

TUTORIAL MEMBUAT PROTOTIPE PREDIKSI KETINGGIAN AIR (PKA) UNTUK PENDETEKSI BANJIR PERINGATAN DINI

TUTORIAL MEMBUAT PROTOTIPE PREDIKSI KETINGGIAN AIR (PKA) UNTUK PENDETEKSI BANJIR PERINGATAN DINI

**Mohamad Nurkamal Fauzan, S.T., M.T.
Lalita Chandiany Adiputri
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 Tunnisah
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. Sariasisih 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

MOHAMAD NURKAMAL FAUZAN, S.T., M.T., LALITA CHANDIANY ADIPUTRI,
Informatics Research Center., Politeknik Pos Indonesia, Bandung, Indonesia

CONTENTS IN BRIEF

1 Pembuatan Prototipe	1
2 Pembuatan <i>Prototype</i>	3
3 Implementasi Coding Komponen Prototipe	79

DAFTAR ISI

Daftar Gambar	xi
Daftar Tabel	xvii
Foreword	xxi
Kata Pengantar	xxiii
Acknowledgments	xxv
Acronyms	xxvii
Glossary	xxix
List of Symbols	xxxi
Introduction	xxxiii
<i>Mohamad Nurkamal Fauzan, S.T., M.T., Lalita Chandiany adiputri.</i>	
1 Pembuatan Prototipe	1
2 Pembuatan Prototype	3
2.1 Sejarah Prototype	3
2.2 Tujuan Prototype	4
	ix

2.3	Bahasa Pemograman C	4
2.3.1	Sejarah Bahasa C	4
2.3.2	Pengertian Bahasa Pemograman C	4
2.3.3	Kelebihan dan Kekurangan Bahasa C	5
2.4	Penggunaan Bahasa C menggunakan Arduino IDE	5
2.4.1	Arduino IDE	5
2.4.2	Bagian-Bagian Arduino IDE	6
2.4.3	<i>Sketch</i> Arduino	8
2.5	Komponen yang Digunakan Untuk Pembuatan <i>Prototype</i> PKA	14
2.6	Pembuatan <i>Prototype</i>	45
2.7	Merakit <i>Kerangka Prototype</i>	50
2.8	Perakitan dan Memprogram Komponen <i>Prototype</i>	59
2.8.1	Perakitan Komponen	59
3	Implementasi Coding Komponen Prototipe	79
3.1	Penjelasan Implementasi dan Coding	79
3.1.1	Implementasi	79
3.1.2	<i>Coding</i>	80
3.2	Implementasi Coding Prototipe	80
3.2.1	Cara <i>Install</i> Arduino IDE dan Penggunaanya	81
3.2.2	Contoh Pemograman Arduino	83
3.2.3	Cara <i>Import Library</i>	85
3.3	Memprogram Komponen <i>Prorotype PKA</i>	89
3.3.1	Memprogram Sensor Ultasonic	89
3.3.2	Memprogram Led	91
3.3.3	Memprogram Buzzer	92
3.3.4	Memprogram Sensor, Led dan Buzzer	93
3.3.5	Memprogram Komponen dan Antares	95
	Daftar Pustaka	103

DAFTAR GAMBAR

2.1	Arduino IDE	6
2.2	Tampilan Arduino IDE	7
2.3	Contoh Void Setup	9
2.4	Gambar Contoh Fungsi Setup dan Loop	10
2.5	NodeMCU	15
2.6	Spesifikasi NodeMCU	16
2.7	Skema Posisi Pin NodeMCU	17
2.8	Sensor HC-SR04	18
2.9	Prinsip Pengukuran Jarak Sensor HC SR04	19
2.10	Rumus Perhitungan Jarak	20
2.11	Konfigurasi Sensor Ultrasonic HC-SR04	20
2.12	Gambar Sensor Ultrasonic US 100	22
2.13	Prinsip Kerja Sensor Ultrasonic	23

2.14	Rumus perhitungan Jarak Sensor Ultrasonic	23
2.15	Pin Konfigurasi	24
2.16	Skema diagram sensor ultrasonic US 100	25
2.17	Buzzer	26
2.18	Konfigurasi Pin Buzzer	27
2.19	Keterangan Konfigurasi Pin Buzzer	28
2.20	Diagram model 2D Buzzer	29
2.21	Led	30
2.22	Simbol Led	30
2.23	Cara Kerja Led	31
2.24	Cara Melihat Polaritas	32
2.25	Logo Antares	33
2.26	Tampilan Antares Antares	33
2.27	Reistor	34
2.28	Macam-Macam Resistor Sesuai Warna	35
2.29	Resistor Yang Digunakan	36
2.30	Logo Telegram	37
2.31	Bentuk Akrilik	38
2.32	Kabel Jumper	39
2.33	Kabel Jumper Famale	40
2.34	Kabel Jumper Male	41
2.35	Kabel Jumper Male-Famale	42
2.36	Breadboard Mini	43
2.37	Breadboard Medium	44
2.38	Breadboard Large	44
2.39	Contoh Akrilik yang digunakan	45
2.40	Contoh Lem Korea	46
2.41	Contoh Lem Akrilik	47

2.42	Contoh Sticky Note	48
2.43	Contoh Double TApe	49
2.44	Contoh Styrofoam Papan	50
2.45	SkeecthUp Make	51
2.46	Tampilan SkeecthUp Make	51
2.47	DesignWorkshop Lite	52
2.48	Tampilan DesignWorkshop Lite	52
2.49	Blender	53
2.50	Tampilan Blender	54
2.51	Free CAD	54
2.52	Tampilan Free CAD	55
2.53	TinkerCAD	55
2.54	Tampilan TinkerCAD	56
2.55	OpenSCAD	57
2.56	Tampilan OpenSCAD	57
2.57	Desain Prototipe	58
2.58	Kerangka Prototipe	58
2.59	Kerangka Prototipe Tempan Penyimpanan Komponen	59
2.60	Tampilan Frizting	60
2.61	Tampilan Frizting	61
2.62	Tampilan Frizting	61
2.63	Tampilan LiveWire	62
2.64	Tampilan LiveWire	62
2.65	Tampilan LiveWire	63
2.66	Tampilan <i>Electronics Workbench</i> (EWB)	63
2.67	Tampilan <i>Electronics Workbench</i> (EWB)	64
2.68	Tampilan <i>Electronics Workbench</i> (EWB)	64
2.69	Tampilan NI Multisim	65

2.70	Tampilan NI Multisim	65
2.71	Tampilan NI Multisim	66
2.72	Tampilan Autodesk Circuits	66
2.73	Tampilan Autodesk Circuits	67
2.74	Tampilan Autodesk Circuits	67
2.75	Tampilan Proteus ISIS	68
2.76	Tampilan Proteus ISIS	68
2.77	Tampilan Proteus ISIS	69
2.78	Tampilan Awal Fritzing	70
2.79	Ubah Ukuran Breadboard	70
2.80	Memasukan NodeMCU	71
2.81	Sambungkan Vin ke <i>Breadboard</i>	71
2.82	Mengubah Warna Wire	72
2.83	Sambungkan GND ke <i>Breadboard</i>	72
2.84	Menambahkan Komponen Sensor Ultrasonic	73
2.85	Menambahkan Komponen Led, Resistor dan Buzzer	73
2.86	Rangkaian Alat	74
2.87	Rangka Prototipe Depan	75
2.88	Rangka Prototipe Samping	76
2.89	Rangka Prototipe Komponen	76
2.90	Rangka Prototipe Level Ketinggian Air	77
3.1	Tampilan Jendela Instalasi	81
3.2	Jendela Dialog Tujuan Instalasi Dari Arduino IDE	82
3.3	Jendela Instalasi Arduino IDE	82
3.4	Tampilan Arduino IDE Blink	83
3.5	Tampilan Di Serial Monitor	85
3.6	Memilih Menu Manage Librarie	86
3.7	Tampilan Sketch Manager	87

3.8	Tampilan Library Yang Terinstall	87
3.9	Tampilan Library ADD ZIP	88
3.10	Pilih File ZIP	88
3.11	Tampilan Library Berhasil Ditambahkan	89
3.12	Tampilan serial monitor	91
3.13	Halaman Awal Antares	95
3.14	Tampilan Registrasi Antares	96
3.15	Verifikasi email	97
3.16	Login Antares	97
3.17	Tampilan Awal Antares	98
3.18	Halaman <i>Overview</i>	98
3.19	Halaman <i>Aplication</i>	99
3.20	Halaman <i>Widgets</i>	99
3.21	Halaman <i>Documentation</i>	100
3.22	Halaman <i>Documentation</i>	100
3.23	Halaman <i>Documentation Tutorial</i>	101
3.24	Halaman <i>DocumentationTutorial</i>	101

DAFTAR TABEL

Listings

src/void.c	10
src/boolean.c	11
src/char.c	11
src/unchar.c	11
src/byte.c	11
src/int.c	11
src/unint.c	11
src/long.c	12
src/unlong.c	12
src/short.c	12
src/float.c	12
src/double.c	12
src/variabel1.c	13
src/variabel2.c	13
src/hello.c	84
src/sensor.c	89
src/led.c	91
src/buzzer.c	92

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.

M. NURKAMAL .F, LALITA CHANDIANY. A

*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

B Number of Beats

INTRODUCTION

MOHAMAD NURKAMAL FAUZAN, S.T., M.T , LALITA CHANDIANY ADIPUTRI.

Informatics Research Center
Bandung, Jawa Barat, Indonesia

Indonesia sekrang 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 PROTOTIPE

BAB 2

PEMBUATAN *PROTOTYPE*

2.1 Sejarah *Prototype*

Pada tahun 1960-an Herbert Voelcker, 1970: Voelcker mengembangkan alat dasar matematika yang dengan jelas menggambarkan tiga aspek dimensi dan menghasilkan teori-teori awal teorialgoritma dan matematika untuk pemodelan solid.Pada tahun 1987, Carl Deckard, membentuk tim peneliti dari University of Texas. datang dengan ide yang revolusioner yang baik. *Prototype* adalah salah satu pendekatan dalam rekayasa perangkat lunak yang secara langsung mendemonstrasikan bagaimana sebuah perangkat lunak atau komponen-komponen perangkat lunak akan bekerja dalam lingkungannya sebelum tahapan konstruksi aktual dilakukan (Howard, 1997) .

Selain itu juga *PyPrototype* adalah tahapan yang ditujukan untuk mentransformasi sifat-sifat abstrak dari sebuah ide menjadi lebih berwujud. Tahapan ini tidak hanya berupa proses visualisasi ide tetapi juga proses pembangunan ide. Secara umum, Prototype memiliki dua kategori: low-fidelity dan high-fidelity. Proses prototyping yang digunakan di dalam Design Thinking adalah *low-fidelity* atau *Rapid Prototyping*. Proses ini menekankan kepada pembuatan proses pembuatan yang cepat, mudah, murah dan basic .

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 Pemrograman 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 diciptakan oleh Brian W. Kernighan dan Dennis M. Ritchie lalu dikembangkan 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 digunakan 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.

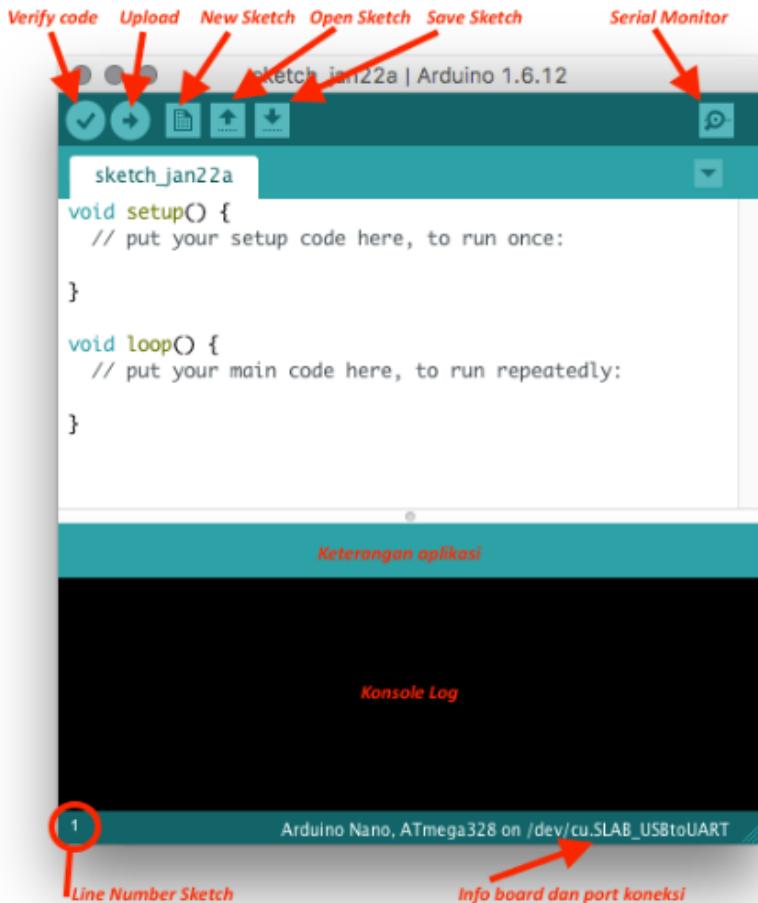
Dalam Software Arduino IDE ini dapat memprogram kode tidak hanya *microcontroller* arduino saja, tetapi pada software ini dapat digunakan untuk memprogram *microcontroller* yang lain salah satu contohnya yaitu Nodemcu.

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, biasakan 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 mikrokontroller.
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 konstantata) 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 variable*, *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 berturu-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

```

int ledPin = 13;

void setup()
{
  pinMode(ledPin, OUTPUT);
}

void loop()
{
  digitalWrite(ledPin, LOW);
}

```

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 nilanya 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.

```
| 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.

```
| 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.

```
| 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).

```
| 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).

```
| double num = 45.253 ;
```

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 :

- 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 pendekripsi banjir peringatan dini yaitu sebagai berikut :

1. NodeMCU

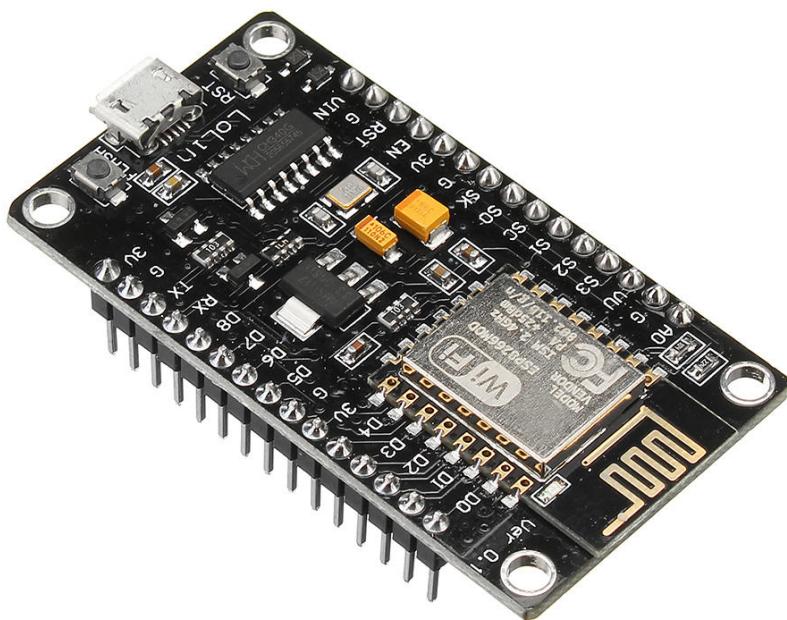
NodeMCU adalah ESP8266 (khususnya seri ESP-12, termasuk ESP-12E) maka fitur fitur yang dimiliki NodeMCU akan kurang lebih sama ESP-12 (juga ESP-12E untuk NodeMCU v.2 dan v.3) kecuali NodeMCU telah dibungkus oleh API sendiri yang dibangun berdasarkan bahasa pemrograman eLua, yang kurang lebih cukup mirip 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 dilengkapi 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 digunakan adalah firmware NodeMCU.



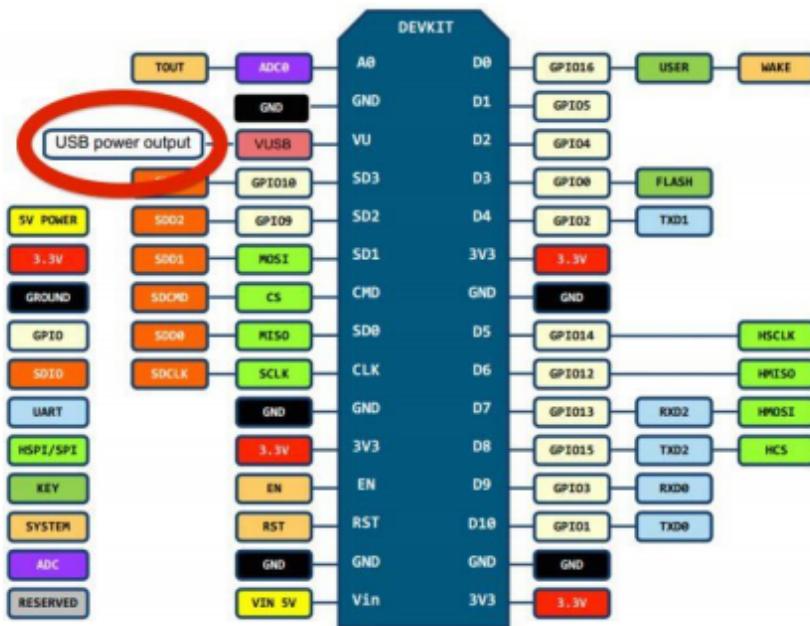
Gambar 2.5 NodeMCU

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

2. Spesifikasi *Datasheet* NodeMCU

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

Gambar 2.6 Spesifikasi NodeMCU



Gambar 2.7 Skema Posisi Pin NodeMCU

3. Protokol Hypertext Transfer Protocol (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 dispesifikasi 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.

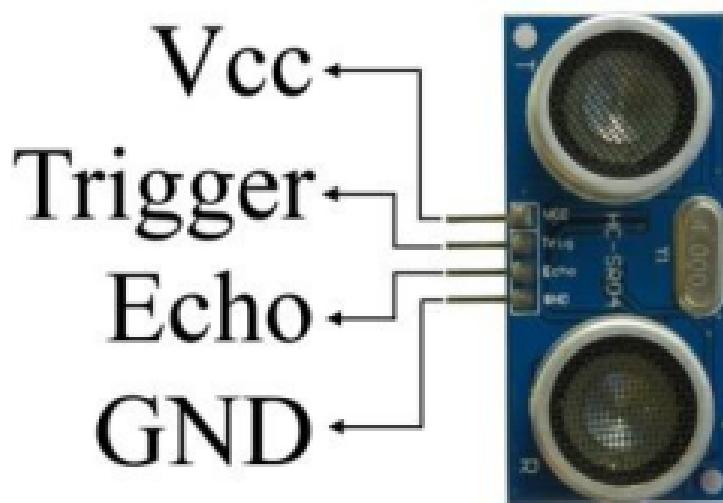
Pada protokol HTTP terdapat 3 jenis hubungan dengan perantara proxy, gateway, dan tunnel. Proxy bertindak sebagai agent penerus, menerima request dalam bentuk Uniform Resource Identifier (URI) absolut, mengubah format request dan mengirimkan request ke server yang ditunjukan oleh URI. Gateway bertindak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

server yang dilayaniinya. 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

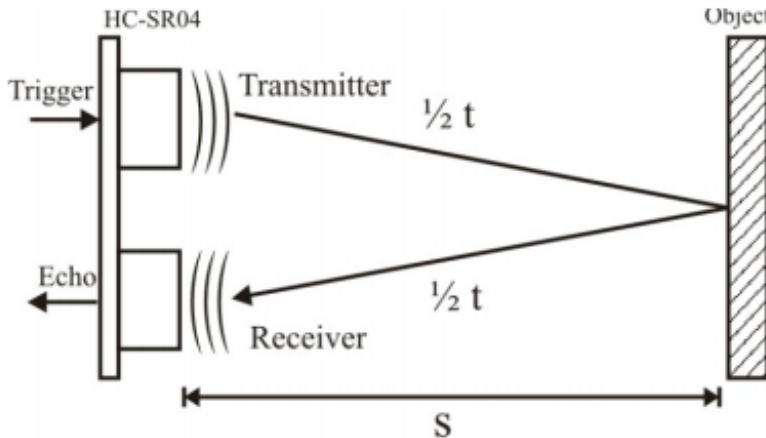
Sensor jarak ultrasonic HC-SR04 adalah sensor 40 KHz. HC-SR04 merupakan sensor ultrasonik yang dapat digunakan untuk mengukur jarak antara penghalang dan sensor. Konfigurasi pin dan tampilan sensor HC-SR04 diperlihatkan pada gambar dibawah :



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 m/s, maka jarak antara sensor dengan objek dapat dihitung dengan menggunakan Persamaan berikut :

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

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 proporsional
- (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 berhubung 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 :



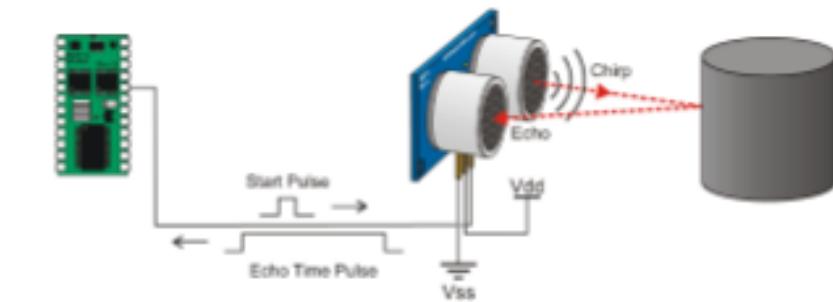
Gambar 2.12 Gambar Sensor Ultrasonic US 100

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

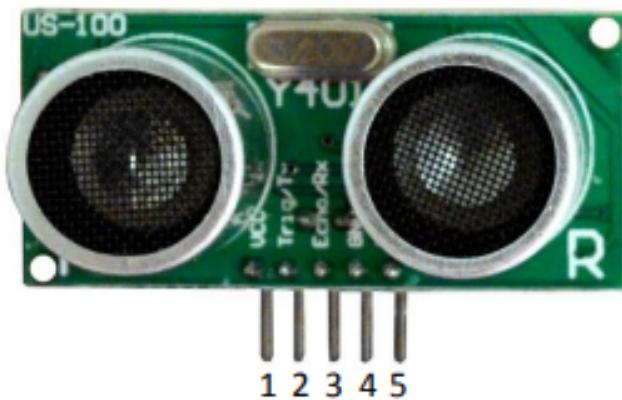
Pemancar ultrasonik mentransmisikan gelombang ultrasonik, gelombang ini bergerak di udara dan ketika ia keberatan dengan bahan apa pun itu dipantulkan kembali ke sensor gelombang. Untuk menghitung jarak menggunakan rumus di atas, harus mengetahui kecepatan dan waktu. Karena menggunakan gelombang Ultrasonik, 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

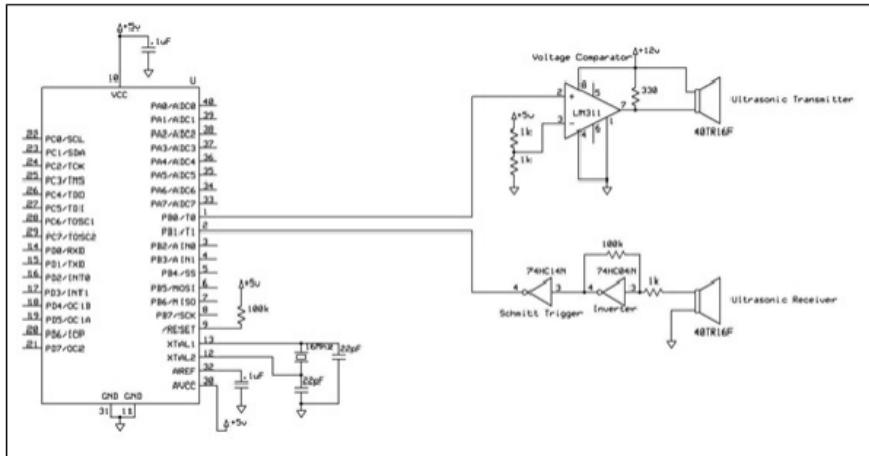
Adapun pin konfigurasi dari sensor ultrasonic Us 100 yaitu :



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 mengocek 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*.

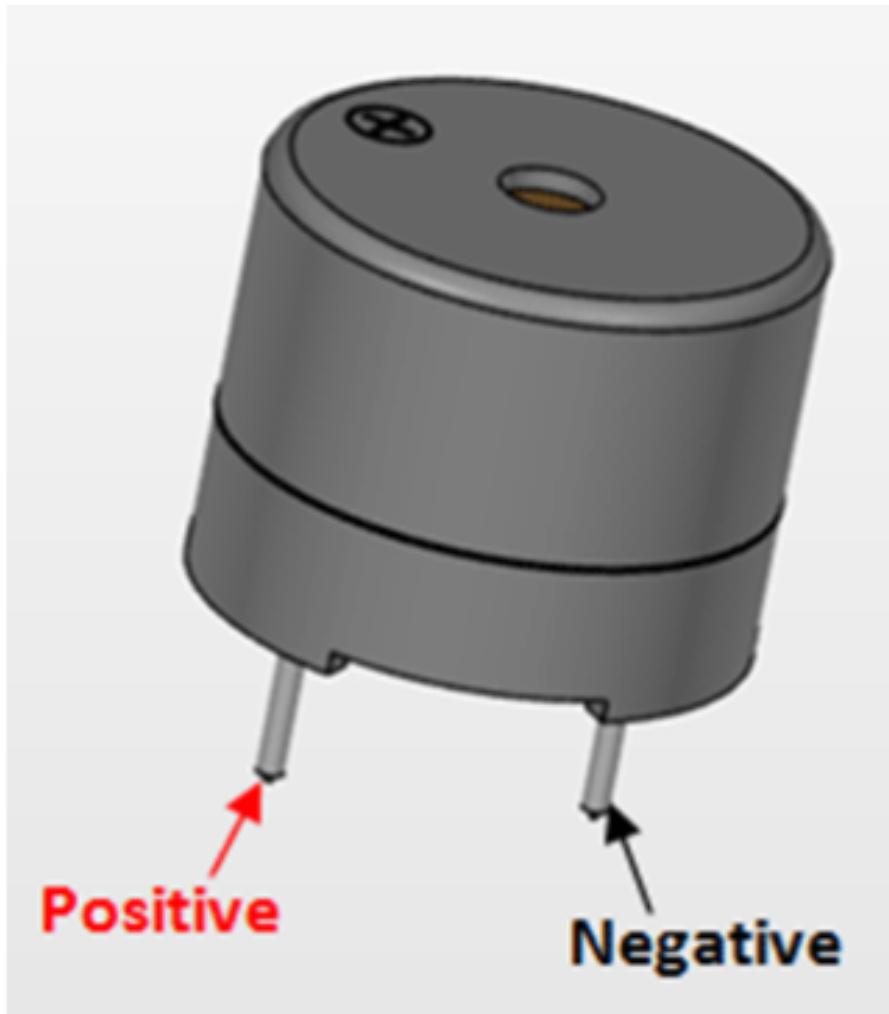


Gambar 2.17 Buzzer

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*: $\pm 30\text{mA}$
- (d) Tipe Suara: Bunyi Kontinu
- (e) Frekuensi resonansi: 2300 Hz
- (f) Kecil dan *package* rapih

13. Cara menggunakan Buzzer

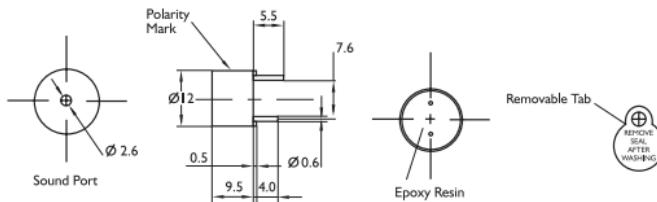
Buzzer adalah komponen kecil namun efisien untuk menambahkan fitur suara ke proyek / sistem. Ini adalah struktur 2-pin yang sangat kecil dan kompak sehingga dapat dengan mudah digunakan pada Breadboard dan bahkan pada PCB yang menjadikannya komponen yang banyak digunakan dalam sebagian besar aplikasi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



Dimensions : Millimetres
Tolerance : $\pm 0.5\text{mm}$

Part Number Table

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

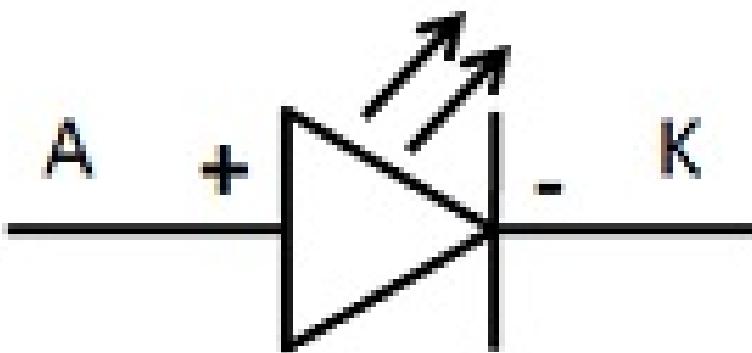
Gambar 2.20 Diagram model 2D Buzzer

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



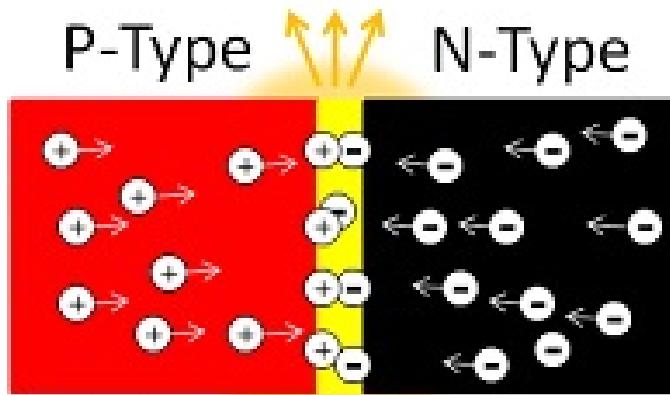
Gambar 2.22 Simbol Led

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.

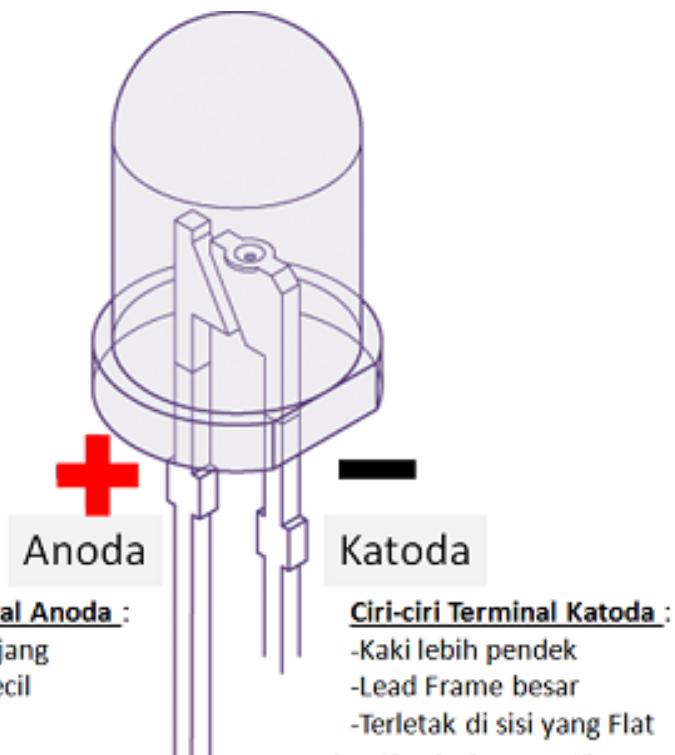
LED terdiri dari sebuah chip semikonduktor yang di doping sehingga menciptakan junction P dan N. Yang dimaksud dengan proses doping dalam semikonduktor adalah proses untuk menambahkan ketidakmurnian (impurity) pada semikonduktor yang murni sehingga menghasilkan karakteristik kelistrikan yang diinginkan.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



Gambar 2.25 Logo Antares

Antares merupakan sebuah *platform IoT* lokal milik Telkom yang telah mendapat pengakuan dari dunia 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

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 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 melaluiinya. 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

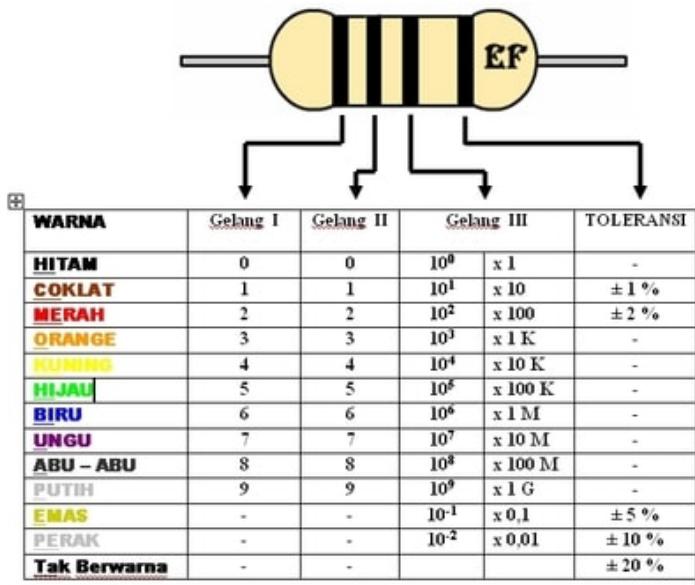
Satuan Resistor adalah Ohm (simbol: Ω) yang merupakan satuan SI untuk resistansi listrik. Dalam sejarah, kata ohm itu diambil dari nama salah seorang fisikawan hebat asal German bernama George Simon Ohm. Beliau juga yang menceetuskan keberadaan hukum ohm yang masih berlaku hingga sekarang.

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



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 mengeluaran Rp.1000,-

22. Bot Telegram



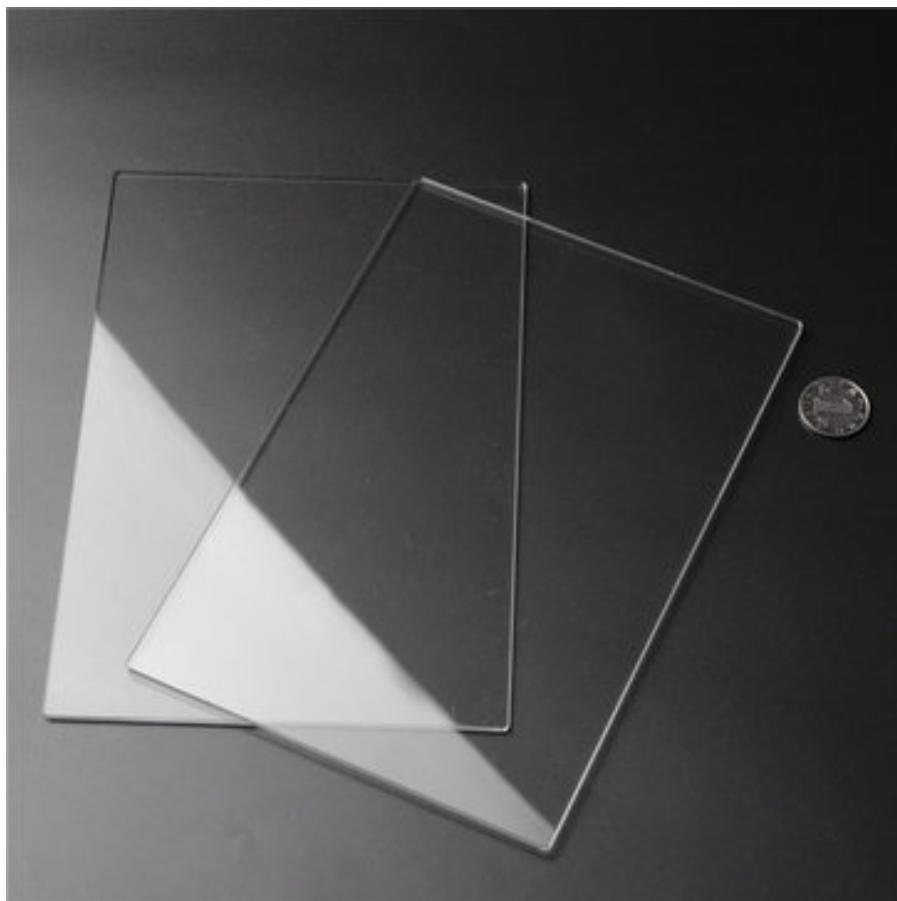
Gambar 2.30 Logo Telegram

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. Krilik adalah bagian dari plastik sintesis yang berasa dari asam akrilik, akrilik terbuat dari 100 MMA (metil metakrilat monomer), sudah paham kan temen-temen yuk kita bahas lebih dalam lagi mengenai akri-

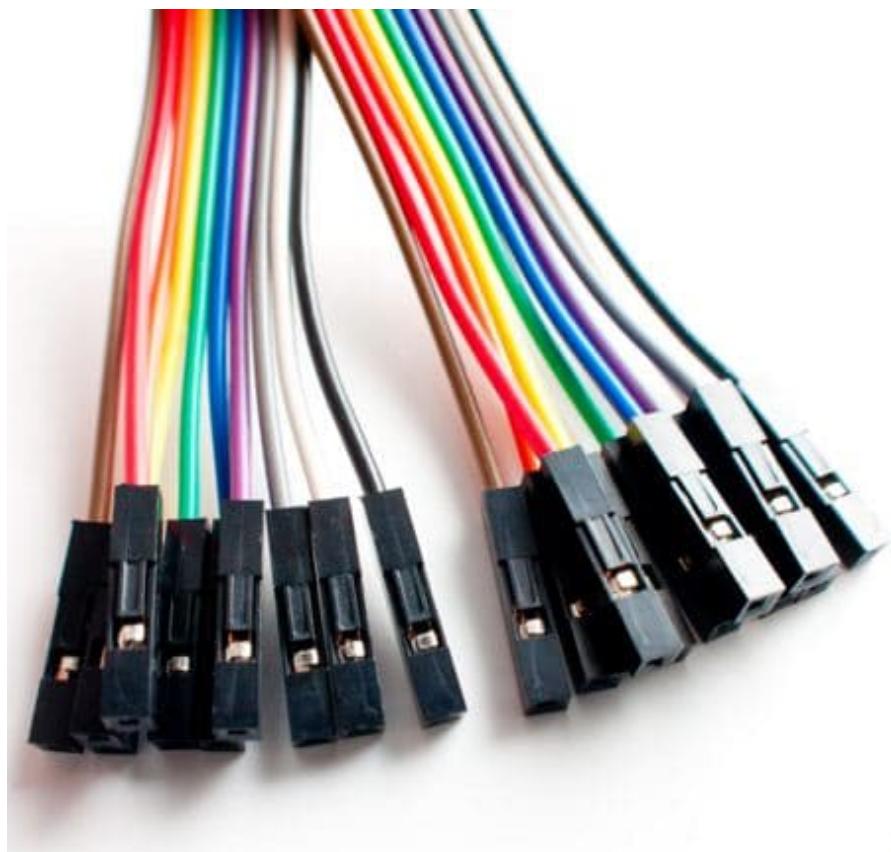
lik. Akrilik memiliki nilai estetika yang tinggi selain tidak mudah pecah dan tahan cuaca, akrilik juga tidak akan berkerut atau mengembang, Akrilik lebih kuat dari kaca, sehingga lebih tahan dan tidak pecah sehingga lebih aman.

24. Kabel Jumper

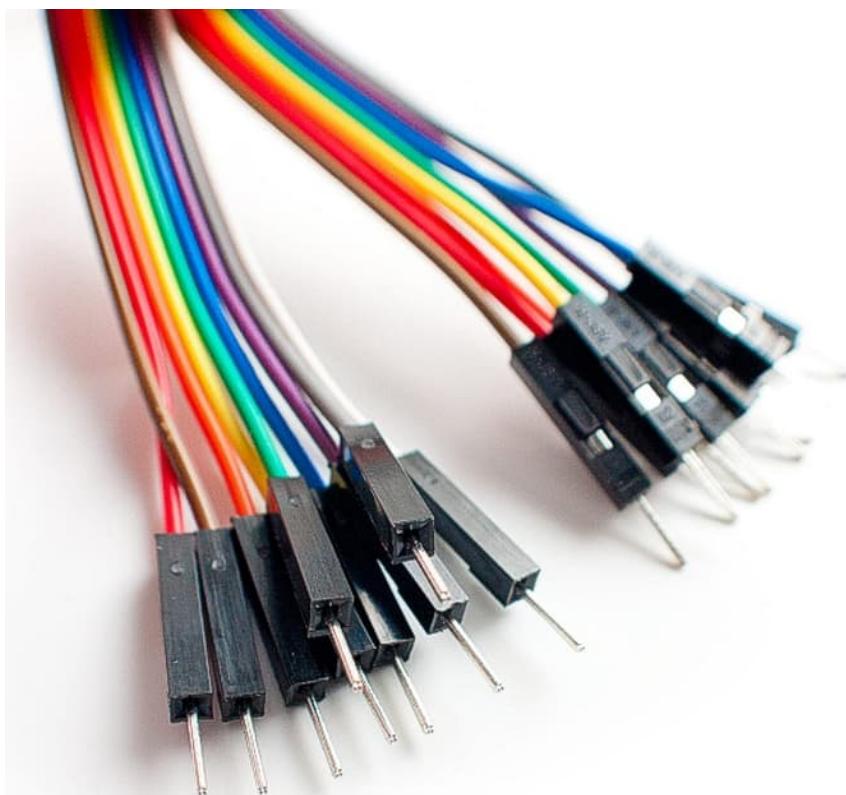


Gambar 2.32 Kabel Jumper

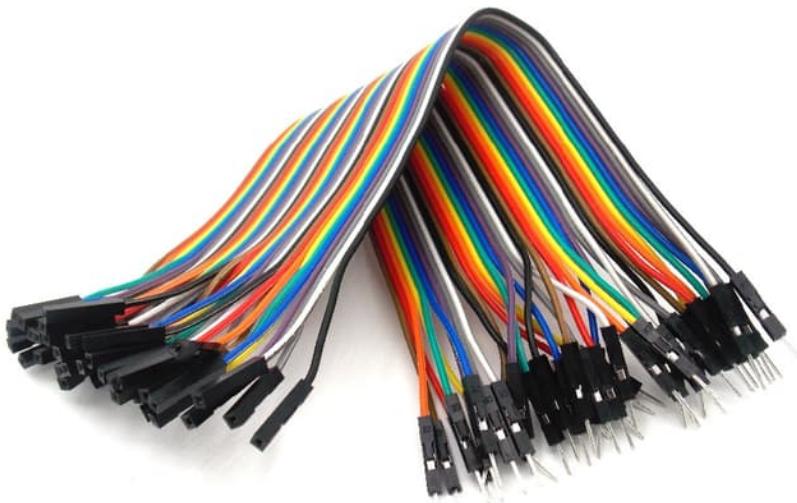
Kabel Jumper digunakan untuk menghubungkan komponen satu dengan komponen lainnya. Selain itu kabel jumper ini juga untuk menghubungkan komponen dengan *microcontroller*. Ada beberapa jenis kabel jumper yaitu sebagai berikut :



Gambar 2.33 Kabel Jumper Famale



Gambar 2.34 Kabel Jumper Male

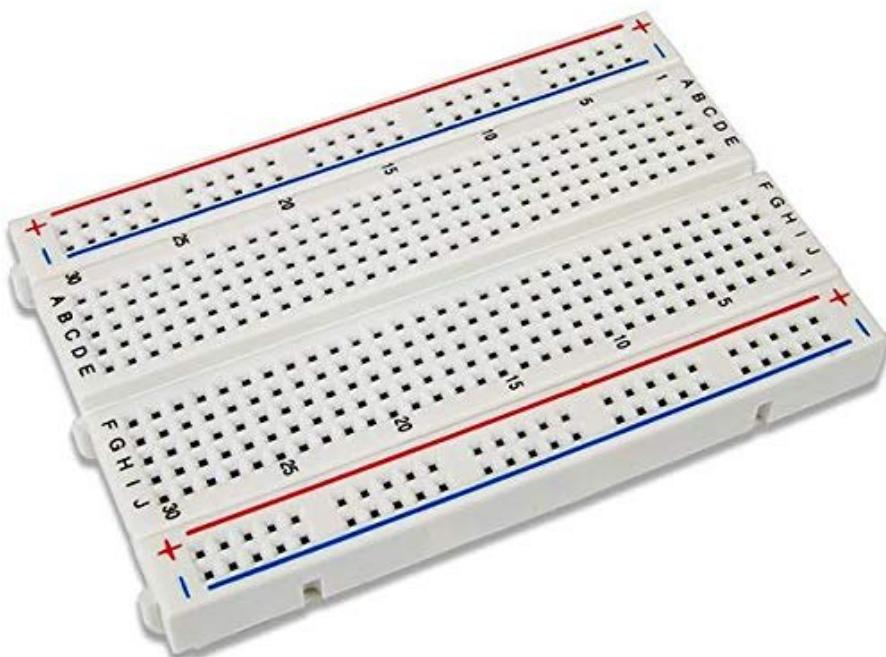


Gambar 2.35 Kabel Jumper Male-Female

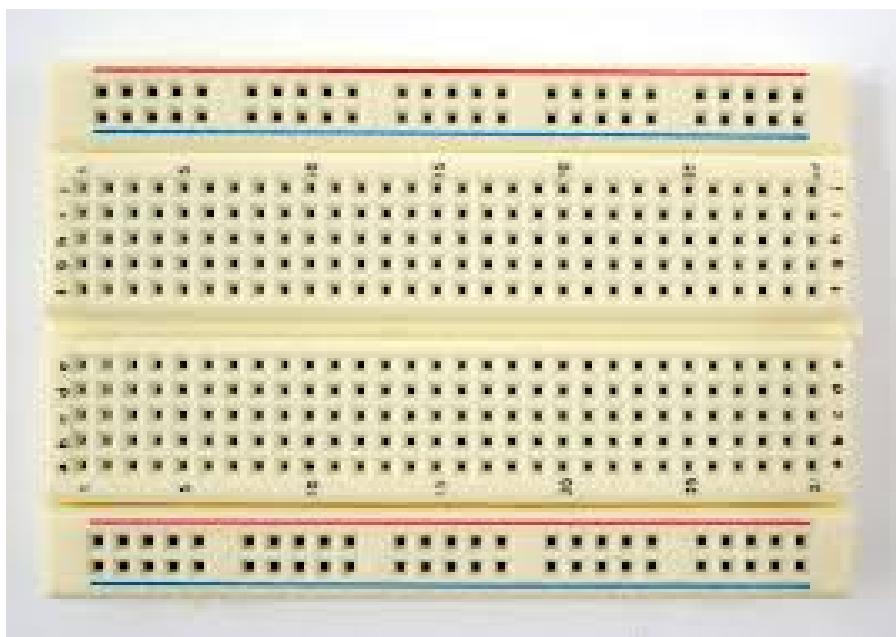
25. BreadBoard

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 diatasnya. Lubang-lubang pada breadboard diatur sedemikian rupa membentuk pola sesuai dengan pola jaringan koneksi di dalamnya.

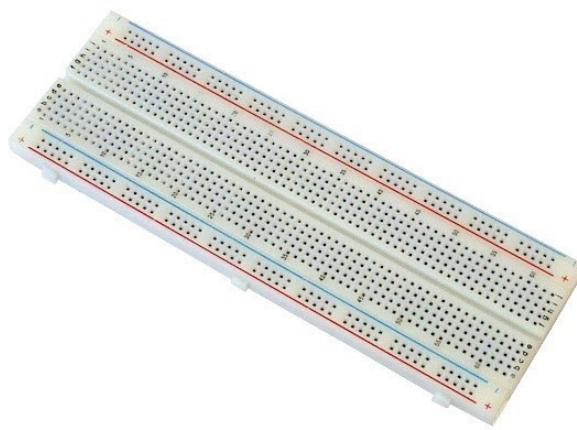
Breadboard yang tersedia di pasaran umumnya terbagi atas 3 ukuran: mini breadboard, medium breadboard atau large breadboard. Mini breadboard memiliki 170 titik koneksi (bisa juga lebih). Kemudian medium breadboard memiliki 400 titik koneksi. Dan large breadboard memiliki 830 titik koneksi.



Gambar 2.36 Breadboard Mini



Gambar 2.37 Breadboard Medium

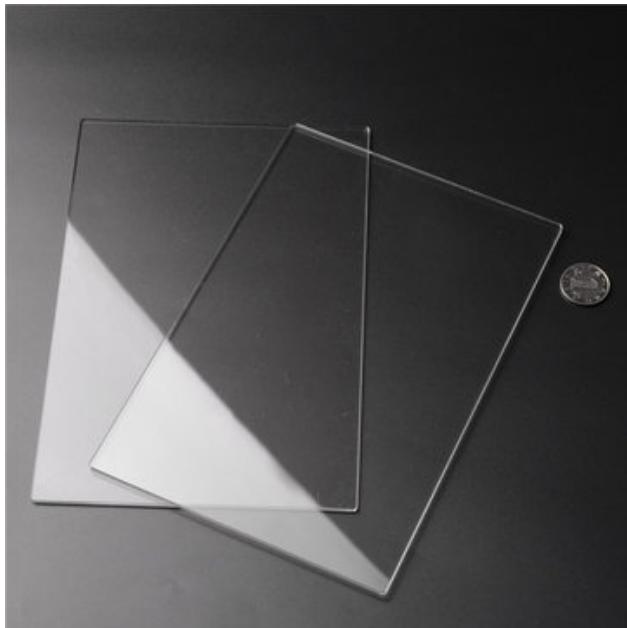


Gambar 2.38 Breadboard Large

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 menempalkan 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 berceran 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 karena 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 menepelkan 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 rangka prototipe. Sebelum kita membuat sebuah kerangka prototipe agar lebih memudahkan proesses ini kita harus membuat sebuah desain dari prototipe .

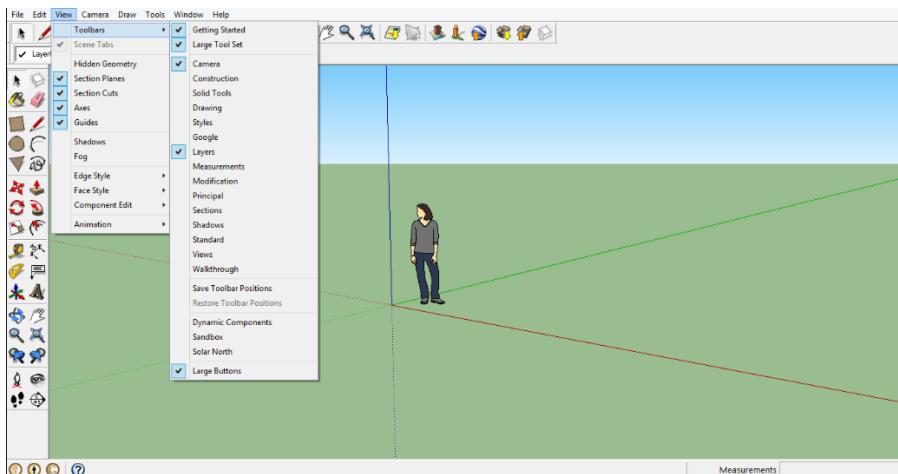
Untuk membuat sebuah desain dari rangka prototipe ini ada beberapa software untuk mendesain yaitu :

1. SketchUp Make



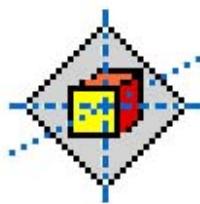
Gambar 2.45 SketchUp Make

SketchUp adalah sebuah perangkat lunak desain grafis yang dikembangkan oleh Trimble. Pendesain grafis ini dapat digunakan untuk membuat berbagai jenis model, dan model yang dibuat dapat diletakkan di Google Earth atau dipamerkan di 3D Warehouse. Tersedia dua versi SketchUp, yaitu SketchUp Make (gratis) dan SketchUp Pro (harga: USD 590.00).



Gambar 2.46 Tampilan SketchUp Make

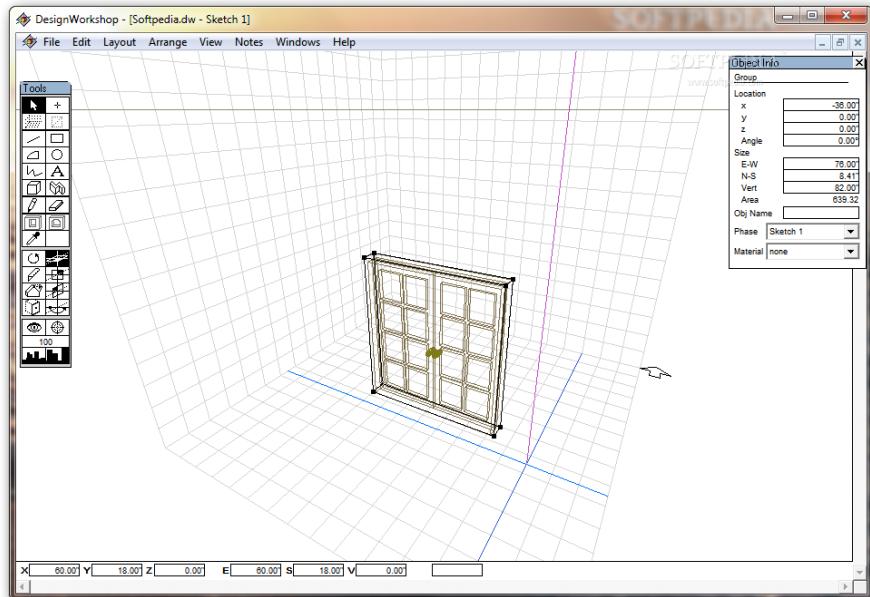
2. DesignWorkshop Lite



DesignWorkshop® 1.5 Lite

Gambar 2.47 DesignWorkshop Lite

DesignWorkshop Lite merupakan paket aplikasi untuk Desain Interior yang menyediakan semua yang kamu butuhkan untuk membangun model desain dan visualisasi arsitektur 3D, lanskap, atau desain tata ruang. Dengan DesignWorkshop Lite kamu dapat melihat model siap pakai dari berbagai ukuran seperti dari website free 3D building models, DXF, DesignWorkshop, atau file dengan format 3DMF.



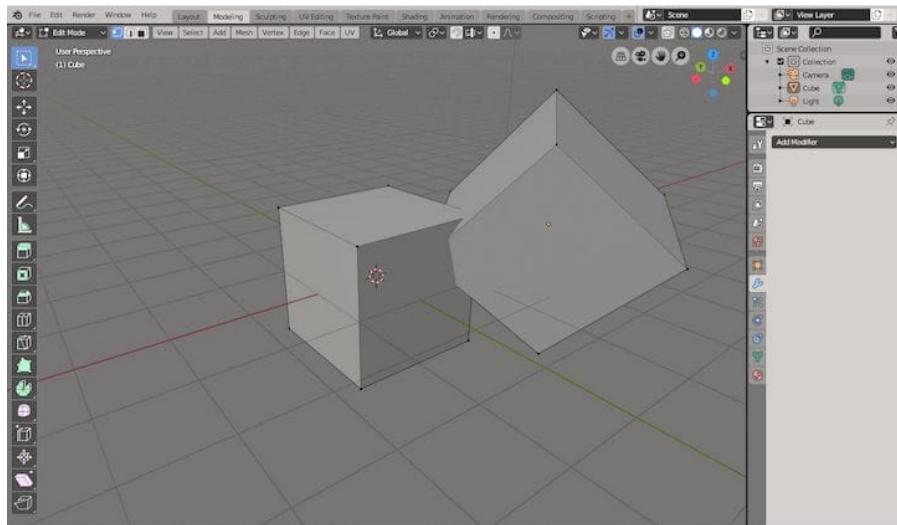
Gambar 2.48 Tampilan DesignWorkshop Lite

3. Blender



Gambar 2.49 Blender

Blender merupakan aplikasi pemodelan komplit dan komprehensif untuk menciptakan, mengubah dan mengedit model 3D dengan mudah. Transformasi model menjadi karakter yang menakjubkan menjadi lebih mudah dari sebelumnya dengan menggunakan Blender. Perangkat lunak ini digunakan untuk membuat film animasi, efek visual, model cetak 3D, aplikasi 3D interaktif dan permainan video. Blender memiliki beberapa fitur termasuk pemodelan 3D, penteksturan, penyunting gambar bitmap, penulangan, simulasi cairan dan asap, simulasi partikel, animasi, penyunting video, pemahat digital, dan rendering.



Gambar 2.50 Tampilan Blender

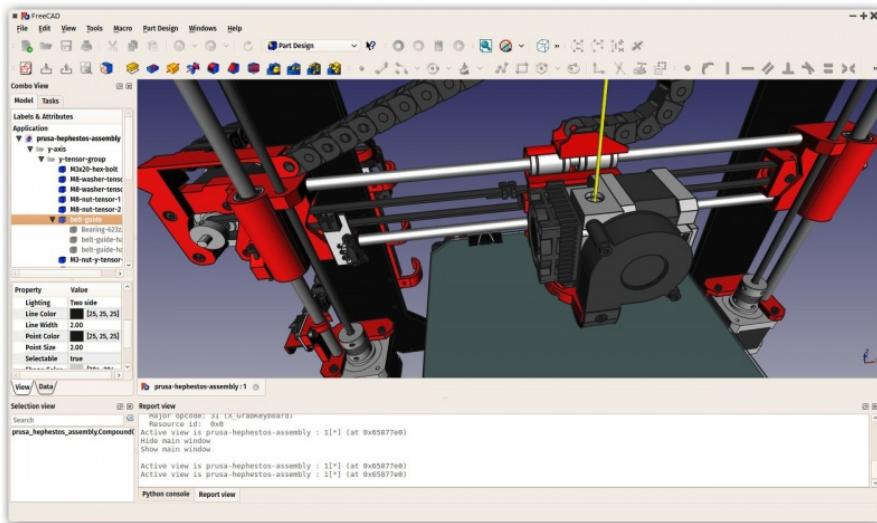
4. Free CAD



Gambar 2.51 Free CAD

FreeCAD adalah software 3D modeling serba-gratis dengan banyak fitur yang bisa disesuaikan, diubah, dan ditambah. FreeCAD bisa digunakan untuk membuat sketsa objek 2 dimensi, seperti bentuk Box, Sphere, Cone, Cylinder, dan Torus; lengkap dengan Tools modifikasi seperti Thickness, Cut, Fillet, Extrude,

Fuse, Boolean.Bisa juga digunakan untuk membuat simulasi gerakan robot, menciptakan model game 3D, model animasi, model arsitektur, model industrial, hingga menampilkan dan menyesuaikan G code.



Gambar 2.52 Tampilan Free CAD

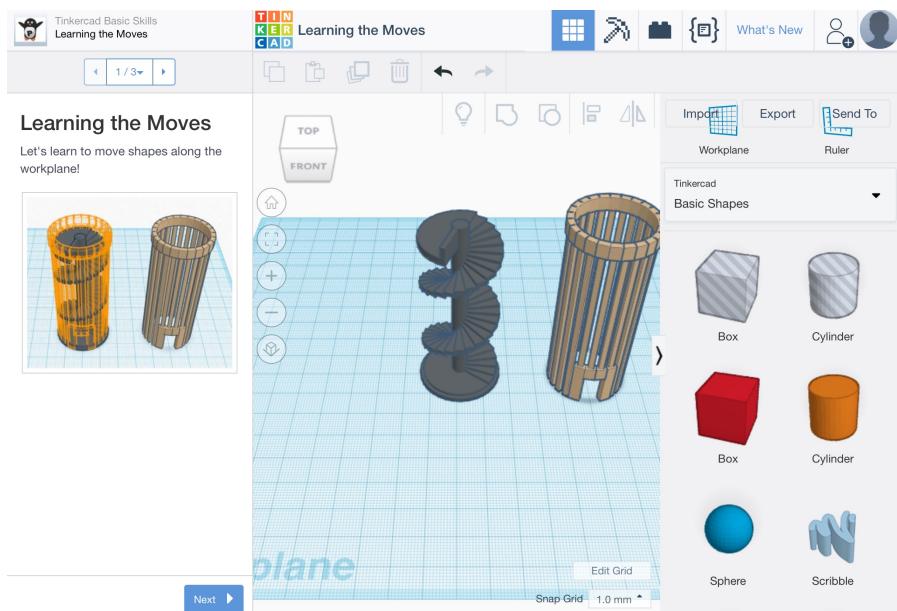
5. TinkerCAD



Gambar 2.53 TinkerCAD

TinkerCAD bisa digunakan untuk membuat 3D model dari dasar menggunakan basic modeling operations, tapi dengan tambahan fungsi untuk menggabungkan atau memisahkan bentuk model yang sedang dibuat.

Setelah menguasai teknik dasar 3D modeling seperti ini, maka pengguna TinkerCAD akan mampu menciptakan model 3D yang mengesankan seperti layaknya RaspberryPi.



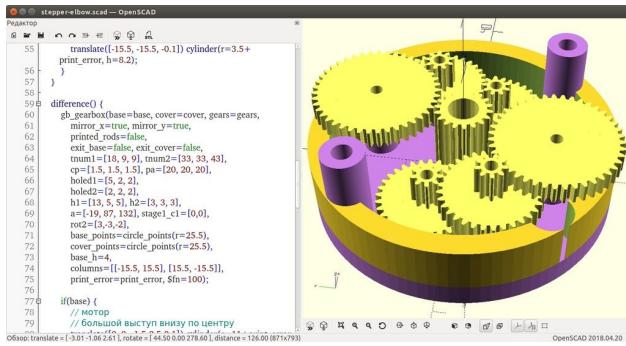
Gambar 2.54 Tampilan TinkerCAD



Gambar 2.55 OpenSCAD

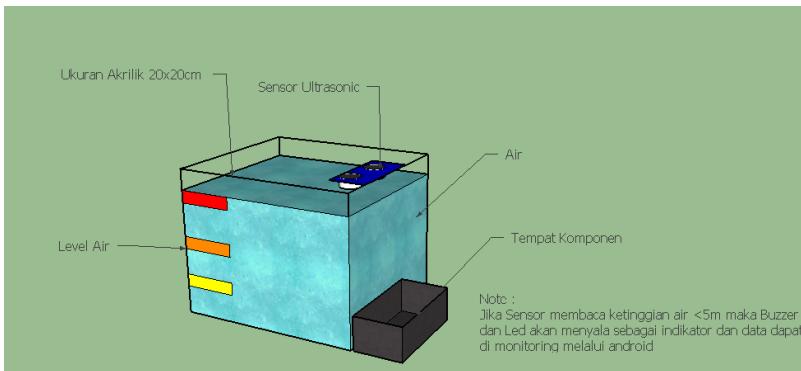
OpenSCAD adalah aplikasi software 3D modeling gratis, tapi bukanlah software yang menarik, dengan cara penggunaan yang berbeda dari software lain. Software OpenSCAD merupakan kompilator 3D berdasarkan bahasa deskripsi tekstual.

Dalam kalimat yang sederhana, model 3D dibuat menggunakan kode. Baik kode yang dirangkai dari dasar ataupun menggunakan kode yang telah digunakan untuk membuat proyek serupa.



Gambar 2.56 Tampilan OpenSCAD

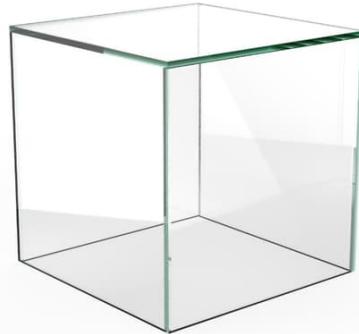
Setelah pembahasan diatas mengenai software pembuatan desain 3D, maka kita dapat memilih software yang akan digunakan . Untuk pembuatan desain prototipe ini menggunakan sketchup make. Mengapa menggunakan sketchup make? karena sketchup make mudah untuk dipahami dan softwarenya gratis. Berikut desain untuk rangka prototipenya :



Gambar 2.57 Desain Prototipe

Setalah kita membuat desain prototipe maka kita membuat rangkanya dengan sesuai desain yang telah kita buat sebelumnya. Cara membuat rangka prototipe 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 2.58 Kerangka Prototipe

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



Gambar 2.59 Kerangka Prototipe Tempan Penyimpanan Komponen

Setalah 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*

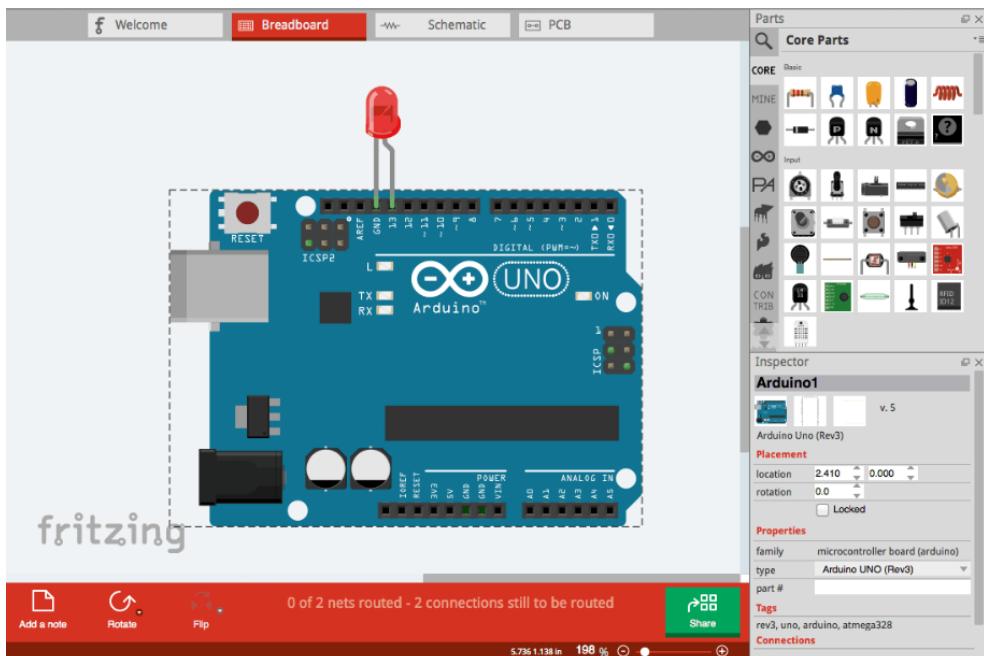
2.8.1 Perakitan Komponen

Komponen elektronika adalah komponen yang paling kompleks dan sensitif, tak jarang terjadi kerusakan apabila komponen yang digunakan tak seui semestinya,

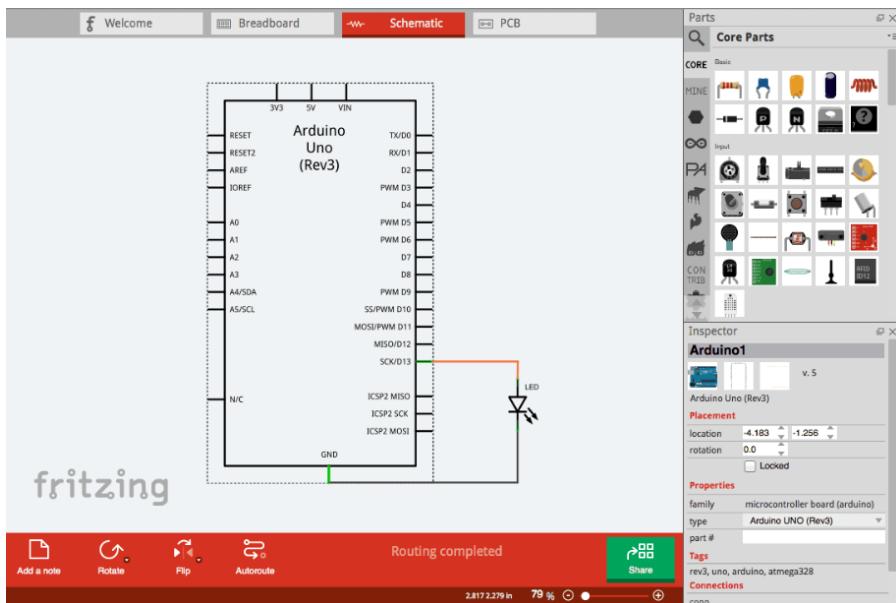
misalkan kelebihan daya, arus, short, dll. Selain itu tidak semua komponen ataupun perangkat elektronika dapat dijumpai dan dibeli dengan mudah, salah satu contohnya adalah osiloskop, spektrum analizer, function generator, dll.

Maka dari itu software simulasi dapat membantu proses merancang rangkaian elektronika sebelum dipatenkan langsung pada komponen. Bukan hanya itu, kita pun 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 :

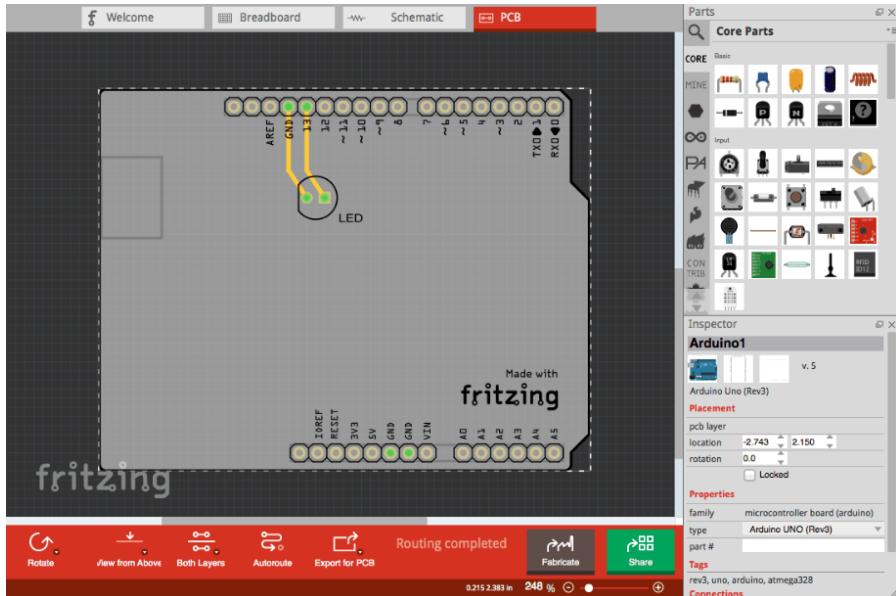
1. Fritzing merupakan aplikasi berbasis free software yang dapat kita unduh secara gratis. Aplikasi ini berfungsi untuk membuat sebuah skema rangkaian elektronika secara nyata (real). Karena komponen yang berada pada rangkaian tersebut digambarkan dengan sangat mirip dengan komponen aslinya. Aplikasi 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 2.60 Tampilan Fritzing

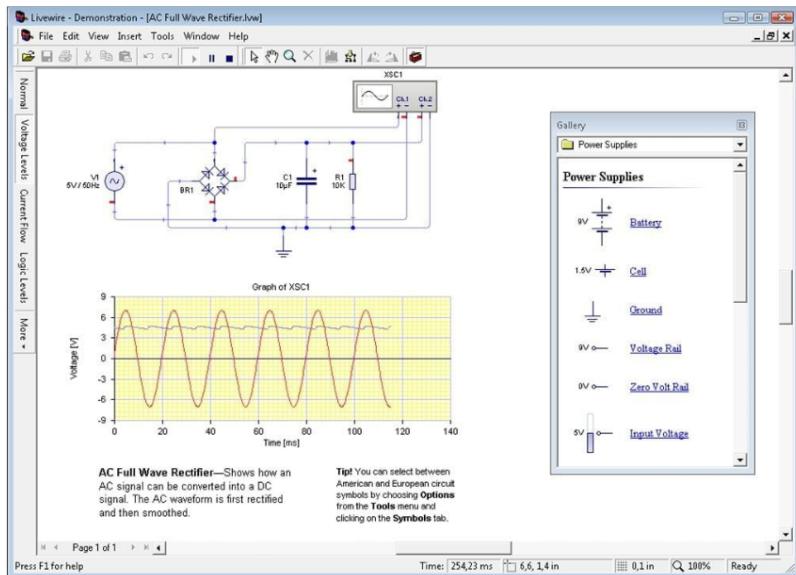


Gambar 2.61 Tampilan Fritzing

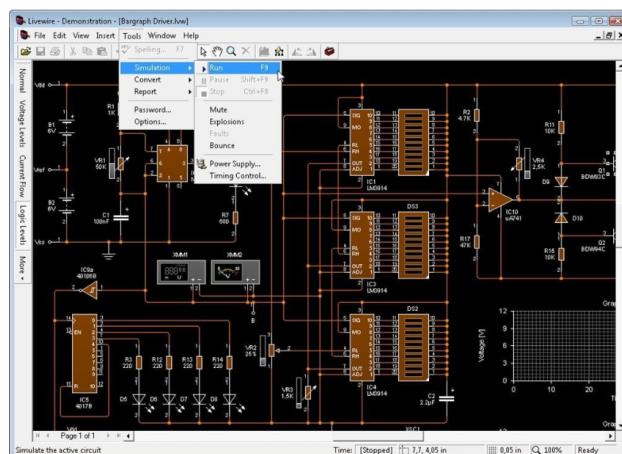


Gambar 2.62 Tampilan Fritzing

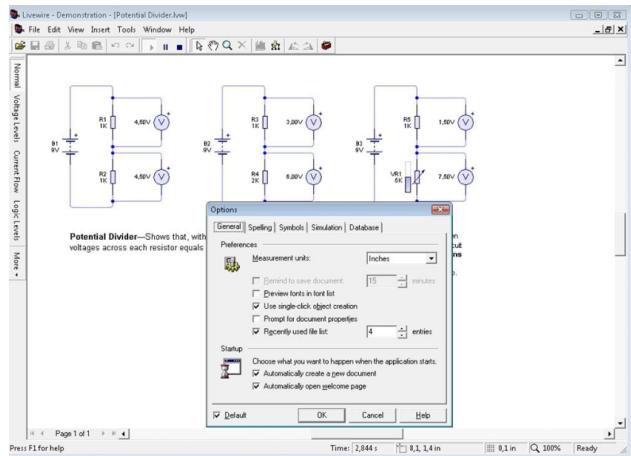
2. LiveWire LiveWire merupakan aplikasi simulasi rangkaian elektronika yang mudah digunakan, khususnya bagi pemula yang baru belajar elektronika. Dalam aplikasi ini sudah disediakan komponen-komponen dasar seperti resistor, kapasitor, op amp, dll. Dengan begitu kita dapat merangkai skema yang akan kita buat (misalkan rangkaian RLC) lalu mensimulasikannya, sehingga kita dapat melihat arus yang mengalir pada rangkaian tersebut.



Gambar 2.63 Tampilan LiveWire

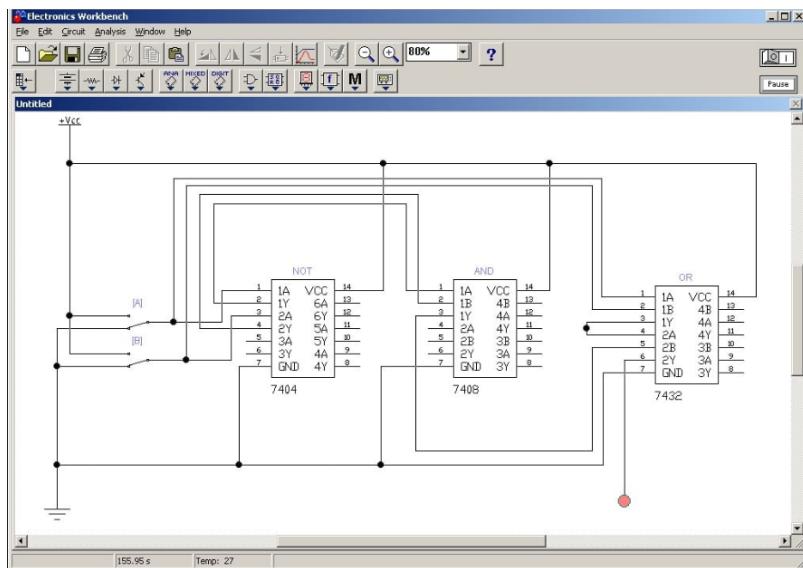


Gambar 2.64 Tampilan LiveWire

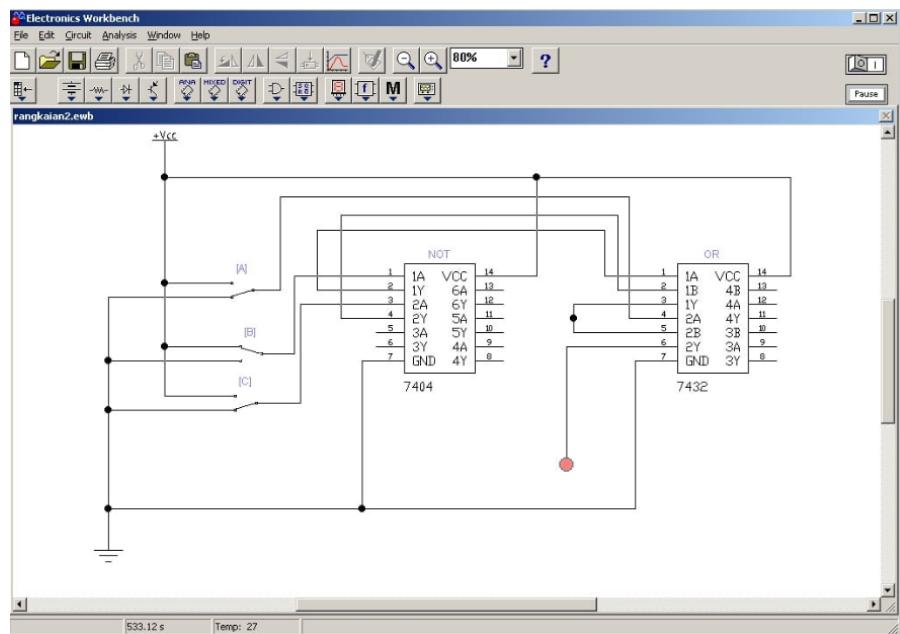


Gambar 2.65 Tampilan LiveWire

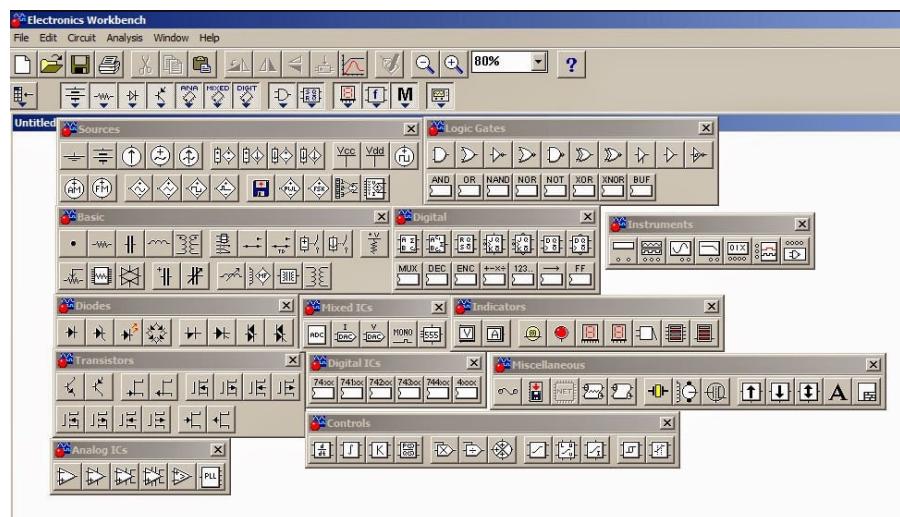
3. Electronics Workbench (EWB) *Electronics Workbench* atau yang sering dikenal dengan EWB merupakan aplikasi yang tidak jauh berbeda dengan LiveWire. Aplikasi ini sangat mudah untuk digunakan, ditambah dengan beberapa feature yang lebih lengkap seperti adanya simulasi osiloskop, spektrum analizer, dan *function generator*.



Gambar 2.66 Tampilan *Electronics Workbench* (EWB)



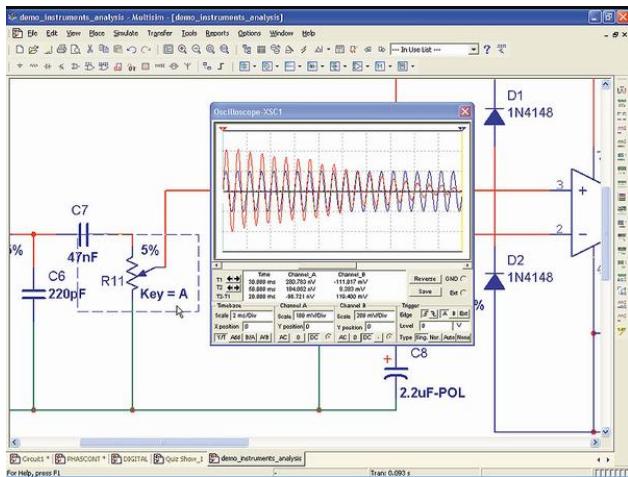
Gambar 2.67 Tampilan *Electronics Workbench* (EWB)



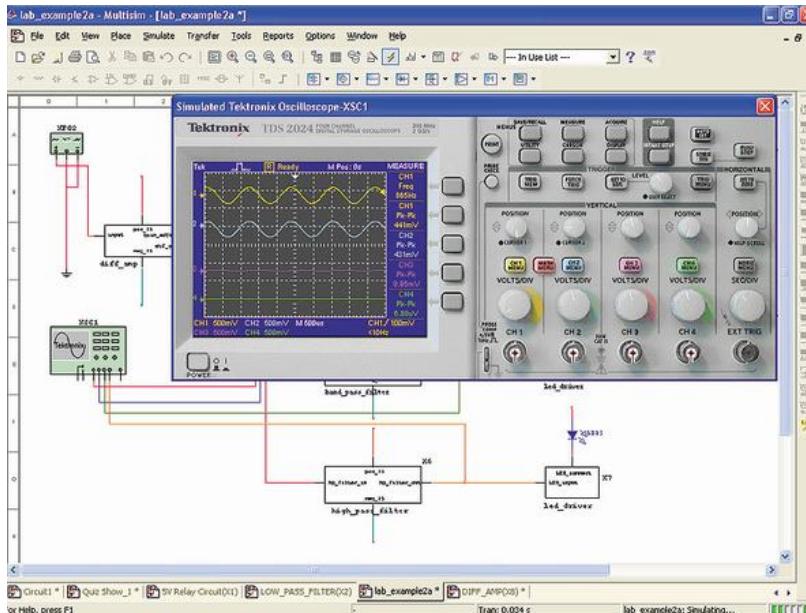
Gambar 2.68 Tampilan *Electronics Workbench* (EWB)

4. NI Multisim merupakan aplikasi yang masih satu pengembangan dengan EWB, bisa dikatakan aplikasi ini adalah pengembangan versi terbaru yang lebih lengkap

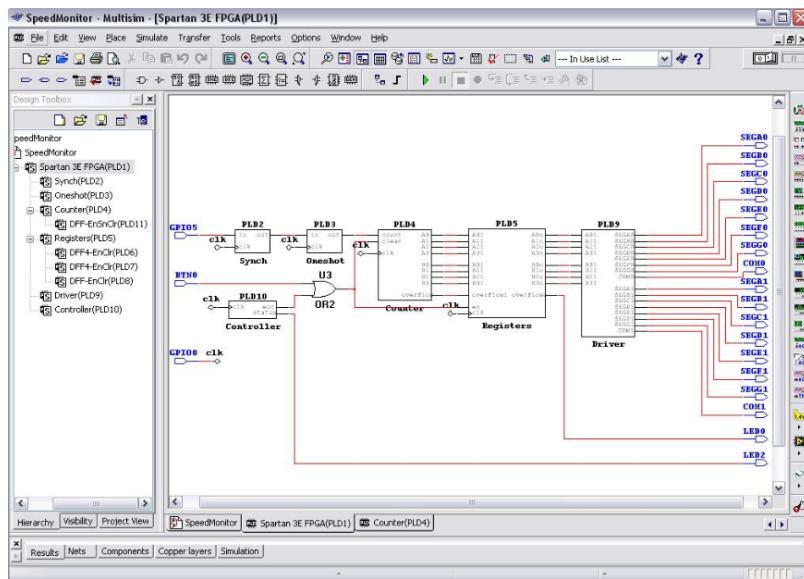
dari versi EWB yang sebelumnya. Aplikasi ini dilengkapi dengan komponen-komponen yang memiliki fungsi yang sama dengan aslinya, begitupun dengan perangkat elektronika seperti osiloskop, function generator, spektrum analizer, dll.



Gambar 2.69 Tampilan NI Multisim



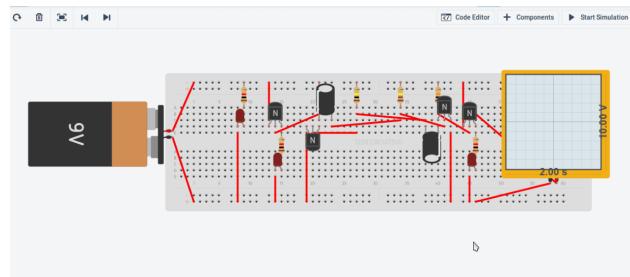
Gambar 2.70 Tampilan NI Multisim



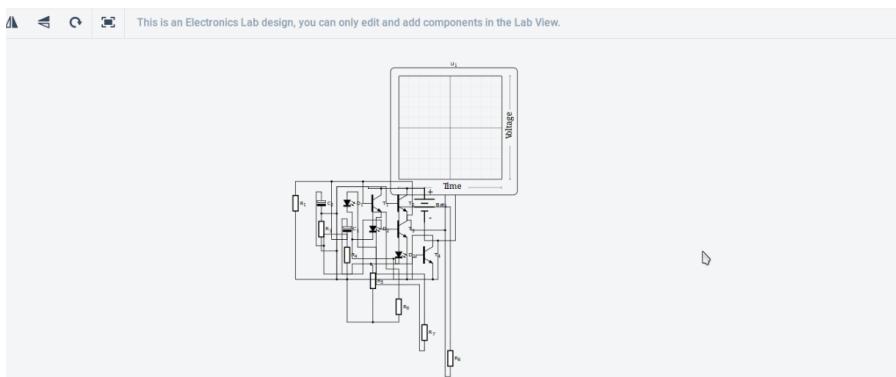
Gambar 2.71 Tampilan NI Multisim

5. Autodesk Circuits

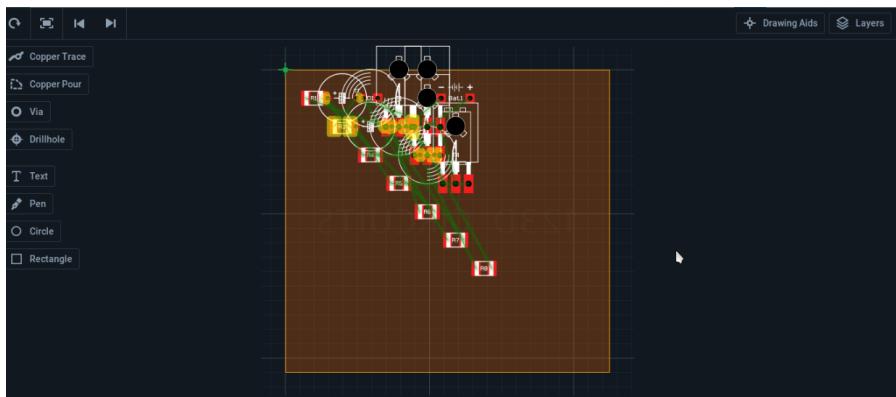
Autodesk Circuits merupakan aplikasi berbasis online, jadi apabila ingin menggunakan aplikasi ini, kita harus terhubung pada jaringan internet terlebih dahulu. Untuk menggunakannya kamu dapat mengunjungi halaman Autodesk Circuits berikut. Fitur yang dimiliki oleh aplikasi ini memang beragam diantaranya Electronics Lab, PCB Design, Library Manager, Circuit Scribe, dan Eagel. Aplikasi ini sangatlah wajib untuk dicoba, skema rangkaian pada aplikasi ini dibuat hampir sama dengan Fritzing, ditambah rangkaian tersebut bisa kita simulasi langsung seperti Arduino, ESP8266, dll. Akan tetapi komponen yang berada pada librarynya memang terbatas, jadi tidak semua komponen bisa kita coba.



Gambar 2.72 Tampilan Autodesk Circuits



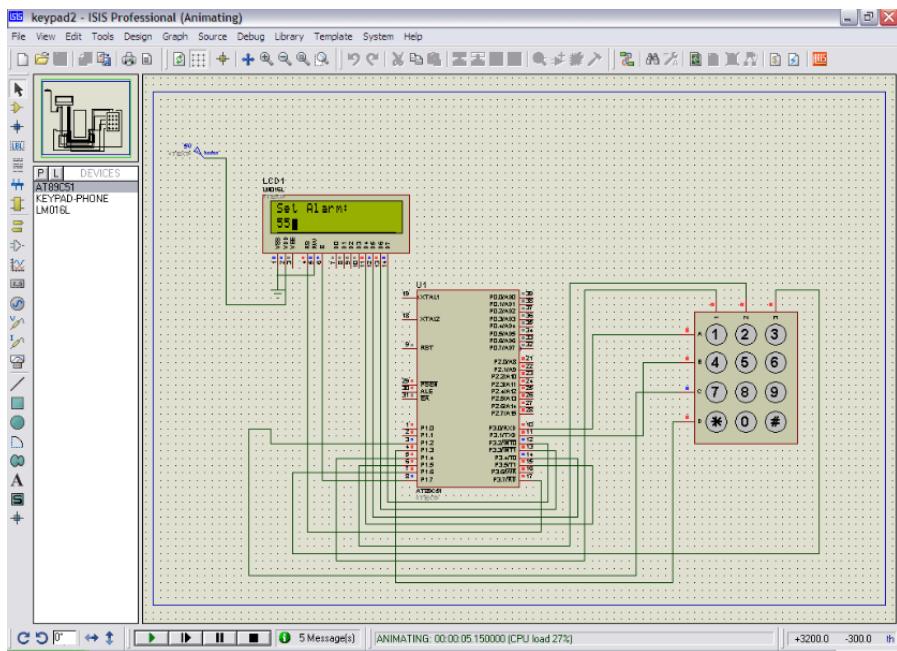
Gambar 2.73 Tampilan Autodesk Circuits



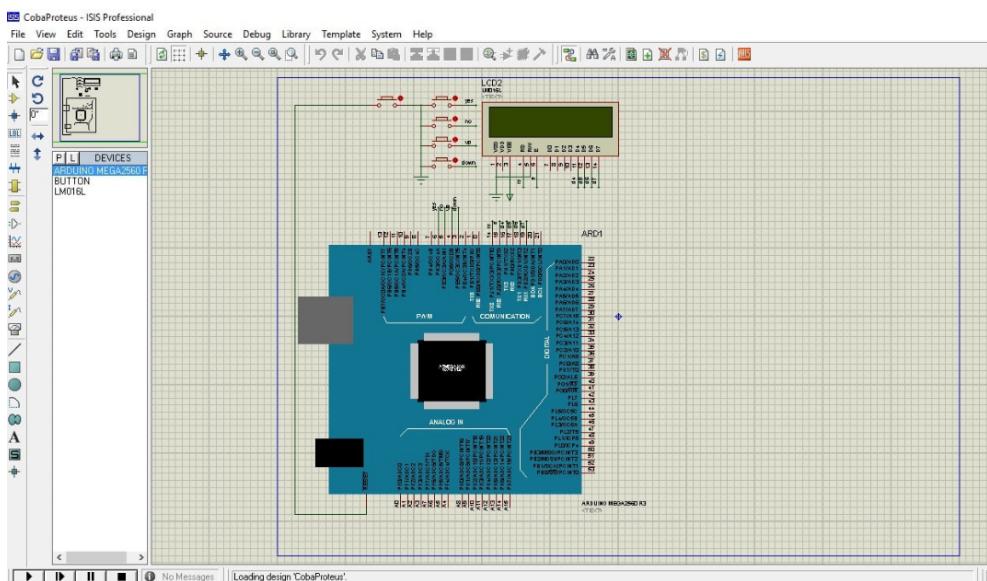
Gambar 2.74 Tampilan Autodesk Circuits

6. Proteus ISIS

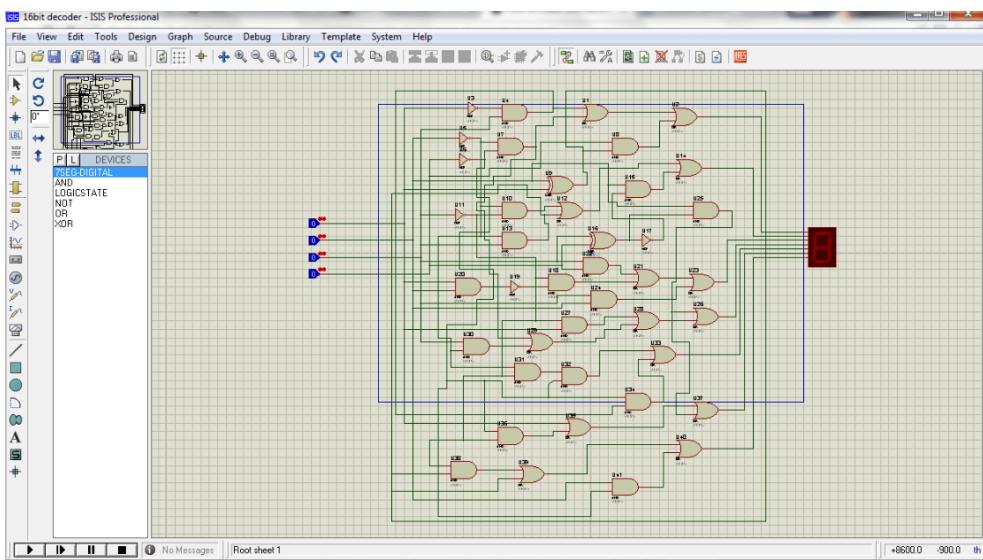
Proteus ISIS merupakan aplikasi yang saat ini menjadi yang terbaik dan paling direkomendasikan, selain dari library komponen yang lengkap dan bahkan bisa kita tambahkan, aplikasi ini juga dapat mensimulasikan semua rangkaian yang kita buat, aplikasi ini dilengkapi dengan text editor yang bisa kita gunakan untuk programming mikrokontroler. Kita juga dapat melakukan PCB layout pada rangkaian yang sudah kita buat, sehingga setelah kita membuat rangkaian untuk disimulasikan dan ternyata berhasil, kita dapat langsung merancangnya untuk di print pada layout PCB.



Gambar 2.75 Tampilan Proteus ISIS



Gambar 2.76 Tampilan Proteus ISIS



Gambar 2.77 Tampilan Proteus ISIS

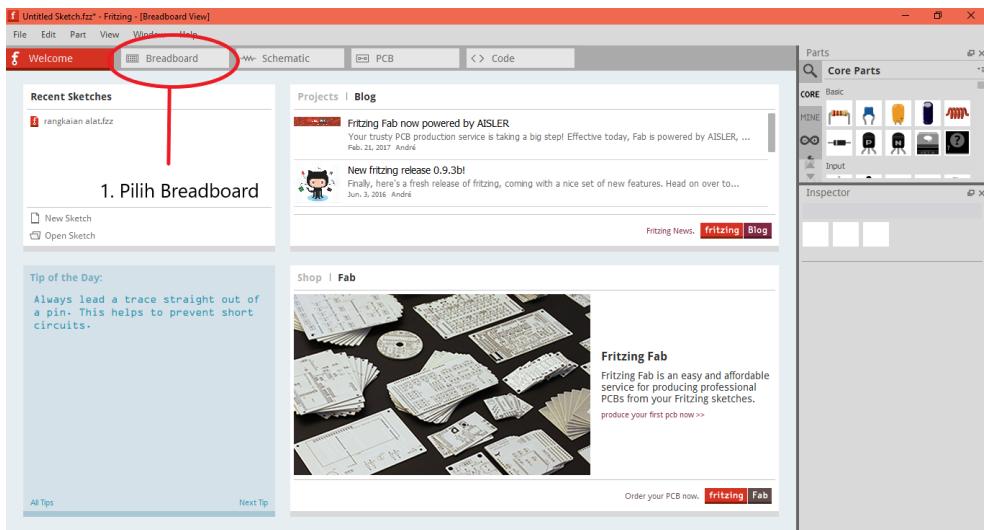
Dengan adanya penjelasan tentang software pembuatan skema komponen, kita dapat memilih software untuk pembuatan komponen sesuai yang kita inginkan. Tetapi kita juga harus memilih software yang akan digunakan sesuai kebutuhan.

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.

Untuk membuat skema komponen prototipe prediksi ketinggian air (PKA) untuk mendeteksi banjir peringatan dini ini menggunakan software frizting. Mengapa software frizting ? Karena software frizting ini sangat mudah dipahami bagi pemula. Selain itu komponen yang tersedia cukup banyak sehingga tidak perlu lagi untuk mengimport komponen dan salah satu keuntungan menggunakan frizting ini yaitu software ini free.

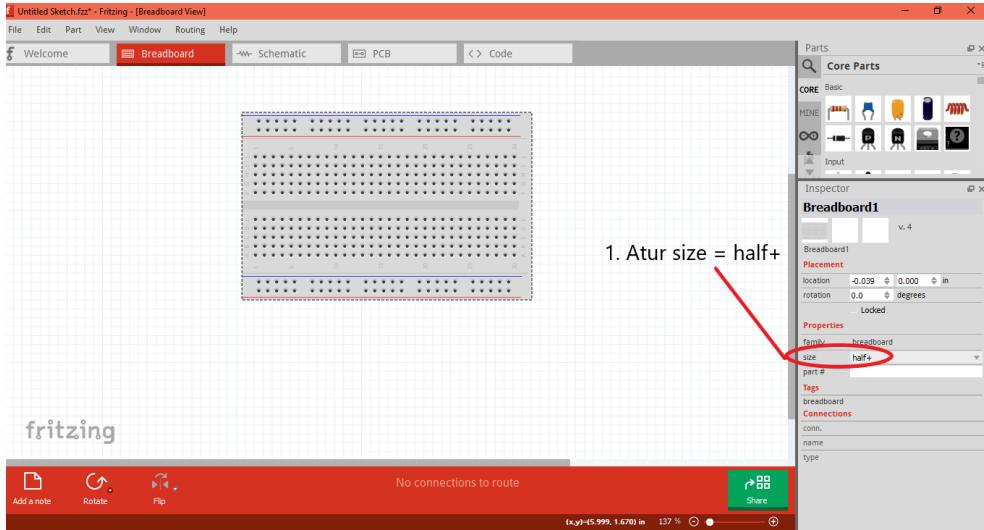
Berikut langkah-langkah untuk membuat skema prototipe prediksi ketinggian air (PKA) untuk mendeteksi banjir peringatan dini menggunakan software frizting :

1. Buka software frizting kemudian pilih breadboard. Seperti pada gambar dibawah ini :



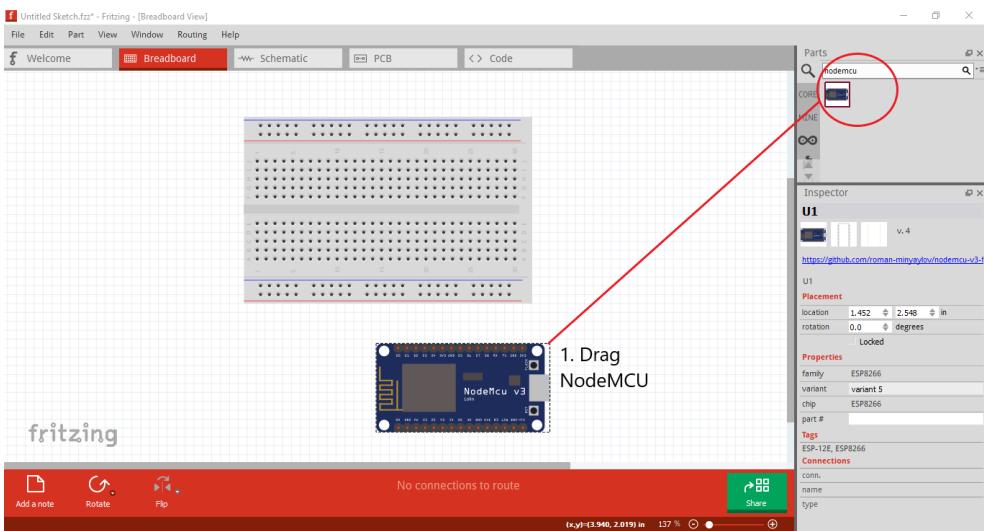
Gambar 2.78 Tampilan Awal Fritzing

2. Atur ukuran atau *size breadboard* menjadil ukuran kecil karena pada pembuatan skema ini tidak memerlukan space yang besar. Dengan cara ubah ukuran menjadi *half+* pada bagian *size* seperti berikut :



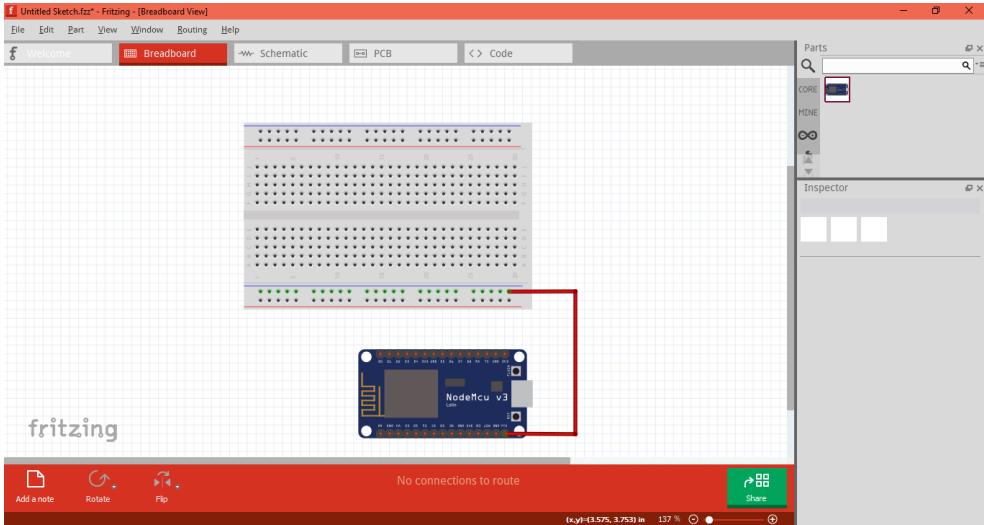
Gambar 2.79 Ubah Ukuran Breadboard

3. Cari komponen NodeMCU pada bagian *search parts* kemudian drag NodeMCU ke dekat *breadboard*.



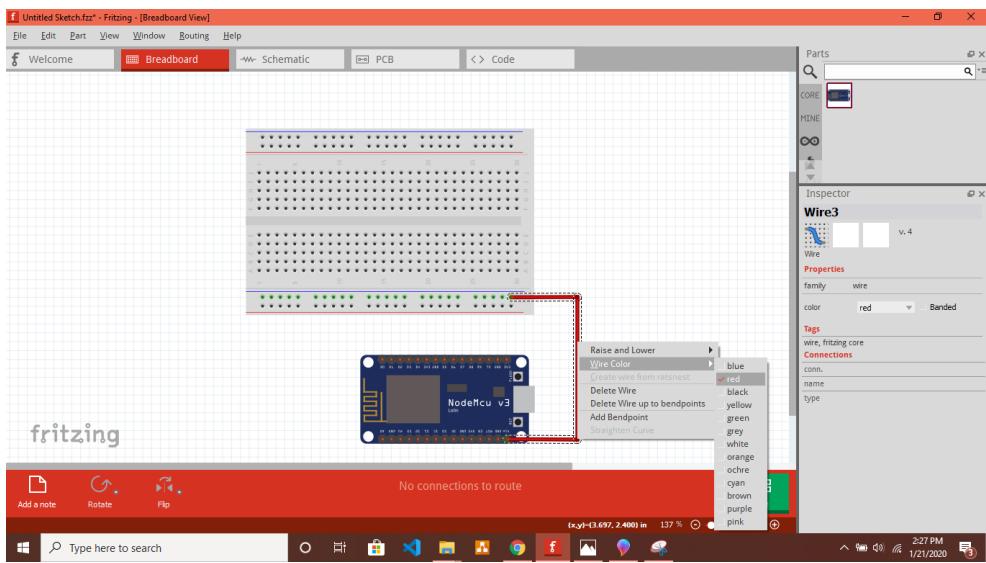
Gambar 2.80 Memasukan NodeMCU

- Setelah itu sambungkan Vin pada bagian positif (+) *breadboard* dengan cara klik dua kali pada bagian Vin maka akan muncul wire dan arahkan ke bagian positif *breadboard*.



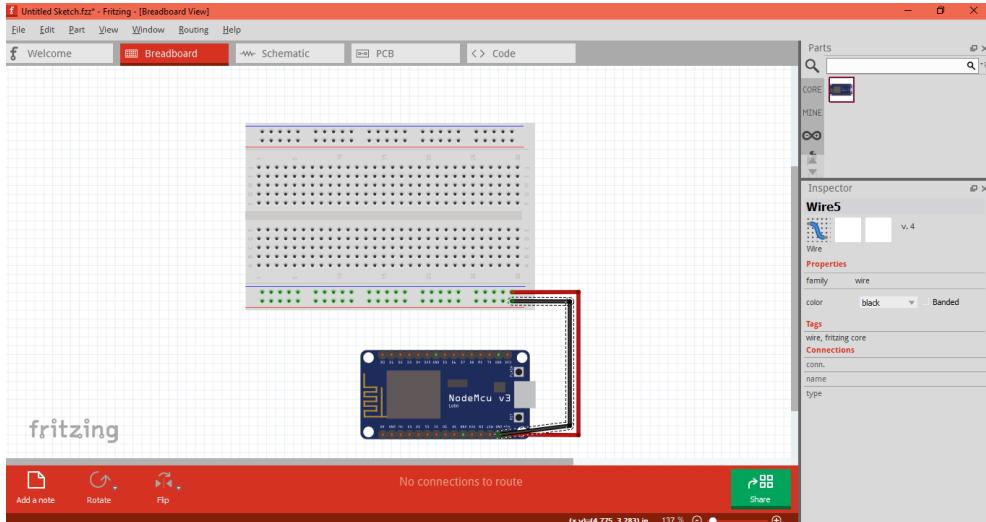
Gambar 2.81 Sambungkan Vin ke Breadboard

- Untuk mengubah warna wire untuk menjadi merah dengan cara klik kanan pada wire. Kemudian pilih *wire colour* dan pilih warna merah atau *red*.



Gambar 2.82 Mengubah Warna Wire

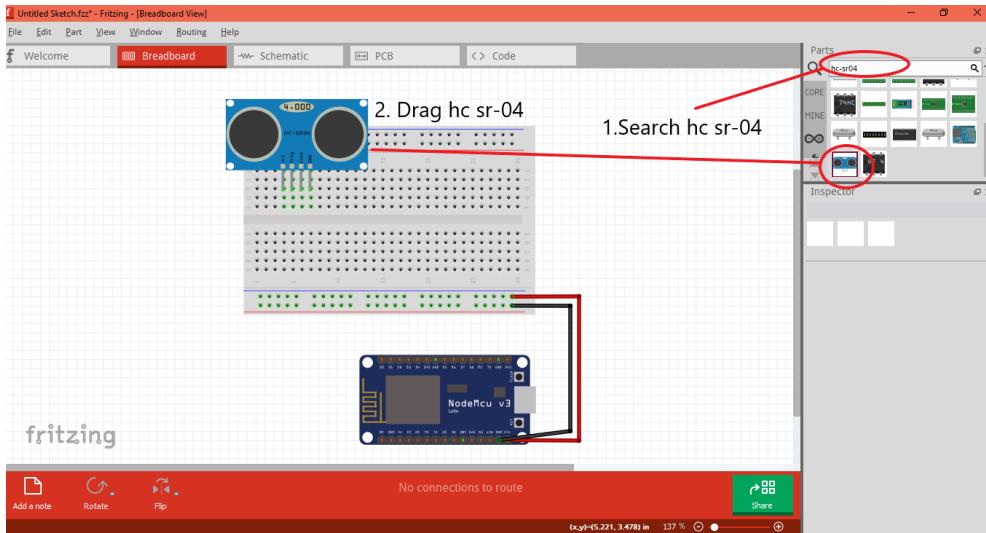
6. Sambungkan GND pada bagian negatif (-) *breadboard* dengan cara klik dua kali pada bagian Vin maka akan muncul *wire* dan arahkan ke bagian negatif *breadboard*.



Gambar 2.83 Sambungkan GND ke *Breadboard*

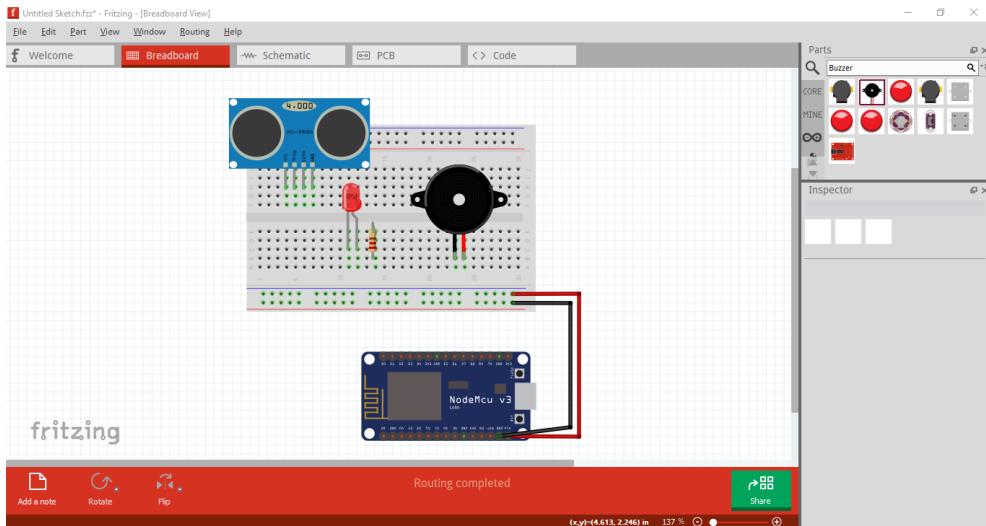
Untuk mengubah warnanya sama seperti langkah sebelumnya.

7. Masukan komponen sensor ultrasonic dengan cara ketik hc sr-04 pada bagian *search parts*. Setelah itu drag sensor ke *breadboard*



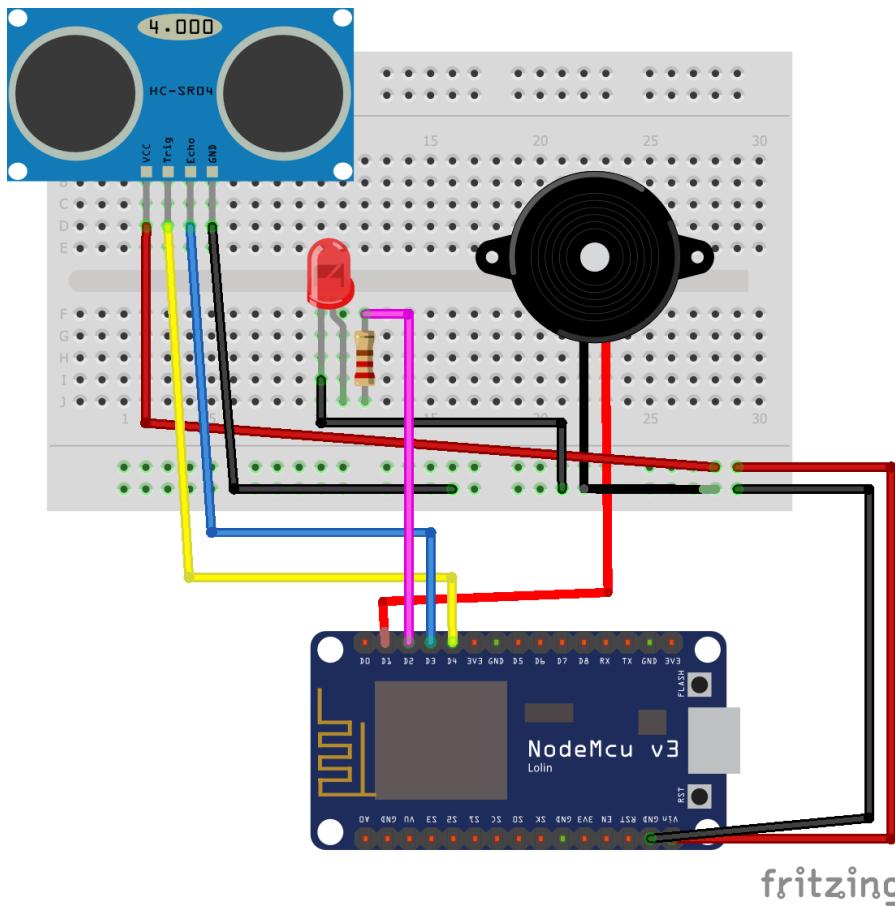
Gambar 2.84 Menambahkan Komponen Sensor Ultrasonic

8. Tambahkan komponen led, resistor, dan buzzer dan drag kedalam *breadboard* seperti gambar dibawah :



Gambar 2.85 Menambahkan Komponen Led, Resistor dan Buzzer

Berikut hasil akhir dari skema prototipe prediksi ketinggian air (PKA) untuk mendeteksi banjir peringatan dini yang telah dibuat menggunakan software fritzing .



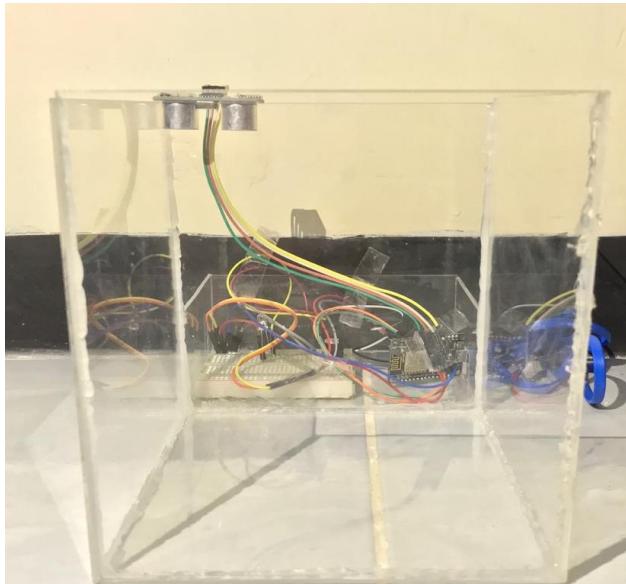
Gambar 2.86 Rangkaian Alat

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

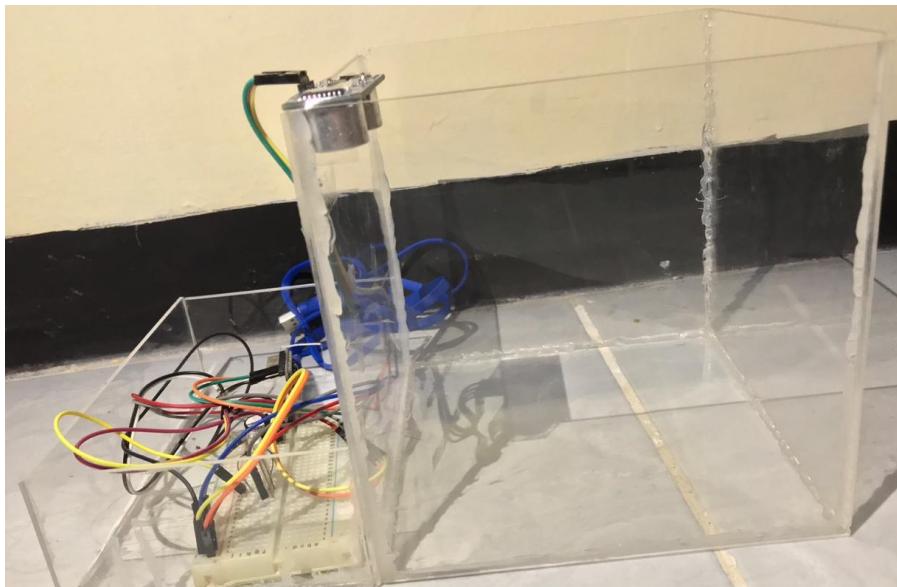
1. Sambungkan Vin NodeMCU menggunakan kabel jumper pada *breadboard* bagian positif (+). Vin merupakan Power atau VCC dengan memiliki daya 5V. Jadi jika tidak ada 5V dalam sebuah *microcontroller* dapat di gantikan dengan vin.
2. Sambungkan GND (*ground*) NodeMCU menggunakan kabel jumper pada *breadboard* bagian negative (-). GND merupakan sistem pengamanan pada instalasi listrik dimana jika terjadi kebocoran listrik .

3. pin vcc yang berada di sensor ultrasonic US 100 hubungkan pada *breadboard* bagian positif (+) menggunakan kabel jumper yang sebelumnya sudah di sambungkan dengan vin pada NodeMCU. Sehingga jika kita colokan di bagian positif pada *breadboard* otomatis akan sudah teraliri vin.
4. pin GND yang berada di sensor ultrasonic US 100 hubungkan pada *breadboard* bagian negatif (-) menggunakan kabel jumper yang sebelumnya sudah di sambungkan dengan GND pada NodeMCU. Sehingga jika kita colokan di bagian negatif pada *breadboard* otomatis akan sudah teraliri GND.
5. pin echo pada sensor ultrasonic US 100 hubungkan pada D3 NodeMCU menggunakan kabel jumper.
6. pin trig pada sensor ultrasonic US 100 hubungkan pada D4 NodeMCU menggunakan kabel jumper.
7. Letakan atau colokan led pada *breadboard*. Pada bagian positif (+) led dihubungkan dengan resistor dan pin D2. Kemudian untuk bagian negatif (-) dihubungkan ke bagian GND.
8. Letakan atau colokan buzzer pada *breadboard*. Pada bagian Positif (+) hubungkan dengan D1. Kemudian untuk bagian negatif (-) dihubungkan ke bagian GND.

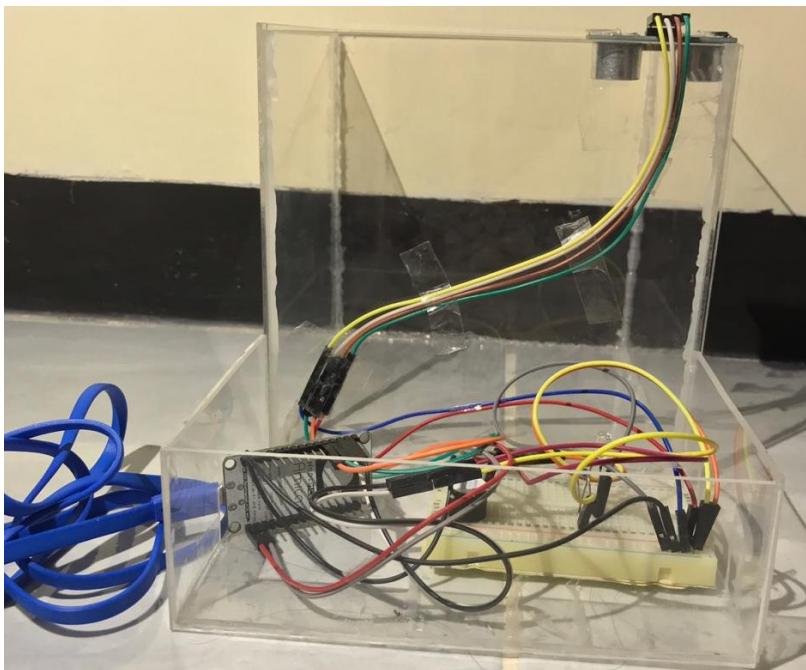
Setalah Kita mengikuti langkah-langkah tersebut maka simpan komponen pada rangka prototipe yang telah dibuat sebelumnya seperti pada gambar beikut:



Gambar 2.87 Rangka Prototipe Depan

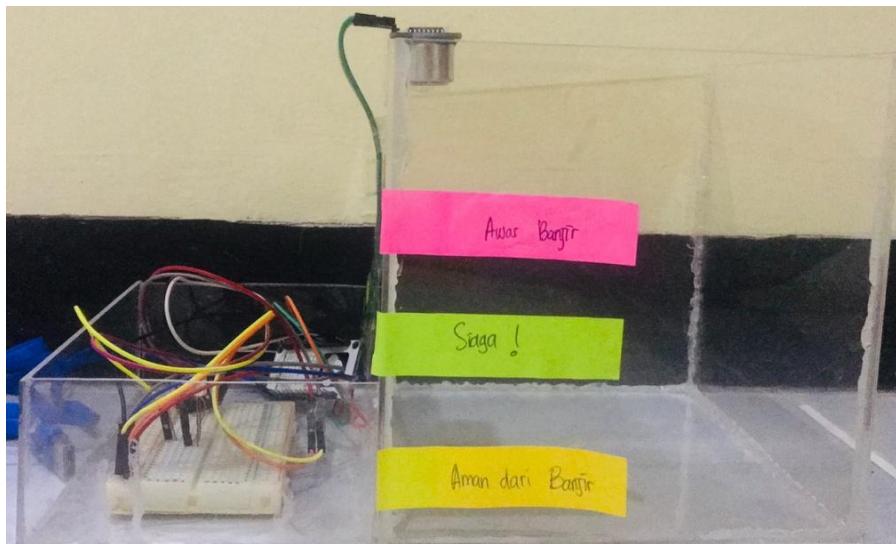


Gambar 2.88 Rangka Prototipe Samping



Gambar 2.89 Rangka Prototipe Komponen

Kemudian setelah rangka selesai dibuat dan komponen sudah disimpan pada rangka prototipe, hal yang perlu dilakukan adalah menempelkan sticky note yang bertujuan untuk memberikan level ketinggian air. Seperti gambar dibawah ini :



Gambar 2.90 Rangka Prototipe Level Ketinggian Air

BAB 3

IMPLEMENTASI CODING KOMPONEN PROTOTIPE

3.1 Penjelasan Implementasi dan Coding

3.1.1 Implementasi

Arti implementasi menurut KBBI (Kamus Besar Bahasa Indonesia) yaitu pelaksanaan / penerapan. Sedangkan pengertian umum adalah suatu tindakan atau pelaksana rencana yang telah disusun secara cermat dan rinci (matang). Kata implementasi sendiri berasal dari bahasa Inggris *to implement* artinya mengimplementasikan. Tak hanya sekedar aktivitas, implementasi merupakan suatu kegiatan yang direncanakan serta dilaksanakan dengan serius juga mengacu pada norma-norma tertentu guna mencapai tujuan kegiatan.

Pendapat lain mengatakan bahwa pengertian implementasi adalah suatu tindakan atau bentuk aksi nyata dalam melaksanakan rencana yang telah dirancang dengan matang. Dengan kata lain, implementasi hanya dapat dilakukan jika sudah ada perencanaan dan bukan hanya sekedar tindakan semata.

Dari penjelasan tersebut kita dapat melihat bahwa implementasi bermuara pada mekanisme suatu sistem. Penerapan implementasi harus sesuai dengan perencanaan yang telah dibuat agar hasil yang dicapai sesuai dengan yang diharapkan.

Seperti yang disebutkan sebelumnya, implementasi merupakan aktivitas yang dilakukan secara sistematis dan terikat oleh mekanisme untuk mencapai tujuan tertentu. Mengacu pada pengertian implementasi tersebut, adapun beberapa tujuan implementasi adalah sebagai berikut:

1. Tujuan utama implementasi adalah untuk melaksanakan rencana yang telah disusun dengan cermat, baik oleh individu maupun kelompok.
2. Untuk menguji serta mendokumentasikan suatu prosedur dalam penerapan rencana atau kebijakan.
3. Untuk mewujudkan tujuan-tujuan yang hendak dicapai di dalam perencanaan atau kebijakan yang telah dirancang.
4. Untuk mengetahui kemampuan masyarakat dalam menerapkan suatu kebijakan atau rencana sesuai dengan yang diharapkan.
5. Untuk mengetahui tingkat keberhasilan suatu kebijakan atau rencana yang telah dirancang demi perbaikan atau peningkatan mutu.

3.1.2 Coding

Coding yaitu menulis sekumpulan code sesuai syntax (aturan penulisan) tergantung bahasa pemrograman yang dipakai (python, php, ruby, java, atau yang lainnya) dengan bantuan text editor seperti sublime text, atom, notepad, dll. Dengan coding, kita memberikan daftar instruksi kepada komputer sesuai tujuan kita membuat. Contoh coding yaitu kita melakukan penulisan code untuk membuat website, aplikasi android, dan lain sebagainya.

Tujuan dari Pengkodean (coding) adalah menjadikan setiap karakter data dalam sebuah informasi digital ke dalam bentuk biner agar dapat ditransmisikan dan bisa melakukan komunikasi data. Kode-kode yang digunakan dalam komunikasi data pada system computer memiliki perbedaan dari generasi ke generasinya, karena semakin besar dan kompleksnya data yang akan dikirim atau digunakan digunakan.

3.2 Implementasi Coding Prototipe

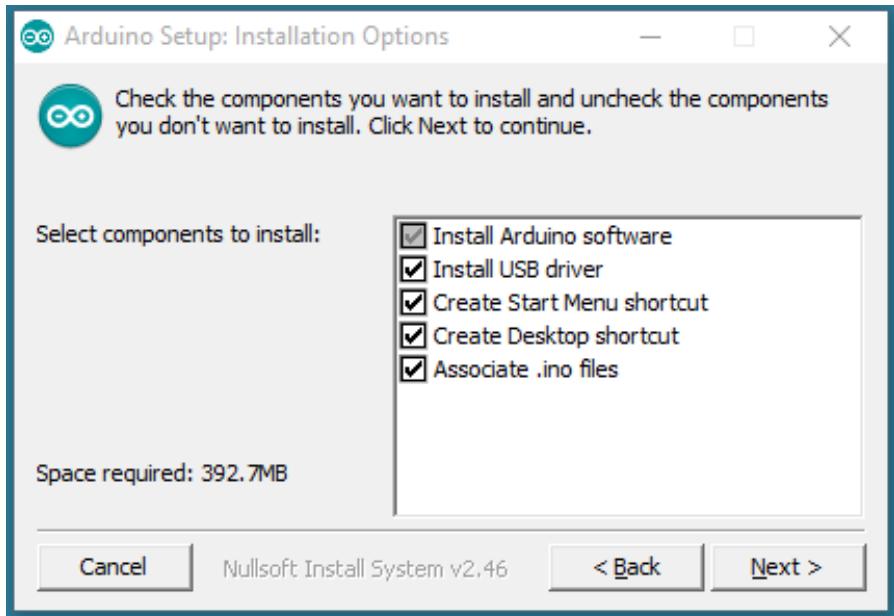
Dalam mengimplementasikan codingan untuk membuat prototipe prediksi ketinggian air (PKA) untuk mendeteksi banjir peringatan banjir ini menggunakan bahasa pemograman C. Dimana bahasa pemograman C ini telah di jelaskan pada bab sebelumnya.

Untuk lebih mempermudah pemahaman mengenai pemograman bahasa C menggunakan *software* Arduino IDE. Maka terlebih dahulu akan memberikan contoh tentang pemograman bahasa C dasar serta cara install dan penggunaan *software* Arduino IDE.

3.2.1 Cara Install Arduino IDE dan Penggunaanya

Hal pertama kali yang harus kita butuhkan adalah board development Arduino, yang berfungsi untuk dapat menginstall Arduino IDE. Dalam tutorial ini menggunakan Arduino IDE untuk sistem operasi Windows 10-64 byte yang berformat .exe . Perhatikan langkah-langkah berikut ini:

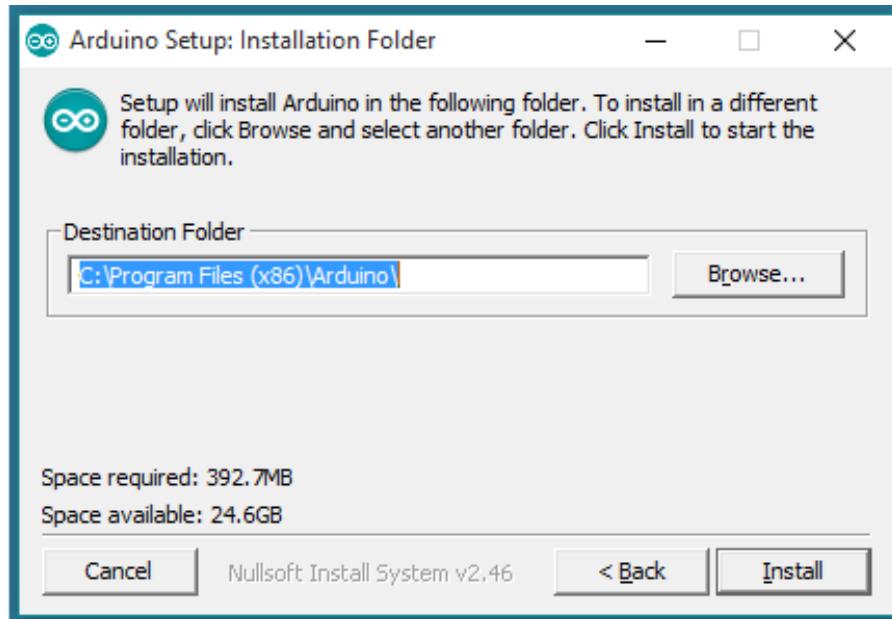
1. Setelah diunduh, klik dua kali file tersebut sehingga muncul jendela instalasi sebagai berikut :



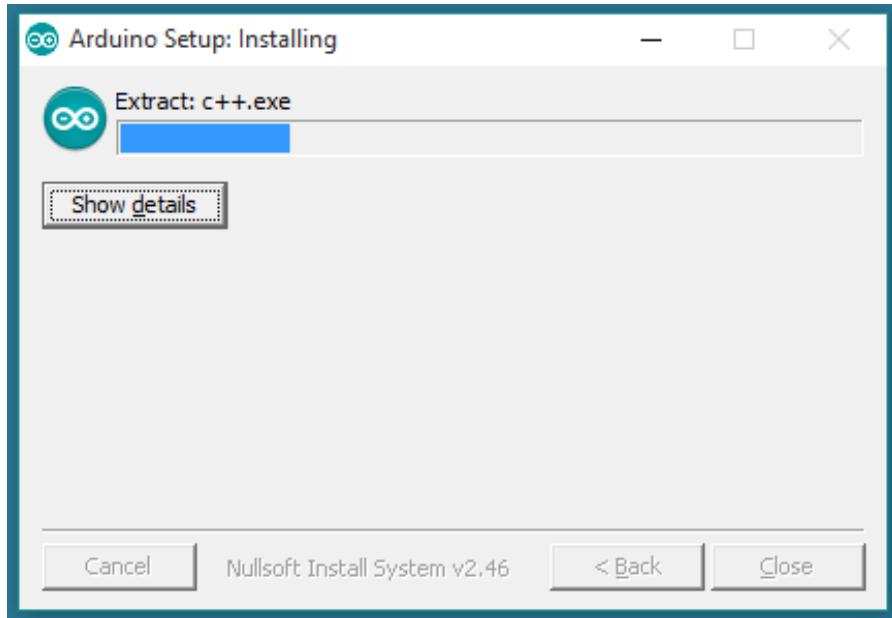
Gambar 3.1 Tampilan Jendela Instalasi

Pilih komponen yang sudah diinstal. Sesuaikan dengan kebutuhan anda, pastikan pilihan install USB driver telah tercentang karena ini terkait nanti dengan driver usb Arduino. Setelah beres klik Next.

2. Pada jendela berikutnya akan ditampilkan pada jendela dialog tujuan instalasi dari arduino IDE. Pilih default saja dan langsung klik Install.



Gambar 3.2 Jendela Dialog Tujuan Instalasi Dari Arduino IDE

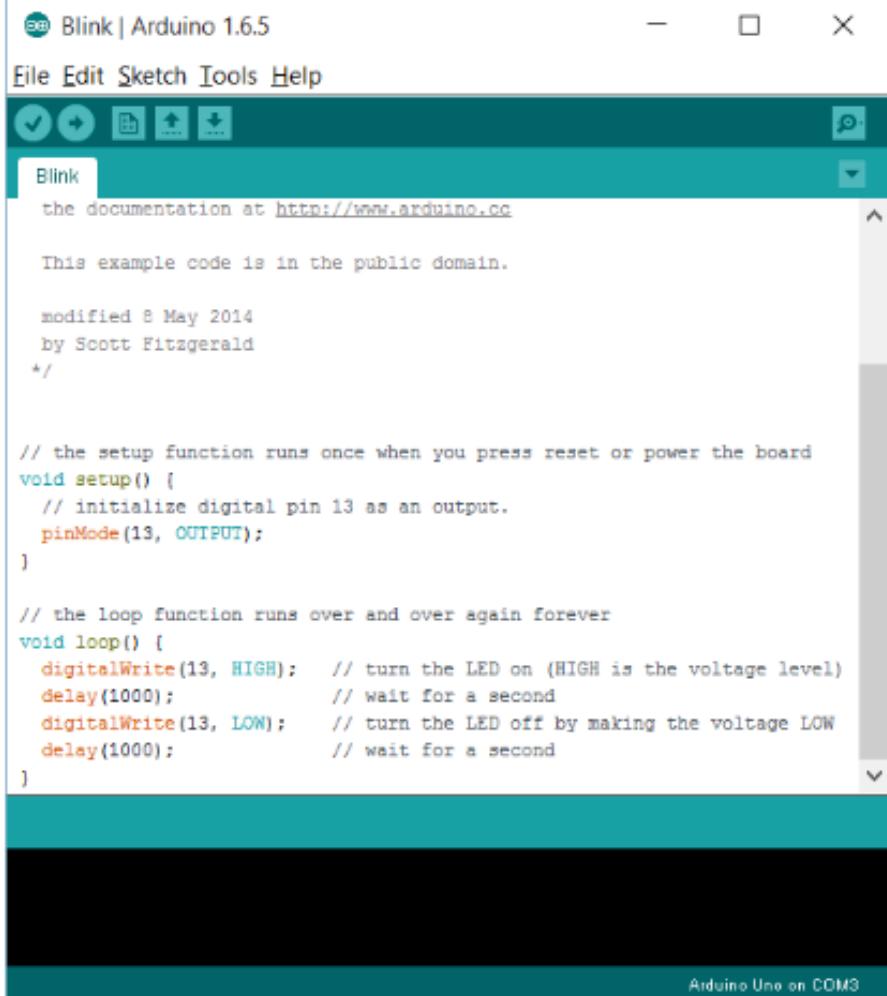


Gambar 3.3 Jendela Instalasi Arduino IDE

Tunggu hingga proses instalasi selesai. Setelah Arduino IDE selesai terinstall, kini sambungkan kabel USB Arduino pada port USB komputer/laptop. Jika anda menggunakan file installer yang sama di tunjukan pada pembahasan sebelumnya, maka arduino akan otomatis terbaca di COMX (X adalah nomer port serial, misalkan COM1 atau COM2 dan seterusnya).

3.2.2 Contoh Pemrograman Arduino

Kini saatnya menjalankan Arduino IDE, pilih File → Examples → Basic → Blink . Maka akan muncul pada jendela yang berisi source sebagai berikut:



The screenshot shows the Arduino IDE interface with the title bar "Blink | Arduino 1.6.5". The menu bar includes File, Edit, Sketch, Tools, and Help. Below the menu is a toolbar with icons for save, upload, and other functions. A scrollable code editor displays the "Blink" sketch. The code is as follows:

```
the documentation at http://www.arduino.cc
This example code is in the public domain.

modified 8 May 2014
by Scott Fitzgerald
 */

// the setup function runs once when you press reset or power the board
void setup() {
    // initialize digital pin 13 as an output.
    pinMode(13, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
    digitalWrite(13, HIGH);      // turn the LED on (HIGH is the voltage level)
    delay(1000);                // wait for a second
    digitalWrite(13, LOW);       // turn the LED off by making the voltage LOW
    delay(1000);                // wait for a second
}
```

In the bottom right corner of the code editor, it says "Arduino Uno on COM3".

Gambar 3.4 Tampilan Arduino IDE Blink

Program ini memerintahkan Arduino untuk memberi sinyal digital bergantian dari 0 V dan 5V pada port 13. Kenapa Port 13? Karena di Port 13 Arduino Uno, terdapat LED orange kecil yang tertanam dalam board sehingga untuk mengetes apakah arduino kita berfungsi atau tidak, cukup program port 13 untuk mengirimkan sinyal digital dan lihat hasilnya secara visual pada LED tersebut. Langkah selanjutnya perhatikan berikut ini :

1. Pilih Port Serial

Sebelum kita mengupload program ke dalam board, pilih port usb dimana Arduino Uno terbaca. Pilih Tools → Port untuk menyambungkan port.

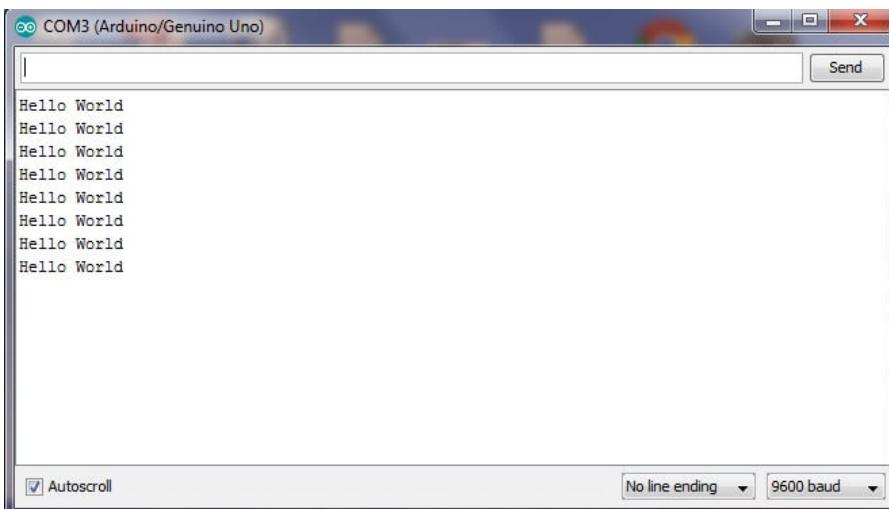
2. Upload Program

Setelah koneksi ke port serial sudah oke, klik tombol *Upload* pada Arduino IDE. dan tunggu beberapa saat hingga muncul pesan *Done Uploading* pada status bar. Jika berhasil, kita akan melihat LED orange arduino Uno akan berkedip-kedip dengan delay tertentu.

Setelah Blink kita akan membuat "Hello Word" menggunakan Arduino IDE yang nantinya akan di tampilkan di serial monitor. Berikut code pemogramannya :

```
1 // Test Program Dasar Serial Monitor Menggunakan Arduino
2 // by Fungky King
3 void setup()
4 {
5   Serial.begin(9600);
6 }
7
8 void loop()
9 {
10  Serial.print("Hello World");
11  delay(2000);
12 }
```

Setelah program diatas diketikkan pada software Arduino IDE, maka tahap selanjutnya lakukan proses *Upload* kedalam board Arduinonya tunggu sampai proses selesai sampai ada tanda *Done Uploading*. Untuk melihat hasilnya klik menu Tools kemudian Serial Monitor maka hasilnya akan muncul seperti gambar berikut.



Gambar 3.5 Tampilan Di Serial Monitor

Penjelasan tentang fungsi-fungsi dari tiap-tiap bagian program diatas adalah sebagai berikut:

- (a) Void Setup() adalah fungsi yang dijalankan secara otomatis pertama kali oleh board Arduino, dimana Semua kode program yang ada dalam void setup akan dibaca sekali oleh Arduino. Biasanya isinya berupa kode perintah untuk menentukan fungsi pada sebuah pin atau deklarasi INPUT/OUTPUT.
- (b) begin() digunakan untuk mengatur baudrate / kecepatan transmisi data. Beberapa pilihan kecepatan komunikasi data yang dapat digunakan pada board arduino adalah 300, 1200, 2400, 4800, 9600, 14400, 19200, 28800, 38400, 57600 atau 115200. Pengaturan baudrate dilakukan pada bagian setup().
- (c) Void loop() melakukan proses dimana semua kode yang ada disini akan dibaca berulang kali (terus menerus) oleh Arduino.
- (d) Perintah Serial.print digunakan untuk menampilkan data ke serial monitor. Data yang ditampilkan dapat berupa karakter, bytes, atau angka.
- (e) delay(2000) merupakan pernyataan untuk melakukan penundaan selama 2000 milidetik atau 2 detik dari proses akhir pembacaan program sebelumnya dimana Arduino akan mengulang proses pembacaan program dari awal kembali.

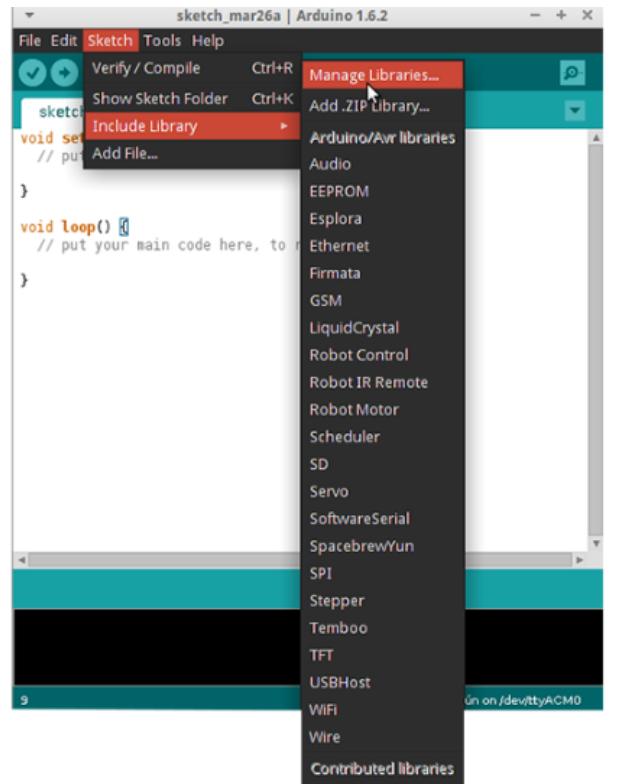
3.2.3 Cara Import Library

Library adalah kumpulan code yang biasanya terkumpul dalam sebuah namespace/module/ package (tergantung anda menggunakan di bahasa pemrograman apa)

yang dapat di import/ reuse ke program lain. Untuk mengimport library ada tiga cara yaitu menggunakan *manager library*, mengimport library .zip hasil download dan Instalasi Manual.

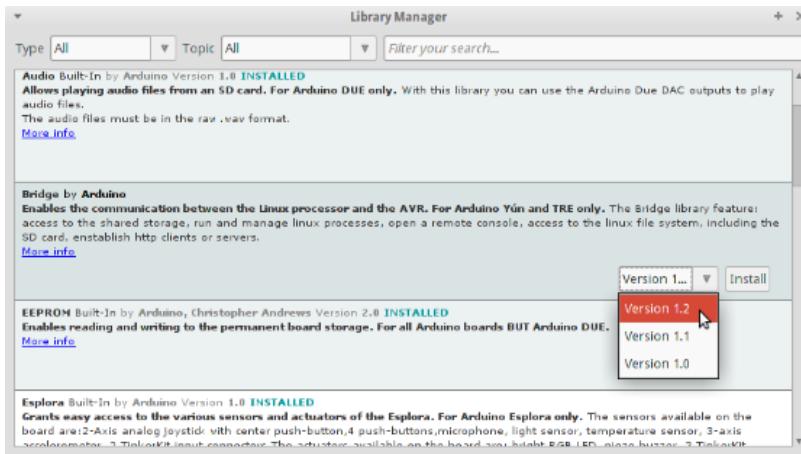
1. Import Library Menggunakan Manager

Untuk menginstal sebuah Library baru ke dalam IDE Arduino,dapat menggunakan Library Manager (tersedia sejak Arduino IDE versi 1.6.2). Buka Arduino IDE dan klik ke menu Sketch dan kemudian Include Library ; Manage Libraries



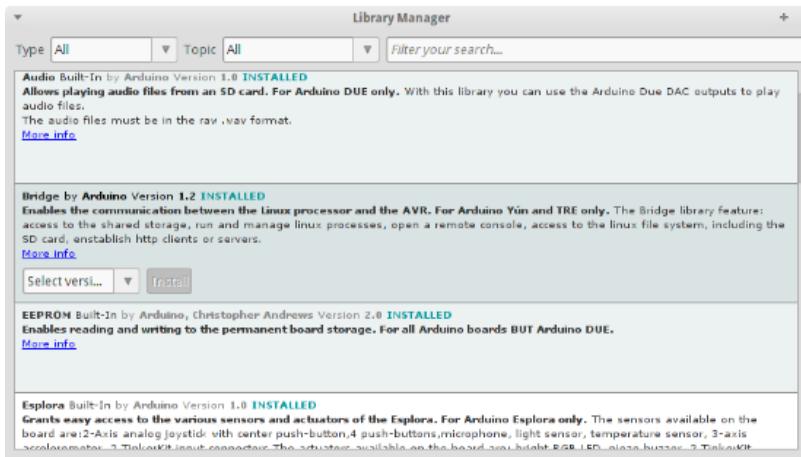
Gambar 3.6 Memilih Menu Manage Librarie

Kemudian Sketch Manager akan terbuka dan akan menemukan daftar Library yang sudah terpasang ataupun Library baru yang siap untuk di instalasi. Dalam contoh ini kita akan menginstal library Bridge. Gulir daftar untuk menemukannya, lalu pilih versi Library yang ingin Anda instal. (beberapa Library terkadang hanya ada satu versi Library yang tersedia).



Gambar 3.7 Tampilan Sketch Manager

Langkah selanjutnya adalah klik instal dan tunggu IDE untuk menginstal library baru. Setelah selesai, sebuah tag Installed akan muncul di sebelah Library Bridge. Setelah terinstal, sekarang kita dapat menemukan Library baru yang tersedia di menu Include Library.



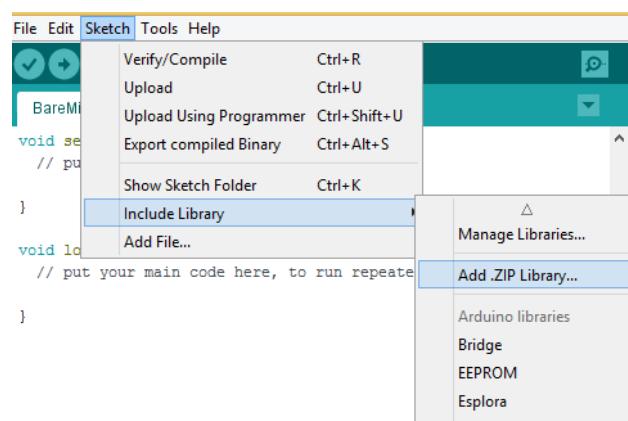
Gambar 3.8 Tampilan Library Yang Terinstall

2. Mengimport Libray .zip hasil Download.

Terkadang Library hasil buatan sesorang banyak dan sering yang didistribusikan sebagai file ZIP atau folder sehingga dapat kita unduh, di GitHub misalnya.

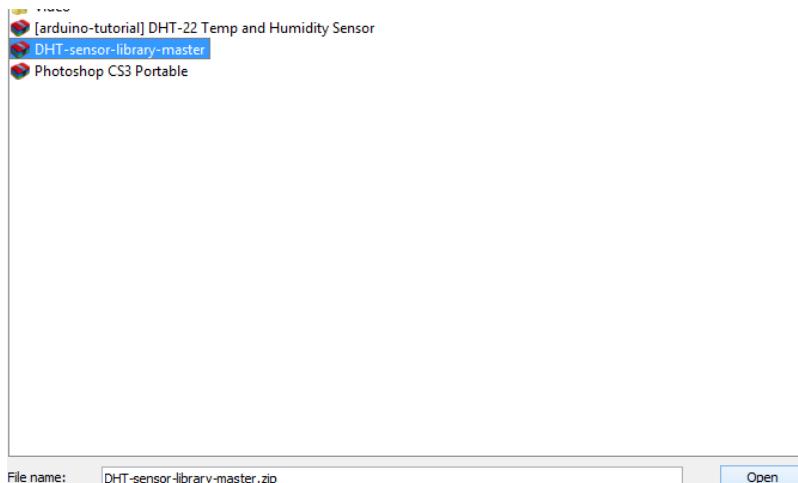
Nama folder adalah nama Library. Di dalam folder tersebut akan ada file .cpp, file .h , Folder Contoh Sketch, dan file lainnya yang dibutuhkan oleh Library. Dimulai dengan versi 1.0.5, Kita bisa menginstal library pihak ke-3 di IDE. Jangan unzip Libray yang telah didownload, biarkan seperti apa adanya.

Buka aplikasi Arduinonya, lalu Masuk ke menu SKETCH, pilih INCLUDE LIBRARY, pilih ADD. ZIP Library , seperti gambar berikut :



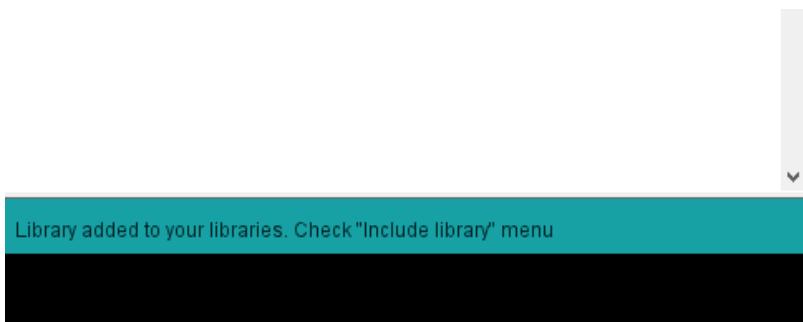
Gambar 3.9 Tampilan Library ADD ZIP

Kemudian akan diminta untuk memilih Library yang ingin ditambahkan. klik add .ZIP library, lalu cari file ZIP yang sudah didownload.



Gambar 3.10 Pilih File ZIP

Jika berhasil, aplikasi Arduino kamu akan muncul keterangan seperti dibawah ini:



Gambar 3.11 Tampilan Library Berhasil Ditambahkan

3. Instalasi Manual

Bila ingin menambahkan Library secara manual, perlu mendownload sebagai file ZIP, lalu extract file tersebut. (Klik Kanan pada File ZIP lalu pilih extract to nama file), setelah folder hasil extrakan tersedia, selanjutnya kita harus memasukannya kedalam folder di C:User-NamaPC-Documents-Arduino-libraries.

3.3 Memprogram Komponen *Prorotype PKA*

Setalah kita mengetahui cara penggunaan Arduino IDE,*import library*, dan struktur pemogramannya yang telah dibahas sebelum nya maka, tahapan selanjutnya yaitu memrogram komponen untuk *prototype PKA* ini. Dengan cara memprogram serta testing komponen satu persatu kemudian digabungkan menjadi kesatuan yang utuh.

3.3.1 Memprogram Sensor Ultasonic

komponen pertama yang akan di program atau di tes yaitu sensor ultrasonic US 100 yang nantinya berfungsi untuk membaca jarak ketinggian air. Berikut kodinganya :

```
1 #define triggerPin  D4
2 #define echoPin      D3
3 void setup() {
4     Serial.begin (9600);
5     pinMode(triggerPin , OUTPUT);
6     pinMode(echoPin , INPUT);
7 }
8 void loop() {
9     long duration , jarak;
10    digitalWrite(triggerPin , LOW);
11    delayMicroseconds(2);
```

```

13  digitalWrite ( triggerPin , HIGH );
14  delayMicroseconds ( 10 );
15  digitalWrite ( triggerPin , LOW );
16  duration = pulseIn ( echoPin , HIGH );
17  jarak = ( duration / 2 ) / 29.1;
18  Serial . println ( " jarak : " );
19  Serial . print ( jarak );
20  Serial . println ( " cm " );
21  delay ( 1000 );
22 }
```

Penjelasan coding diatas yaitu :

1. define triggerPin D4 define echoPin D3

Script tersebut untuk menginsiasi pin yang akan digunakan. Pin D4 digunakan untuk pin trigger dan D3 digunakan untuk pin echo.

2. Serial.begin (9600); ,pinMode(triggerPin, OUTPUT); ,pinMode(echoPin, INPUT);

Script diatas berada di fungsi void setup(). Serial.begin() digunakan untuk memulai serial. selanjutnya pinMode digunakan untuk menentukan fungsi dari pin yang akan digunakan (triggerPin dan echoPin). triggerPin berfungsi sebagai pin output sedangkan echoPin sebagai pin input.

3. long duration, jarak;

Script diatas digunakan untuk deklarasi variabel duration dan jarak. Duration dan jarak merupakan variabel bertipe long. Long merupakan tipe data serupa dengan int tetapi memiliki range yang lebih panjang.

4. digitalWrite(triggerPin, LOW);, delayMicroseconds(2);, digitalWrite(triggerPin, HIGH);, delayMicroseconds(10); digitalWrite(triggerPin, LOW);

Script tersebut menjelaskan bahwa pin trigger menembakkan pulsa sinyal ultrasonik selama 10 micro second.

5. duration = pulseIn(echoPin, HIGH);

Selanjutnya ketika gelombang ultrasonik itu ditangkap kembali setiap pulsanya oleh pin echo maka waktu tangkap tersebut diubah menjadi variabel duration.

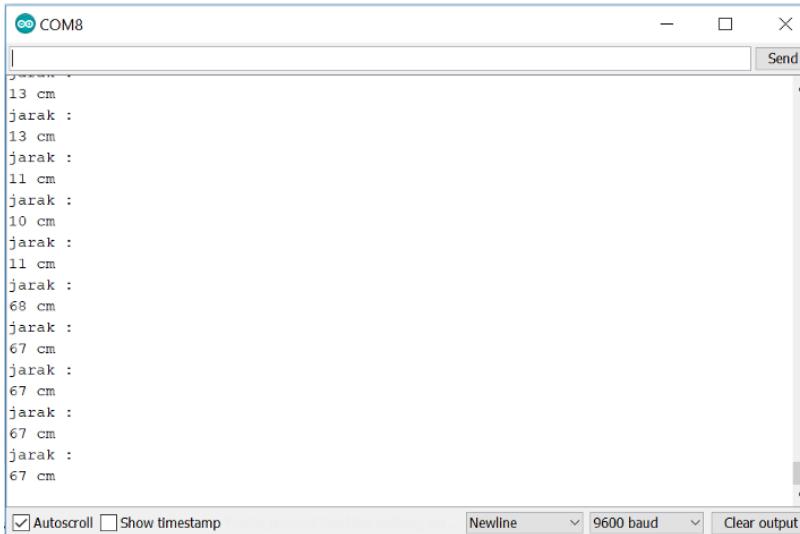
6. jarak = (duration/2) / 29.1;

Dengan persamaan diatas nilai waktu duration akan dikonversi menjadi nilai jarak dalam bentuk cm.

7. Serial.println("jarak :");, Serial.print(jarak);, Serial.println(" cm ");, delay(1000);

Selanjutnya nilai jarak yang didapatkan akan ditampilkan di serial monitor setiap 1 detik (delay(1000)).

Untuk melihat hasil pembacaan sensor lewat serial monitor pada arduino IDE. Jika sukses maka akan seperti gambar berikut :



Gambar 3.12 Tampilan serial monitor

3.3.2 Memprogram Led

Komponen kedua yang akan di program atau di tes yaitu led yang nantinya berfungsi untuk sebagai indikator.Berikut Kodingannya :

```

1 const int ledPin = D2;
2 void setup() {
3   pinMode(2, OUTPUT);
4   void loop() {
5     digitalWrite(2, LOW);
6     delay(1000);
7     digitalWrite(2, HIGH);
8     delay(2000);
9 }
```

Penjelasan coding diatas yaitu :

1. const int ledPin = D2;

Script tersebut untuk menginsiasi pin yang akan digunakan.

2. pinMode(2, OUTPUT);

Menetapkan pin GPIO 2/LED sebagai OUTPUT.

3. digitalWrite(2, LOW);

Perintah untuk mematikan Lampu atau memberikan nilai LOW(0) pada pin GPIO 2.

4. delay(1000);

Matikan lampu selama 1 detik.

5. digitalWrite(2, HIGH);

Perintah memberikan nilai HIGH atau menyalakan Lampu.

6. delay(2000);

Nyalakan lampu selama 2 detik.

3.3.3 Memprogram Buzzer

Komponen ketiga yang akan di program atau di tes yaitu buzzer yang nantinya berfungsi untuk sebagai indikator.Berikut Kodingannya :

```
1 const int ledPin = D1;
2 void setup() {
3   pinMode(1, OUTPUT);
4   void loop() {
5     digitalWrite(1, LOW);
6     delay(1000);
7     digitalWrite(1, HIGH);
8     delay(2000);
9 }
```

Penjelasan coding diatas yaitu :

1. const int ledPin = D1;

Script tersebut untuk menginsiasi pin yang akan digunakan.

2. pinMode(1, OUTPUT);

Menetapkan pin GPIO 1/Buzzer sebagai OUTPUT.

3. digitalWrite(2, LOW);

Perintah untuk mematikan Buzzer atau memberikan nilai LOW(0) pada pin GPIO 1.

4. delay(1000);

Matikan buzzer selama 1 detik.

5. digitalWrite(1, HIGH);

Perintah memberikan nilai HIGH atau menyalakan buzzer.

6. delay(2000);

Nyalakan buzzer selama 2 detik.

3.3.4 Memprogram Sensor, Led dan Buzzer

Setelah memprogram komponen satu persatu maka selanjutnya menggabungkan komponen yang telah diprogram sebelumnya. Komponen yang akan digabungkan yaitu sensor, led , buzzer terlebih dahulu. Mengapa kita harus memprogram satu persatu terlebih dahulu tidak langsung ? alasan terlebih dahulu memprogram komponen satu persatu jika adanya error pada saat proses penggabungan komponen akan mengetahui komponen manakah yang error. Berikut codingan dari komponen yang digabungkan :

```
1 // defines pins numbers
2
3 const int trigPin = 2; //D4
4 const int echoPin = 0; //D3
5 const int ledPin = 4; //D2
6 const int buzzerPin = 5; //D1
7
8
9
10 // defines variables
11 long duration;
12 int distance;
13 int safetyDistance;
14
15 void setup() {
16
17
18 pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
19 pinMode(echoPin, INPUT); // Sets the echoPin as an Input
20 pinMode(buzzerPin, OUTPUT);
21 pinMode(ledPin, OUTPUT);
22 Serial.begin(115200); // Starts the serial communication
23 }
24
25 void loop() {
26
27 // Clears the trigPin
28 digitalWrite(trigPin, LOW);
29 delayMicroseconds(2);
30
31 // Sets the trigPin on HIGH state for 10 micro seconds
32 digitalWrite(trigPin, HIGH);
33 delayMicroseconds(10);
34 digitalWrite(trigPin, LOW);
35
36 // Reads the echoPin, returns the sound wave travel time in
37 // microseconds
38 duration = pulseIn(echoPin, HIGH);
39
40 // Calculating the distance
41 distance= duration*0.034/2;
42 // Prints the distance on the Serial Monitor
43 // Serial.print("Distance: ");
44 // Serial.println(distance);
45 // delay(2000);
```

```

45
46 safetyDistance = distance;
47 if (safetyDistance <= 5){
48   digitalWrite(buzzerPin , HIGH);
49   digitalWrite(ledPin , HIGH);
50 }
51 else{
52   digitalWrite(buzzerPin , LOW);
53   digitalWrite(ledPin , LOW);
54 }
55
56 // Prints the distance on the Serial Monitor
57 Serial.print("Distance: ");
58 Serial.println(distance);
59 Serial.print("pulseIn ");
60 Serial.print("duration ");
61 }
```

Penjelasan coding diatas yaitu :

1. const int trigPin = 2; //D4 , const int echoPin = 0; //D3 , const int ledPin = 4; //D2, const int buzzerPin = 5; //D1

Script tersebut untuk menginsiasi pin yang akan digunakan. Pin D4 digunakan untuk pin trigger,D3 digunakan untuk pin echo, D2 digunakan untuk ledppin, dan D1 digunakan untuk buzzerpin.

2. // defines variables , long duration; , int distance;, int safetyDistance;

Script diatas merupakan sebuah pendefinisian variabel.

3. pinMode(trigPin, OUTPUT);, pinMode(echoPin, INPUT); , pinMode(buzzerPin, OUTPUT); , pinMode(ledPin, OUTPUT);, Serial.begin(115200);

Pada script tersebut merupakan sebuah bagian void setup yang berfungsi sebagai output dan input dari komponen. Selain itu pada script tersebut adanya baudrate yang digunakan 115200 yang berfungsi untuk komunikasi serial.

4. digitalWrite(trigPin, LOW); , delayMicroseconds(2);

Script diatas merupakan sebuah perintah untuk mematikan (*Low* pada trigpin disensor ultrasonic.

5. digitalWrite(trigPin, HIGH , delayMicroseconds(10); , digitalWrite(trigPin, LOW);

Pada bagian script diatas merupakan sebuah perintah untuk trigpin pada sensor ultrasonic dan akan adanya jeda sebesar 10 detik.

6. safetyDistance = distance;// if (safetyDistance != 5)// digitalWrite(buzzerPin, HIGH);// digitalWrite(ledPin, HIGH);/// else// digitalWrite(buzzerPin, LOW);// digitalWrite(ledPin, LOW);//

Script tersebut merupakan sebuah logika dari sensor ultrasonic. Dimana jika jarak yang dibaca oleh sensor ultrasonic adalah kurang dari sama dengan 5 maka

buuer dan led akan menyala dan jika tidak membaca jarak kurang sama dengan 5 buzzr dan led mati.

7. Serial.print("Distance: "); , Serial.println(distance); , Serial.print("pulseIn "); , Serial.print("duration "); ,

Bagian script tersebut merupakan sebuah komunikasi serial yang nantinya jarak yang dibaca oleh sensor akan di tampilkan di serial monitor.

3.3.5 Memprogram Komponen dan Antares

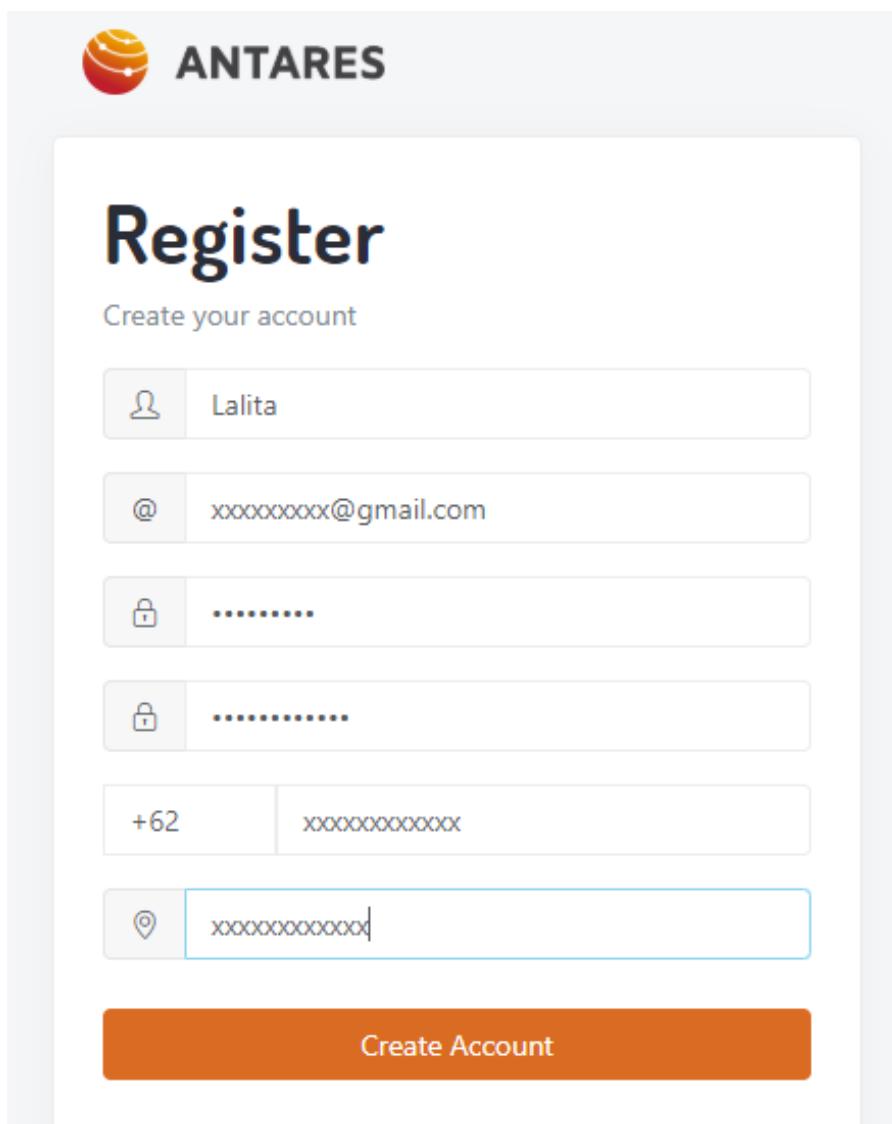
Setelah semua komponen yang digabungkan sudah diprogram maka tahapan selanjutnya yaitu menambahkan pemograman untuk antaresnya. Antares tersebut akan digunakan sebagai media penyimpan data yang dibaca oleh sensor ultrasonic. Sebelum memprogram ada langkah-langkah yang harus dilakukan untuk tahapan antares ini yaitu :

1. Registrasi di Antares menggunakan gmail. Untuk membuka halaman antares cukup ketikan <https://antares.id/id/index.html> pada browser maka akan muncul tampilan seperti gambar berikut.



Gambar 3.13 Halaman Awal Antares

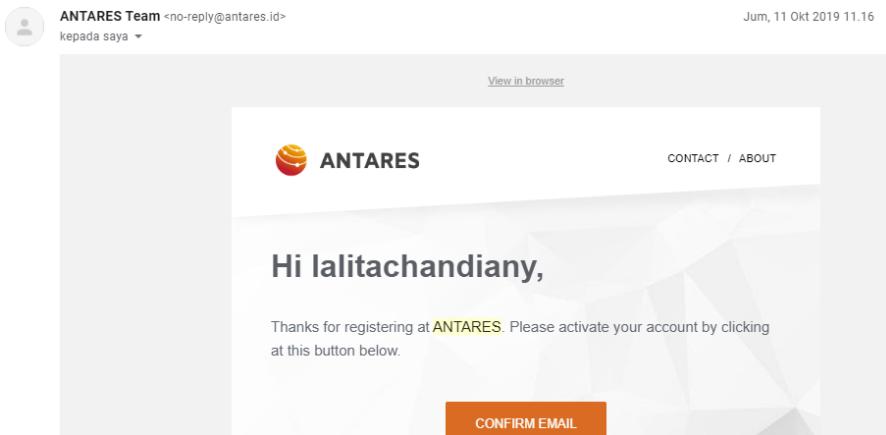
Setelah halaman tersebut muncul, pilih *Signup* kemudian masukan email yang aktif.



Gambar 3.14 Tampilan Registrasi Antares

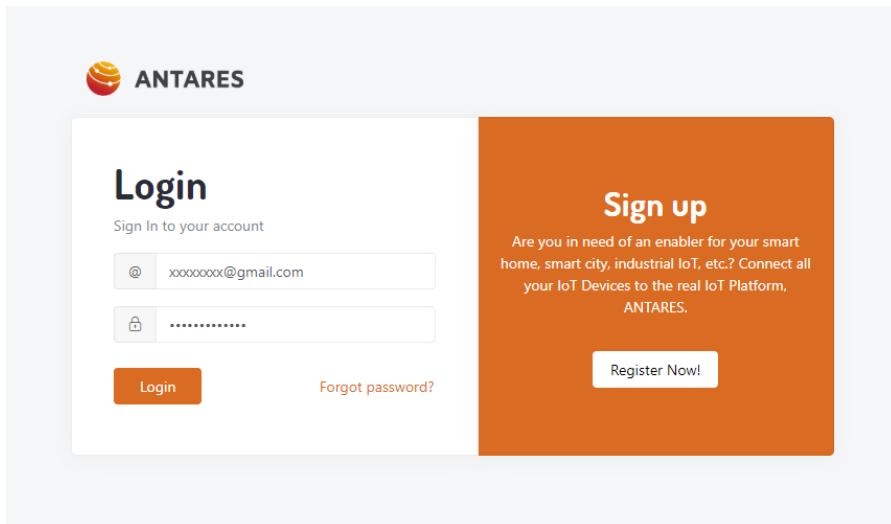
Masukan nama , alamat email (alamat email aktif), password, no telepon dan alamat. Alamat email harus aktif karena ini berguna untuk *verifikasi account* dan apabila ada info mengenai antares akan dikirmkan melalui email.

Activation Kotak Masuk



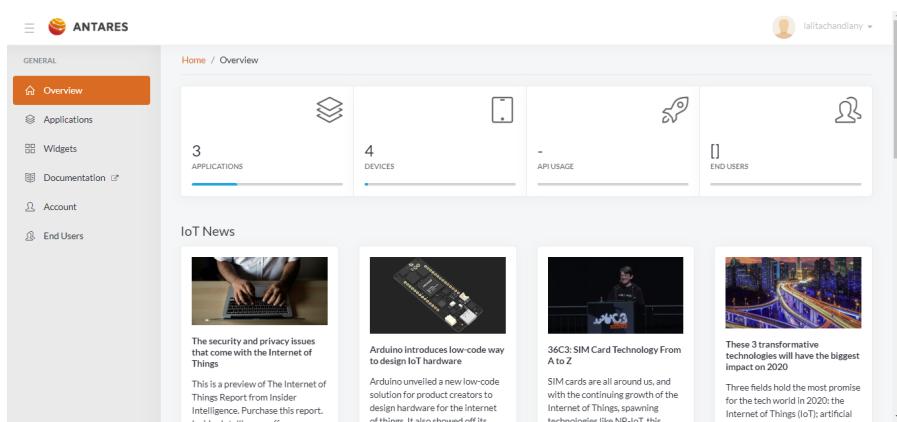
Gambar 3.15 Verifikasi email

Setelah melakukan registrasi di antares maka akan adanya email *verifikasi* pada email yang sebelumnya email tersebut di daftarkan. Untuk mengkonfirmasi *verifikasi* dari pihak antares cukup klik *confirm*



Gambar 3.16 Login Antares

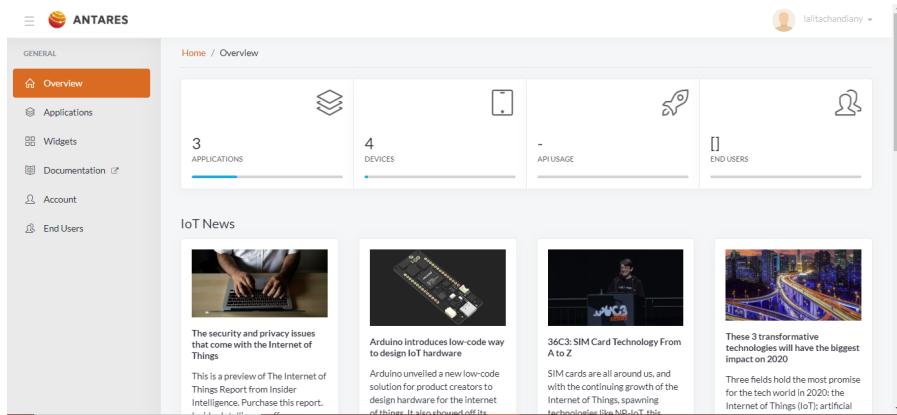
setelah kita meng*confirm* antares di email maka kita harus login di antares dengan mamasukan email yang telah didaftarkan dan memasukan password yang sebelumnya telah dibuat.



Gambar 3.17 Tampilan Awal Antares

Gambar diatas merupakan halaman awal antares ketika kita sudah login. Dalam halaman ini adanya beberapa menu yaitu :

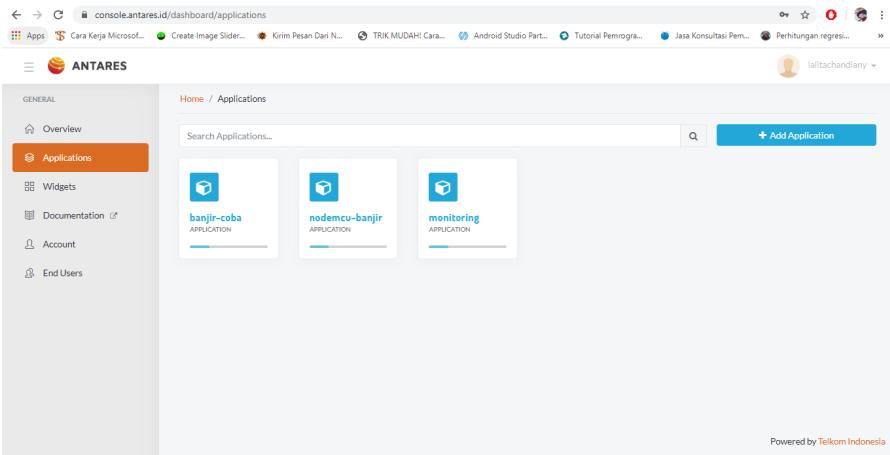
- Overview
- Application
- Widgets
- Documentation
- Account
- Enduser



Gambar 3.18 Halaman Overview

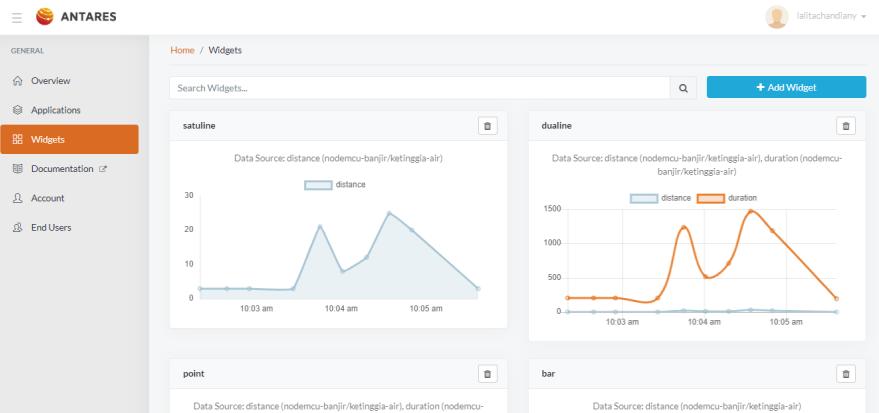
Halaman *Overview* merupakan halaman awal yang ditampilkan pada saat setelah login. Halaman ini menampilkan sebuah berita mengenai IoT (*Internet of*

Thing). Selain mengenai berita IoT halaman ini juga menampilkan project yang sudah kita buat. Adanya *application*, *device*, *API useage* dan *enduser*.



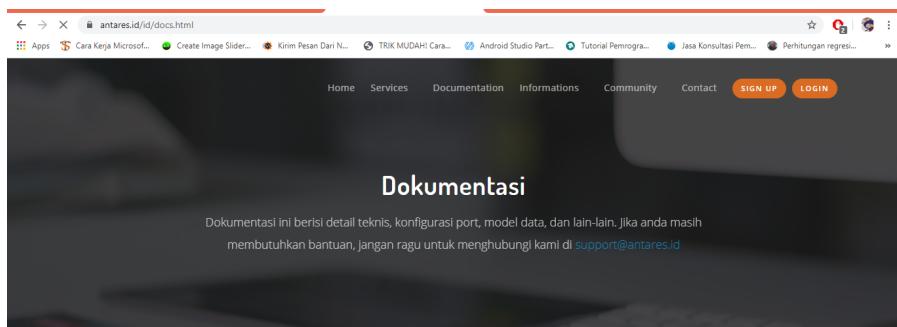
Gambar 3.19 Halaman Application

Pada halaman *application* menampilkan *project* yang telah dibuat. Kemudian pada halaman ini kita dapat membuat sebuah project baru.



Gambar 3.20 Halaman Widgets

Halaman widget merupakan sebuah halaman yang menampilkan sebuah grafik yang telah dibuat sebelumnya. Selain menampilkan sebuah grafik dalam halaman ini dapat membuat sebuah grafik baru.



Gambar 3.21 Halaman *Documentation*

This screenshot shows the 'Pendahuluan' section of the documentation. On the left, there's a sidebar with links: Pendahuluan, Registrasi, Buat App, and Tambah Device ke App. The main content area has a heading 'Pendahuluan' and text explaining that ANTARES is a Horizontal IoT Platform designed to make services accessible across various vertical IoT solutions. It mentions common IoT use cases like smart home, smart metering, and asset tracking.

Gambar 3.22 Halaman *Documentation*

Halaman *Documentation* ini merupakan sebuah halaman yang menampilkan tutorial atau penjelasan mengenai penggunaan antares.

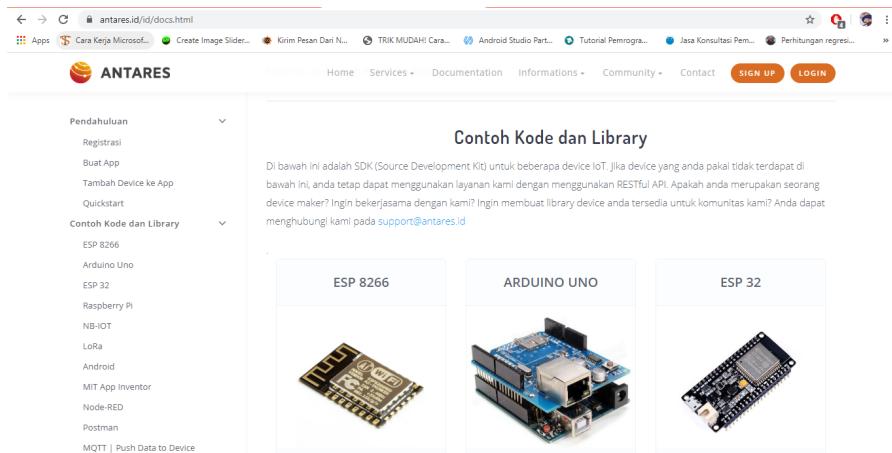
The screenshot shows a web browser window with the URL <https://antares.id/id/docs.html>. The page title is "ANTARES". The left sidebar has a tree view with categories like Pendahuluan, Registrasi, Buat App, Tambah Device ke App, Quickstart, Contoh Kode dan Library, and others. Under "Contoh Kode dan Library", there's a section for "ESP 8266" which contains the following C++ code:

```

3 #define ACCESSKEY "your-access-key"
4 #define WIFISSID "your-wifi-ssid"
5 #define PASSWORD "your-wifi-password"
6
7 #define projectName "your-project-name"
8 #define deviceName "your-device-name"
9
10 AntaresESP8266HTTP antares(ACCESSKEY);
11
12 void setup() {
13   Serial.begin(115200);
14   antares.beginDebug(true);
15   antares.wifiConnection(WIFISSID,PASSWORD);
16 }
17
18 void loop() {
19   int temp = random(25,30);
20   int hum = random(75,90);
21
22   antares.add("temperature", temp);
23   antares.add("humidity", hum);
24
25   antares.send(projectName, deviceName);
26   delay(10000);
27 }

```

Gambar 3.23 Halaman *Documentation Tutorial*



Gambar 3.24 Halaman *DocumentationTutorial*

Pada halaman *documentation* juga terdapat sebuah tutorial mengenai komponen yang dimana menggunakan antares sebagai media penyimpanan datanya. Kemudian pada halaman ini juga menampilkan sebuah contoh kode library yang bisa kita gunakan.

DAFTAR PUSTAKA
