungsi dari kedua sensor ini sama yaitu untuk membaca atau mengukur jarak dari suatu benda. Hanya saja jika teman-teman memilih sensor ultrasonic US 100 ini akan mengecek pengeluaran yang lebih besar dibandingkan dengan sensor ultrasonic HC-SR04.

#### 10. Buzzer

Buzzer merupakan sebuah komponen elektronika yang masuk dalam keluarga transduser, yang dimana dapat mengubah sinyal listrik menjadi getaran suara. Nama lain dari komponen ini disebut dengan *beeper*.

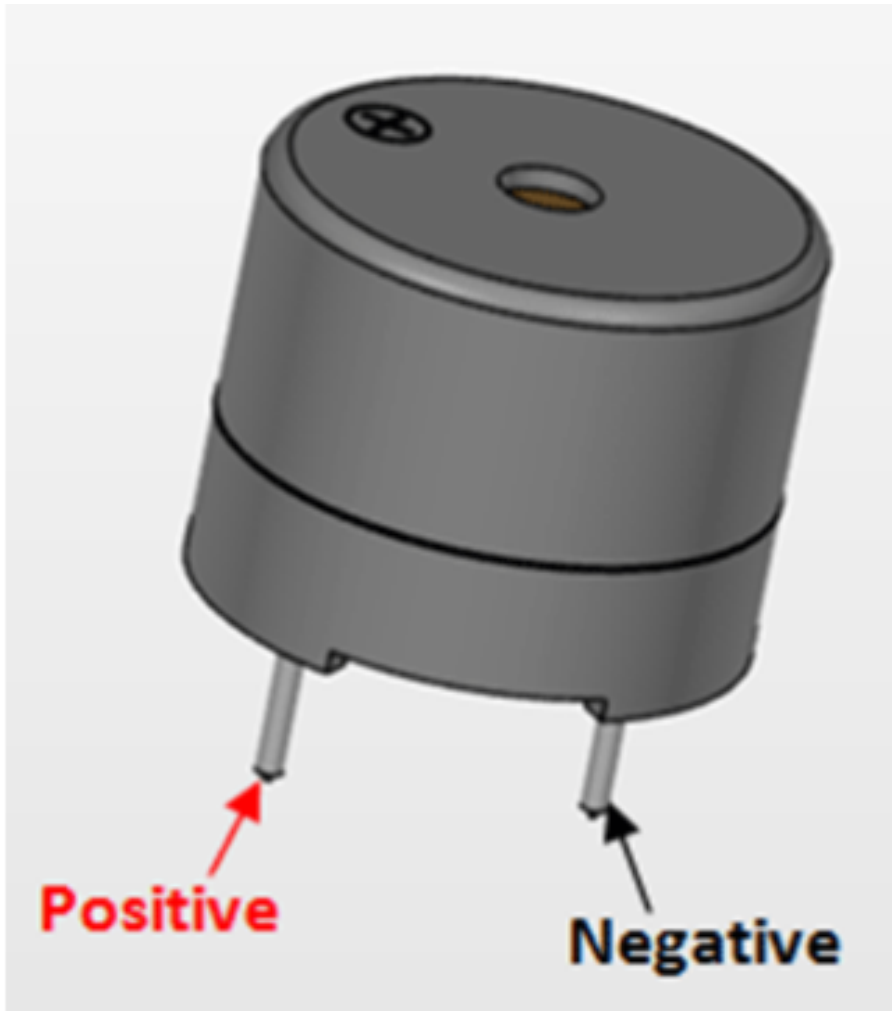


**Gambar 2.17** Buzzer

Dalam kehidupan sehari-hari, umumnya digunakan untuk rangkaian alarm pada jam, bel rumah, perangkat peringatan bahaya, dan lain sebagainya. Jenis-jenis yang sering ditemukan dipasaran yaitu tipe piezoelectric. Dikarenakan tipe ini

memiliki kelebihan seperti harganya yang relatif murah, mudah diaplikasikan ke dalam rangkaian elektronika.

#### 11. Cara Kerja Buzzer



**Gambar 2.18** Konfigurasi Pin Buzzer

Pada saat ada aliran catu daya atau tegangan listrik yang mengalir ke rangkaian yang menggunakan piezoelectric, maka akan terjadi pergerakan mekanis pada piezoelectric tersebut. Yang dimana gerakan tersebut mengubah energi listrik menjadi energi suara yang dapat didengar oleh telinga manusia. Piezoelectric menghasilkan frekuensi di range kisaran antara 1 5 kHz hingga 100 kHz yang

diaplikasikan ke Ultrasound. Tegangan operasional piezoelectric pada umumnya yaitu berkisar antara 3Vdc hingga 12 Vdc. Adapun konfigurasi pin buzzer yaitu sebagai berikut :

Pin Number	Pin Name	Description
1	Positive	Identified by (+) symbol or longer terminal lead. Can be powered by 6V DC
2	Negative	Identified by short terminal lead. Typically connected to the ground of the circuit

**Gambar 2.19** Keterangan Konfigurasi Pin Buzzer

## 12. Spesifikasi(*DataSheet*) Buzzer

Spesifikasi atau datasheet dari buzzer meliputi :

- (a) Nilai Tegangan: 6V DC
- (b) Tegangan Pengoperasian: 4-8V DC
- (c) Nilai *current*: 30mA
- (d) Tipe Suara: Bunyi Kontinu
- (e) Frekuensi resonansi: 2300 Hz
- (f) Kecil dan *package* rapih

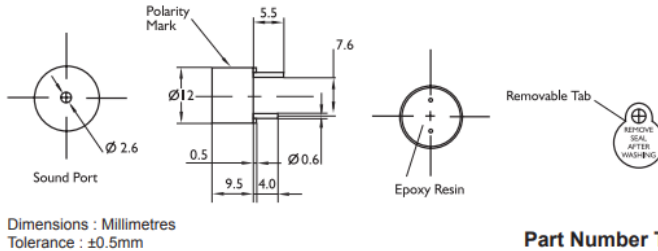
## 13. Cara menggunakan Buzzer

Buzzer adalah komponen kecil namun efisien untuk menambahkan fitur suara ke proyek / sistem. Ini adalah struktur 2-pin yang sangat kecil dan kompak sehingga dapat dengan mudah digunakan pada Breadboard dan bahkan pada PCB yang menjadikannya komponen yang banyak digunakan dalam sebagian besar aplikasi elektronik.

Ada dua jenis buzzers yang umumnya tersedia. Yang ditampilkan di sini adalah buzzer sederhana yang ketika diaktifkan akan membuat suara Continuous Beep... jenis lainnya disebut buzzer readymade yang akan terlihat lebih besar daripada ini dan akan menghasilkan Beep. Berbunyi. Berbunyi. Suara karena rangkaian osilasi internal yang ada di dalamnya. Tapi, yang ditunjukkan di sini paling banyak digunakan karena dapat disesuaikan dengan bantuan sirkuit lain agar sesuai dengan mudah dalam *prototype* kita.

Buzzer ini dapat digunakan hanya dengan menyalakannya menggunakan catu daya DC mulai dari 4V hingga 9V. Baterai 9V sederhana juga dapat digunakan, tetapi disarankan untuk menggunakan catu daya + 5V atau + 6V yang teregulasi. Buzzer biasanya dikaitkan dengan sirkuit *switching* untuk menghidupkan atau mematikan buzzer pada waktu yang diperlukan dan membutuhkan interval. Adapun diagram buzzer atau model 2D buzzer seperti pada gambar dibawah :

**Diagram**



**Part Number Table**

Description	Part Number
Buzzer, Electromech, 6V DC	ABI-009-RC

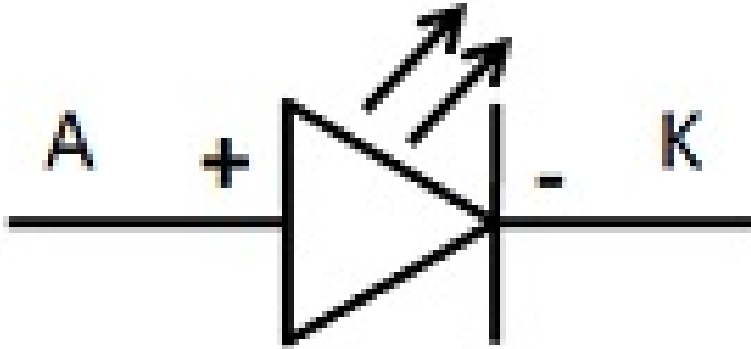
**Gambar 2.20** Diagram model 2D Buzzer

#### 14. Led

LED (*Light Emitting Diode*) 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.



**Gambar 2.21** Led



**Gambar 2.22** Simbol Led

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. Oleh karena itu, saat ini LED (Light Emitting Diode) yang bentuknya kecil telah banyak digunakan sebagai lampu penerang dalam LCD TV yang mengganti lampu tube.

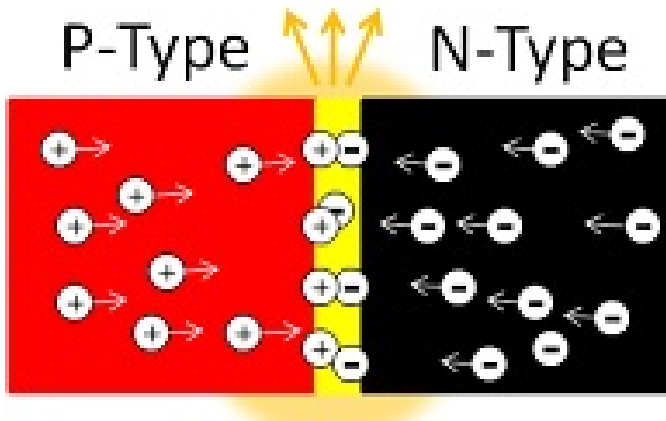
## 15. Cara Kerja Led

Seperti dikatakan sebelumnya, LED merupakan keluarga dari Dioda yang terbuat dari Semikonduktor. Cara kerjanya pun hampir sama dengan Dioda yang memiliki dua kutub yaitu kutub Positif (P) dan Kutub Negatif (N). LED hanya akan memancarkan cahaya apabila dialiri tegangan maju (bias forward) dari Anoda menuju ke Katoda.

LED terdiri dari sebuah chip semikonduktor yang di doping sehingga menciptakan junction P dan N. Yang dimaksud dengan proses doping dalam semikonduktor adalah proses untuk menambahkan ketidakmurnian (impurity) pada semikonduktor yang murni sehingga menghasilkan karakteristik kelistrikan yang diinginkan. Ketika LED dialiri tegangan maju atau bias forward yaitu dari Anoda (P) menuju ke Katoda (K), Kelebihan Elektron pada N-Type material akan berpindah ke wilayah yang kelebihan Hole (lubang) yaitu wilayah yang bermu-



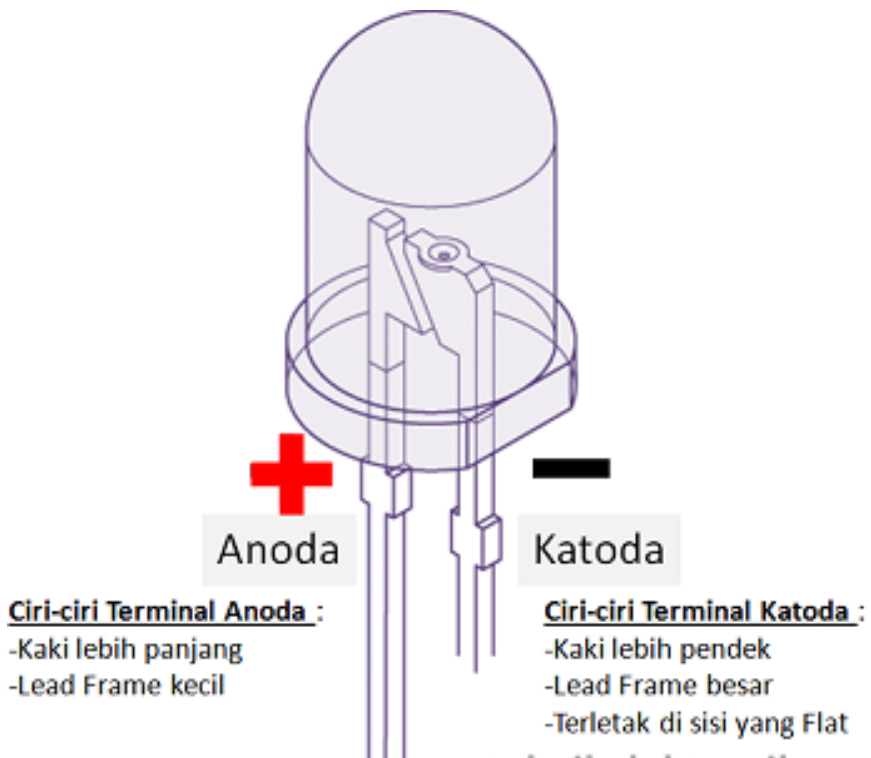
atan positif (P-Type material). Saat Elektron berjumpa dengan Hole akan melepaskan photon dan memancarkan cahaya monokromatik (satu warna).



**Gambar 2.23** Cara Kerja Led

LED atau *Light Emitting Diode* yang memancarkan cahaya ketika dialiri tegangan maju ini juga dapat digolongkan sebagai Transduser yang dapat mengubah Energi Listrik menjadi Energi Cahaya.

## 16. Cara Mengetahui Polaritas Led



**Gambar 2.24** Cara Melihat Polaritas

Untuk mengetahui polaritas terminal Anoda (+) dan Katoda (-) pada LED. Kita dapat melihatnya secara fisik berdasarkan gambar diatas. Ciri-ciri Terminal Anoda pada LED adalah kaki yang lebih panjang dan juga Lead Frame yang lebih kecil. Sedangkan ciri-ciri Terminal Katoda adalah Kaki yang lebih pendek dengan Lead Frame yang besar serta terletak di sisi yang Flat.

## 17. Antares



# ANTARES

**Gambar 2.25** Logo Antares

Antares merupakan sebuah *platform IoT* lokal milik Telkom yang telah mendapat pengakuan dari duna internasional yang dikembangkan oleh departemen media dan digital Telkom MDD. Antares menjadi jembatan dalam solusi IoT yang mendukung berbagai macam protokol yang umum digunakan untuk solusi IoT seperti MQTT, HTTP, websocket, dan CoAP disamping format data JSON dan XML. Selain itu, untuk memudahkan pengembang perangkat lunak dan keras disediakan pula *library* untuk Android dan *microcontroller* berbasis Arduino.



**Gambar 2.26** Tampilan Antares Antares

## 18. Resistor



**Gambar 2.27** Reistor

Resistor merupakan komponen elektronika yang tidak memiliki kutub sehingga dapat dipasang bolak balik yang tidak akan menimbulkan masalah pada peralatan elektronika. Pada dasarnya sangat jarang kerusakan barang elektronik disebabkan karena Resistor yang sudah rusak, tetapi jika memang terjadi kerusakan pada resistor tentunya hanya tinggal diganti yang baru dengan ukuran hambatan dan jenis resistor yang sama. Harga resistor di toko elektronik sangatlah murah dan untuk mengganti resistor yang rusak di PCB juga sangat mudah.

Resistor atau hambatan salah satu komponen elektronika yang memiliki nilai hambatan tertentu, dimana hambatan ini akan menghambat arus listrik yang mengalir melaluinya. Sebuah resistor biasanya terbuat dari bahan campuran Carbon. Namun tidak sedikit juga resistor yang terbuat dari kawat nikrom, sebuah kawat yang memiliki resistansi yang cukup tinggi dan tahan pada arus kuat. Contoh lain penggunaan kawat nikrom dapat dilihat pada elemen pemanas setrika. Jika elemen pemanas tersebut dibuka, maka terdapat seutas kawat spiral yang biasa disebut dengan kawat nikrom

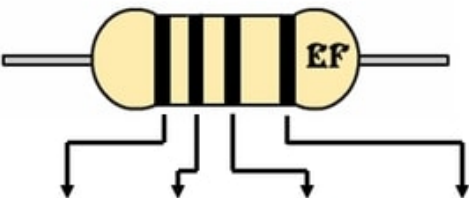
Satuan Resistor adalah Ohm (simbol:  $\Omega$ ) yang merupakan satuan SI untuk resistansi listrik. Dalam sejarah, kata ohm itu diambil dari nama salah seorang fisikawan hebat asal Jerman bernama George Simon Ohm. Beliau juga yang mencetuskan keberadaan hukum ohm yang masih berlaku hingga sekarang.

## 19. Fungsi Resistor

Resistor berfungsi sebagai penghambat arus listrik. Jika ditinjau secara mikroskopik, unsur-unsur penyusun resistor memiliki sedikit sekali elektron bebas. Akibatnya pergerakan elektronnya menjadi sangat lambat. Sehingga arus yang terukur pada multimeter akan menunjukkan angka yang lebih rendah jika dibandingkan rangkaian listrik tanpa resistor.

Namun meskipun misalnya kita menyusun rangkaian listrik tanpa resistor, bukan berarti tidak ada hambatan listrik didalamnya. Karena setiap konduktor pasti memiliki nilai hambatan, meskipun relatif kecil. Namun dalam perhitungan matematis, biasanya kita abaikan nilai hambatan pada konduktor tersebut, dan kita anggap konduktor dalam kondisi ideal. Itu berarti besar resistansi konduktor adalah nol.

## 20. Macam-Macam Resistor Sesuai Warna



WARNA	Gelang I	Gelang II	Gelang III		TOLERANSI
<b>HITAM</b>	0	0	$10^0$	x 1	-
<b>COKLAT</b>	1	1	$10^1$	x 10	$\pm 1 \%$
<b>MERAH</b>	2	2	$10^2$	x 100	$\pm 2 \%$
<b>ORANGE</b>	3	3	$10^3$	x 1 K	-
<b>KUNING</b>	4	4	$10^4$	x 10 K	-
<b>HIJAU</b>	5	5	$10^5$	x 100 K	-
<b>BIRU</b>	6	6	$10^6$	x 1 M	-
<b>UNGU</b>	7	7	$10^7$	x 10 M	-
<b>ABU – ABU</b>	8	8	$10^8$	x 100 M	-
<b>PUTIH</b>	9	9	$10^9$	x 1 G	-
<b>EMAS</b>	-	-	$10^{-1}$	x 0,1	$\pm 5 \%$
<b>PERAK</b>	-	-	$10^{-2}$	x 0,01	$\pm 10 \%$
<b>Tak Berwarna</b>	-	-			$\pm 20 \%$

**Gambar 2.28** Macam-Macam Resistor Sesuai Warna

Dari gambar diatas kita dapat menyesuaikan penggunaan resistor untuk kebutuhan kita.

## 21. Resistor Yang Digunakan

Resistor yang digunakan untuk pembuatan *prototype* prediksi ketinggian air (PKA) untuk mendeteksi banjir peringatan dini ini yaitu resistor 300 OHM 1/2WATT Carbon Film Resistor seperti pada gambar dibawah ini :



**Gambar 2.29** Resistor Yang Digunakan

Resistor 300 OHM 1/2WATT Carbon Film digunakan untuk menjadi hambatan lampu led, agar pada saat lampu led menyala tidak terlalu terang dan tidak menyebabkan sakit mata pada saat dilihat. Resistor 300 OHM 1/2WATT Carbon Film ini cocok digunakan pada prototipe ini karena cocok sebagai hambatan lampu led karena hanya 300 OHM. Sehingga hambatan yang diberikan oleh resistor ini kepada led tidak terlalu besar.

Oleh karena itu lampu led akan menyala dengan tingkat penerangan yang sedang tidak terlalu bersinar. Jika kita menggunakan resistor yang hambatannya lebih dari 300 OHM dikhawatirkan lampu led akan menyala sangat redup dikarenakan diberikan hambatan pada lampu led sangat besar. Jika kita memberi hambatan kurang dari 300 OHM pada lampu led. Maka hal yang akan terjadi yaitu lampu led akan memancarkan cahaya sangat terang. Cahaya yang dipancarkan oleh lampu led terlalu terang maka akan mengakibatkan rasa sakit dimata ketika kita melihatnya. Oleh karena itulah kita harus mengetahui hambatan yang diperlukan untuk komponen yang digunakan pada saat membuat project. Selain itu resistor ini akan dipergunakan juga pada buzzer. Sehingga suara yang dikeluarkan oleh buzzer tidak terlalu keras yang dapat merusak indra pendengaran kita. selain itu jika buzzer diberikan hambatan akan mengeluarkan suara yang lebih kecil.

Disamping itu harga dari resistor relatif murah. jika kita ingin membeli atau membutuhkan resistor untuk suatu project yang sedang dibuat maupun yang akan dibuat, kalian hanya perlu mengeluarkan biaya *cost* sebanyak Rp.100,-

/buah. jika kalian membutuhkan 10 resistor maka kita hanya perlu mengeluarkan Rp.1000,-

## 22. Bot Telegram



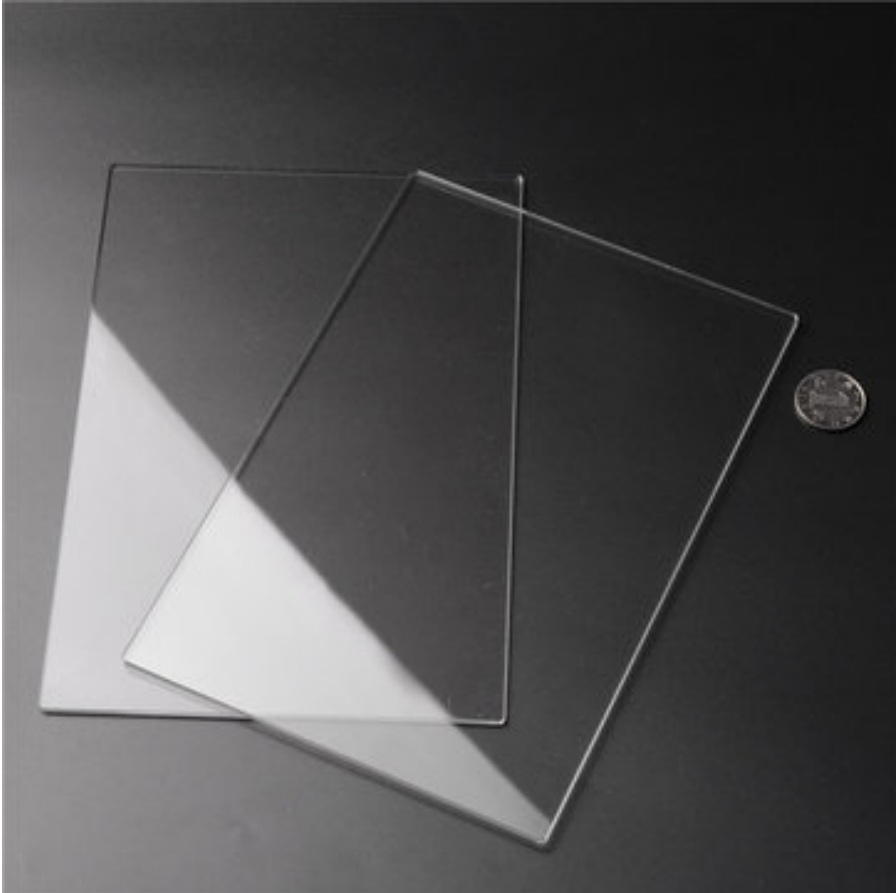
**Gambar 2.30** Logo Telegram

Seiring Messenger Telegram yang mulai diinstall banyak orang dan dipergunakan untuk percakapan sehari-hari. Memang Telegram belum sepopuler WhatsApp, BBM, maupun Line. Namun, bisa jadi suatu saat akan menjadi suatu messenger yang potensial mendapatkan hati dikalangan masyarakat maya. Menurut Cokrojoyo kelebihan dari Telegram ini adalah adanya landasan untuk menggunakan *Application Programming Interface*(API) untuk masyarakat luas. Salah satu API yang disediakan adalah fitur bot. Bot Telegram adalah bot yang saat ini mulai populer dipergunakan.

Keunggulan pertama dari aplikasi Telegram ini adalah fleksibel. Artinya Anda bisa membuat fitur-fitur tambahan yang disertakan dalam aplikasi Telegram ini. Misalkan Anda ingin membuat polling. Anda bisa membuatnya tanpa sendiri dan menambahkan di Telegram Anda tanpa harus menunggu ada update fitur dari Developernya. Sehingga ini membuat Telegram menjadi lebih lebih fleksibel jika dibanding aplikasi sejenis seperti WhatsApp ataupun LineMessenger.

Selain itu Keunggulan ke dua dari Telegram adalah pesan berbasis awan (cloud-based message). Artinya dengan Telegram Anda bisa berkomunikasi dengan siapapun (yang juga punya akun Telegram) tanpa batasan device/ gadget.

### 23. Akrilik



**Gambar 2.31** Bentuk Akrilik

Pada pembuatan rangka *prototype* ini menggunakan akrilik. Akrilik adalah semacam plastik yang menyerupai kaca, namun memiliki sifat yang membuatnya lebih unggul daripada kaca, akrilik itu lembaran plastik yang super keras. Warnanya yang tak cepat pudar dan bobotnya yang ringan menjadi keunggulan akrilik hingga menjadi bahan baku barang kerajinan. Akrilik digunakan untuk membuat berbagai produk. akrilik adalah bagian dari plastik sintesis yang berasal dari asam akrilik, akrilik terbuat dari 100 MMA (metil metakrilat monomer), sudah paham kan temen-temen yuk kita bahas lebih dalam lagi mengenai akri-



lik. Akrilik memiliki nilai estetika yang tinggi selain tidak mudah pecah dan tahan cuaca, akrilik juga tidak akan berkerut atau mengembang, Akrilik lebih kuat dari kaca, sehingga lebih tahan dan tidak pecah sehingga lebih lebih aman.

#### 24. Kabel Jumper

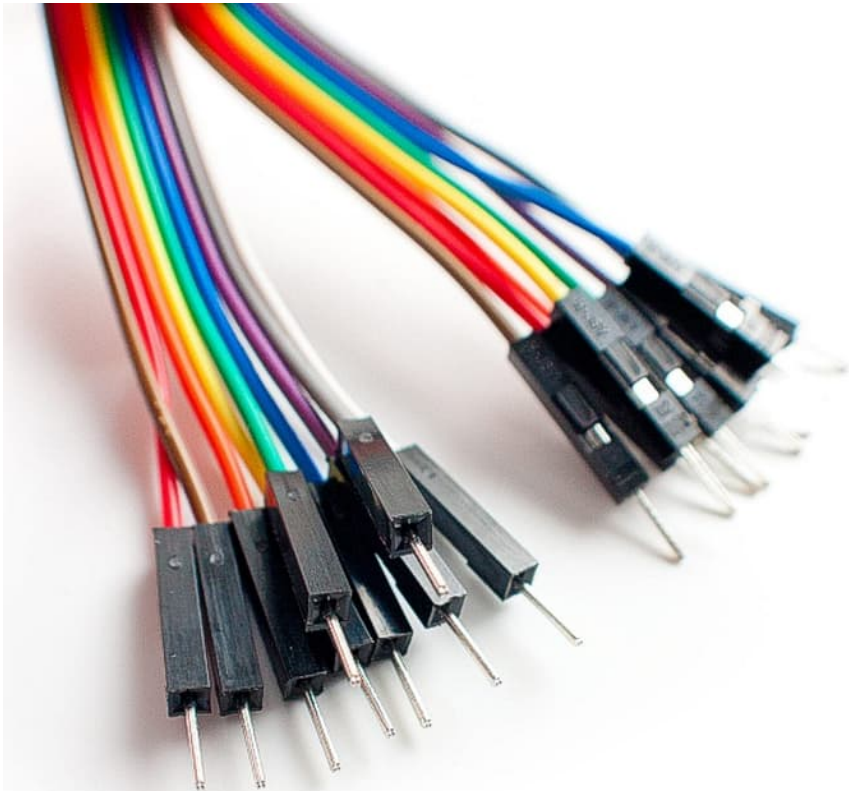


**Gambar 2.32** Kabel Jumper

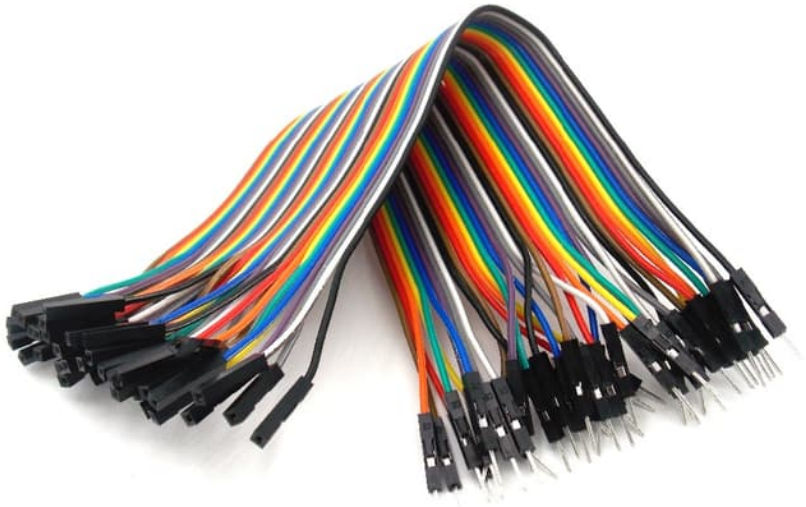
Kabel Jumper digunakan untuk menghubungkan komponen satu dengan komponen lainnya. Selain itu kabel jumper ini juga untuk menghubungkan komponen dengan *microcontroller*. Ada beberapa jenis kabel jumper yaitu sebagai berikut :



**Gambar 2.33** Kabel Jumper Female



**Gambar 2.34** Kabel Jumper Male

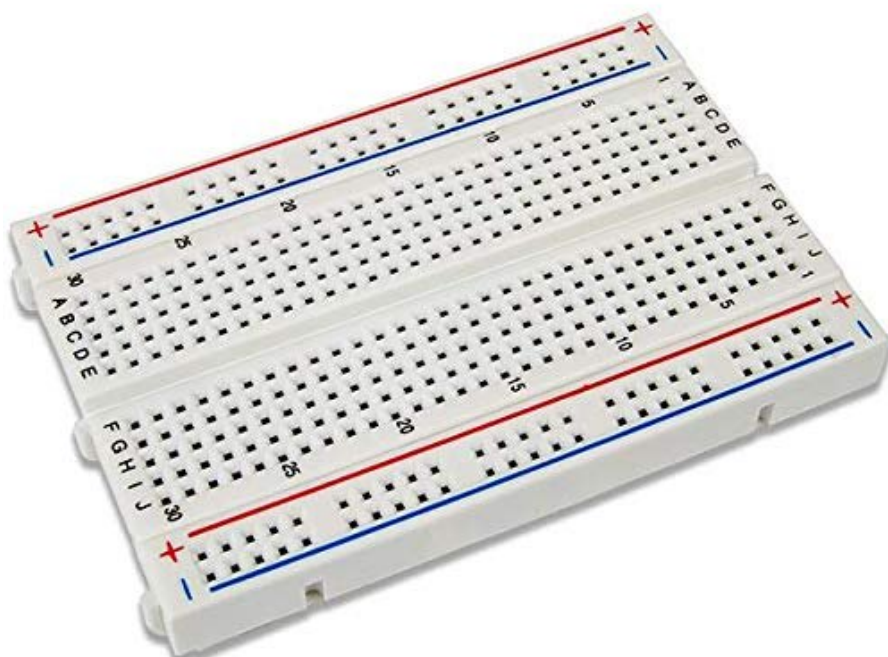


**Gambar 2.35** Kabel Jumper Male-Female

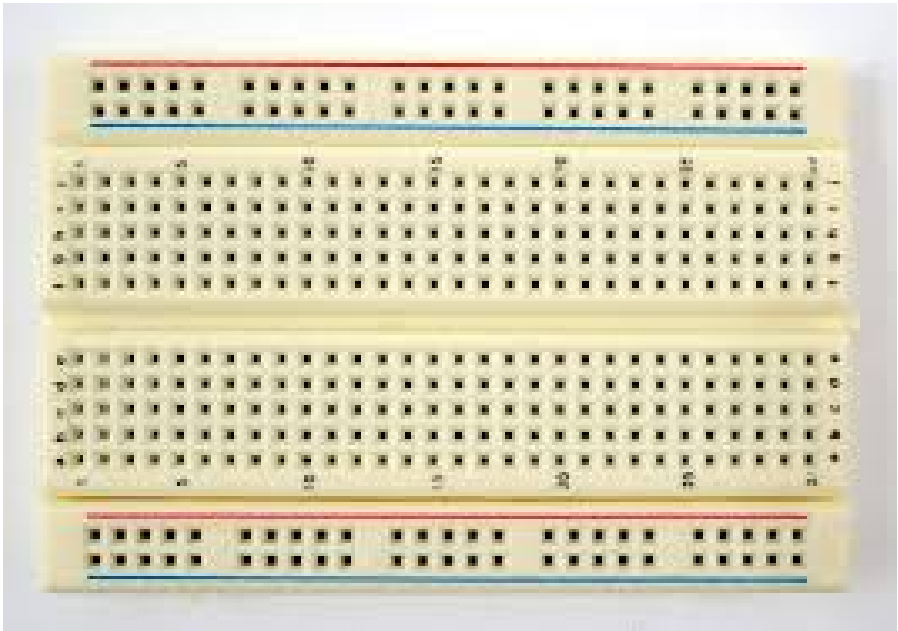
## 25. BreadBoard

Breadboard adalah board yang digunakan untuk membuat rangkaian elektronik sementara dengan tujuan uji coba atau prototipe tanpa harus menyolder. Dengan memanfaatkan breadboard, komponen-komponen elektronik yang dipakai tidak akan rusak dan dapat digunakan kembali untuk membuat rangkaian yang lain. Breadboard umumnya terbuat dari plastik dengan banyak lubang-lubang di atasnya. Lubang-lubang pada breadboard diatur sedemikian rupa membentuk pola sesuai dengan pola jaringan koneksi di dalamnya.

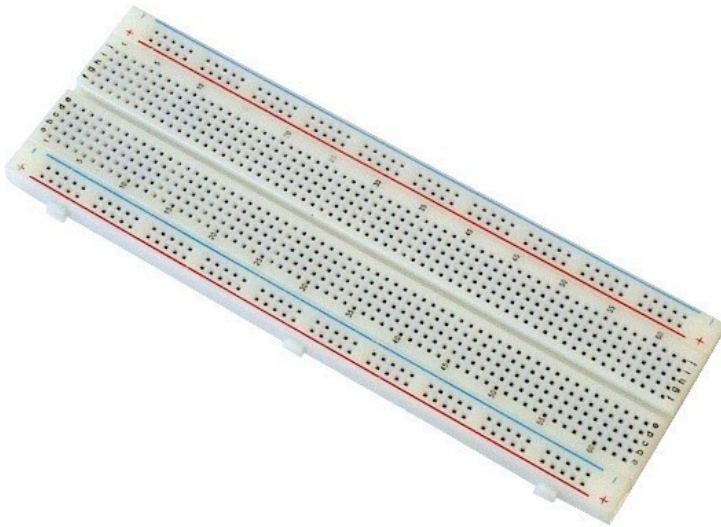
Breadboard yang tersedia di pasaran umumnya terbagi atas 3 ukuran: mini breadboard, medium breadboard atau large breadboard. Mini breadboard memiliki 170 titik koneksi (bisa juga lebih). Kemudian medium breadboard memiliki 400 titik koneksi. Dan large breadboard memiliki 830 titik koneksi.



**Gambar 2.36** Breadboard Mini



**Gambar 2.37** Breadboard Medium

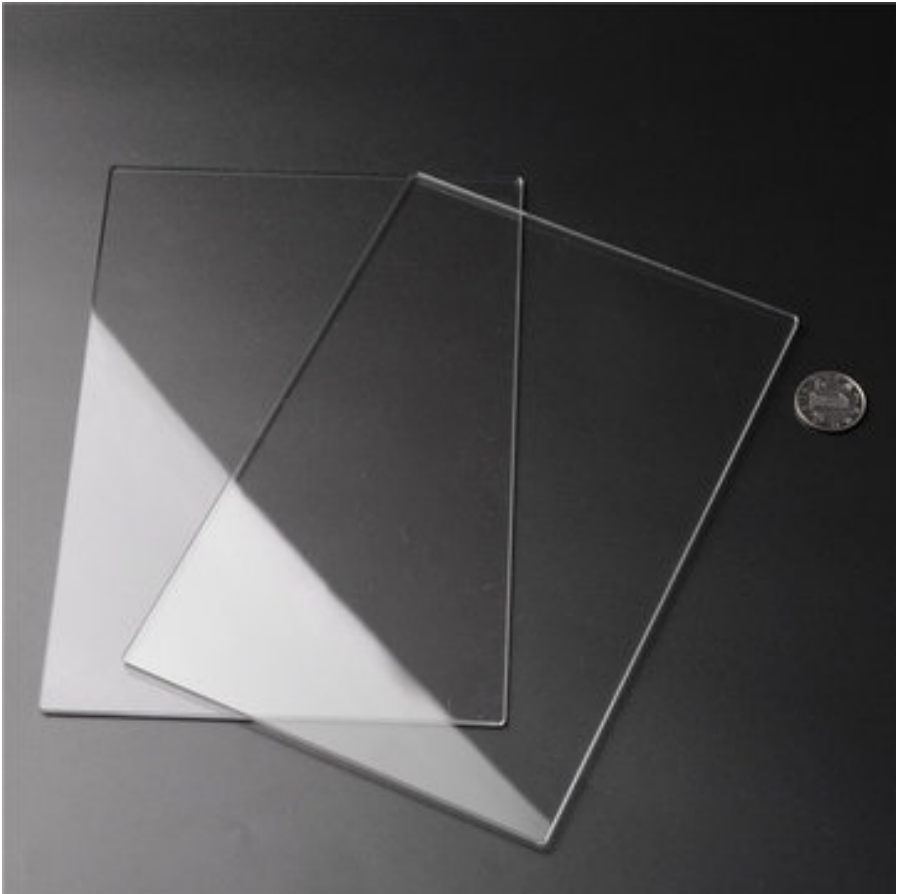


**Gambar 2.38** Breadboard Large

## 2.6 Pembuatan *Prototype*

Setelah membahas tentang komponen yang akan digunakan kemudian kita lanjutkan pada tahapan pembuatan *prototype*. Dalam tahapan pembuatan *prototype* ini tidak boleh ada salah satu step yang terlewatkan karena akan terjadi hal yang fatal. Berikut merupakan langkah-langkah pembuatan :

1. Siapkan 6 buah akrilik bening berbentuk persegi empat berukuran 20cm x 20cm dengan ketebalan 3mm. Dengan memilih akrilik dengan ketebalan 3mm rangka pada *prototype* yang akan dibuat akan lebih kuat dan lebih kokoh. Seperti pada gambar dibawah ini :



**Gambar 2.39** Contoh Akrilik yang digunakan

2. Siap lem korea atau lem akrilik untuk menempelkan akrilik satu dengan akrilik yang lainnya. Contoh lem yang harus digunakan sebagai berikut :





**Gambar 2.40** Contoh Lem Korea



**Gambar 2.41** Contoh Lem Akrilik

Pada pembuatan *prototype* prediksi ketinggian air (PKA) untuk mendeteksi banjir sebagai peringatan dini ini menggunakan lem khusus akrilik, jika teman-teman ingin menggunakan lem korea tentunya bisa saja. Tetapi untuk masing-masing lem itu ada kelebihan maupun kekurangannya.

Kenapa saya lebih memilih lem akrilik dibandingkan lem korea? karena lem akrilik lebih kuat untuk menempelkan akrilik satu dengan akrilik lainnya, hanya dengan sekali olesan. Sedangkan lem korea agar lebih kuat menempelkan akrilik satu dengan yang lainnya harus mengoleskan lemnya beberapa kali. Selain itu jika menggunakan lem korea teksturnya sangat cair sehingga lem akan berceceran yang mengakibatkan tidak rapih pada saat proses menempelkan akrilik.

Selain hal tersebut ada beberapa faktor lagi yaitu jika menggunakan lem akrilik kita tidak perlu membutuhkan lem terlalu banyak karna hanya perlu satu olesan saja untuk merekatkan akrilik seperti yang telah dijelaskan sebelumnya. Sebaliknya jika kita menggunakan lem korea maka kita membutuhkan lem korea yang lebih banyak dibandingkan dengan lem akrilik.

Disamping hal itu jika kita menggunakan lem akrilik ada kekurangannya juga dalam segi *cost* atau pengeluaran dikarenakan lem akrilik lebih mahal dibandingkan dengan lem korea.

3. Siapkan sticky note warna warni. Sticky note ini akan digunakan sebagai level peringatan pada kerangka *prototype*. Contoh sticky note yang digunakan sebagai berikut :



**Gambar 2.42** Contoh Sticky Note

Warna Sticky note yang digunakan hanya 3 warna tidak akan menggunakan semua warna. Warna yang digunakan yaitu warna merah, hijau, dan kuning. Warna merah digunakan untuk level awas banjir, warna hijau digunakan untuk level siaga banjir dan warna kuning digunakan untuk level aman dari banjir.

Jika teman-teman ingin mengganti warna-warna sticky note untuk level-level banjir tentu saja boleh . Tapi harus ada beberapa hal yang harus diperhatikan karena setiap warna memiliki arti atau *filosofinya* yang berbeda. Seperti contohnya teman-teman memilih warna coklat untuk level awas banjir itu sangat tidak *sinkron*.

4. Siapkan double tape, karena double tape ini akan digunakan untuk menempelkan sticky note pada kerangka *prototype*.



**Gambar 2.43** Contoh Double TApe

Disini menggunakan double tape dengan ukuran yang kecil karena disesuaikan dengan ukuran sticky note yang digunakan. Jika teman-teman menggunakan sticky note yang berukuran sedang maka harus menggunakan double tape yang berukuran sedang agar sticky note dapat menempel dengan rapih.

## 5. *styrofoam* papan



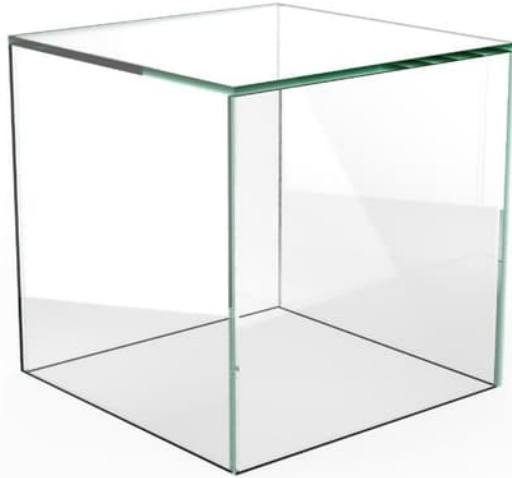
**Gambar 2.44** Contoh Styrofoam Papan

*styrofoam* papan ini digunakan untuk alas komponen pada kerangka *prototype* prediksi ketinggian air (PKA) untuk mendeteksi banjir peringatan dini. Fungsi diberikan alas untuk komponen yaitu agar komponen tidak bersentuhan langsung dengan permukaan akrilik.

## 2.7 Merakit *Kerangka Prototype*

Setelah semua komponen dan bahan- bahan yang dibutuhkan untuk membuat prototipe ini tersedia. Maka tahapan selanjutnya yaitu perakitan alat. perhatikan langkah- langkah berikut ini :

1. Tempelkan akrilik menjadi berbentuk seperti kubus, tetapi pada bagian atas atau atap nya tidak perlu ditutup sehingga seperti pada gambar dibawah ini :



**Gambar 2.45** Kerangka Prototipe

2. Setelah itu membuat kotak kotak kecil yang nantinya sebagai tempat komponen. Sehingga seperti gambar dibawah ini :



**Gambar 2.46** Kerangka Prototipe Tempaan Penyimpanan Komponen

Setelah kedua rangka prototipe tersebut dibuat maka rekatkan kerangka tersebut dengan kerangka prototipe yang sebelumnya yang telah dibuat. Caa merekatkannya yaitu dengan lem korea atau lem akrilik sesuai dengan pembahasan sebelumnya.

## **2.8 Perakitan dan Memprogram Komponen *Prototype***

### **2.8.1 Perakitan Komponen**

Komponen elektronika adalah komponen yang paling kompleks dan sensitif, tak jarang terjadi kerusakan apabila komponen yang digunakan tak sesuai semestinya, misalkan kelebihan daya, arus, short, dll. Selain itu tidak semua komponen ataupun perangkat elektronika dapat dijumpai dan dibeli dengan mudah, salah satu contohnya adalah osiloskop, spektrum analyzer, function generator, dll.

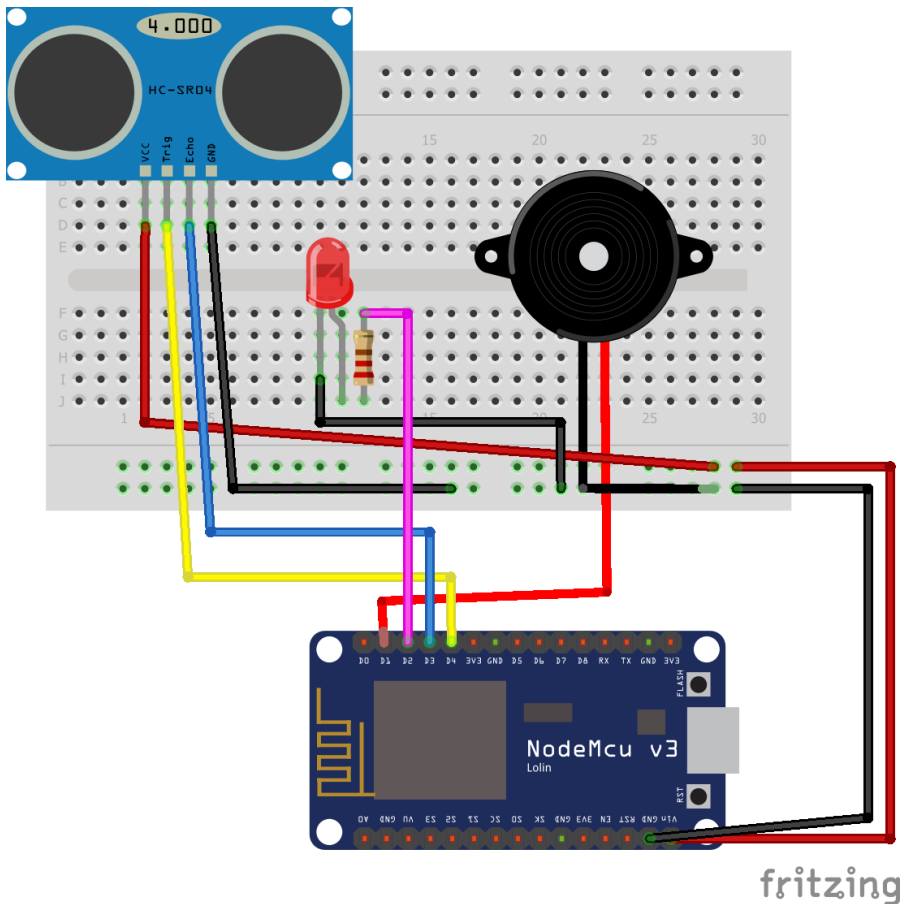
Maka dari itu software simulasi dapat membantu proses merancang rangkaian elektronika sebelum dipatenkan langsung pada komponen. Bukan hanya itu, kitapun dapat menggunakan perangkat elektronika yang tidak dapat kita jangkau (beli dan dijumpai) seperti halnya osiloskop dalam software simulasi tersebut.

Untuk membuat suatu rangkaian atau skema kita bisa menggunakan beberapa software seperti :

## 1. Fritzing

Fritzing merupakan aplikasi berbasis free software yang dapat kita unduh secara gratis. Aplikasi ini berfungsi untuk membuat sebuah skema rangkaian elektronika secara nyata (real) Karena komponen yang berada pada rangkaian tersebut digambarkan dengan sangat mirip dengan komponen aslinya. Aplikasi ini tidak hanya dilengkapi dengan skema rangkaian yang sangat baik, banyak fitur lainnya yang dapat kita coba seperti PCB layout dan programming Arduino langsung pada aplikasi ini. Sayangnya aplikasi ini hanya dapat membuat skema saja, kita tidak dapat melakukan simulasi pada rangkaian yang sudah kita buat.



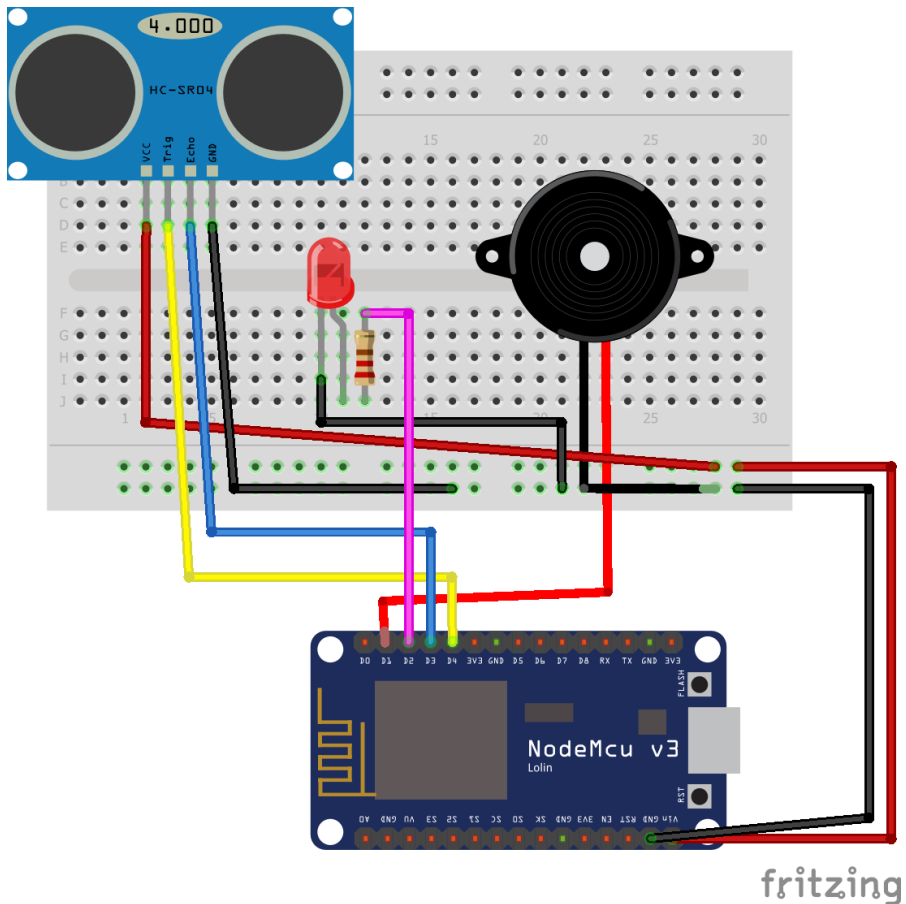


**Gambar 2.47** Rangkaian Alat

2. LiveWire
3. Electronics Workbench (EWB)
4. NI Multisim
5. Autodesk Circuits
6. Proteus ISIS

Tahapan selanjutnya yaitu membuat atau merakit komponen untuk prototipe. Tetapi ada satu hal sebelum kita masuk ke proses perakitan yaitu kita harus membuat sebuah rangkaian atau skema komponen. Tujuan dibuatnya suatu rangkaian atau skema ini untuk mempermudah kita pada saat proses perakitan.

perhatikan langkah - langkah berikut ini :



**Gambar 2.48** Rangkaian Alat

Agar mempermudah pada saat proses perakitan komponen maka harus mengikuti gambar rangkaian diatas. Untuk penjelasannya sebagai berikut :

1. Sambungkan Vin pada NodeMCU pada *breadboard* bagian positif (+). Vin merupakan Power atau VCC dengan memiliki daya 5V. Jadi jika tidak ada 5V dalam sebuah *microcontroller* dapat di gantikan dengan vin.



## BAB 3

---

# JUDUL BAGIAN KEDUA

---

### 3.1 Variabel

Variabel adalah sebuah tempat untuk menampung value dimemori, dapat dimisalkan seperti sebuah ruangan atau wadah, variabel dibagi dua berdasarkan ruang lingkup yaitu variabel lokal dan global, untuk menentukan variabel global atau lokal itu tergantung dari tempat dideklarasikannya variabel pada program yang sedang dibuat. Variabel global yaitu variabel yang dapat diakses di semua lingkup dalam program yang sedang dibuat, dalam kata lain variabel global ini dapat dikenali oleh semua fungsi dan prosedur, sementara variabel lokal yaitu variabel yang dapat diakses hanya di lingkup khusus, dalam kata lain variabel lokal ini hanya bisa diakses pada fungsi/prosedur dimana variabel itu dideklarasikan.

Berikut merupakan standar-standar dalam penulisan variabel:

1. Nama variabel diawali dengan huruf atau garis bawah, contoh: nama, \_nama, namaKu, nama\_variabel.
2. Karakter selanjutnya dapat berupa huruf, garis bawah atau angka, contoh: \_\_nama, nama1, p1.

3. Nama variabel tidak boleh diawali dengan angka
4. Karakter bersifat case-sensitive (huruf besar dan huruf kecil dibedakan), contoh: Nama dan NAMA keduanya memiliki arti yang berbeda dan merupakan variabel yang berbeda.
5. Nama variabel tidak boleh menggunakan kata kunci yang ada pada bahasa pemrograman python, contoh: if, else, while

## 3.2 Input dan Output

Input & output bertujuan agar pengguna dan program dapat berinteraksi, Perintah input() berguna untuk meminta inputan dari user, sehingga memungkinkan user untuk menginputkan data.

Perintah print() berguna untuk menampilkan output dari data yang diinputkan oleh user, sehingga data yang diinputkan user dapat ditampilkan ke layar.

Contoh dari penggunaan input dan output adalah sebagai berikut:

```
1 #Input yang ditujukan untuk user
2 nama= informatics research center
3
4 #output yang didapatkan user
5 print( Halo , nama, selamat datang )
```

## 3.3 Operasi Aritmatika

Python memiliki operasi aritmatika, antara lainnya seperti :

1. penjumlahan (+)
2. pengurangan (-)
3. perkalian (\*)
4. pembagian (/)
5. sisa bagi/modulus (%)
6. pemangkatan (\*\*)

Penggunaan dari simbol simbol ini sama hal nya dengan fungsi aritmatika pada umumnya.

## 3.4 Perulangan

Dalam membuat sebuah program, terkadang kita memerlukan satu baris atau satu blok kode yang sama secara berulang, disini fungsi perulangan dipakai sehingga kita tidak perlu menulis baris atau blok kode yang sama secara terus menerus, dalam python perulangan dibagi menjadi 2, yaitu for dan while.

### 3.4.1 For

For merupakan perulangan yang akan mengulang kondisi true sampai batas yang telah ditentukan, biasanya digunakan untuk perulangan yang mana parameter pengulangannya menggunakan list atau range. Berikut ini merupakan contoh penggunaan sintaks perulangan for.

```
1 for i in range (0 ,10):
2     print ( i )
```

### 3.4.2 While

While merupakan perulangan yang akan terjadi apabila kondisinya True, perulangan akan terus berjalan hingga diperoleh kondisi False. Berikut ini merupakan contoh penggunaan sintaks perulangan while.

```
1 #perulangan while
2 hitung = 0
3 while (hitung < 9):
4     print ( hitungan ke :      , hitung )
5     hitung = hitung + 1
6
7 print ( "Good bye!" )
```

## 3.5 Kondisi

Pengambilan keputusan kadang diperlukan dalam sebuah program untuk menentukan tindakan apa yang akan dilakukan sesuai dengan kondisi yang terjadi, contoh kasus misalkan ada seorang anak bernama idam, seorang manusia yang membutuhkan makan, jika idam lapar maka idam akan makan. Maka dapat dijabarkan seperti dibawah ini :

Kondisi, jika :

Idam lapar

Maka :

Idam akan makan

Namun terkadang kondisi juga diberikan tambahan opsi sebuah kondisi tambahan, misalkan jika idam makan maka idam kenyang, namun jika tidak maka idam akan kelaparan. Penjabarannya dapat dilihat sebagai berikut :

Kondisi, jika :

Idam makan

Maka :

Idam akan kenyang

Jika tidak :

Idam akan kelaparan

Contoh diatas dapat ditulis dalam syntax python dengan menggunakan kondisi, pengkondisian dalam python dibagi menjadi 4, yaitu : IF, IF ELSE, ELIF, nested IF. Berikut merupakan pembahasannya.

**3.5.0.1 IF** IF adalah suatu struktur yang memiliki suatu perlakuan jika terjadi suatu kondisi. Akan tetapi, tidak terjadi sesuatu yang lain atau terjadi apa-apa ketika berada di dalam luar kondisi tersebut. IF hanya menjalankan satu kondisi dan menampilkan satu output. Contoh: kondisi dimana variabel a lebih besar dari variabel b, maka tampilkan hasil bahwa a lebih besar dari b.

```
1 #if statement
2 a = 330
3 b = 200
4 if a > a:
5     print("a lebih besar dari b")
```

**3.5.0.2 IF ELSE** IF ELSE digunakan apabila kondisi yang terjadi bernilai salah, maka lakukan else. Contoh: kondisi dimana variabel a lebih besar dari variabel b, maka jika b lebih besar dari a, tampilkan hasil b lebih besar dari a, jika salah maka tampilkan a lebih besar dari pada b

```
1 #else
2 a = 200
3 b = 33
4 if b > a:
5     print("b is greater than a")
6 else:
7     print("a is greater than b")
```

**3.5.0.3 ELIF** Kondisi ELIF merupakan suatu strktur logika majemuk yang memiliki banyak pilihan aksi terhadap berbagai kemungkinan kejadian yang terjadi. ELIF digunakan apabila kondisi pertama tidak benar maka lakukan kondisi lain (alternatif). Contoh: kondisi dimana variabel a sama dengan variabel b, maka jika b lebih besar dari a, tampilkan hasil b lebih besar dari a, namun jika a dan b bernilai sama, maka tampilkan a sama dengan b

```
1 #elif
2 a = 33
3 b = 33
4 if b > a:
5     print("b lebih besar dari a")
6 elif a == b:
7     print("a sama dengan b")
```

**3.5.0.4 Nested IF** Nested if merupakan if didalam if (if bersarang), terdapat dua if didalam satu kondisi. Contoh: variabel x sama dengan 41, kondisi pertama yaitu jika x besar dari 10 maka tampilkan lebih besar dari 10, kondisi kedua yaitu jika x besar dari 20, maka tampilkan lebih besar dari 20, jika salah maka tampilkan tidak melebihi 20.

```

1 #nested if
2 x = 41
3
4 if x > 10:
5     print("lebih besar dari 10,")
6     if x > 20:
7         print("lebih besar dari 20!")
8     else:
9         print("tidak melebihi 20.")

```

### 3.6 Error

1. NameError, terjadi apabila kode mengeksekusi nama yang tidak terdefinisikan. Contoh:

```

1 nama = "Dinda Majesty"
2 print>Nama)

```

Maka akan menghasilkan output NameError: name Nama is not dened. error ini dapat diatasi dengan mengubah variabel yang di print sesuai dengan variabel yang didefinisikan, karena penulisan pada python bersifat case-sensitive

2. SyntaxError, terjadi apabila kode python mengalami kesalahan saat penulisan. Contoh: menuliskan variabel yang didahului angka (1nama = Dinda Majesty) maka akan muncul error SyntaxError: invalid syntax. error ini dapat diatasi dengan memperhatikan tata cara penulisan kode pada bahasa pemrograman python.
3. Logic error merupakan kesalahan yang terjadi karena kesalahan pembacaan data pada command perintah seperti data tidak terbaca atau tidak ada, dan tidak sesuai dengan aturannya. Contoh kesalahan tipe data yaitu

```

1 a= 4
2 b=6
3
4 print(a+b)

```

4. TypeError, terjadi apabila kode melakukan operasi atau fungsi terhadap tipe data yang tidak sesuai. Contoh: melakukan penjumlahan terhadap tipe data string dan integer. error ini dapat diatasi dengan mengubah tipe data string menjadi integer.

```

1 a = "10"
2 b = 5
3
4 print(a + b)

```

Maka akan menghasilkan output error TypeError: can only concatenate str (not int) to str

5. IndentationError, terjadi apabila kode perulangan atau pengkondisian tidak menjorok kedalam (tidak menggunakan indentasi), error ini dapat diatasi dengan menambahkan tab atau spasi. Contoh



```
1 a = 200
2 b = 330
3
4 if b > a:
5     print("b lebih besar dari a")
```

Maka akan menghasilkan output error `IndentationError: expected an indented block`

### 3.7 Try Except

Try Except merupakan salah satu bentuk penanganan error di dalam bahasa pemrograman python, perintah try except ini memiliki fungsi untuk menangkap sebuah error dan tetap menjalankan program kita, sehingga program yang sedang dijalankan akan mengeksekusi program hingga akhir. Contohnya terdapat pada listing berikut

```
1 a="1"
2 b=2
3
4 try:
5     a+b
6 except:
7     print("Error , kedua tipe data berbeda")
```

## BAB 4

---

# FUNGSI DAN KELAS

---

### 4.1 Teori

#### 4.1.1 Fungsi

Fungsi adalah sebuah blok kode yang memiliki nama fungsi dan kode program didalamnya jika dijalankan maka fungsi itu akan mengembalikan nilai. Fungsi dapat dipanggil berkali-kali sesuai dengan nama fungsi yang telah didefinisikan. Fungsi memiliki nilai kembalian (return). Contoh fungsi

```
1 def nambahinAngka(angka1, angka2):  
2     hasil = angka1 + angka2  
3     return hasil
```

Apabila kita dapat memberikan nilai ke `angka1` dan `angka2`, dan apa bila sudah diberi nilai dan program sudah dijalankan, maka program pun akan mengembalikan nilai berupa hasil dari penjumlahan `angka 1` dan `angka 2`.

## 4.2 Package

Package merupakan sekumpulan modul yang dikemas oleh programmer dengan tujuan agar mempermudah dalam pembuatan kode program. Kita dapat membuat sebuah kode program atau fungsi didalamnya dan dapat secara mudah menggunakan kode program itu dengan cara memanggilnya pada kode program lainnya atau import package. Contoh nya adalah sebagai berikut

```
1 def my_biodata(nama, umur):
2     bio = "nama saya " + nama + " umur saya " + umur
3     return bio
4
5 def my_study(kampus, prodi):
6     study = "saya kuliah di " + kampus + " program studi " + prodi
7     return study
```

Kode diatas merupakan isi dari le fungsi.py, sedangkan saya ingin menjalankan program fungsi.py pada main.py sehingga kode program pada le main.py akan dituliskan seperti berikut:

```
1 import fungsi
2 nama = "Dinda Majesty"
3 umur = "19 Tahun"
4 biodata = my_biodata(nama, umur)
5 print(biodata)
6
7 kampus = "Politeknik Pos Indonesia"
8 prodi = "D4-Teknik Informatika"
9 kuliah = my_study(kampus, prodi)
10 print(kuliah)
```

Kode program pada le main.py akan mengimport kode program yang ada pada le fungsi.py, sehingga dengan adanya fungsi dan package kita dapat dengan mudah melakukan pemanggilan fungsi yang telah kita deskripsikan sebelumnya, walaupun berada pada le python yang berbeda.

## 4.3 Class, Object, Atribute, and Method

Class atau Kelas merupakan sebuah blueprint/kerangka dari objek yang berisi fungsi dan dibuat untuk mendefenisikan objek dengan atribut yang sesuai dengan kelas yang telah dibuat yang nantinya akan diinisiasikan. Objek adalah sebuah wujud yang dapat kita lakukan perintah sesuai dengan methodnya, Sebuah kelas harus memiliki objek yang nantinya akan di kodekan sesuai dengan fungsi yang telah dibuat pada kelas, tanpa adanya objek sebuah kelas tidak akan bisa menjalankan fungsi-fungsi didalamnya. Atribut berisi variabel yang memiliki tipe data dan dapat kita berikan pada objek, atribut ada 2 yaitu kelas atribut dan instansi atribut, perbedaannya hanya di letak, kalau kelas atribut ada di bawah kelas, dan instansi atribut ada didalam fungsi, atribut itu sebuah variabel yang dimiliki oleh parentnya seperti fungsi atau class. .Method merupakan kode program yang berisi tindakan atau perintah untuk menjalankan objek.

```
1 class Fungsi(object):  
2  
3 def Nama(self, namakamu):  
4     self.kamu = namakamu
```

## 4.4 Pemanggilan Class

Pemanggilan library kelas dapat dilakukan dengan cara import dan membuat objek dari kelas tersebut. Contohnya, kita memiliki le python yang diberi nama ngitung dan didalamnya terdapat class Ngitung yang memiliki banyak fungsi didalamnya. Untuk melakukan pemanggilan class maka kita bisa mengetikkan kode seperti berikut.

```
1 import Fungsi
```

## 4.5 Pemakaian Package Fungsi Apabila File Didalam Folder

Pemakaian Package fungsi apabila le terdapat didalam sebuah folder maka kita bisa menggunakan from folder import le dan from le import fungsi. Contohnya, kita memiliki folder src yang didalamnya terdapat le fungsi.py dan didalam fungsi.py terdapat fungsi Berhitung, untuk mengimportkan fungsi maka kita dapat mengetikkan kode seperti berikut.

```
1 from src import fungsi  
2 from fungsi import Berhitung
```

## 4.6 Pemakaian Package Kelas Apabila File didalam Folder

Pemakaian package kelas apabila le terdapat didalam sebuah folder maka kita bisa menggunakan from folder import le dan from le import kelas. Contohnya, kita memiliki folder src yang didalamnya terdapat le fungsi.py dan didalam fungsi.py terdapat kelas Ngitung, maka untuk melakukan import kelas kita dapat mengetikkan kode sebagai berikut.

```
1 from src import fungsi  
2 Kelas = fungsi.Nama(namakamu)
```



## DAFTAR PUSTAKA

---

