

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
4 Membuat Aplikasi Monitoring	149

DAFTAR ISI

Daftar Gambar	xiii
Daftar Tabel	xxiii
Foreword	xxvii
Kata Pengantar	xxix
Acknowledgments	xxxi
Acronyms	xxxiii
Glossary	xxxv
List of Symbols	xxxvii
Introduction	xxxix
<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

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 Di Arduino IDE	89
3.2.3	Cara <i>Import Library</i>	91
3.3	Memprogram Komponen <i>Prorotype PKA</i>	95
3.3.1	Memprogram Sensor Ultasonic	95
3.3.2	Memprogram Led	97
3.3.3	Memprogram Buzzer	98
3.3.4	Memprogram Sensor, Led dan Buzzer	99
3.3.5	Memprogram Komponen dan Antares	101
4	Membuat Aplikasi Monitoring	149
4.1	Pengenalan Android Dan Android Studio	150
4.1.1	Android	150
4.1.2	Perkembangan Android	151
4.1.3	Android Studio	154
4.1.4	Struktur <i>Project</i>	155
4.1.5	Antarmuka Pengguna	156
4.1.6	Jendela Fitur	158
4.1.7	Pelengkapan kode	159

4.1.8	Navigasi	159
4.1.9	Gaya dan Pemformatan	160
4.1.10	<i>Sistem build Gradle</i>	160
4.1.11	Fitur Profil dan Debug	161
4.1.12	Anotasi di Android Studio	162
4.1.13	Pesan Log	162
4.1.14	<i>Interfaces</i> dan Struktur <i>Project</i> Android Studio	163
4.1.15	Struktur Folder Project Android Studio	167
4.1.16	Bahasa Pemograman Java	168
4.2	Membuat Aplikasi Monitoring <i>Prototype</i>	169
4.2.1	Membuat <i>Splash Screen</i>	169
4.2.2	Membuat Halaman Menu	175
4.2.3	Membuat Menu <i>About</i>	189
4.2.4	Membuat Menu Ketinggian Air	192
4.2.5	Membuat Menu Keterangan Air	199
4.2.6	Membuat Menu <i>Don't Panic</i>	201
4.3	Tampilan Aplikasi Monitoring	203
	Daftar Pustaka	209

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 Google	83
3.5	Tampilan Hasil Pencarian	83
3.6	Tampilan Cp210 Driver	84
3.7	Tampilan Cp210 Driver Dwonload	84

3.8	Tampilan Hasil Dwonload Driver	85
3.9	Tampilan File Hasil <i>extract</i>	85
3.10	Tampilan File Cp210 Driver	86
3.11	Tampilan Hasil Pencarian	86
3.12	Tampilan Hasil Pencarian	87
3.13	Tampilan Ch340x Driver Dwonload	87
3.14	Tampilan Hasil Dwonload Driver	88
3.15	Tampilan File Hasil <i>extract</i>	88
3.16	Tampilan File CH340x Driver	89
3.17	Tampilan Arduino IDE Blink	89
3.18	Tampilan Di Serial Monitor	91
3.19	Memilih Menu Manage Librarie	92
3.20	Tampilan Sketch Manager	93
3.21	Tampilan Library Yang Terinstall	93
3.22	Tampilan Library ADD ZIP	94
3.23	Pilih File ZIP	94
3.24	Tampilan Library Berhasil Ditambahkan	95
3.25	Tampilan serial monitor	97
3.26	Halaman Awal Antares	101
3.27	Tampilan Registrasi Antares	102
3.28	Verifikasi email	103
3.29	Login Antares	103
3.30	Tampilan Awal Antares	104
3.31	Halaman <i>Overview</i>	104
3.32	Halaman <i>Aplication</i>	105
3.33	Halaman <i>Widgets</i>	105
3.34	Halaman <i>Documentation</i>	106
3.35	Halaman <i>Documentation</i>	106

3.36	Halaman <i>Documentation Tutorial</i>	107
3.37	Halaman <i>DocumentationTutorial</i>	107
3.38	Halaman <i>Account</i>	108
3.39	Halaman <i>Enduser</i>	108
3.40	Halaman <i>DocumentationTutorial</i>	109
3.41	Dwownload Library	109
3.42	Dwownload Library di Github	110
3.43	Dwownload Library	110
3.44	Halaman <i>DocumentationTutorial</i>	111
3.45	Tampilan di Github	111
3.46	Tampilan <i>content</i>	112
3.47	Tampilan Using Boards Manager	112
3.48	Tampilan Arduino IDE	113
3.49	Tampilan Arduino IDE	114
3.50	Tampilan Preferences	115
3.51	Tampilan Arduino IDE Boards Mnager	115
3.52	Tampilan Boards Manager	116
3.53	Tampilan Boards Manager Antares	116
3.54	Tampilan Arduino IDE Include Library	117
3.55	Tampilan Library Manager	118
3.56	Tampilan Library Manager Antares	118
3.57	Tampilan Arduino IDE Manage Library	119
3.58	Tampilan Arduino IDE Library Antares	120
3.59	Tampilan Antares	121
3.60	Tampilan Login Antares	121
3.61	Tampilan Awal Antares	122
3.62	Tampilan Application Antares	122
3.63	Tampilan Membuat Application Antares	123

3.64	Tampilan Application Yang Dibuat	123
3.65	Tampilan Membuat Device Antares	124
3.66	Tampilan Membuat Device Antares	124
3.67	Tampilan Device Di Application Antares	125
3.68	Tampilan Halaman Device Antares	125
3.69	Tampilan Antares	126
3.70	Tampilan Users Antares	127
3.71	Verifikasi Arduiono IDE	130
3.72	Upload Arduiono IDE	131
3.73	Upload Arduiono IDE	132
3.74	Halaman Utama Antares	132
3.75	Tampilan <i>Application</i>	133
3.76	Tampilan <i>Project</i>	133
3.77	Tampilan Data Sensor Ultrasonic	134
3.78	Tampilan Telegram	135
3.79	Tampilan Bot Telegram	136
3.80	Tampilan Bot Telegram	137
3.81	Tampilan Bot Telegram	138
3.82	Tampilan IDBot Telegram	139
3.83	Tampilan IDBot Telegram	140
3.84	Tampilan IDBot Telegram	141
3.85	Tampilan Bot Monitoring Di Telegram	142
3.86	Tampilan Dwonload CTbot	142
3.87	Tampilan Dwonload Arduino Json	143
3.88	Tampilan Notifikasi DI Telegram	147
4.1	Logo Android	150
4.2	Perkembangan Android	151
4.3	Logo Android Studio	154

4.4	File project Dalam Tampilan Android	155
4.5	File Project Dalam Tampilan Problems	156
4.6	Jendela utama Android Studio	157
4.7	Pintasan keyboard Ke Beberapa Jendela Alat yang Berguna.	158
4.8	Pintasan keyboard untuk Pelengkapan Kode.	159
4.9	Kode Setelah Pemformatan	160
4.10	Nilai Variabel Inline	161
4.11	Hasil pemeriksaan Lint di Android Studio	162
4.12	<i>Interfaces</i> Android Studio	163
4.13	Menu Bar Android Studio	163
4.14	Tool Bar Android Studio	163
4.15	Struktur <i>Project</i> Android Studio	164
4.16	Pallet Android Studio	164
4.17	Design Android Studio	165
4.18	Tree Android Studio	165
4.19	Properties Android Studio	166
4.20	Status Bar Android Studio	167
4.21	Struktur Folder Projectr Android Studio	167
4.22	Tampilan Awal Android Studio	169
4.23	Tampilan Memilih Activity	170
4.24	Membuat Activity Spalsh Activity	170
4.25	Konfigurasi Project	171
4.26	Konfigurasi Project	172
4.27	Proses Gradle	172
4.28	Build Gradle	173
4.29	Workspace	173
4.30	Drawable	174
4.31	Library Slider	175

4.32	Activity Baru	176
4.33	Tampilan Slider	177
4.34	Library Cardview	178
4.35	Library Antares	192
4.36	Cara Masuk ke Folder Libs	192
4.37	Tampilan Folder Libs	193
4.38	Tampilan Aplikasi DI Emulator	203
4.39	Splash Screen	204
4.40	Tampilan Menu Utama	204
4.41	Tampilan Menu About	205
4.42	Tampilan Ketinggian Air	205
4.43	Tampilan Menu Keterangan Air	206
4.44	Tampilan Menu <i>Don't Panic</i>	206
4.45	Tampilan Notifikasi Telegram	207

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	90
src/sensor.c	95
src/led.c	97
src/buzzer.c	98

src/gabung.c	99
src/nodemcuantares.c	126
src/nodemcuantares2.c	126
src/nodemcuantares3.c	127
src/nodemcuantares4.c	127
src/nodemcuantares5.c	128
src/nodemcuantares6.c	128
src/nodemcuantares7.c	128
src/bot.c	143
src/bot2.c	143
src/bot3.c	143
src/bot4.c	143
src/bot5.c	144
src/bot6.c	144
src/bot7.c	144
src/splash.java	173
src/tampilansplash.xml	175
src/slider.java	176
src/tampilanslider.xml	177
src/menu.java	178
src/homefull.java	179
src/tampilanmenu.xml	181
src/menufull.xml	185
src/about.java	189
src/aboutfull.xml	190
src/manifest.java	193
src/monitoring.java	194
src/tampilanmon.xml	197
src/air.java	199
src/tampilanair.xml	199
src/panic.java	201
src/tampilanpanic.xml	202

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.

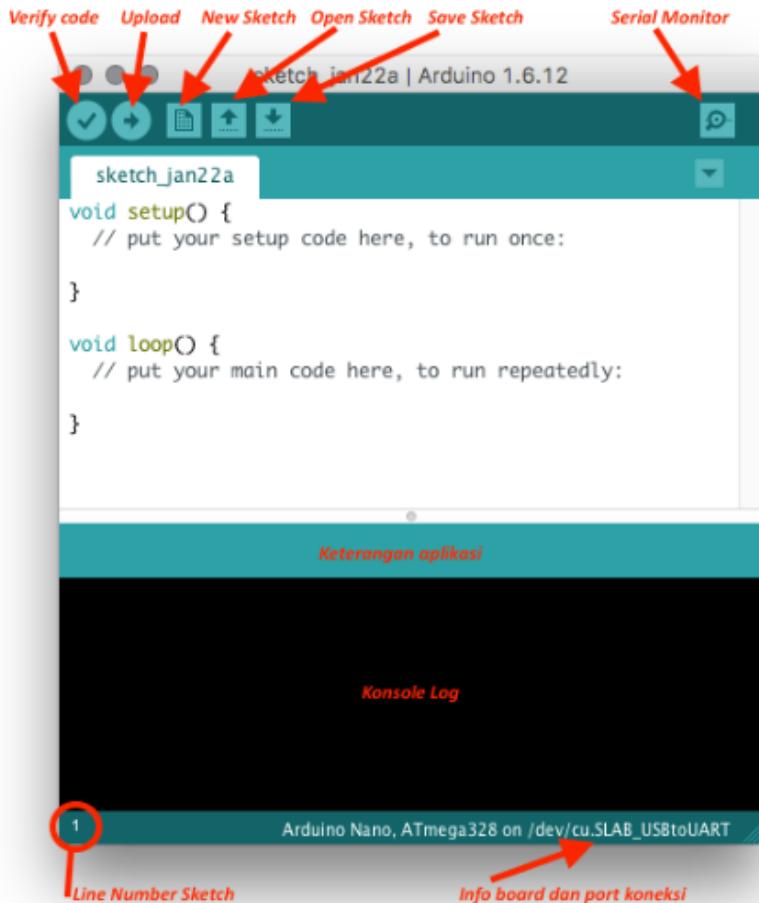
Dalam Software Arduino IDE ini dapat memprogram kode tidak hanya *microcontroller* arduino saja, tetapi pada software ini dapat digunakan untuk memprogram *microcontroller* yang lain salah satu contohnya yaitu Nodemcu.

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 *inisialisai 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 *inisialisai 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

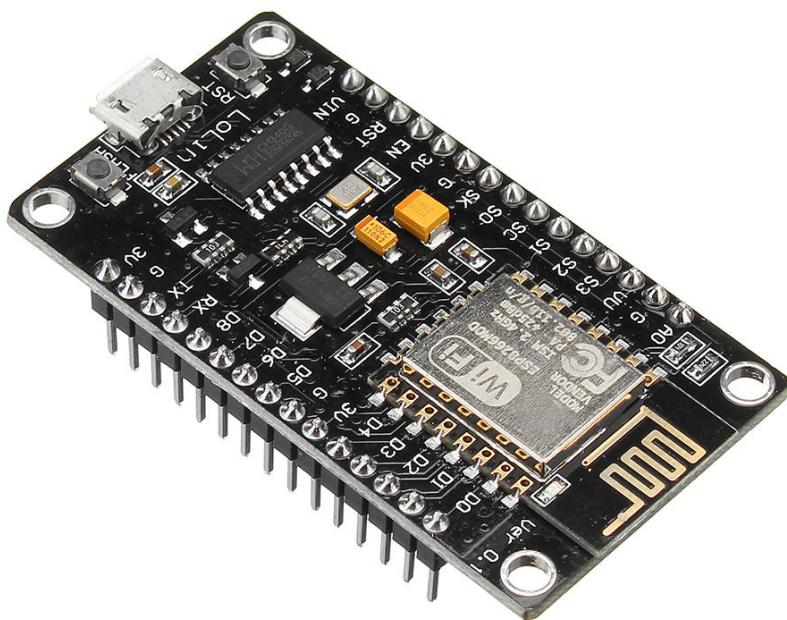
NodeMCU adalah ESP8266 (khususnya seri ESP-12, termasuk ESP-12E) maka fitur fitur yang dimiliki NodeMCU akan kurang lebih sama ESP-12 (juga ESP-12E untuk NodeMCU v.2 dan v.3) kecuali NodeMCU telah dibungkus oleh API sendiri yang dibangun berdasarkan bahasa pemrograman eLua, yang kurang lebih cukup mirip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



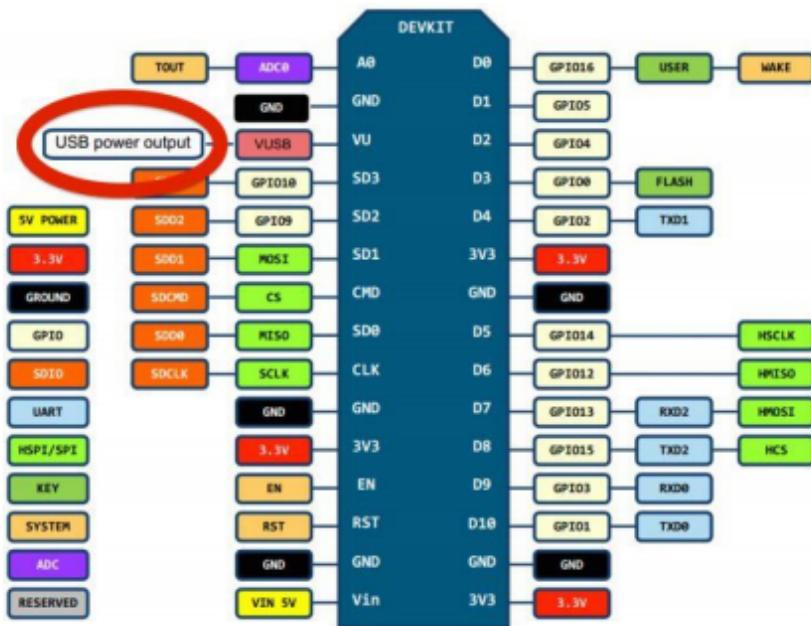
Gambar 2.5 NodeMCU

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.

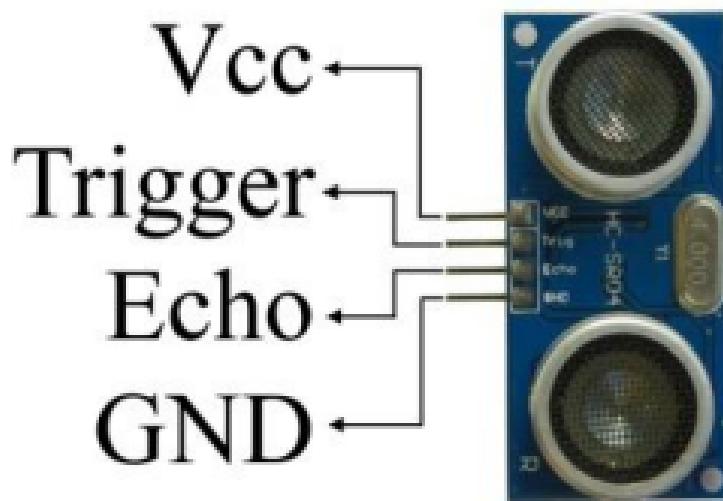
Pada protokol HTTP terdapat 3 jenis hubungan dengan perantara proxy, gateway, dan tunnel. Proxy bertindak sebagai agent penerus, menerima request dalam bentuk Uniform Resource Identifier (URI) absolut, mengubah format request dan mengirimkan request ke server yang ditunjukan oleh URI. Gateway bertindak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

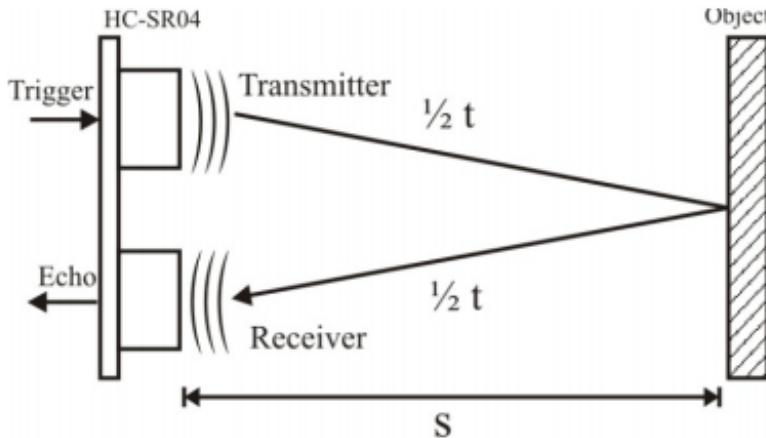
Sensor jarak ultrasonic HC-SR04 adalah sensor 40 KHz. HC-SR04 merupakan sensor ultrasonik yang dapat digunakan untuk mengukur jarak antara penghalang dan sensor. Konfigurasi pin dan tampilan sensor HC-SR04 diperlihatkan pada gambar dibawah :



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 :



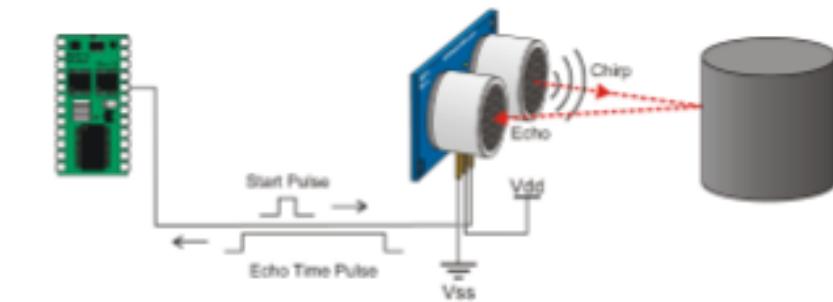
Gambar 2.12 Gambar Sensor Ultrasonic US 100

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

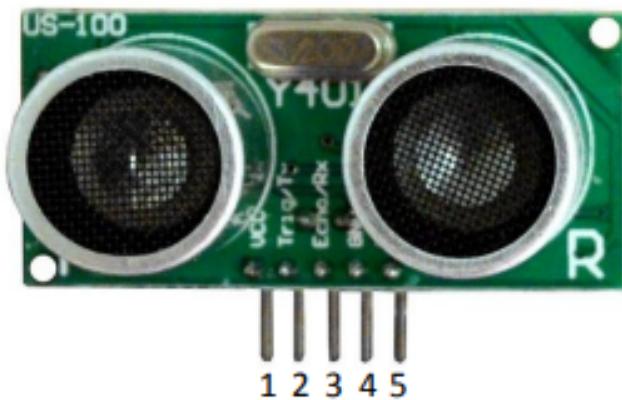
Pemancar ultrasonik mentransmisikan gelombang ultrasonik, gelombang ini bergerak di udara dan ketika ia keberatan dengan bahan apa pun itu dipantulkan kembali ke sensor gelombang. Untuk menghitung jarak menggunakan rumus di atas, harus mengetahui kecepatan dan waktu. Karena menggunakan gelombang Ultrasonik,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

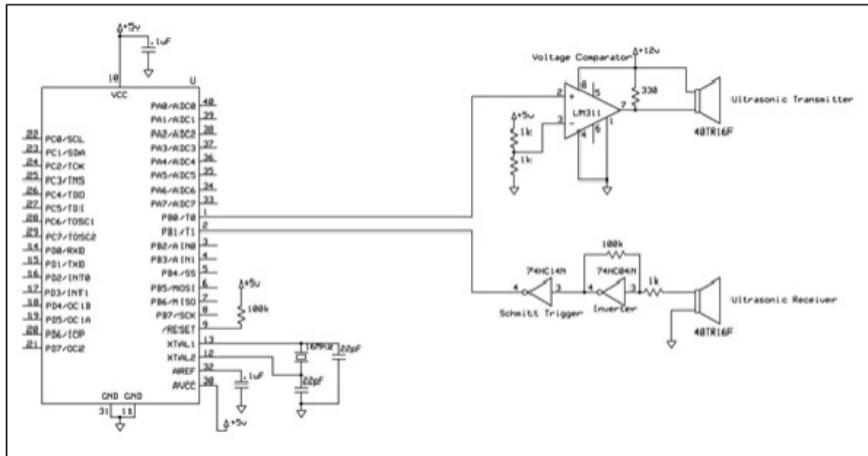
Adapun pin konfigurasi dari sensor ultrasonic Us 100 yaitu :



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 ultrasonik US 100

8. Cara *Test Sensor Ultrasonic US 100*

Komponen yang akan digunakan adalah:

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

Setelah hal diatas sudah terpenuhi maka ikuti langkah verikut ini :

- 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.
- Setelah koneksi perangkat keras, masukkan sketsa sampel ke dalam Arduino IDE.
- Menggunakan kabel USB, sambungkan porta dari mikrokontroler ke komputer.
- Upload* program.
- 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*.

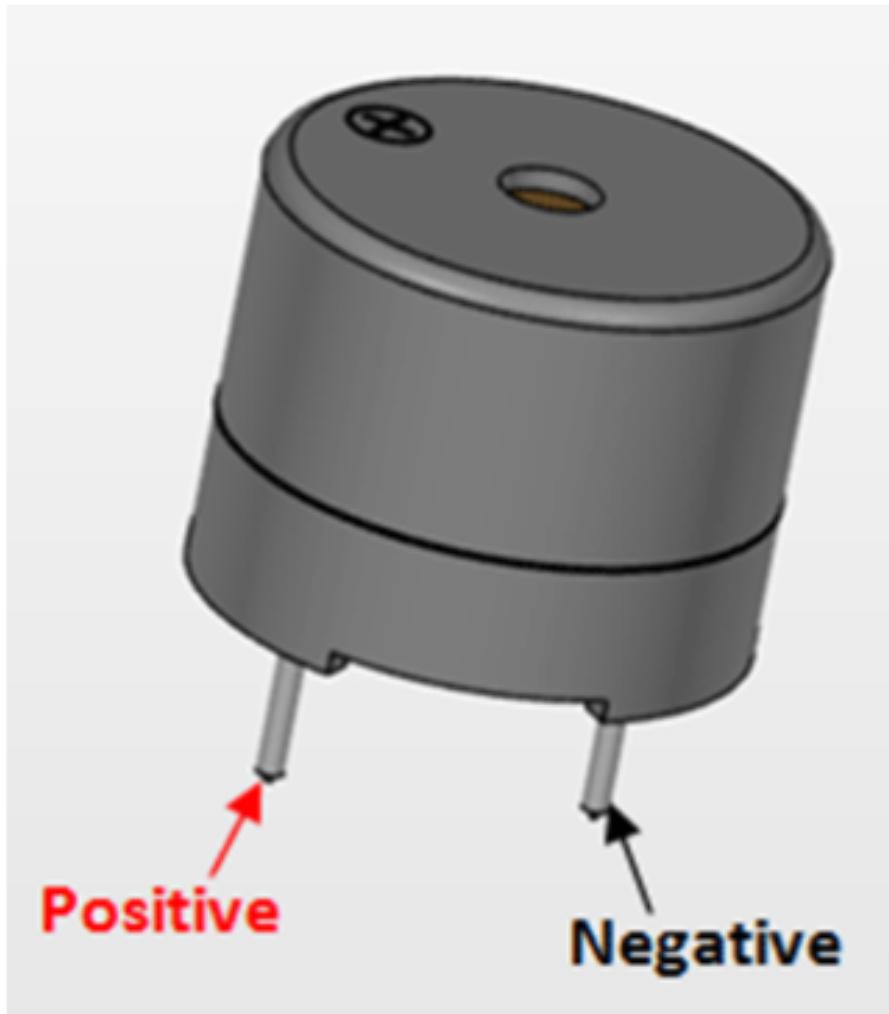


Gambar 2.17 Buzzer

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

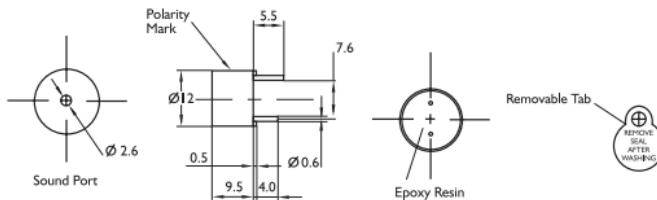
Buzzer adalah komponen kecil namun efisien untuk menambahkan fitur suara ke proyek / sistem. Ini adalah struktur 2-pin yang sangat kecil dan kompak sehingga dapat dengan mudah digunakan pada Breadboard dan bahkan pada PCB yang menjadikannya komponen yang banyak digunakan dalam sebagian besar aplikasi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

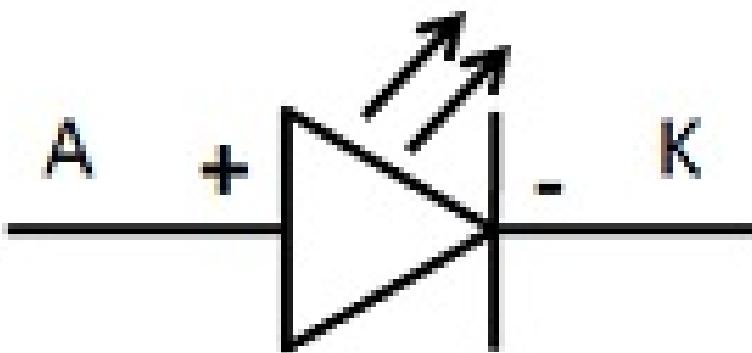
Gambar 2.20 Diagram model 2D Buzzer

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



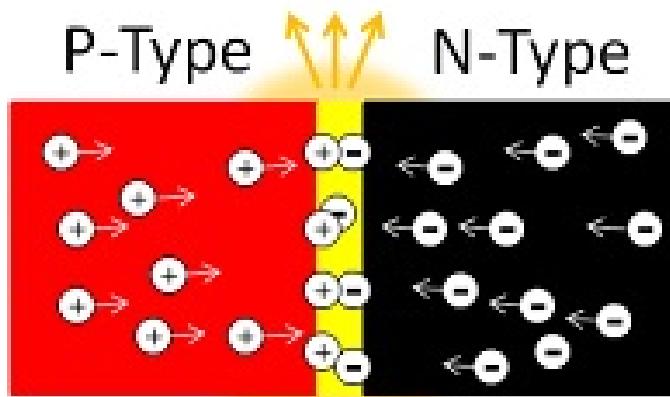
Gambar 2.22 Simbol Led

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.

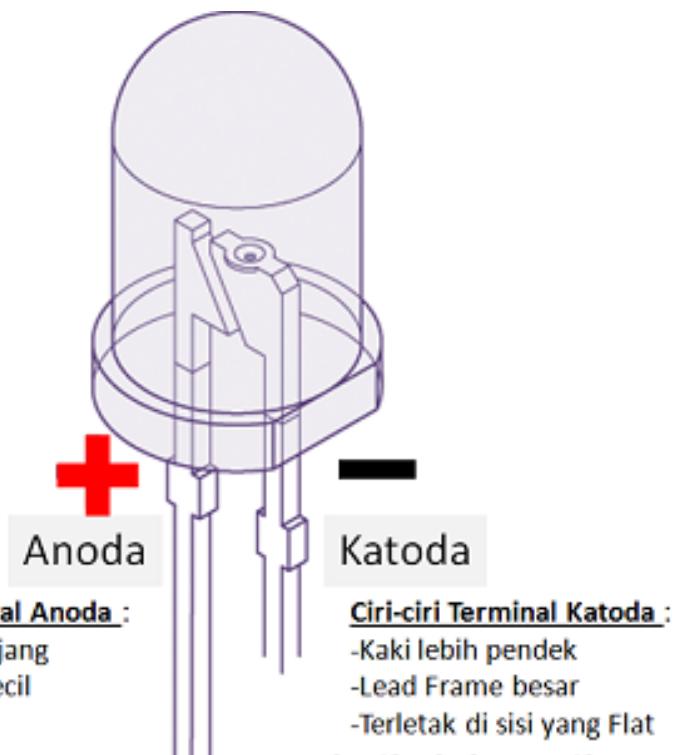
LED terdiri dari sebuah chip semikonduktor yang di doping sehingga menciptakan junction P dan N. Yang dimaksud dengan proses doping dalam semikonduktor adalah proses untuk menambahkan ketidakmurnian (impurity) pada semikonduktor yang murni sehingga menghasilkan karakteristik kelistrikan yang diinginkan. 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

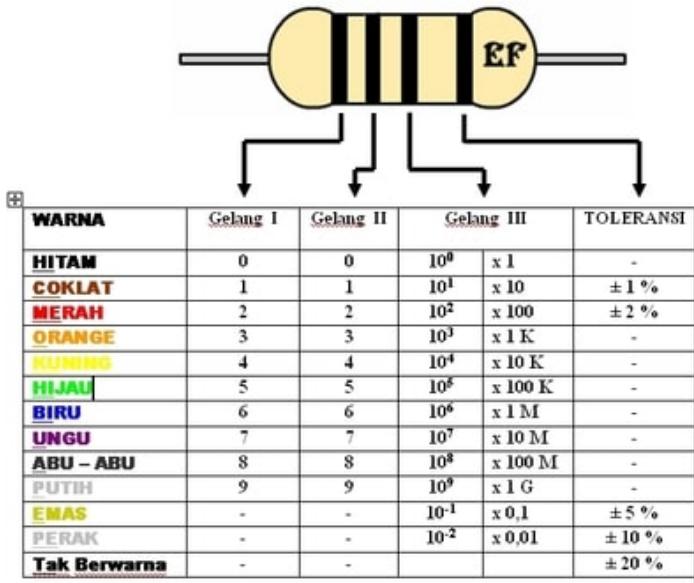
Satuan Resistor adalah Ohm (simbol: Ω) yang merupakan satuan SI untuk resistansi listrik. Dalam sejarah, kata ohm itu diambil dari nama salah seorang fisikawan hebat asal German bernama George Simon Ohm. Beliau juga yang menceetuskan keberadaan hukum ohm yang masih berlaku hingga sekarang.

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



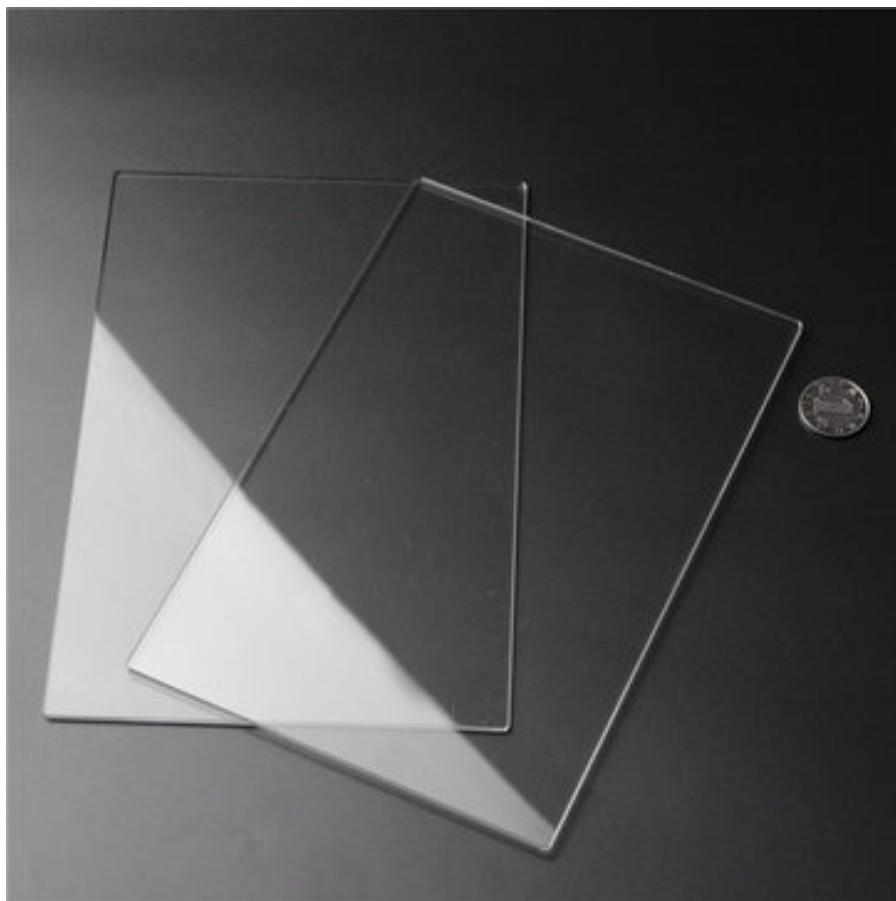
Gambar 2.30 Logo Telegram

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

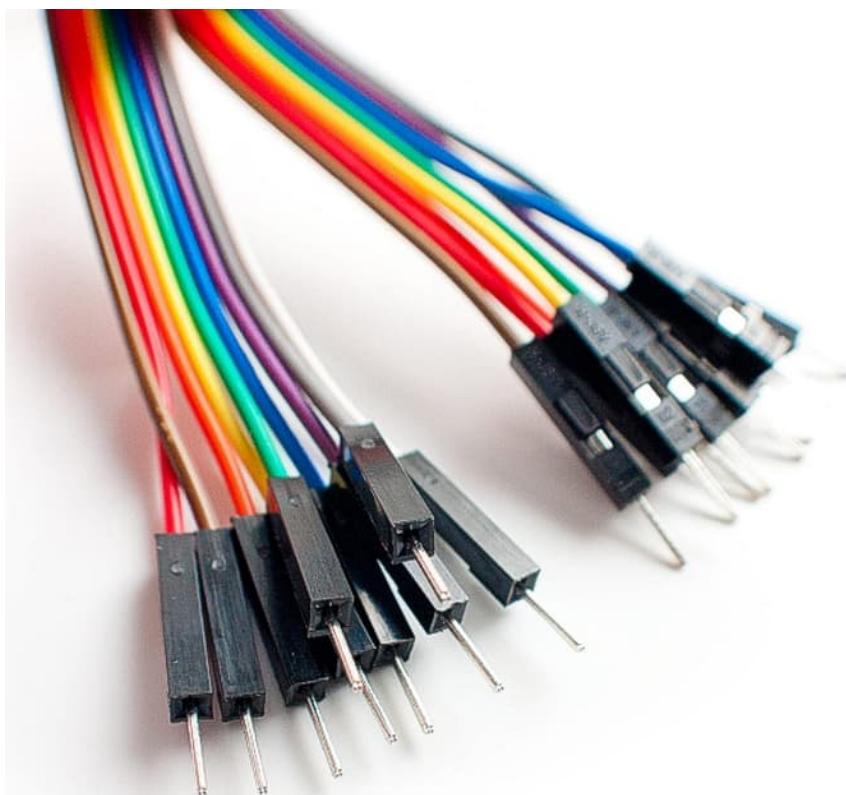


Gambar 2.32 Kabel Jumper

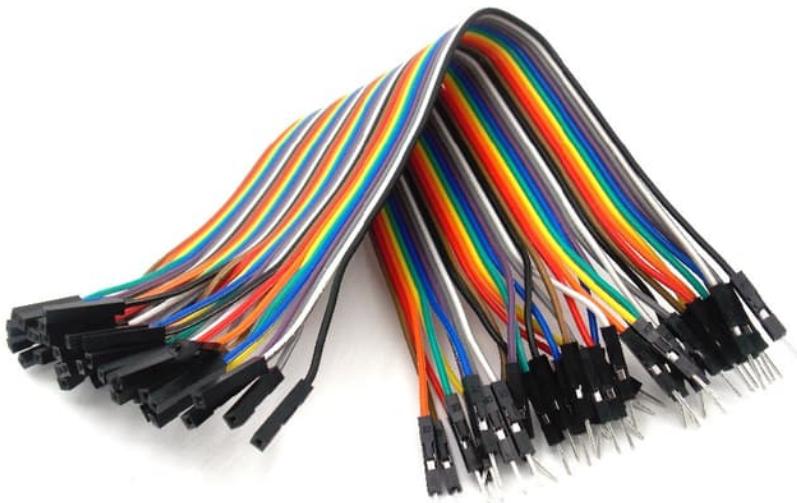
Kabel Jumper digunakan untuk menghubungkan komponen satu dengan komponen lainnya. Selain itu kabel jumper ini juga untuk menghubungkan komponen dengan *microcontroller*. Ada beberapa jenis kabel jumper yaitu sebagai berikut :



Gambar 2.33 Kabel Jumper Famale



Gambar 2.34 Kabel Jumper Male

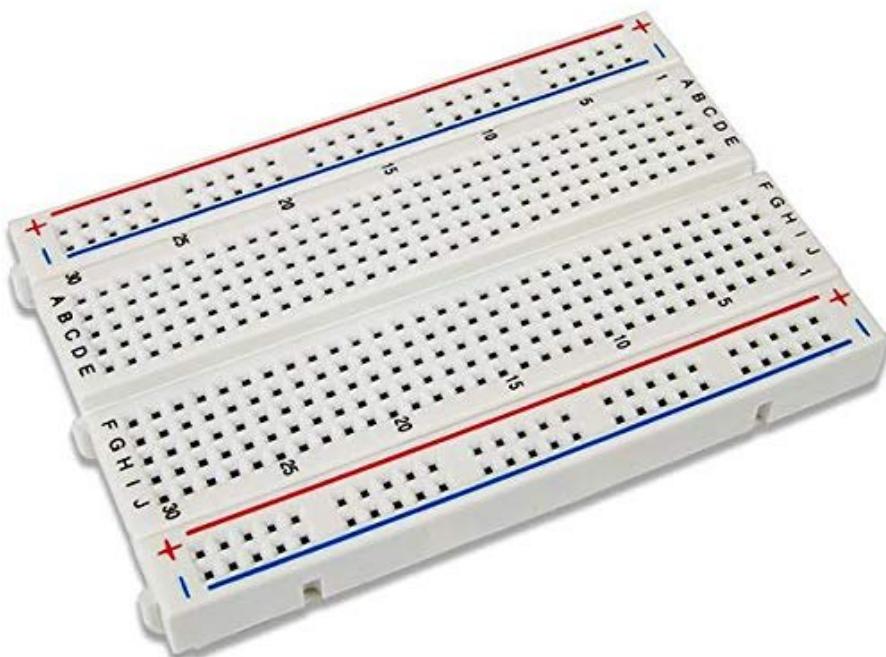


Gambar 2.35 Kabel Jumper Male-Female

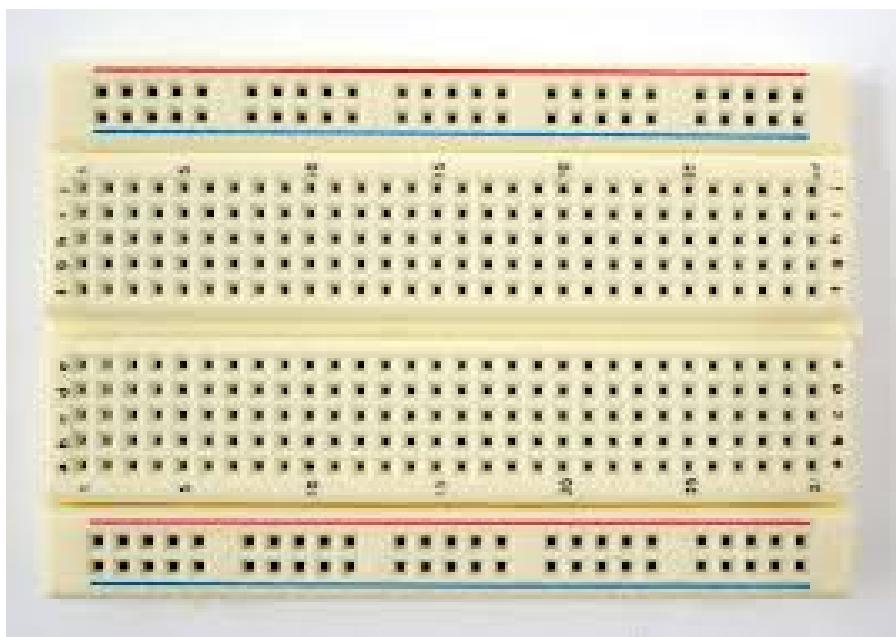
25. BreadBoard

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.

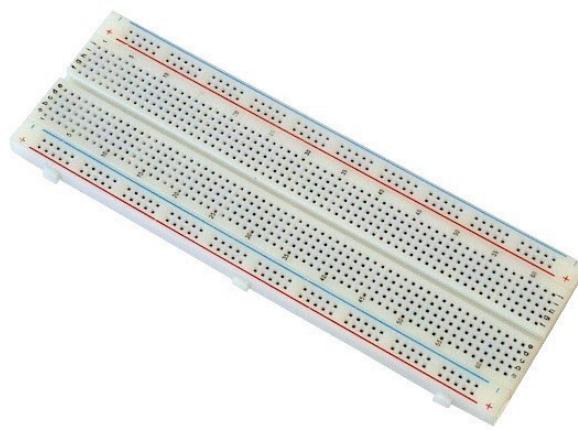
Breadboard yang tersedia di pasaran umumnya terbagi atas 3 ukuran: mini breadboard, medium breadboard atau large breadboard. Mini breadboard memiliki 170 titik koneksi (bisa juga lebih). Kemudian medium breadboard memiliki 400 titik koneksi. Dan large breadboard memiliki 830 titik koneksi.



Gambar 2.36 Breadboard Mini



Gambar 2.37 Breadboard Medium

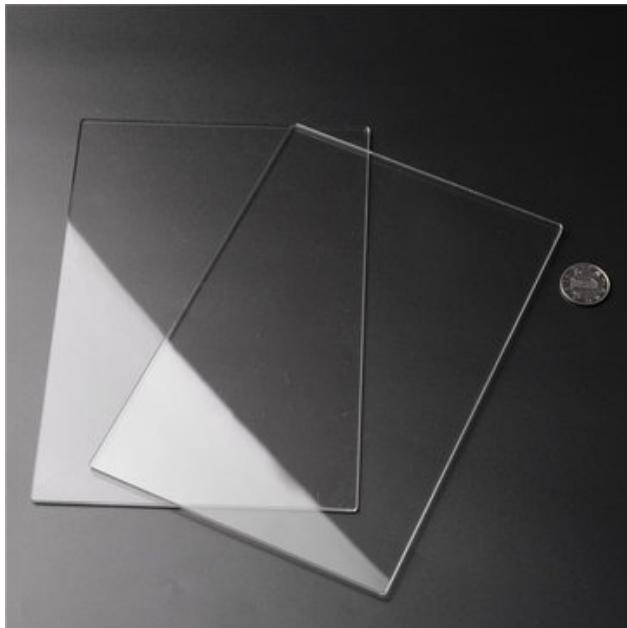


Gambar 2.38 Breadboard Large

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 .

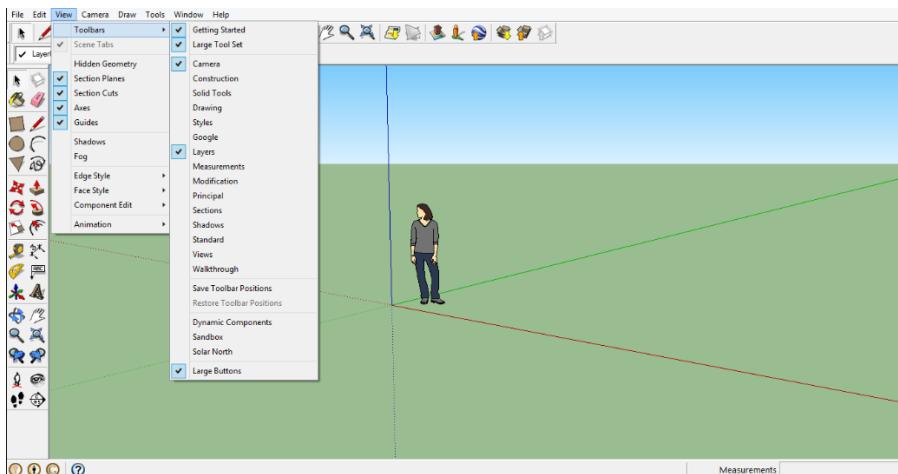
Untuk membuat sebuah desain dari rangka prototipe ini ada beberapa software untuk mendesain yaitu :

1. SketchUp Make



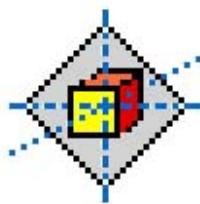
Gambar 2.45 SketchUp Make

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

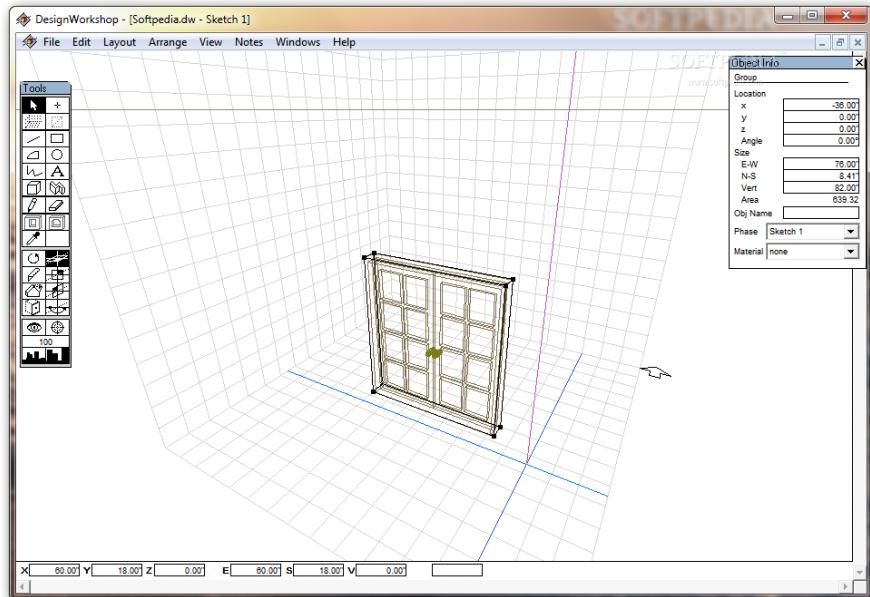
2. DesignWorkshop Lite



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.



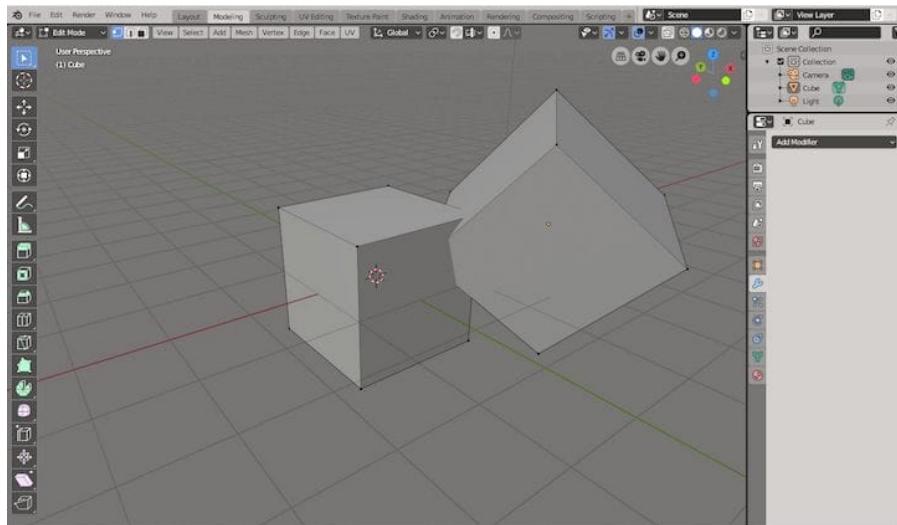
Gambar 2.48 Tampilan DesignWorkshop Lite

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

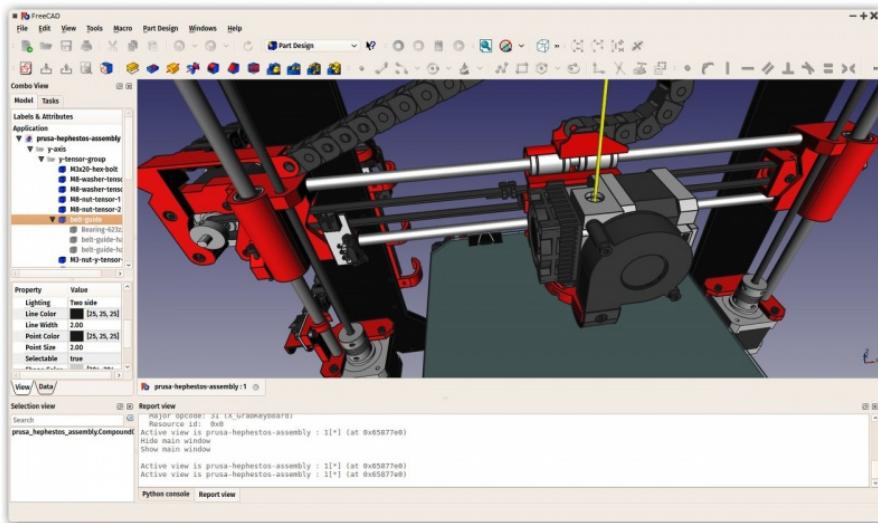
4. Free CAD



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

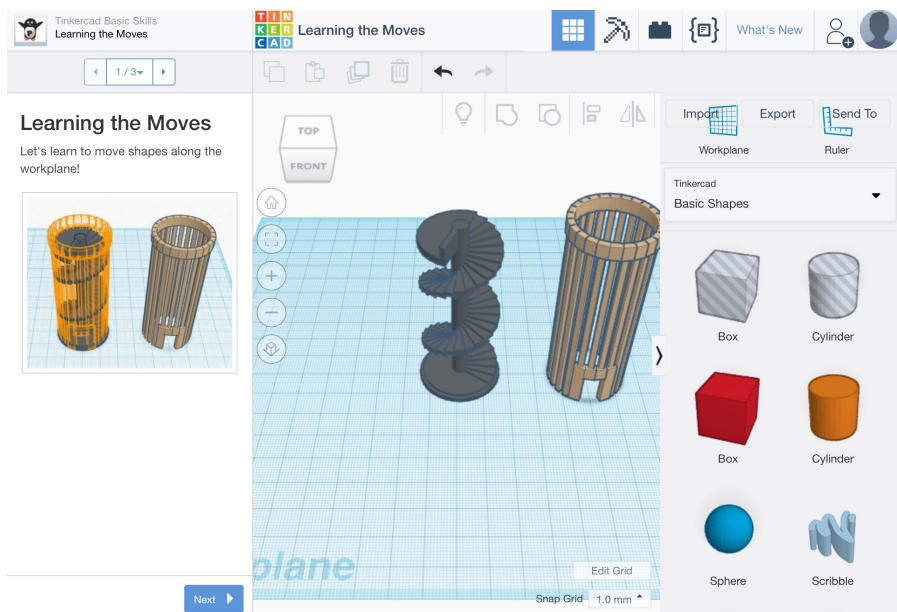
5. TinkerCAD



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.



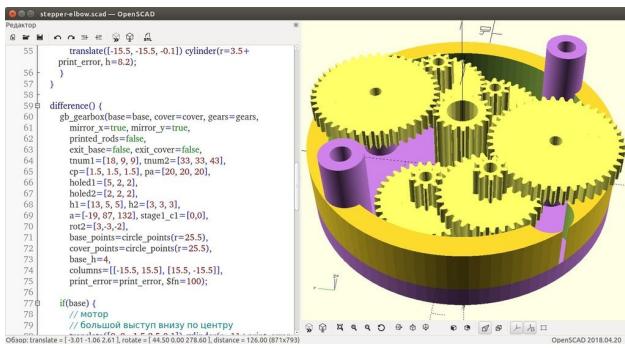
Gambar 2.54 Tampilan TinkerCAD



Gambar 2.55 OpenSCAD

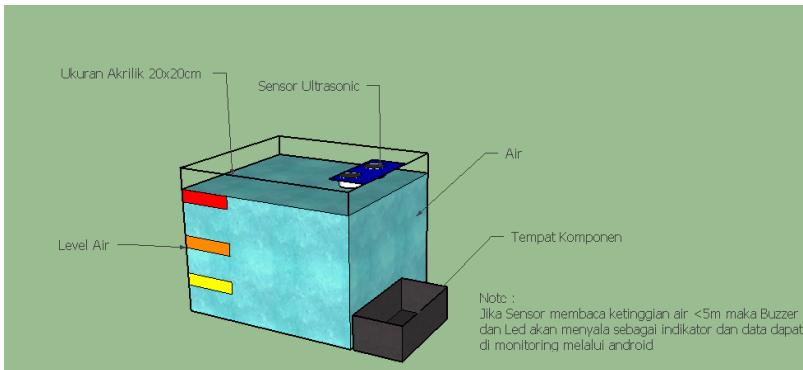
OpenSCAD adalah aplikasi software 3D modeling gratis, tapi bukanlah software yang menarik, dengan cara penggunaan yang berbeda dari software lain. Software OpenSCAD merupakan kompilator 3D berdasarkan bahasa deskripsi tekstual.

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

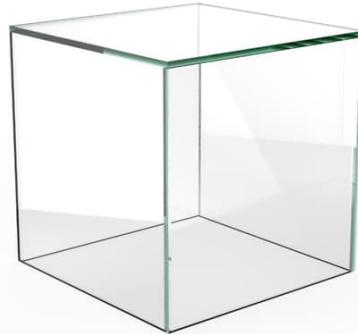
Setelah pembahasan diatas mengenai software pembuatan desain 3D, maka kita dapat memilih software yang akan digunakan . Untuk pembuatan desain prototipe ini menggunakan sketchup make. Mengapa menggunakan sketchup make? karena sketchup make mudah untuk dipahami dan softwarenya gratis. Berikut desain untuk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

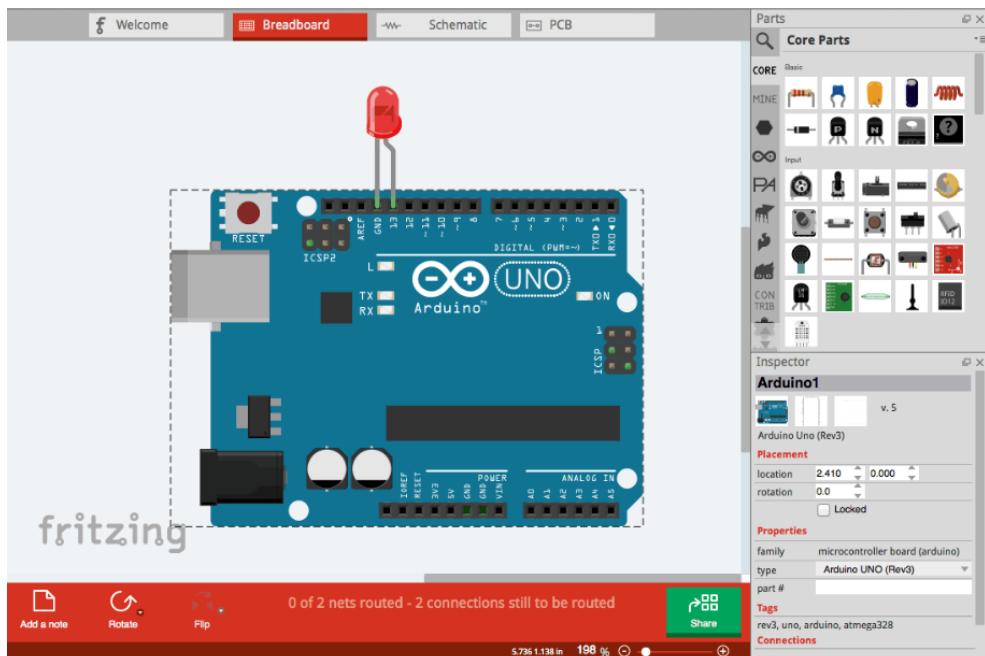
2.8.1 Perakitan Komponen

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

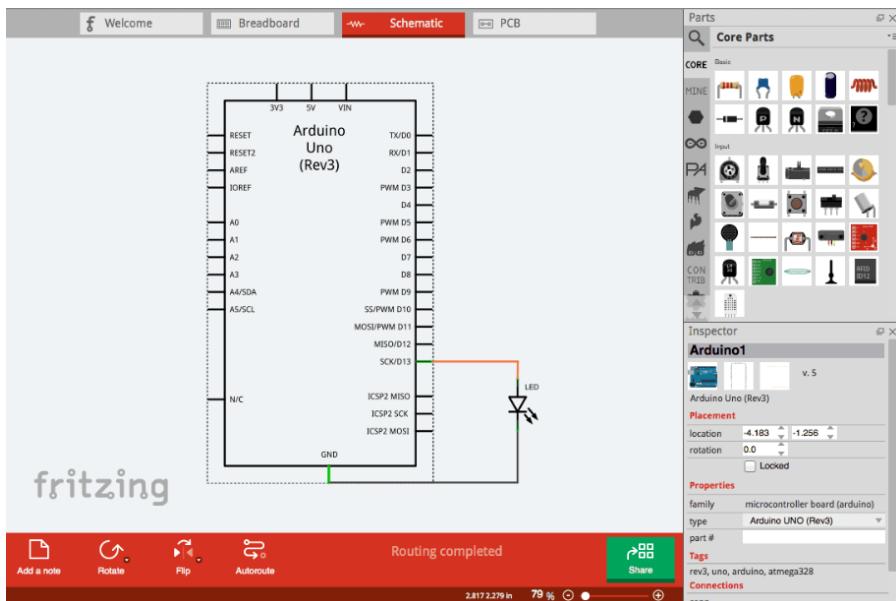
misalkan kelebihan daya, arus, short, dll. Selain itu tidak semua komponen ataupun perangkat elektronika dapat dijumpai dan dibeli dengan mudah, salah satu contohnya adalah osiloskop, spektrum analizer, function generator, dll.

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 :

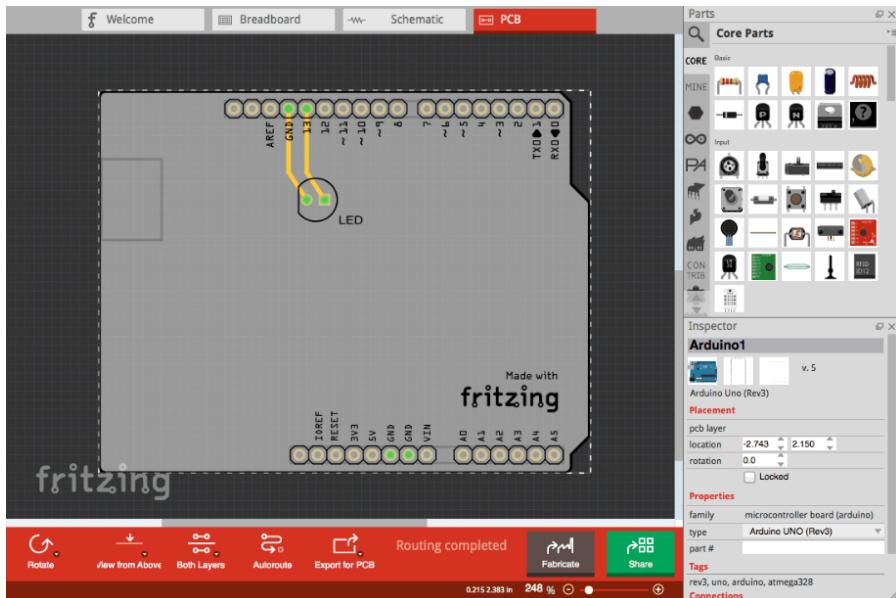
1. Fritzing merupakan aplikasi berbasis free software yang dapat kita unduh secara gratis. Aplikasi ini berfungsi untuk membuat sebuah skema rangkaian elektronika secara nyata (real). Karena komponen yang berada pada rangkaian tersebut digambarkan dengan sangat mirip dengan komponen aslinya. Aplikasi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

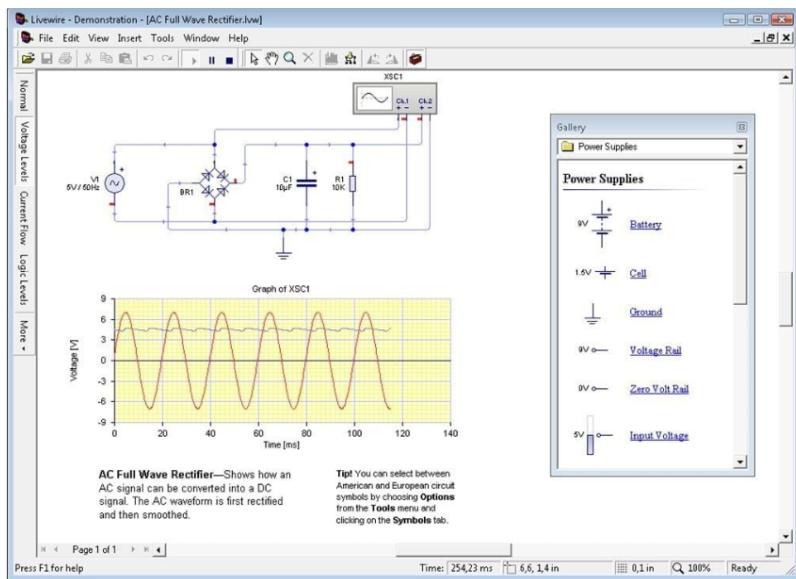


Gambar 2.61 Tampilan Fritzing

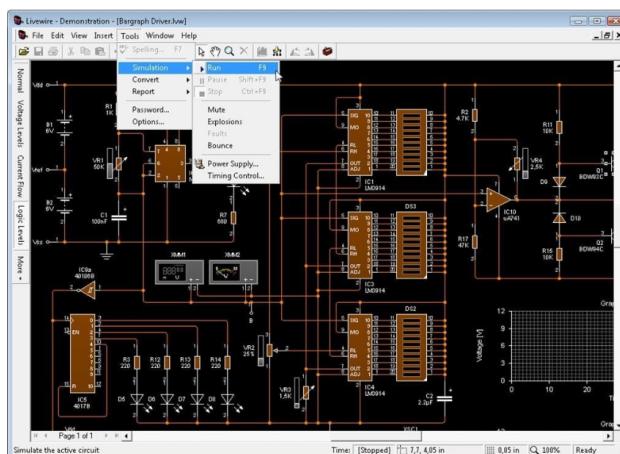


Gambar 2.62 Tampilan Fritzing

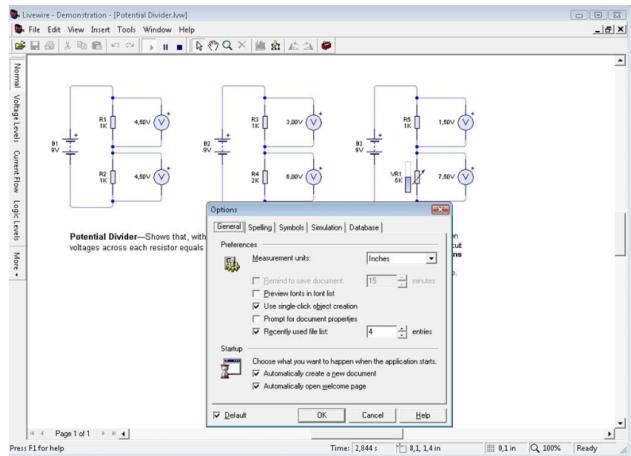
2. LiveWire LiveWire merupakan aplikasi simulasi rangkaian elektronika yang mudah digunakan, khususnya bagi pemula yang baru belajar elektronika. Dalam aplikasi ini sudah disediakan komponen-komponen dasar seperti resistor, kapasitor, op amp, dll. Dengan begitu kita dapat merangkai skema yang akan kita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



Gambar 2.63 Tampilan LiveWire

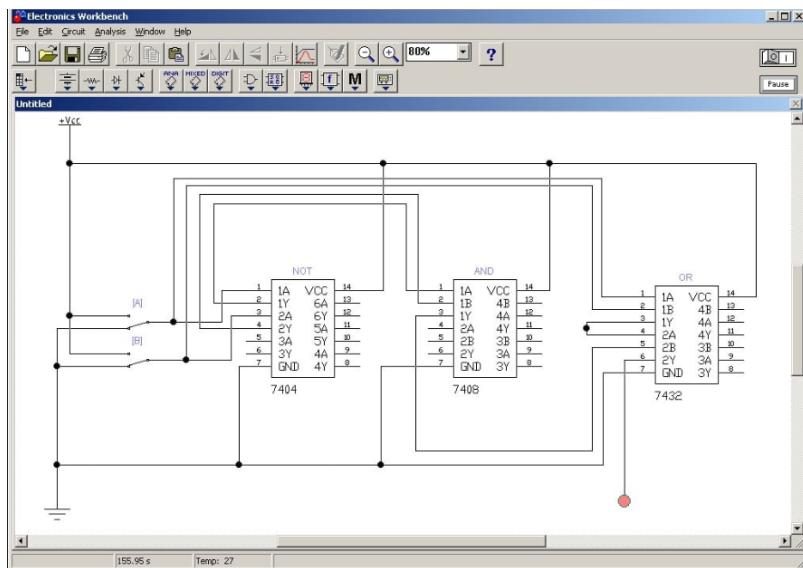


Gambar 2.64 Tampilan LiveWire

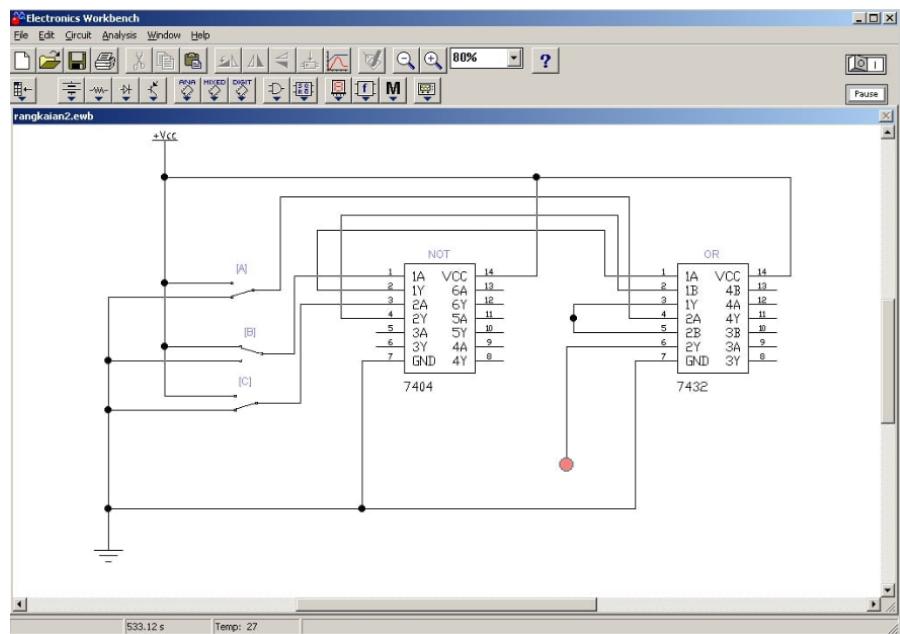


Gambar 2.65 Tampilan LiveWire

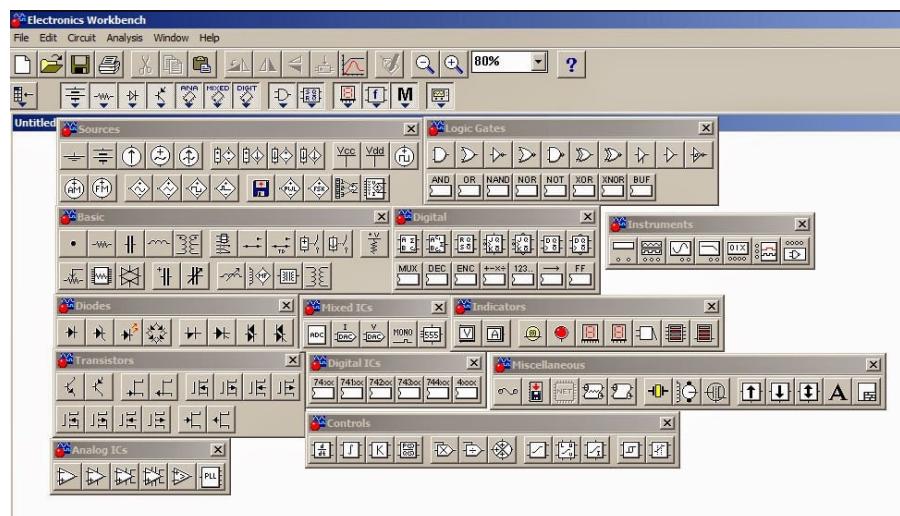
3. Electronics Workbench (EWB) *Electronics Workbench* atau yang sering dikenal dengan EWB merupakan aplikasi yang tidak jauh berbeda dengan LiveWire. Aplikasi ini sangat mudah untuk digunakan, ditambah dengan beberapa feature yang lebih lengkap seperti adanya simulasi osiloskop, spektrum analizer, dan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



Gambar 2.66 Tampilan *Electronics Workbench* (EWB)



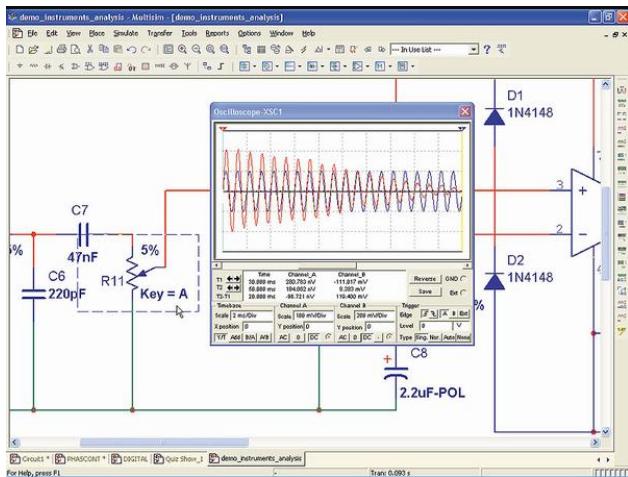
Gambar 2.67 Tampilan *Electronics Workbench* (EWB)



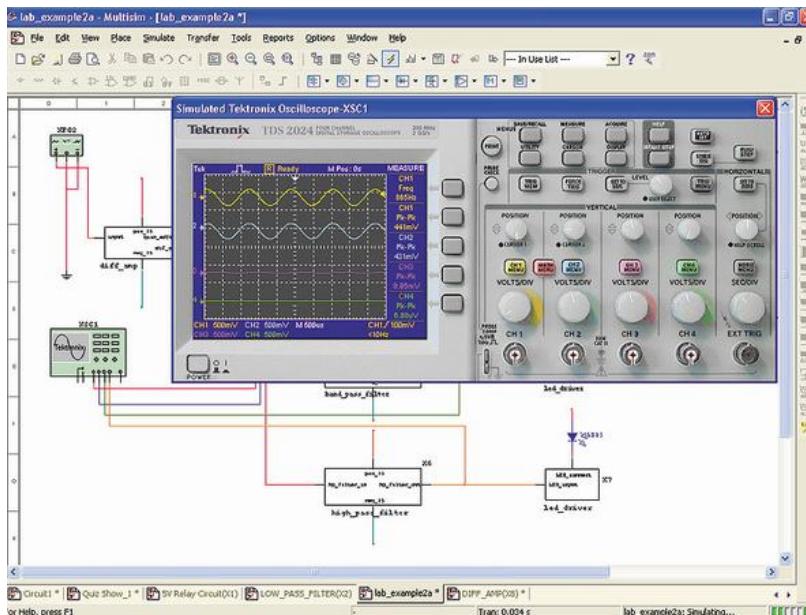
Gambar 2.68 Tampilan *Electronics Workbench* (EWB)

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

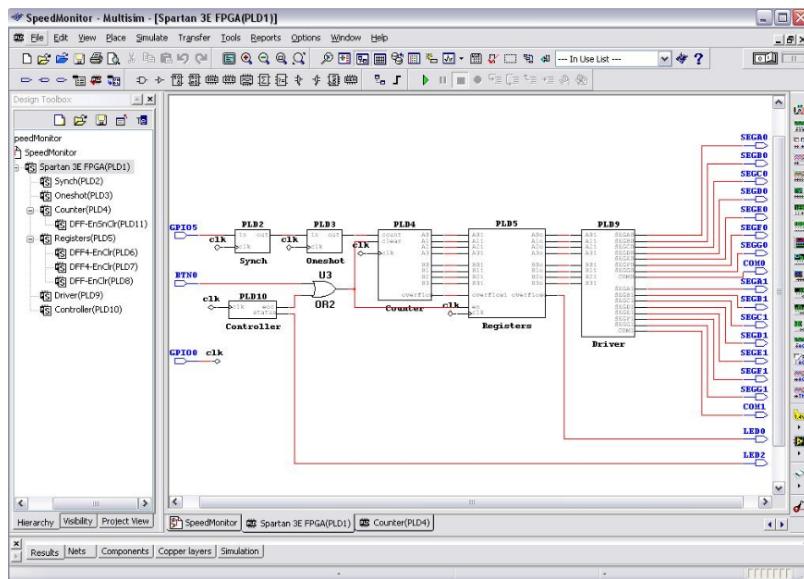
dari versi EWB yang sebelumnya. Aplikasi ini dilengkapi dengan komponen-komponen yang memiliki fungsi yang sama dengan aslinya, begitupun dengan perangkat elektronika seperti osiloskop, function generator, spektrum analizer, dll.



Gambar 2.69 Tampilan NI Multisim



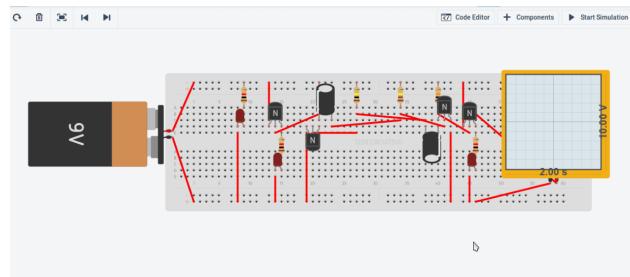
Gambar 2.70 Tampilan NI Multisim



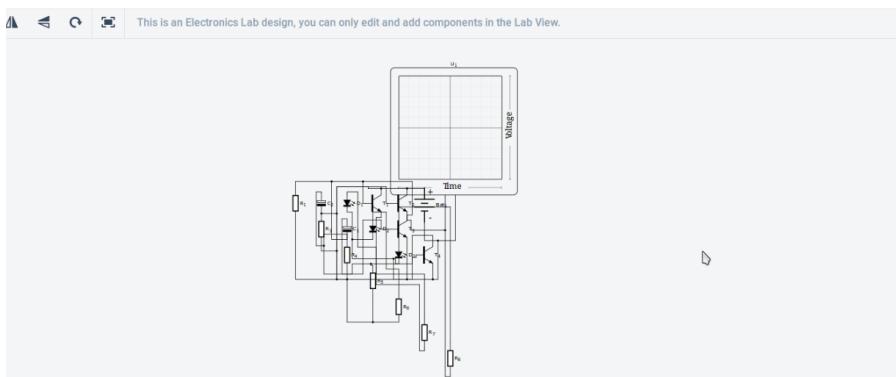
Gambar 2.71 Tampilan NI Multisim

5. Autodesk Circuits

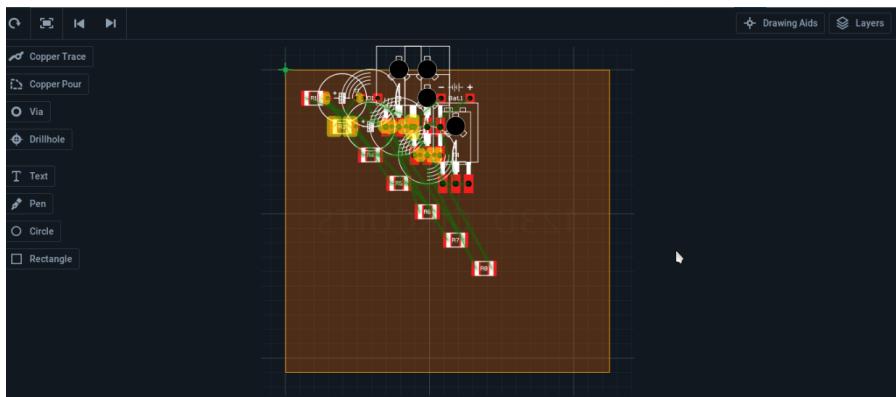
Autodesk Circuits merupakan aplikasi berbasis online, jadi apabila ingin menggunakan aplikasi ini, kita harus terhubung pada jaringan internet terlebih dahulu. Untuk menggunakannya kamu dapat mengunjungi halaman Autodesk Circuits berikut. Fitur yang dimiliki oleh aplikasi ini memang beragam diantaranya 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



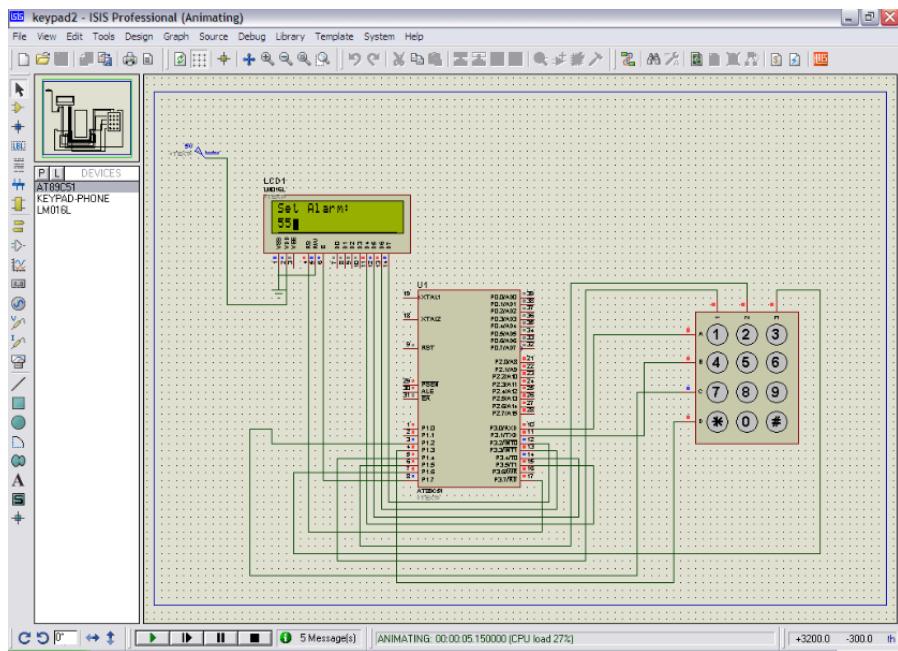
Gambar 2.73 Tampilan Autodesk Circuits



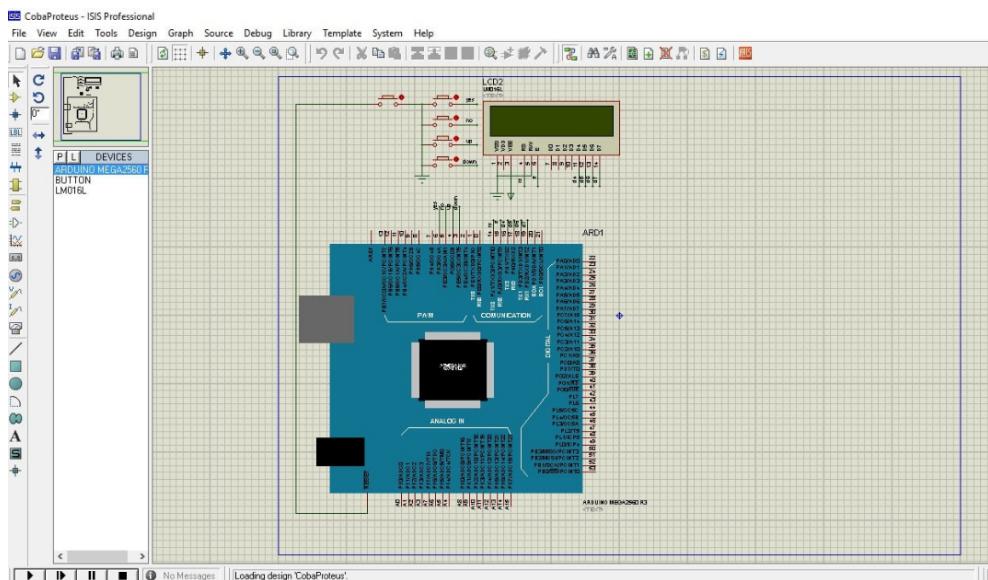
Gambar 2.74 Tampilan Autodesk Circuits

6. Proteus ISIS

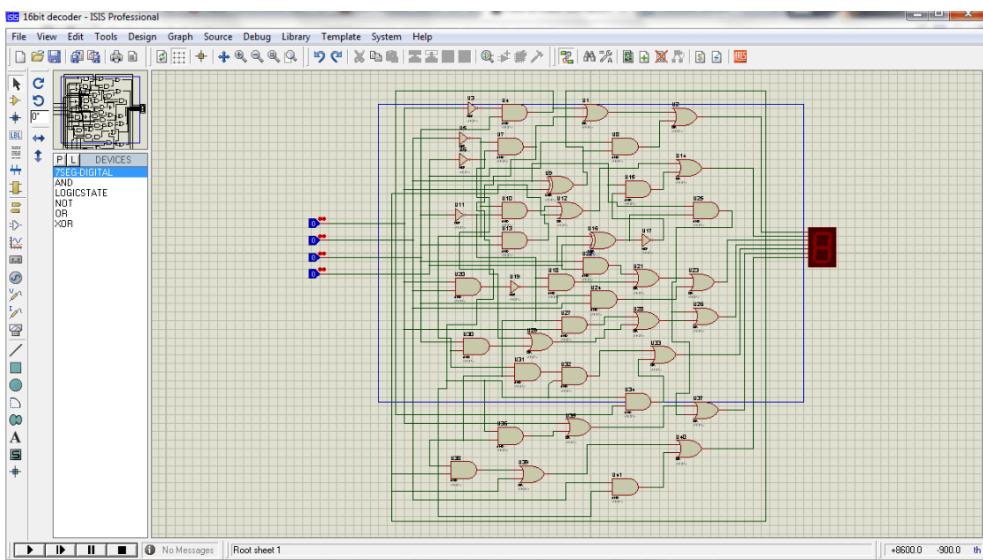
Proteus ISIS merupakan aplikasi yang saat ini menjadi yang terbaik dan paling direkomendasikan, selain dari library komponen yang lengkap dan bahkan bisa kita tambahkan, aplikasi ini juga dapat mensimulasikan semua rangkaian yang kita buat, aplikasi ini dilengkapi dengan text editor yang bisa kita gunakan 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

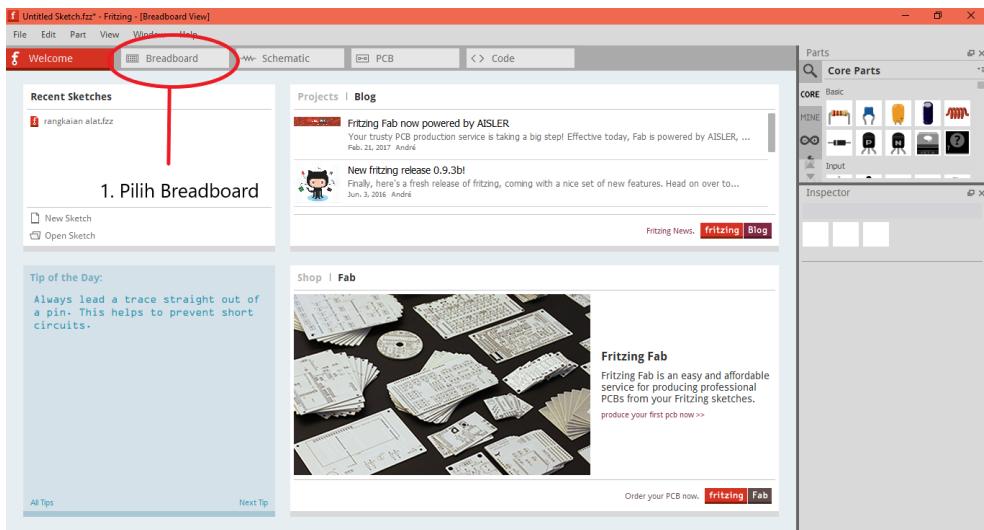
Dengan adanya penjelasan tentang software pembuatan skema komponen, kita dapat memilih software untuk pembuatan komponen sesuai yang kita inginkan. Tetapi kita juga harus memilih software yang akan digunakan sesuai kebutuhan.

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.

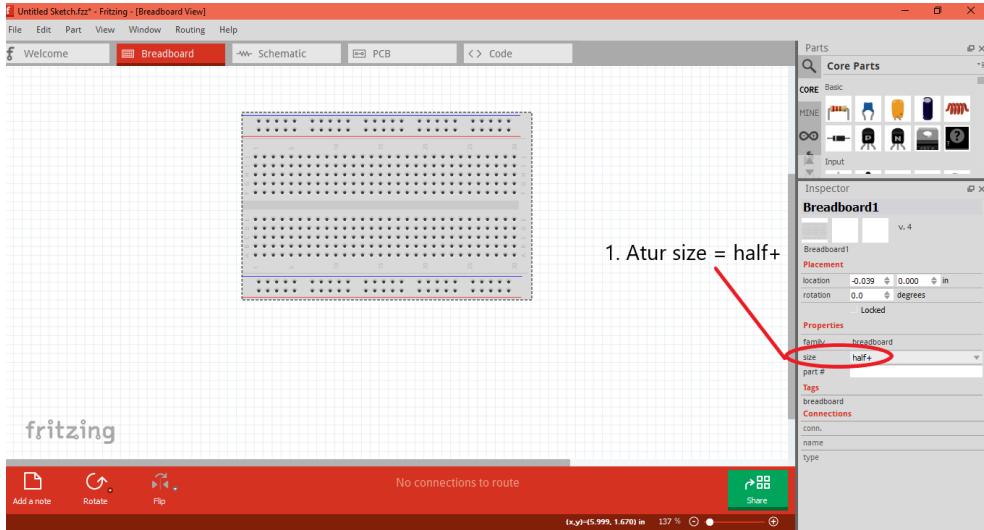
Berikut langkah-langkah untuk membuat skema prototipe prediksi ketinggian air (PKA) untuk mendeteksi banjir peringatan dini menggunakan software frizting :

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



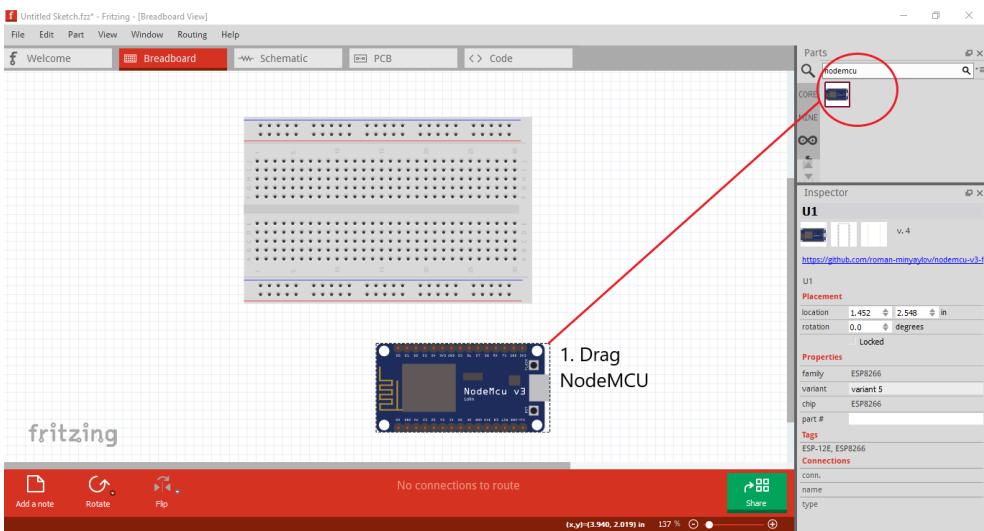
Gambar 2.78 Tampilan Awal Fritzing

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 :



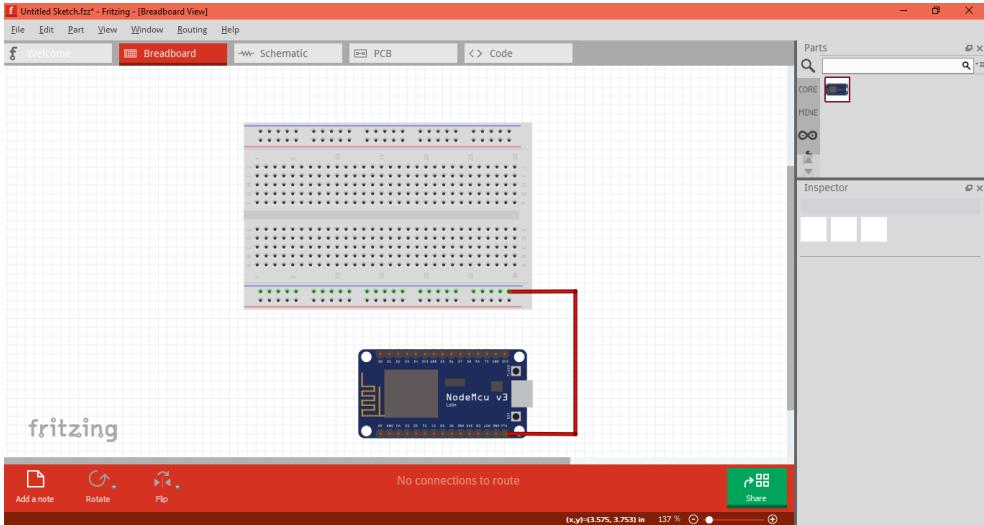
Gambar 2.79 Ubah Ukuran Breadboard

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



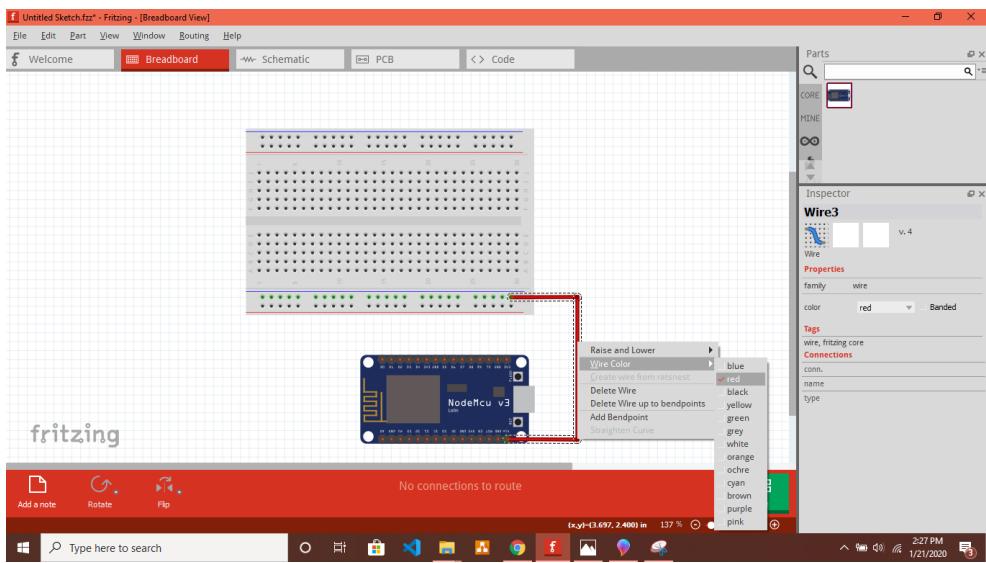
Gambar 2.80 Memasukan NodeMCU

- 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*.



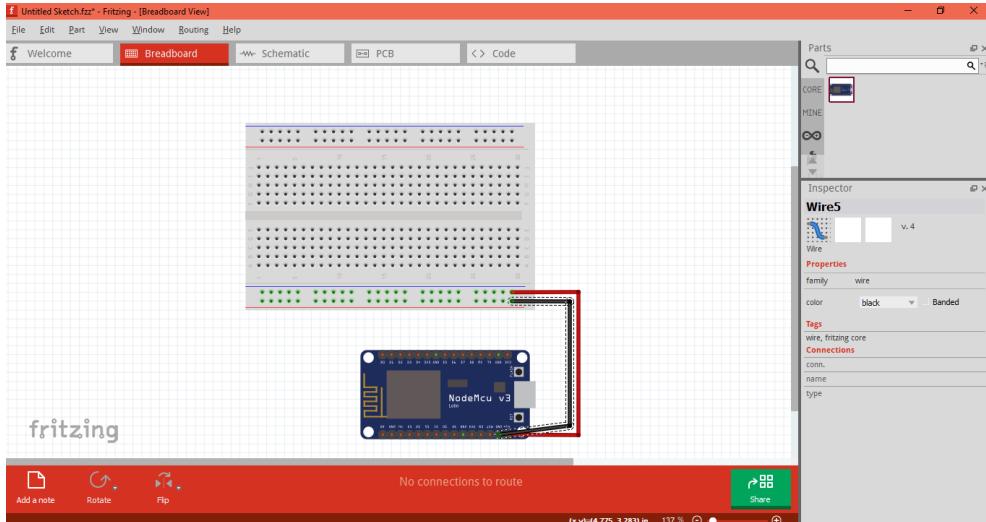
Gambar 2.81 Sambungkan Vin ke Breadboard

- 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

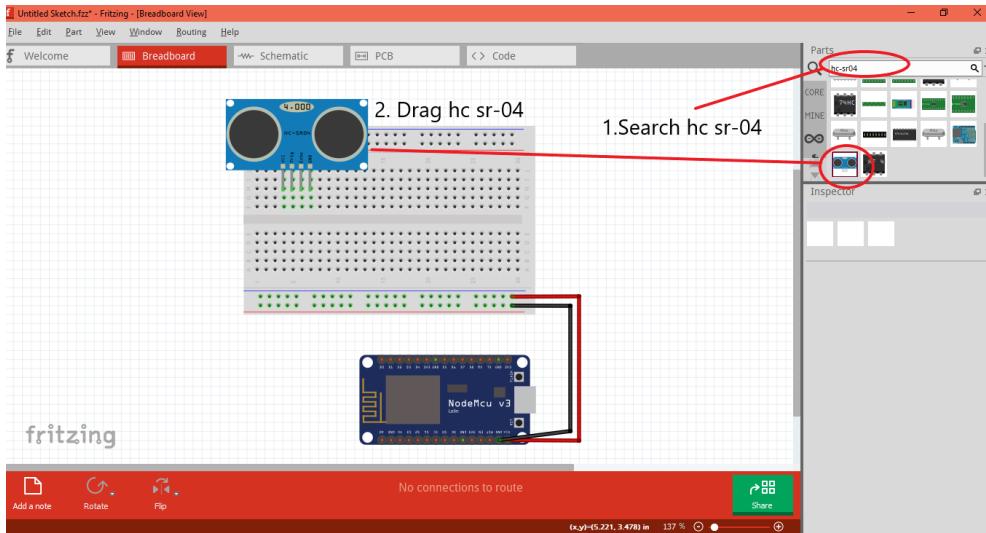
6. Sambungkan GND pada bagian negatif (-) *breadboard* dengan cara klik dua kali pada bagian Vin maka akan muncul *wire* dan arahkan ke bagian negatif *breadboard*.



Gambar 2.83 Sambungkan GND ke *Breadboard*

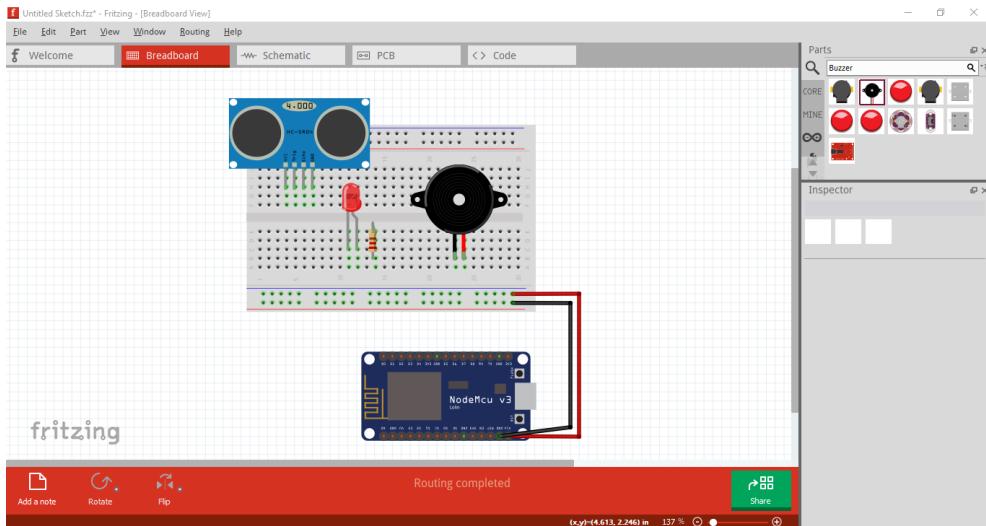
Untuk mengubah warnanya sama seperti langkah sebelumnya.

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



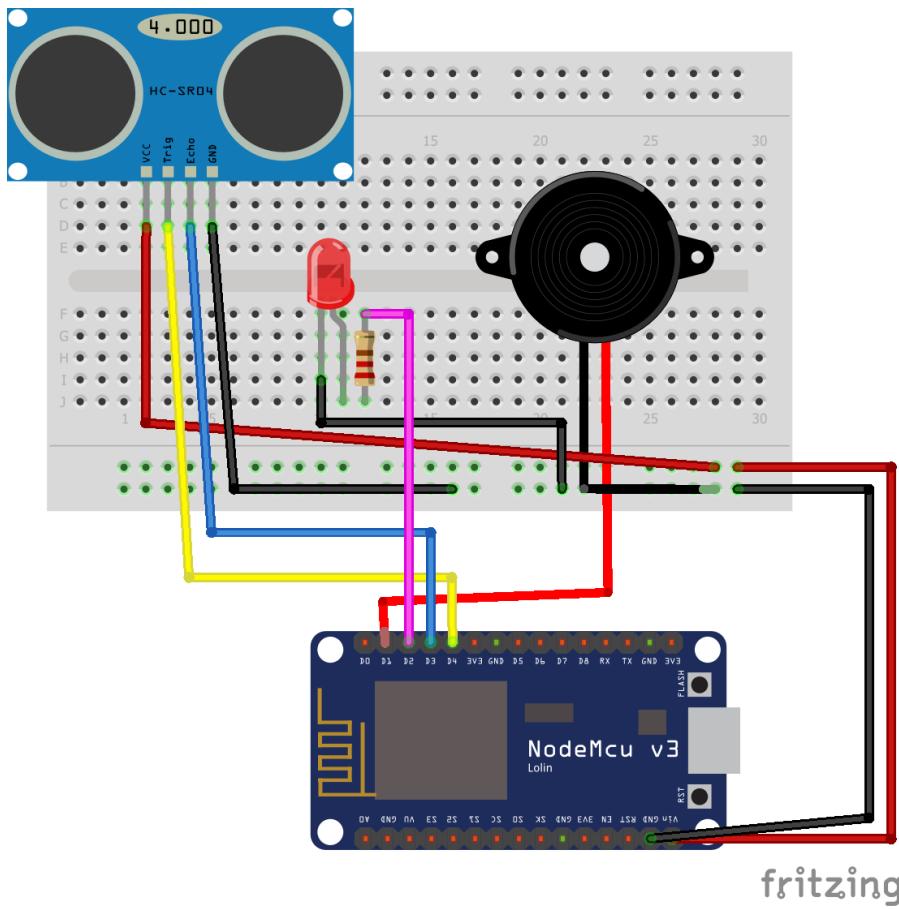
Gambar 2.84 Menambahkan Komponen Sensor Ultrasonic

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 .



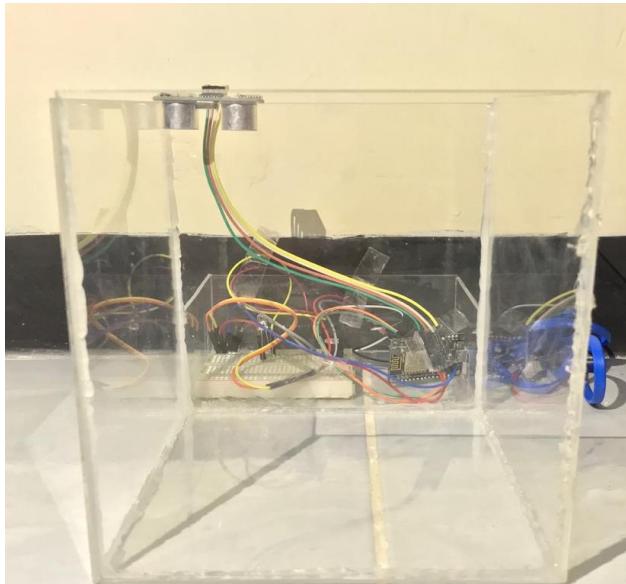
Gambar 2.86 Rangkaian Alat

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

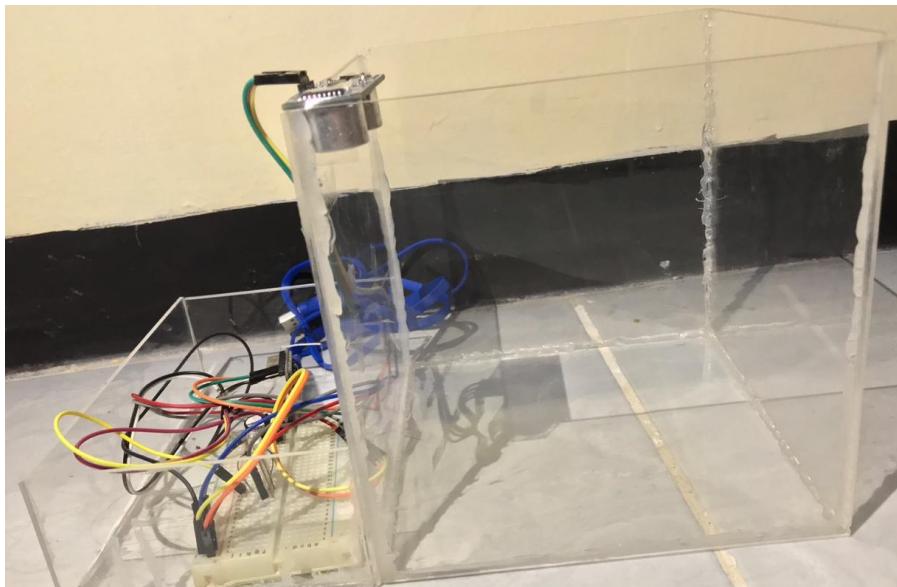
1. Sambungkan Vin NodeMCU menggunakan kabel jumper pada *breadboard* bagian positif (+). Vin merupakan Power atau VCC dengan memiliki daya 5V. Jadi jika tidak ada 5V dalam sebuah *microcontroller* dapat di gantikan dengan vin.
2. Sambungkan GND (*ground*) NodeMCU menggunakan kabel jumper pada *breadboard*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

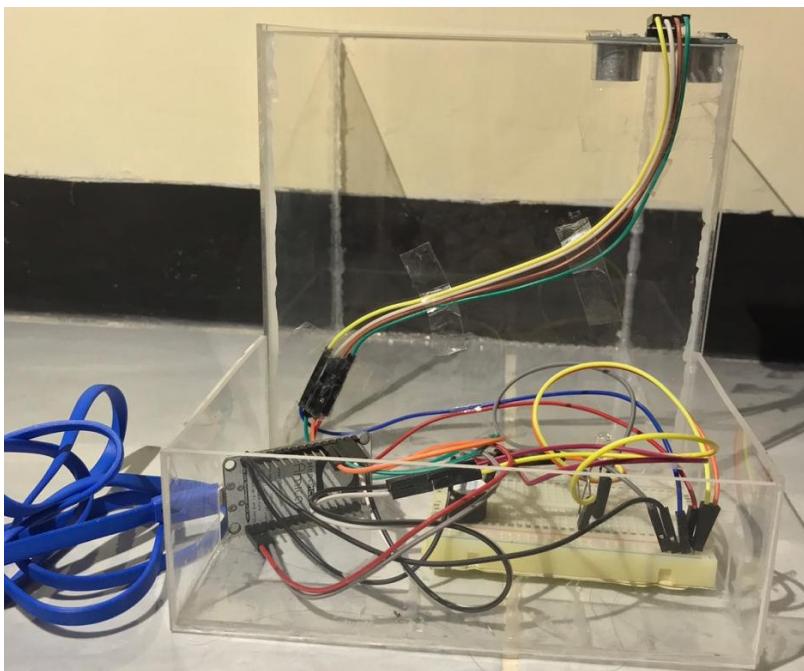
Setalah Kita mengikuti langkah-langkah tersebut maka simpan komponen pada rangka prototipe yang telah dibuat sebelumnya seperti pada gambar beikut:



Gambar 2.87 Rangka Prototipe Depan

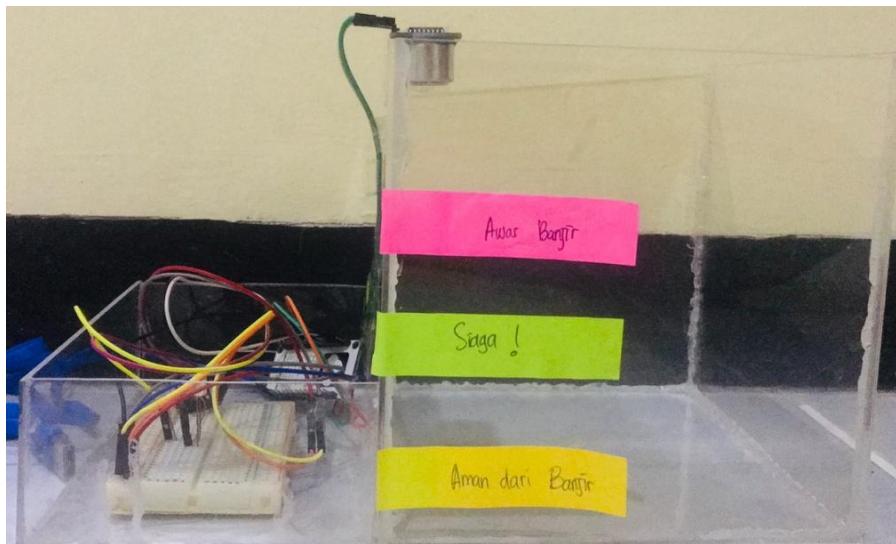


Gambar 2.88 Rangka Prototipe Samping



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

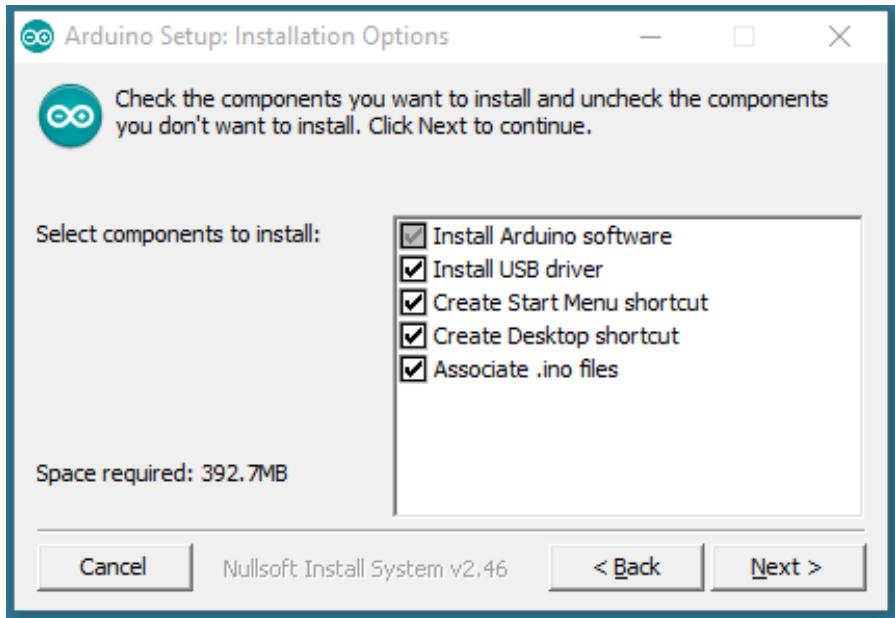
Dalam mengimplementasikan codingan untuk membuat prototipe prediksi ketinggian air (PKA) untuk mendeteksi banjir peringatan banjir ini menggunakan bahasa pemograman C. Dimana bahasa pemograman C ini telah di jelaskan pada bab sebelumnya.

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:

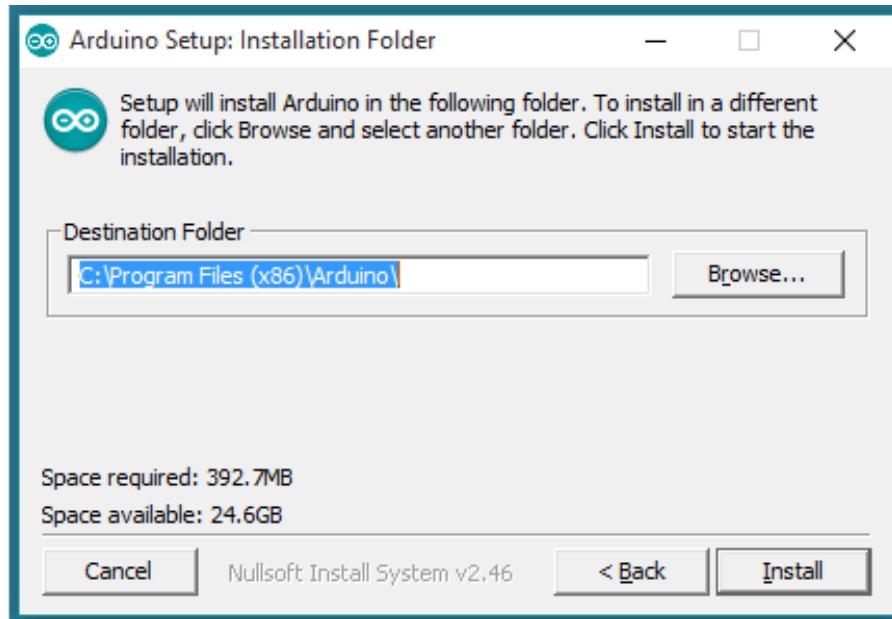
1. Setelah diunduh, klik dua kali file tersebut sehingga muncul jendela instalasi sebagai berikut :



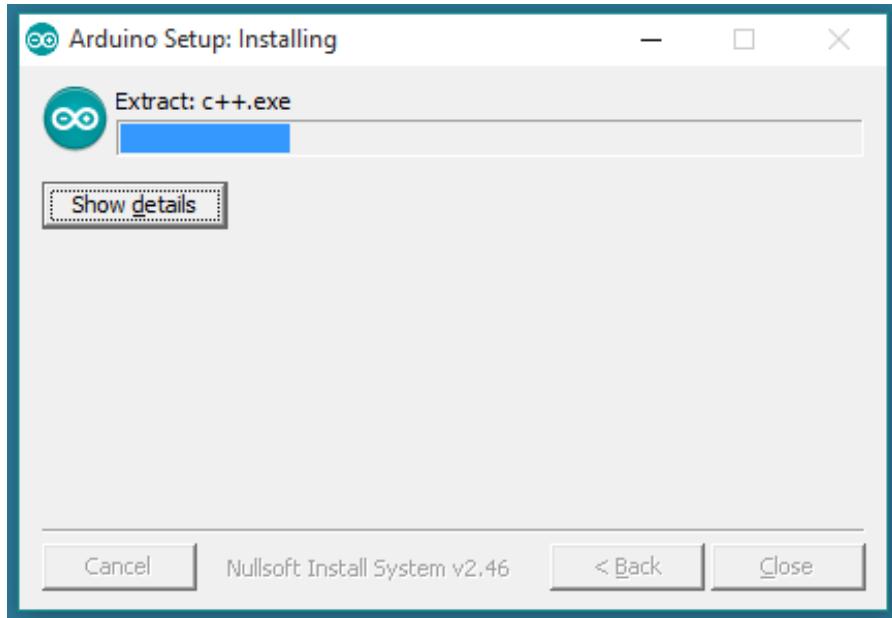
Gambar 3.1 Tampilan Jendela Instalasi

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.



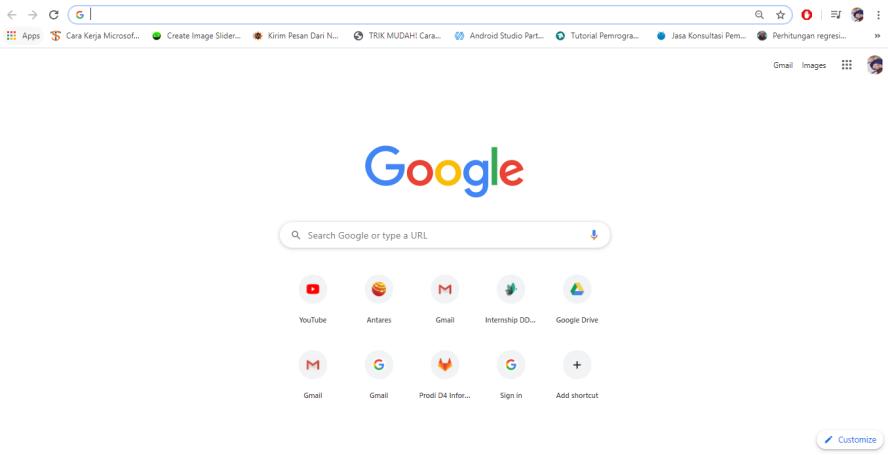
Gambar 3.2 Jendela Dialog Tujuan Instalasi Dari Arduino IDE



Gambar 3.3 Jendela Instalasi Arduino IDE

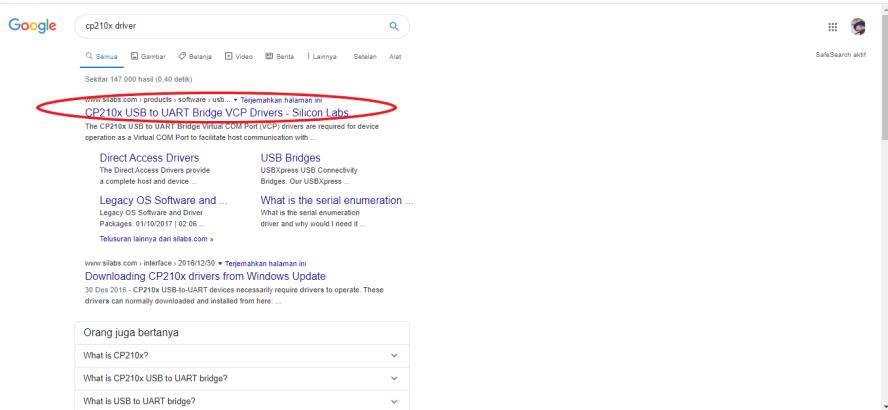
Tunggu hingga proses instalasi selesai. Setelah Arduino IDE selesai terinstall, kini sambungkan kabel USB Arduino pada port USB komputer/laptop. Jika anda menggunakan file installer yang sama di tunjukan pada pembahasan sebelumnya, maka arduino akan otomatis terbaca di COMX (X adalah nomer port serial,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

Setelah Arduino IDE terinstall maka kita harus *install* juga terlebih dahulu yang diperlukan untuk *development board* NodeMCU. Driver yang dibutuhkan yaitu driver CH34x dan CP210x. Cara mendownload dan *install* sebagai berikut :



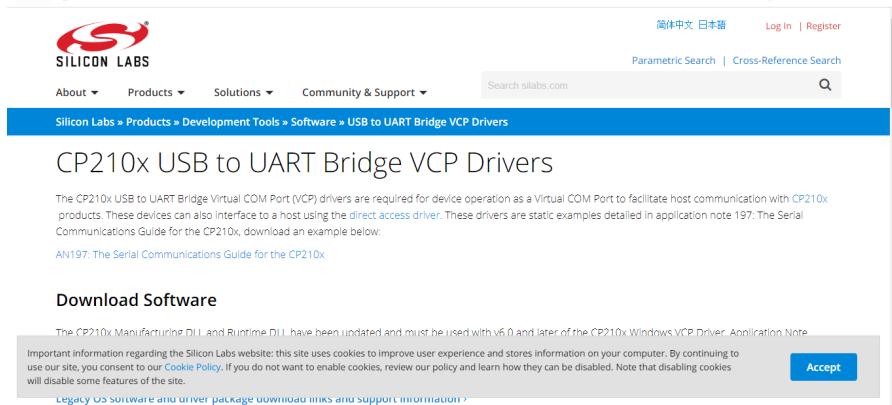
Gambar 3.4 Tampilan Google

Buka browser yang dimiliki, disini menggunakan google. Ketika google di klik maka tampilannya seperti gambar di atas. Setelah itu ketikan Cp210 driver.



Gambar 3.5 Tampilan Hasil Pencarian

Setelah ketik atau mencari Cp210 driver maka google akan menampilkan hasil pencari dari beberapa referensi. Kemudian klik hasil pencarian Cp210 driver paling atas seperti pada gambar diatas.



Gambar 3.6 Tampilan Cp210 Driver

Setelah diklik Cp210 driver paling atas maka tampilan awal dari website tersebut seperti pada gambar.

Note: The latest version of the Universal Driver can be automatically installed from Windows Update.

Platform	Software	Release Notes
Windows 10 Universal	Download VCP (2.3 MB)	Download VCP Revision History

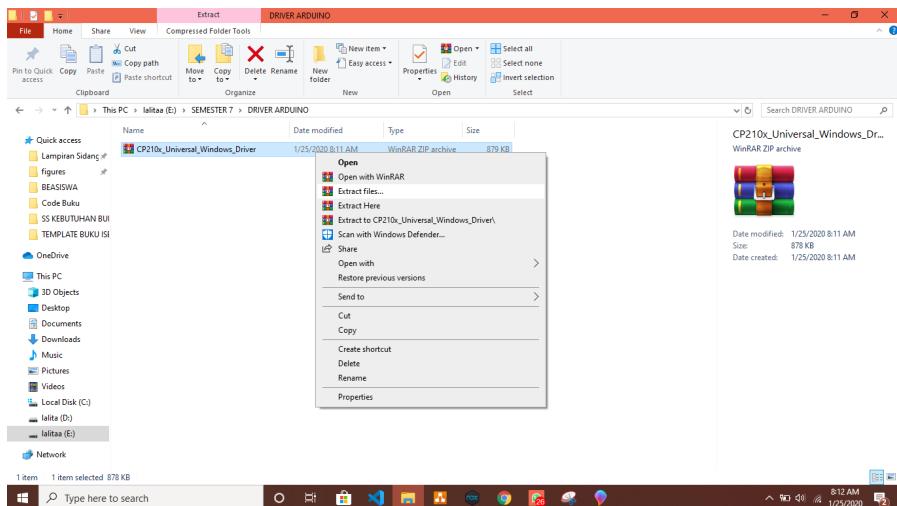
Download for Windows 7/8/8.1 (v6.7.6)

Platform	Software	Release Notes
Windows 7/8/8.1	Download VCP (5.3 MB) (Default)	Download VCP Revision History
Windows 7/8/8.1	Download VCP with Serial Enumeration (5.3 MB) Learn More »	Download VCP Revision History

Download for Windows XP/Server 2003/Vista/7/8/8.1 (v6.1)

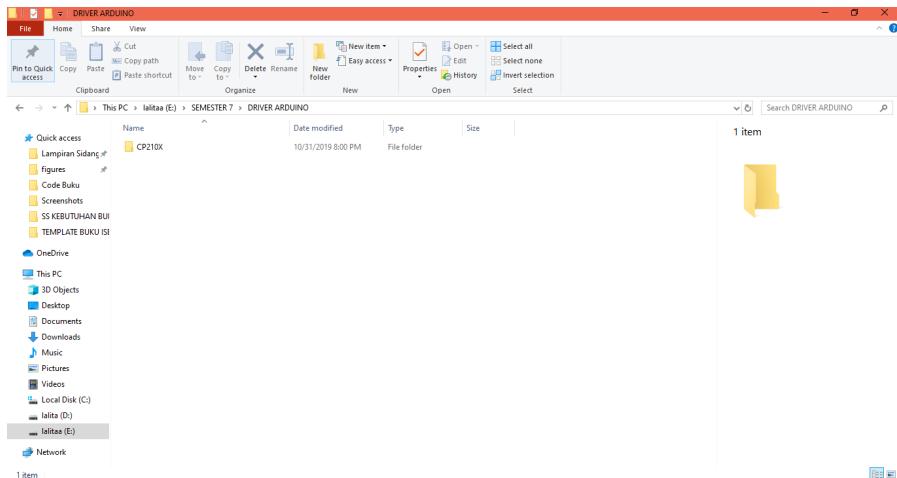
Gambar 3.7 Tampilan Cp210 Driver Dwonload

Setelah masuk ke tampilan Cp210 scroll kebawah sampai ada tulisan dwonload untuk window seperti gambar diatas. Kemudian pilih windows 10 jika menggunakan sistem operasi windows 10 jika tidak memakai maka harus disesuaikan.



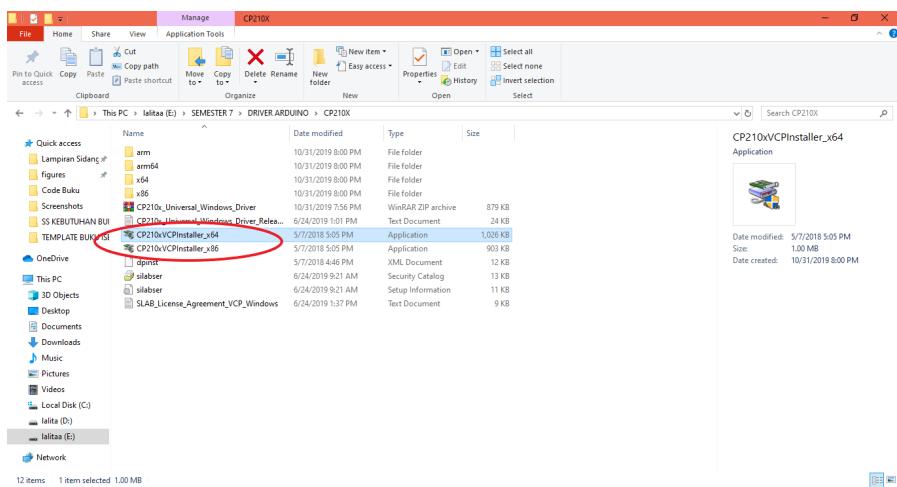
Gambar 3.8 Tampilan Hasil Dwonload Driver

Setelah di *dwnonload* maka file yang didapatkan dari hasil *dwnonload* berupa Rar. Kemudian klik kanan dan pilih extract here.



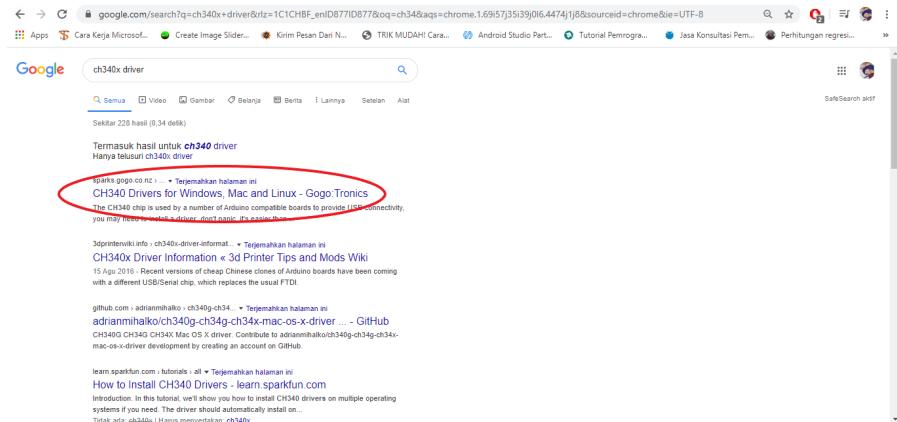
Gambar 3.9 Tampilan File Hasil *extract*

Hasil file yang sudah di extrac akan menjadi sebuah folder seperti pada gambar diatas.



Gambar 3.10 Tampilan File Cp210 Driver

Buka folder tersebut kemudian pilih CP210.exe seperti pada gambar diatas yang bertujuan untuk di *install*. Disini *install* driver Cp210 64 byte dan apabila memakai yang 32 byte maka pilih file yang ujungnya x86. Kemudian jika sudah menginstall driver Cp210 maka selanjutnya harus menginstall driver Ch340x seperti berikut.



Gambar 3.11 Tampilan Hasil Pencarian

Ketik atau mencari Ch340x driver di google. maka Google akan menampilkan hasil pencari dari beberapa referensi. Kemudian klik hasil pencarian Ch340x driver paling atas seperti pada gambar diatas.

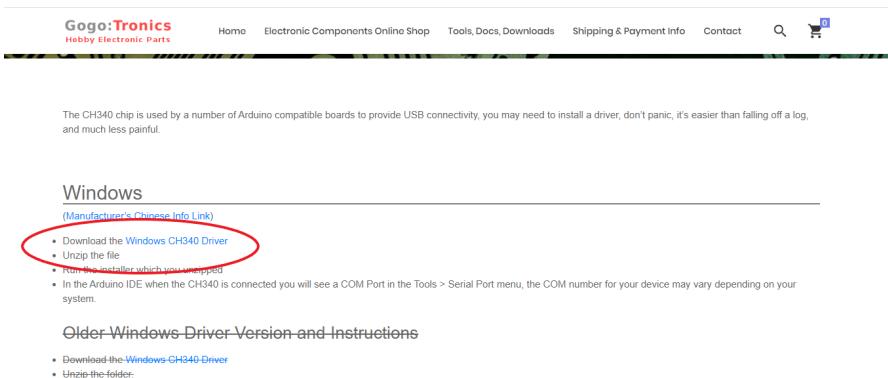


The CH340 chip is used by a number of Arduino compatible boards to provide USB connectivity, you may need to install a driver, don't panic, it's easier than falling off a log, and much less painful.

Windows

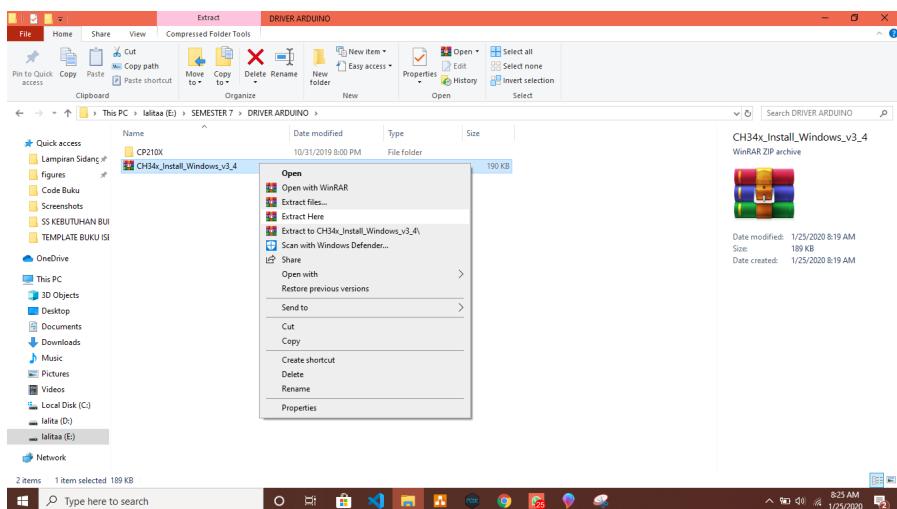
Gambar 3.12 Tampilan Hasil Pencarian

Setelah diklik Ch340x driver paling atas maka tampilan awal dari website tersebut seperti pada gambar.



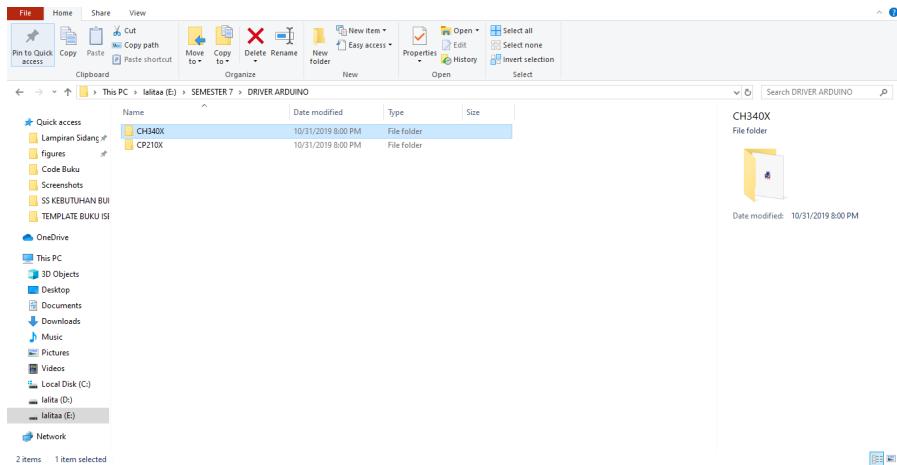
Gambar 3.13 Tampilan Ch340x Driver Dwonload

Setelah masuk ke tampilan Ch340x scroll kebagan bawah sampai ada tulisan dwonload untuk window seperti gambar diatas. Kemudian pilih windows CH340 Driver.



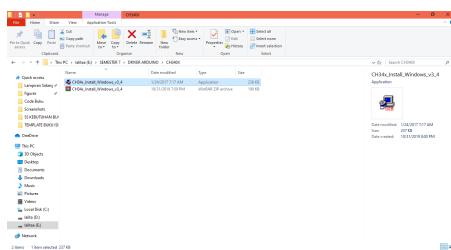
Gambar 3.14 Tampilan Hasil Download Driver

Setelah di *dwnonload* maka file yang didapatkan dari hasil *dwnonload* berupa Rar. Kemudian klik kanan dan pilih extract here.



Gambar 3.15 Tampilan File Hasil *extract*

Hasil file yang sudah di extrac akan menjadi sebuah folder seperti pada gambar diatas.



Gambar 3.16 Tampilan File CH340x Driver

Buka folder tersebut kemudian pilih CH340x.exe seperti pada gambar diatas yang bertujuan untuk di *installs*. Setelah kita selesai menginstall driver yang dibutuhkan maka tahapan selanjutnya yaitu memprogram nodemcu di arduino IDE.

3.2.2 Contoh Pemrograman Di Arduino IDE

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. The main area displays the 'Blink' example code. The code is as follows:

```

Blink
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
}

```

At the bottom of the IDE, it says 'Arduino Uno on COM3'.

Gambar 3.17 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.

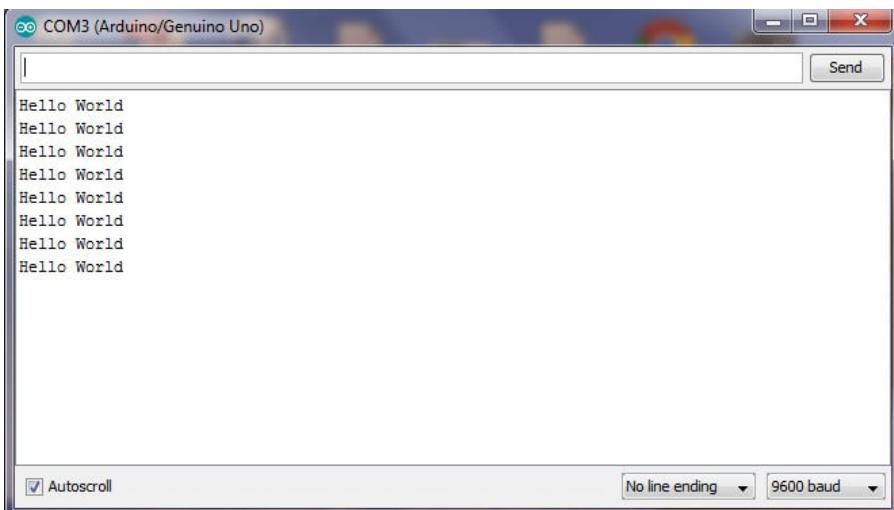
2. Upload Program

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 3.18 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.

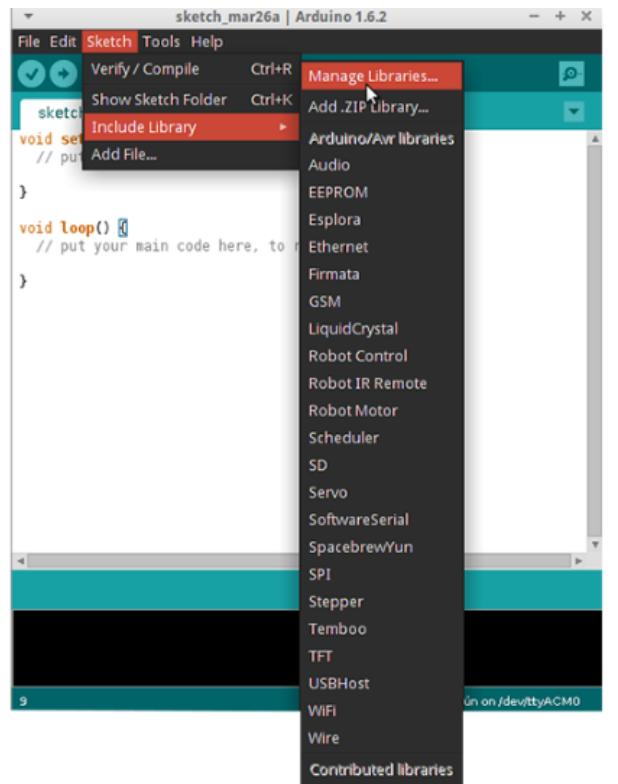
3.2.3 Cara Import Library

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.

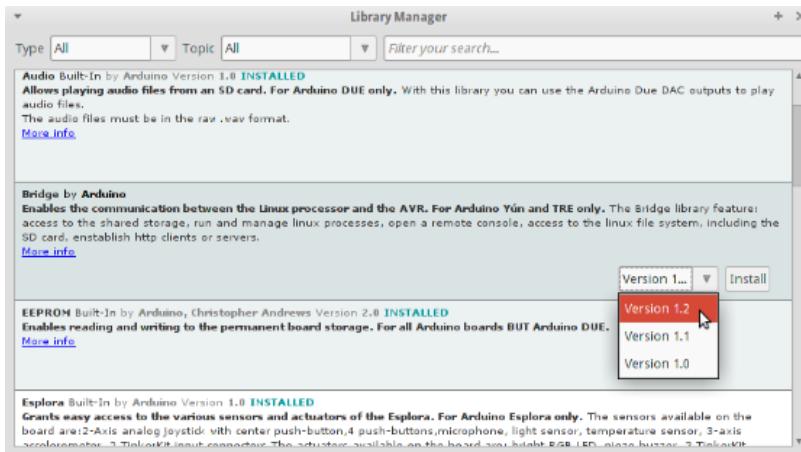
1. Import Library Menggunakan Manager

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



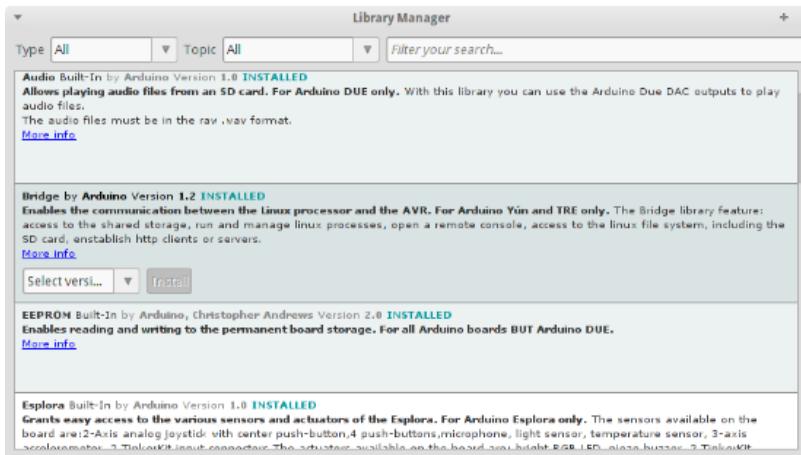
Gambar 3.19 Memilih Menu Manage Librarie

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 3.20 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.



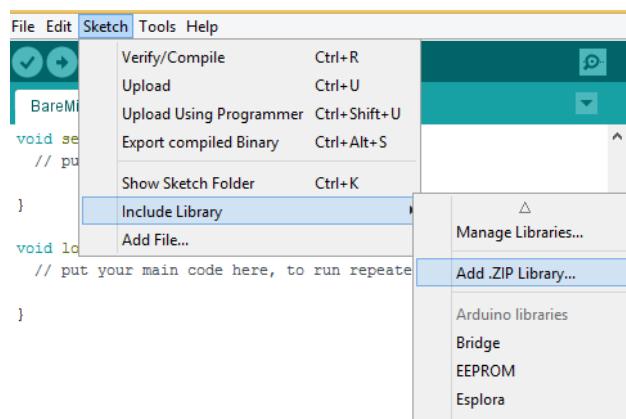
Gambar 3.21 Tampilan Library Yang Terinstall

2. Mengimport Library .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.

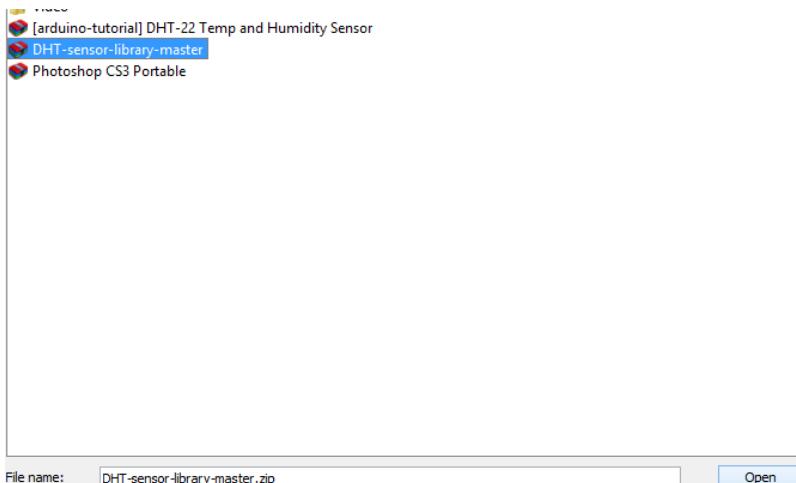
Nama folder adalah nama Library. Di dalam folder tersebut akan ada file .cpp, file .h , Folder Contoh Sketch, dan file lainnya yang dibutuhkan oleh Library. Dimulai dengan versi 1.0.5, Kita bisa menginstal library pihak ke-3 di IDE. Jangan unzip Libray yang telah didownload, biarkan seperti apa adanya.

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



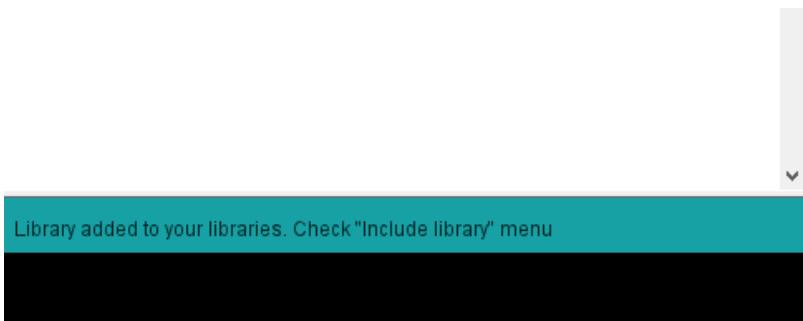
Gambar 3.22 Tampilan Library ADD ZIP

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



Gambar 3.23 Pilih File ZIP

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



Gambar 3.24 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.

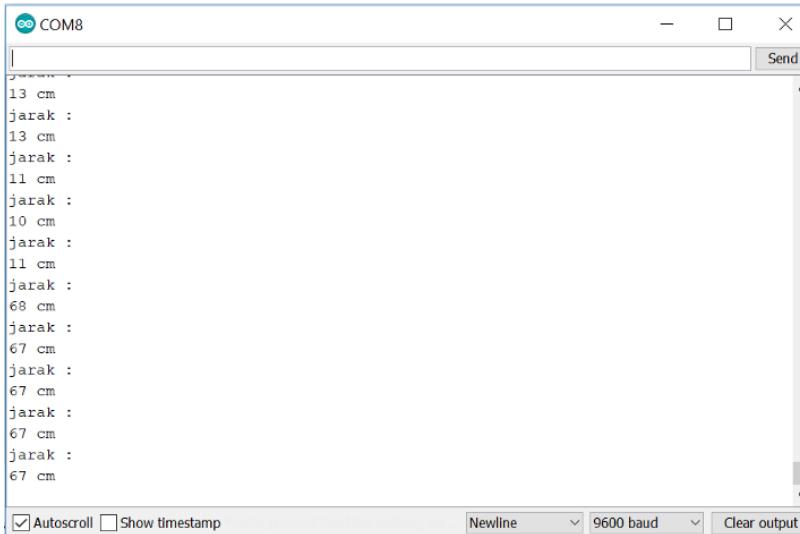
6. jarak = (duration/2) / 29.1;

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 3.25 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

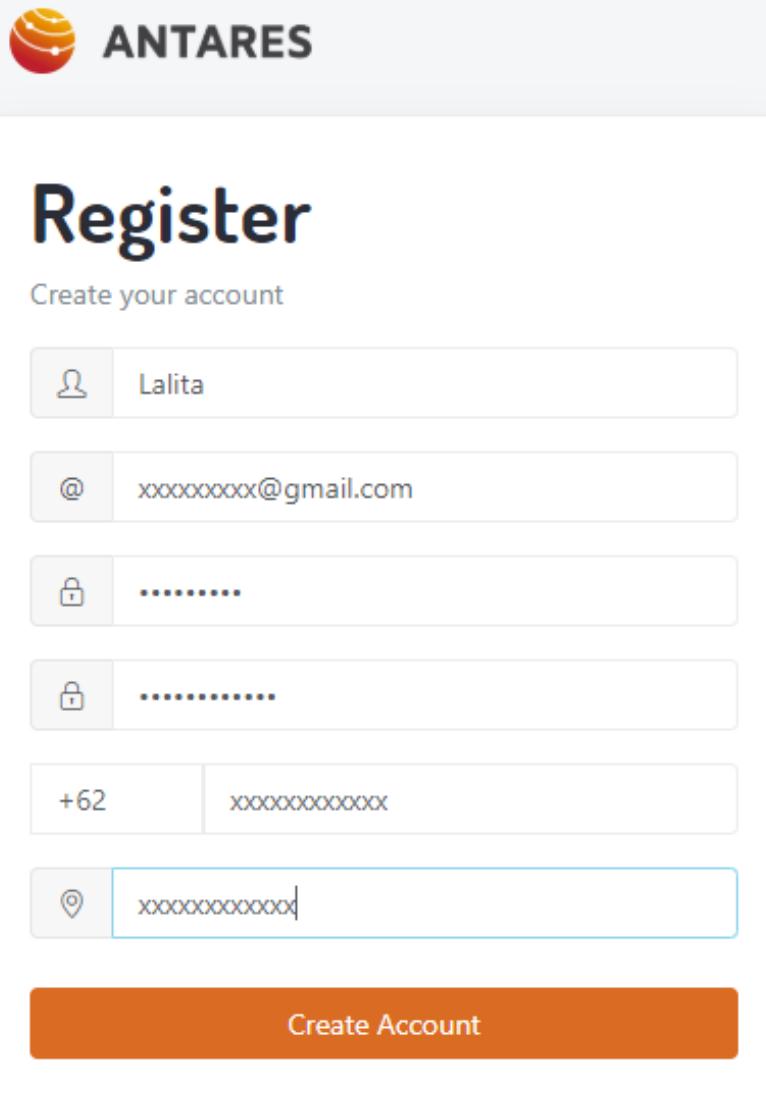
Setelah semua komponen yang digabungkan sudah diprogram maka tahapan selanjutnya yaitu menambahkan pemograman untuk antaresnya. Antares tersebut akan digunakan sebagai media penyimpan data yang dibaca oleh sensor ultrasonic. Sebelum memprogram ada langkah-langkah yang harus dilakukan untuk tahapan antares 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 3.26 Halaman Awal Antares

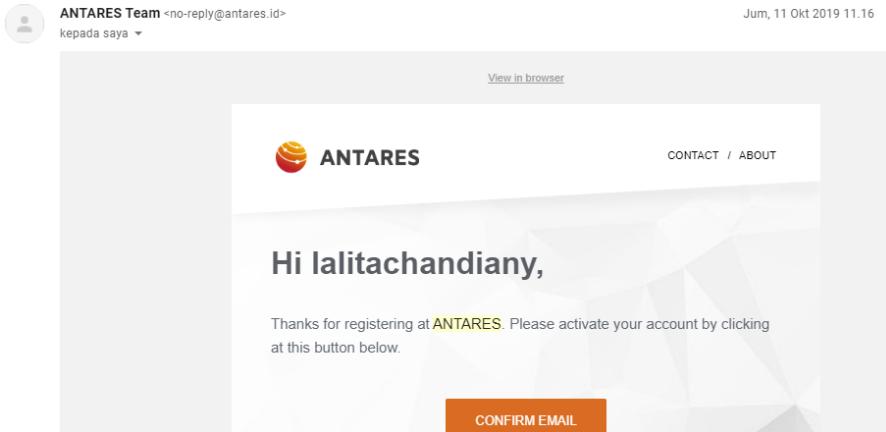
Setelah halaman tersebut muncul, pilih *Signup* kemudian masukan email yang aktif.



Gambar 3.27 Tampilan Registrasi Antares

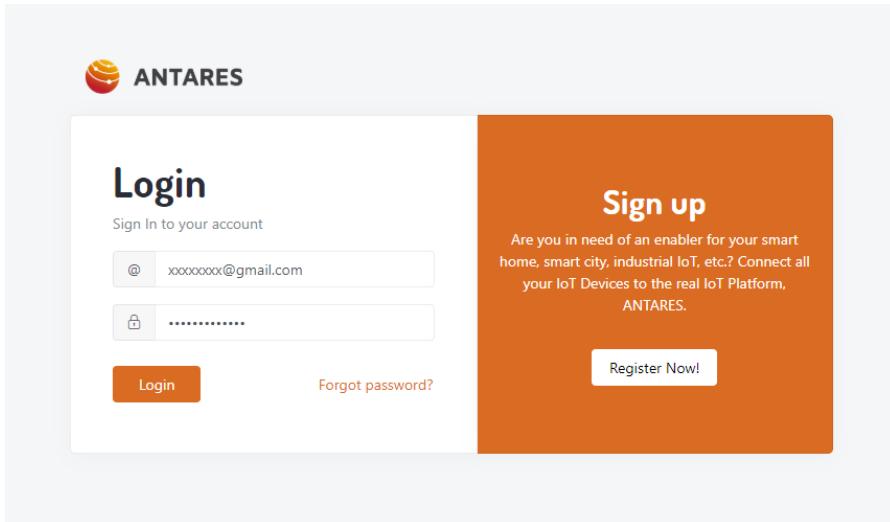
Masukan nama , alamat email (alamat email aktif), password, no telepon dan alamat. Alamat email harus aktif karena ini berguna untuk *verifikasi account* dan apabila ada info mengenai antares akan dikirmkan melalui email.

Activation Kotak Masuk



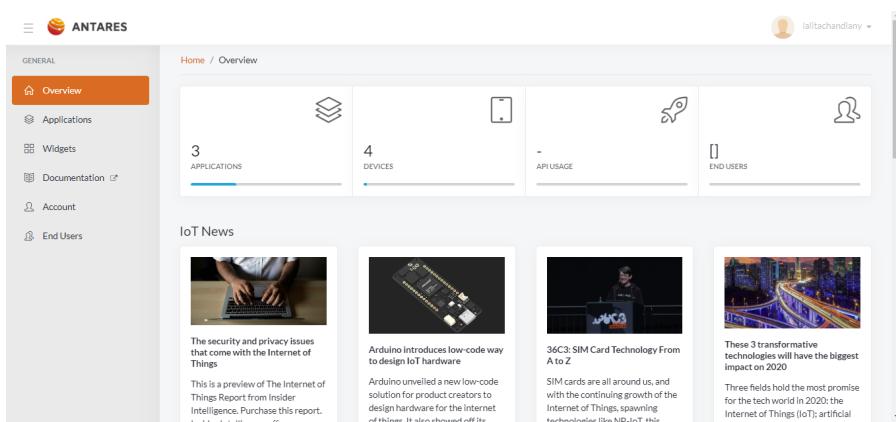
Gambar 3.28 Verifikasi email

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 3.29 Login Antares

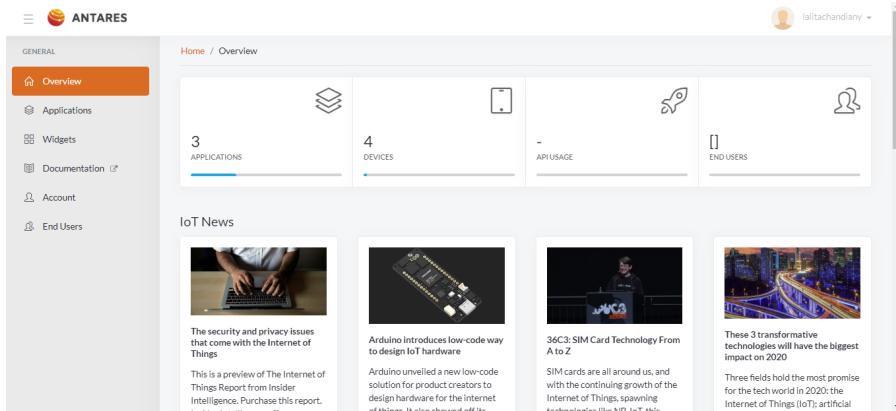
setelah kita meng*confirm* antares di email maka kita harus login di antares dengan mamasukan email yang telah didaftarkan dan memasukan password yang sebelumnya telah dibuat.



Gambar 3.30 Tampilan Awal Antares

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

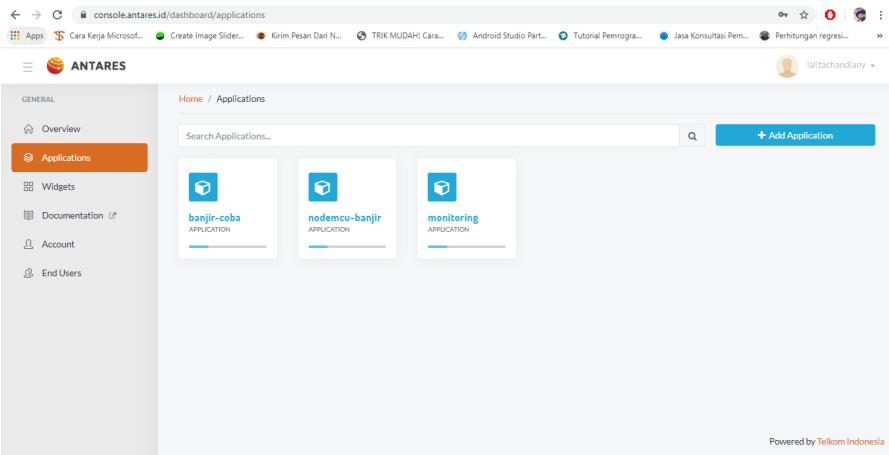
- Overview
- Application
- Widgets
- Documentation
- Account
- Enduser



Gambar 3.31 Halaman Overview

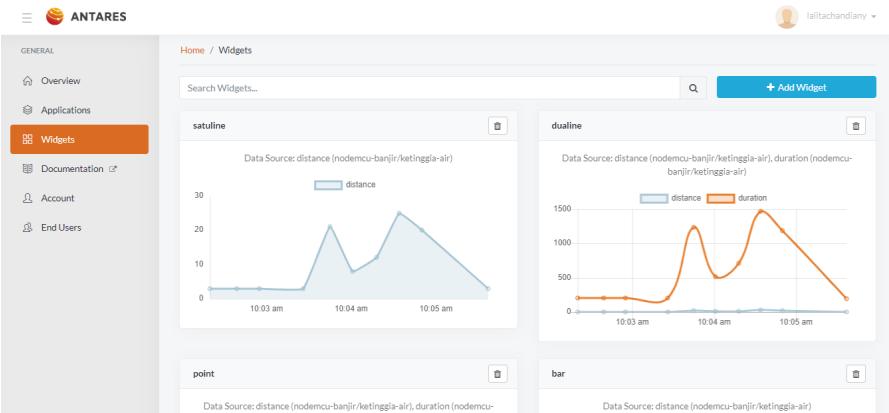
Halaman *Overview* merupakan halaman awal yang ditampilkan pada saat setelah login. Halaman ini menampilkan sebuah berita mengenai IoT (*Internet of*

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



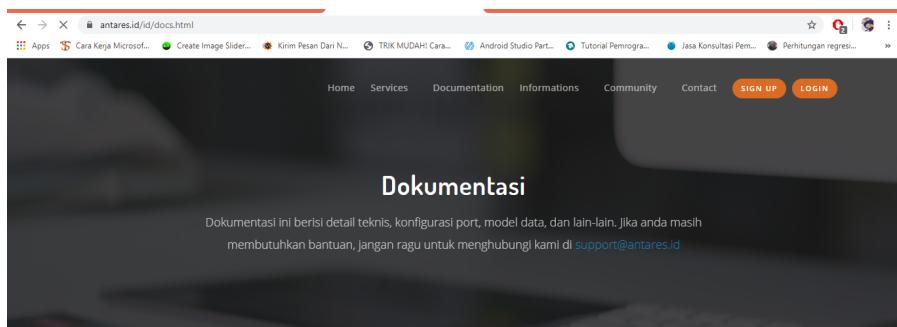
Gambar 3.32 Halaman Application

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

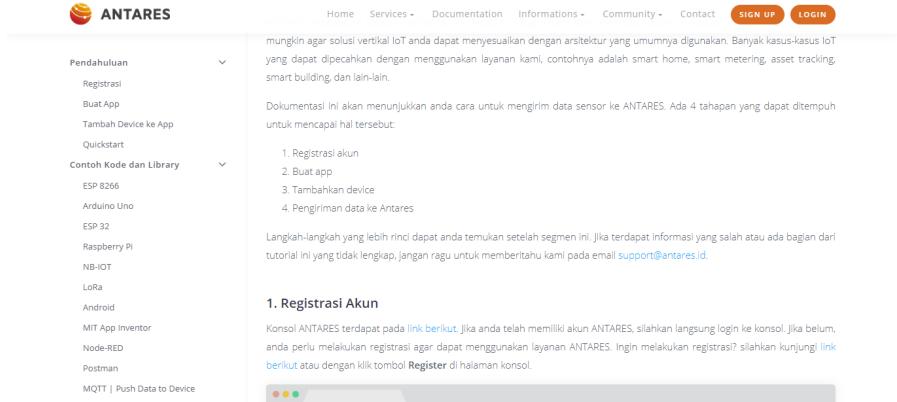


Gambar 3.33 Halaman Widgets

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 3.34 Halaman *Documentation*



Gambar 3.35 Halaman *Documentation*

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

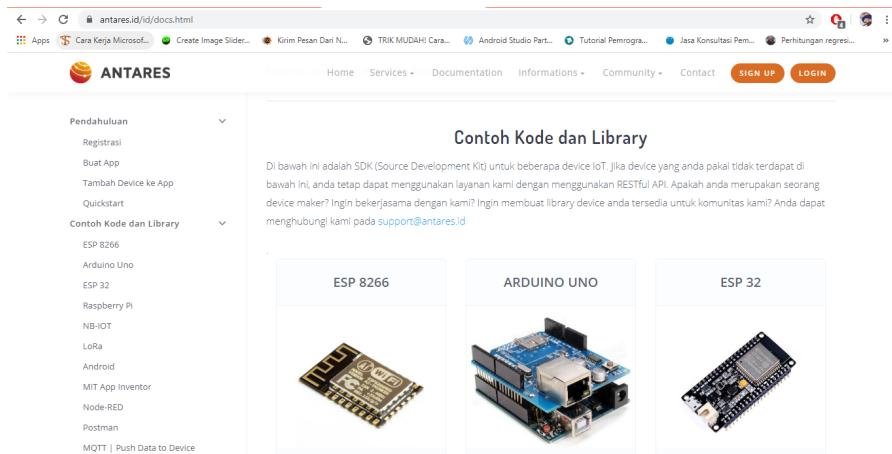
The screenshot shows a web browser window with the URL antares.id/id/docs.html. The page title is "ANTARES". The main navigation menu includes Home, Services, Documentation, Informations, Community, Contact, SIGN UP, and LOGIN. On the left sidebar, there are several categories: Pendahuluan, Registrasi, Buat App, Tambah Device ke App, Quickstart, Contoh Kode dan Library, ESP 8266, Arduino Uno, ESP 32, Raspberry Pi, NB-IOT, LoRa, Android, MIT App Inventor, Node-RED, Postman, and MQTT | Push Data to Device. The "Contoh Kode dan Library" section is expanded, showing a code snippet for an ESP8266 device. The code defines access keys, initializes WiFi, sets up serial communication, and sends sensor data (temperature and humidity) to a cloud service every 10 seconds. The code editor has syntax highlighting for C++.

```

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.setDebug(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 }

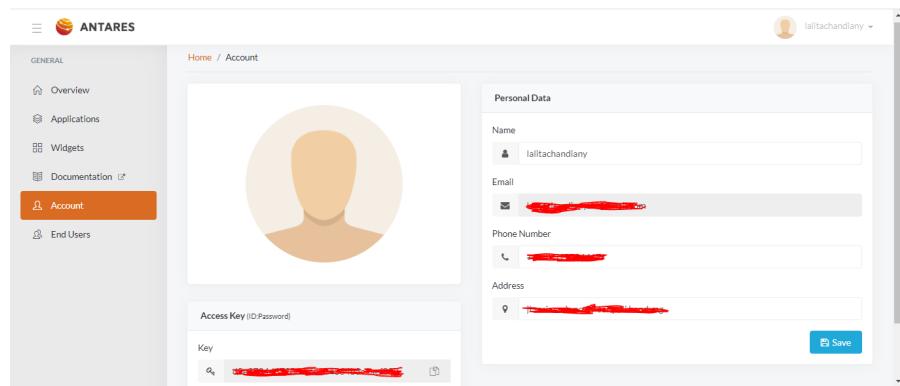
```

Gambar 3.36 Halaman *Documentation Tutorial*



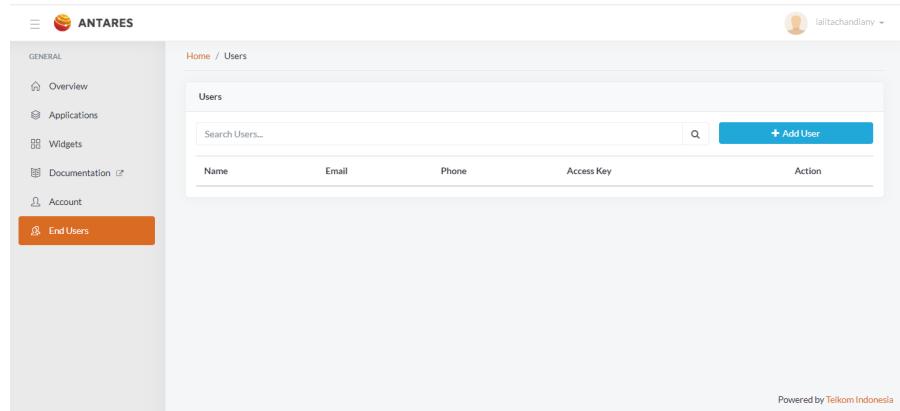
Gambar 3.37 Halaman *Documentation Tutorial*

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.



Gambar 3.38 Halaman *Account*

Halaman *account* ini menampilkan suatu informasi pemilik *account* seperti foto profile, *acces key*, *name*, *email*, *phone number*, *address*.



Gambar 3.39 Halaman *Enduser*

Halaman *enduser* berisi mengenai pembuatan user baru dengan cara memasukan *name*, *email*, *phone*, *acces key* and *action*

2. Memasukan *Library* antares ke *coding*

Sebelum memprogram dan membuat logika tentang antares terlebih dahulu kita harus mendownload *library* yang akan digunakan dengan cara berikut :

Contoh Kode dan Library

Di bawah ini adalah SDK (Source Development Kit) untuk beberapa device IoT. Jika device yang anda pakai tidak terdapat di bawah ini, anda tetap dapat menggunakan layanan kami dengan menggunakan RESTful API. Apakah anda merupakan seorang device maker? Ingin bekerjasama dengan kami? Ingin membuat library device anda tersedia untuk komunitas kami? Anda dapat menghubungi kami pada support@antares.id

ESP 8266	ARDUINO UNO	ESP 32

Gambar 3.40 Halaman *DocumentationTutorial*

Setelah login keantares pilih menu *documentation* kemudian pilih esp8266 go to tutorial.

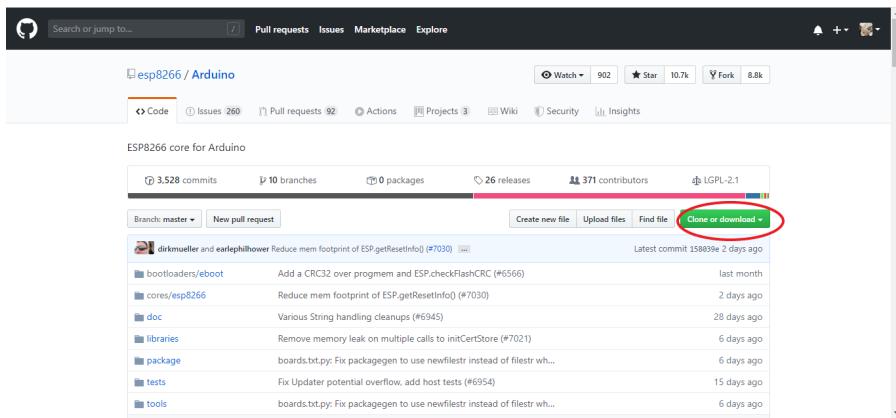
Software

Berikut adalah beberapa software yang dibutuhkan untuk menjalankan contoh-contoh pada dokumentasi ini:

- **LDR**
Sensor untuk mengukur intensitas cahaya
- **Arduino IDE**
Software utama yang akan digunakan untuk memprogram ESP8266.
[Download Arduino IDE](#)
- **Arduino ESP8266**
Arduino IDE tidak mendukung board ESP8266 secara bawaannya, maka dari itu perlu dilakukan instalasi Arduino ESP8266 melalui *Board Manager Arduino IDE*
[Download Arduino ESP8266](#)
- **Library Antares ESP8266**
Komunikasi antara ESP8266 dengan database Antares dapat menggunakan **HTTP** maupun **MQTT**.
Instalasi masing-masing *library* HTTP maupun MQTT dapat dilakukan melalui *Library Manager Arduino IDE*, atau dengan mendownloadnya pada halaman GitHub.
 - [Antares ESP8266 HTTP](#)
 - [Antares ESP8266 MQTT](#)

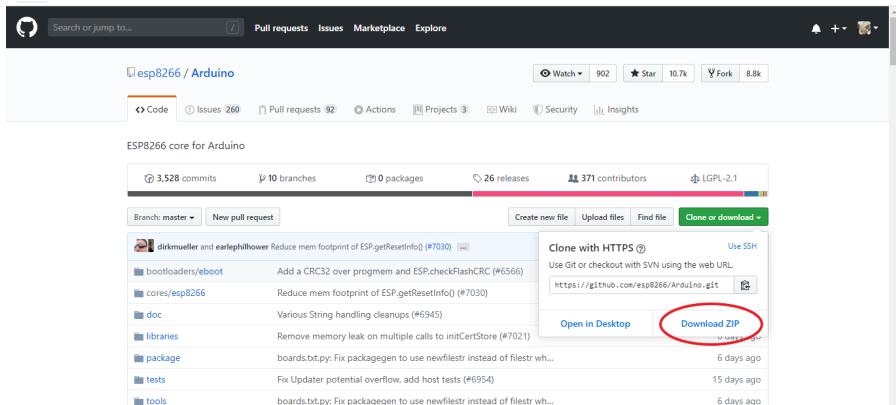
Gambar 3.41 Dwonload Library

Setelah kita memilih *go to tutorial* maka akan muncul tampilan seperti gambar diatas. Lalu pilih atau klik *dwnload* arduino esp8266.



Gambar 3.42 Dwonload Library di Github

Ketika kita sudah klik *dwnload arduino esp8266*. Maka Akan muncul tampilan seperti diatas, dimana tampilan tersebut langsung masuk ke repositori di github. Setelah masuk ke github kemudian kita klik menu *clone or dwonload*.



Gambar 3.43 Dwonload Library

Ketika menu *clone or dwonload* diklik maka akan muncul dua pilihan seperti pada gambar diatas. Untuk dwonload *library* maka pilih *Download ZIP*.

Jika kita ingin kita *import library* menggunakan board manager, maka langkah-langkahnya seperti berikut :

The screenshot shows the Antares website's documentation page for the ESP8266 module. The left sidebar has a 'Contoh Kode dan Library' section expanded, showing options like ESP 8266, Arduino Uno, ESP 32, Raspberry Pi, NB-IOT, LoRa, Android, MIT App Inventor, Node-RED, Postman, and MQTT | Push Data to Device. The main content area has a heading 'Contoh Kode dan Library' and a paragraph about the SDK. Below it are three images of the ESP 8266 module: ESP 8266, ARDUINO UNO, and ESP 32.

Gambar 3.44 Halaman *DocumentationTutorial*

Setelah login keantares pilih menu *documentation* kemudian pilih esp8266 go to tutorial.

The screenshot shows the GitHub repository page for 'esp8266 / Arduino'. It displays basic repository statistics: 3,528 commits, 10 branches, 0 packages, 26 releases, 371 contributors, and an Apache License 2.0 license. A pull request from 'dirkmueller' is shown. The repository has 902 stars and 8.8k forks. The repository description is 'ESP8266 core for Arduino'.

Branch:	master	New pull request	Create new file	Upload files	Find file	Clone or download
dirkmueller and earlephilhower Reduce mem footprint of ESP.getResetInfo() (#7030)		Latest commit 158839e 2 days ago				
bootloaders/eboot	Add a CRC32 over progmem and ESP.checkFlashCRC (#6566)	last month				
cores/esp8266	Reduce mem footprint of ESP.getResetInfo()	2 days ago				
doc	Various String handling cleanups (#6945)	29 days ago				
libraries	Remove memory leak on multiple calls to initCertStore (#7021)	7 days ago				
package	boards.txt.py: Fix packagegen to use newfilestr instead of filestr wh...	7 days ago				
tests	Fix Updater potential overflow. add host tests (#6954)	16 days ago				
tools	boards.txt.py: Fix packagegen to use newfilestr instead of filestr wh...	7 days ago				

Gambar 3.45 Tampilan di Github

Setelah kita memilih *go to tutorial* maka akan muncul tampilan seperti gambar diatas. Lalu scroll kebawah hingga menemukan tulisan *contents*.

- Current "git version" documentation
- Install git version (sources)

Arduino on ESP8266

This project brings support for the ESP8266 chip to the Arduino environment. It lets you write sketches, using familiar Arduino functions and libraries, and run them directly on ESP8266, with no external microcontroller required.

ESP8266 Arduino core comes with libraries to communicate over WiFi using TCP and UDP, set up HTTP, mDNS, SSDP, and DNS servers, do OTA updates, use a file system in flash memory, and work with SD cards, servos, SPI and I2C peripherals.

Contents

- Installing options:
 - Using Boards Manager
 - Using git version
 - Using PlatformIO
 - Building with make
- Documentation
- Issues and support
- Contributing
- License and credits

Gambar 3.46 Tampilan *content*

Kemudian setelah *scroll* ke bawah hingga menemukan tulisan *content maka pilih atau klik using boards manager*.

Installing with Boards Manager

Starting with 1.6.4, Arduino allows installation of third-party platform packages using Boards Manager. We have packages available for Windows, Mac OS, and Linux (32 and 64 bit).

- Install the current upstream Arduino IDE at the 1.8.7 level or later. The current version is on the [Arduino website](#).
- Start Arduino and open the Preferences window.
- Enter `https://arduino.esp8266.com/stable/package_esp8266com_index.json` into the Additional Board Manager URLs field. You can add multiple URLs, separating them with commas.
- Open Boards Manager from Tools > Board menu and install esp8266 platform (and don't forget to select your ESP8266 board from Tools > Board menu after installation).

Latest release [Release 2.0.0](#)

Boards manager https://arduino.esp8266.com/stable/package_esp8266com_index.json

Documentation: <https://arduino-esp8266.readthedocs.io/en/2.0.0/>

Using git version

[Build package](#)

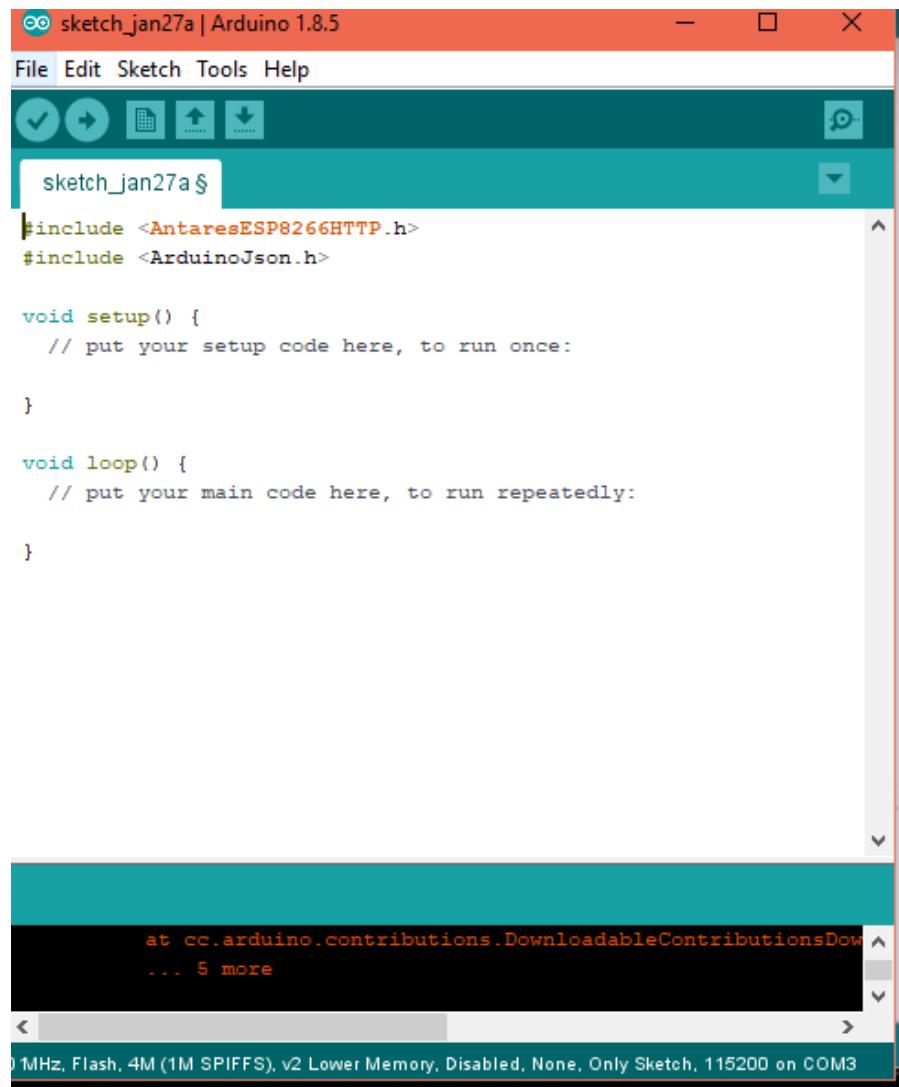
Also known as latest git or master branch.

- Install the current upstream Arduino IDE at the 1.8 level or later. The current version is on the [Arduino website](#).
- Follow the [instructions in the documentation](#).

[Using PlatformIO](#)

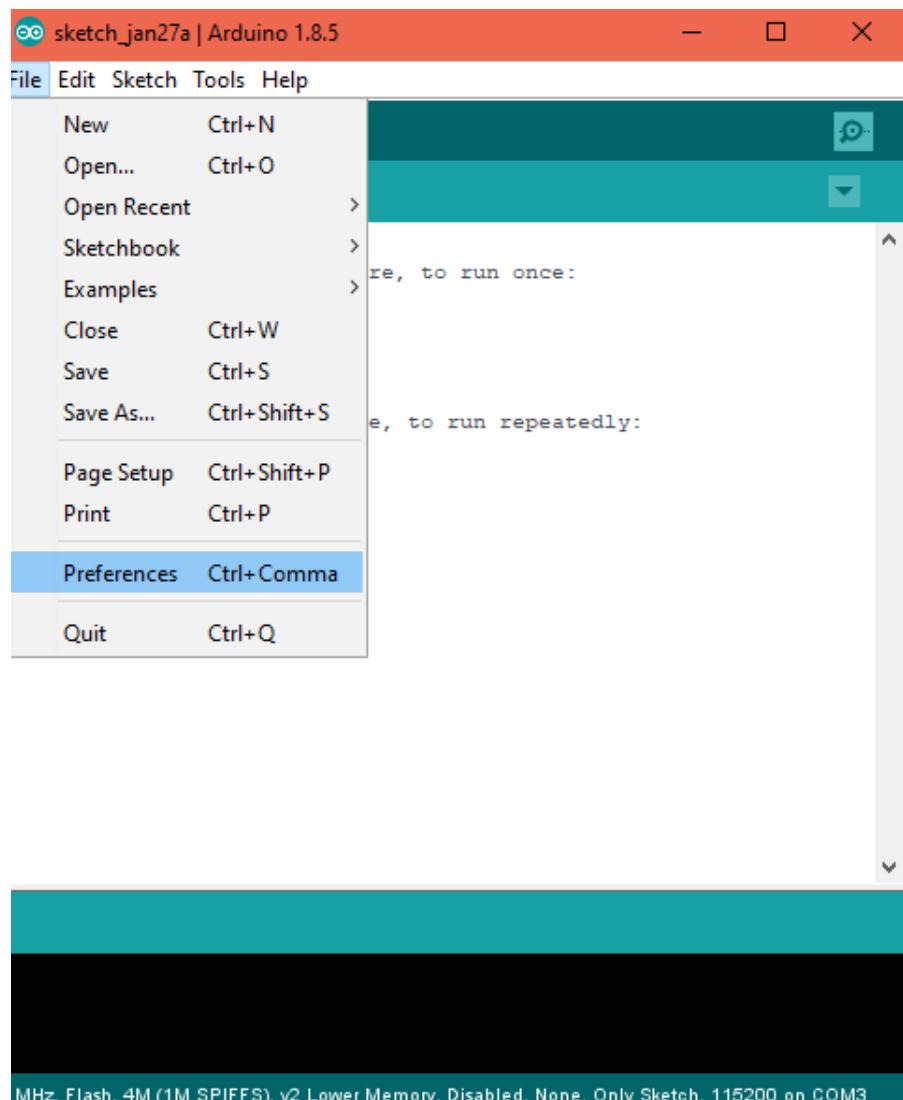
Gambar 3.47 Tampilan Using Boards Manager

Ketika klik *using boards manager* maka tampilan yang akan ditampilkan seperti gambar diatas. Kemudian kita copy alamat link yang tertera.



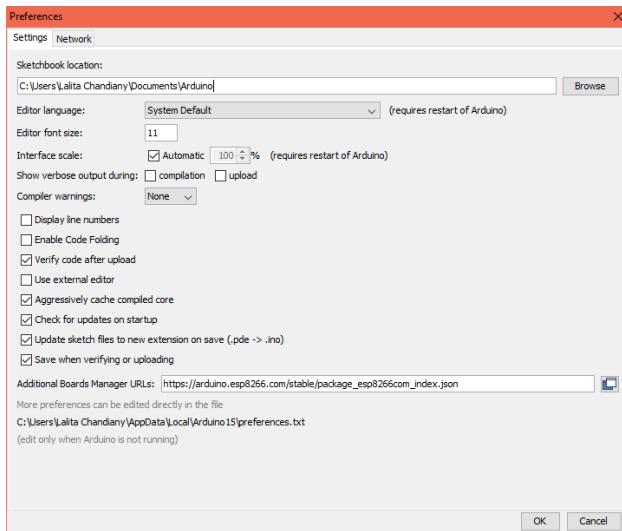
Gambar 3.48 Tampilan Arduino IDE

Setelah mengcopy link dari github,Kita terlebih dahulu membuka *software* arduino IDE.



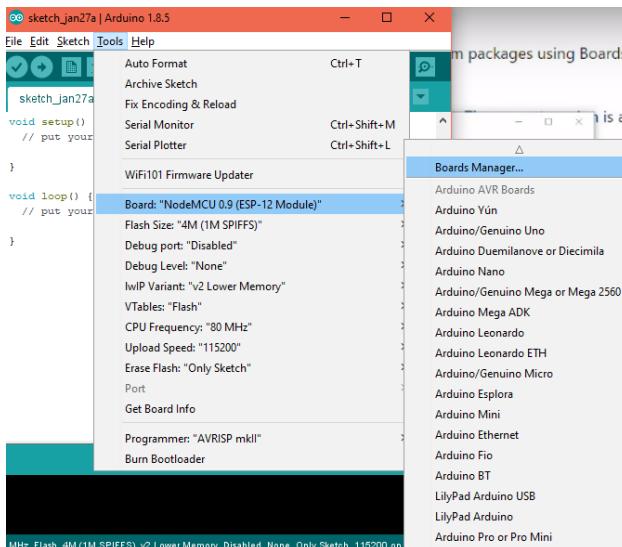
Gambar 3.49 Tampilan Arduino IDE

Ketika arduino IDE telah dibuka. Maka kita pilih File klik Preferencer seperti pada gambar diatas.



Gambar 3.50 Tampilan Preferences

Setelah memilih preferences maka tampilannya yang muncul seperti pada gambar tersebut. Kemudian paste alamat url di bagian *Additional Boards Manager URLs* yang sebelumnya di copy dari github dan klik ok.



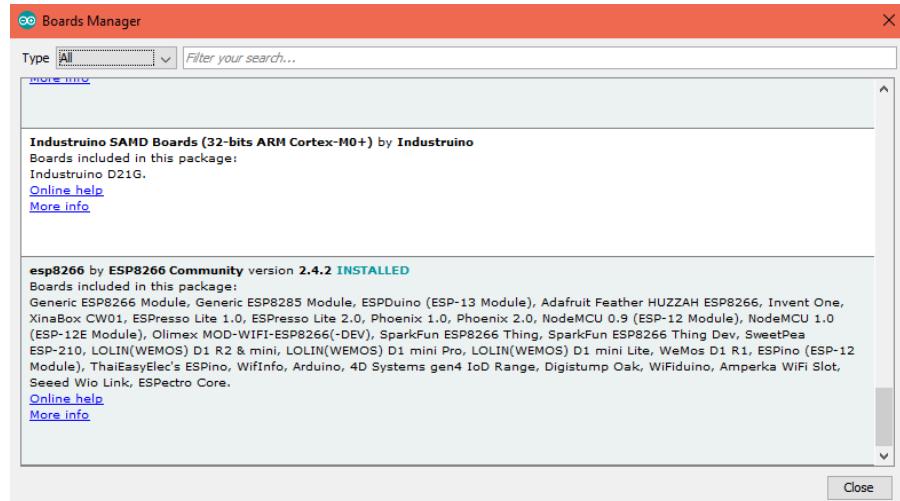
Gambar 3.51 Tampilan Arduino IDE Boards Mnager

Kemudian jika sudah salin alam url nya maka selanjutnya pilih Tools lalu Boards kemudian Boardmanager seperti gambar diatas.



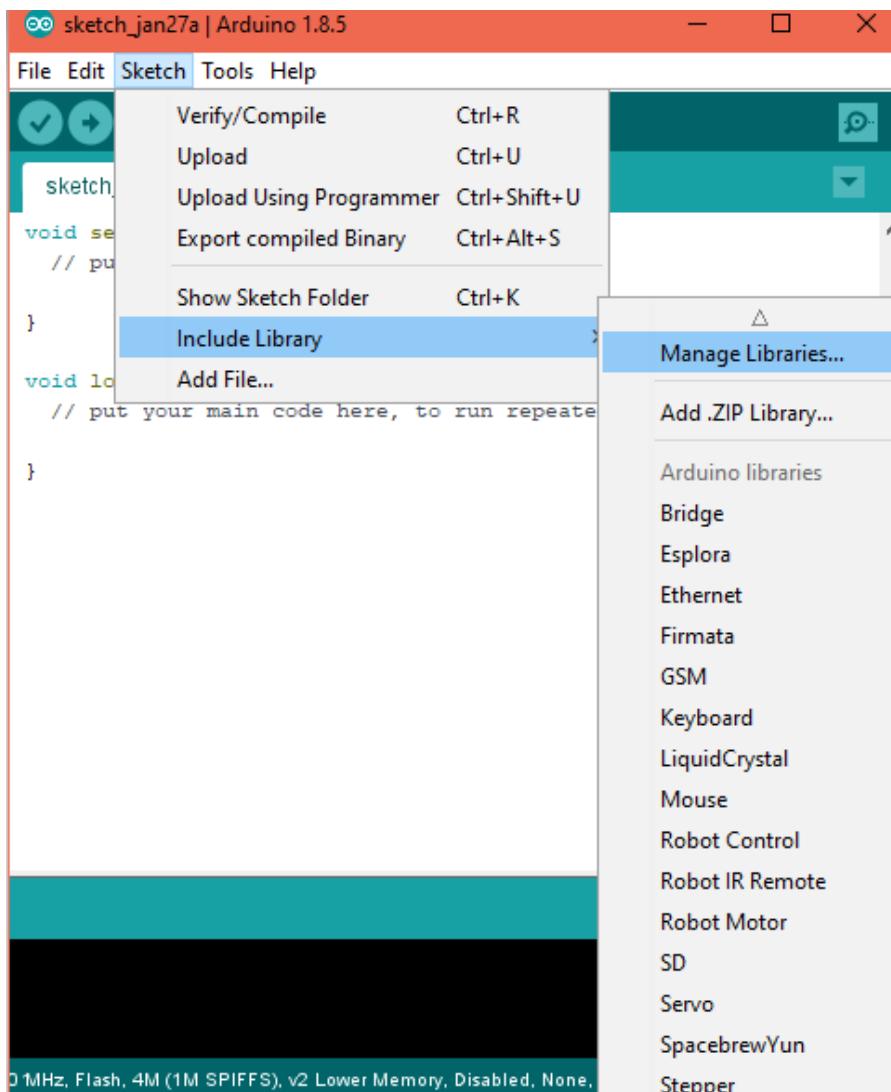
Gambar 3.52 Tampilan Boards Manager

Jika kita telah klik boards manager maka akan muncul tampilan boards manager.



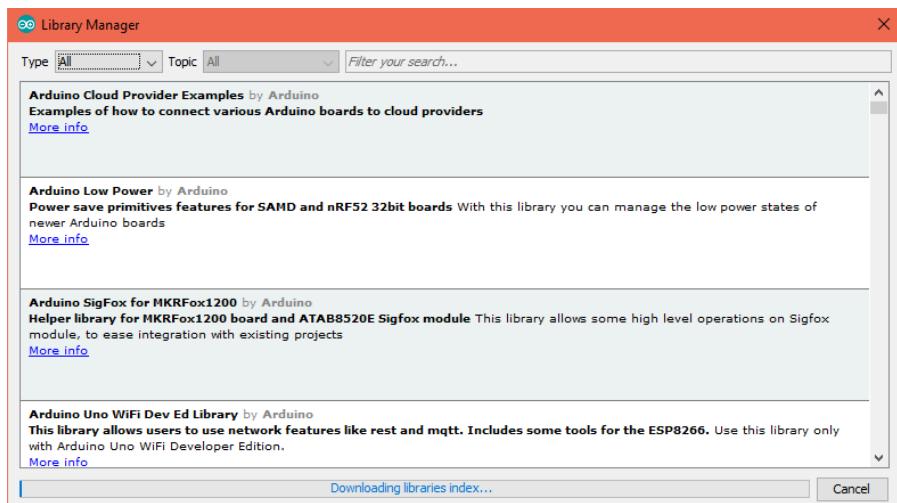
Gambar 3.53 Tampilan Boards Manager Antares

Ketika tampilan boards manager telah muncul kita scroll ke yang paling bawah. Kemudian install esp8266 dan versinya disesuaikan.



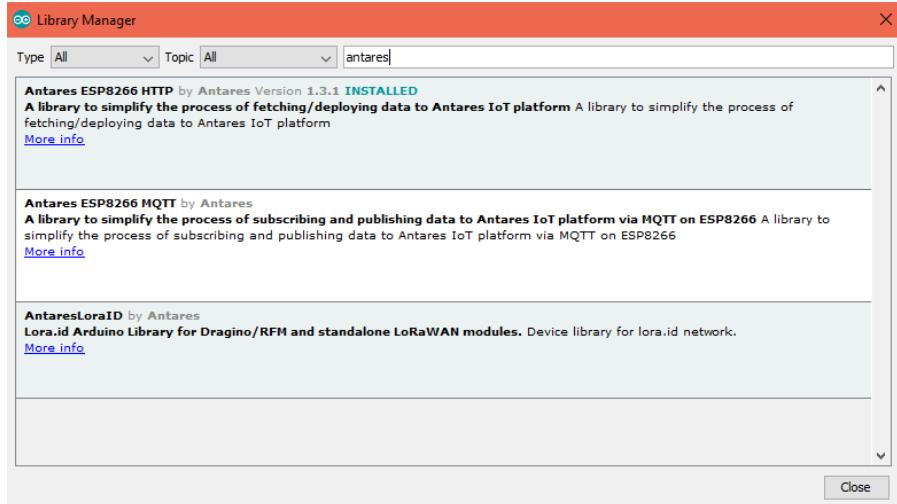
Gambar 3.54 Tampilan Arduino IDE Include Library

Setelah menginstal esp8266 tahapan selanjutnya kita pilih Sketch lalu Include Library kemudian Manage Library. Seperti pada gambar diatas.



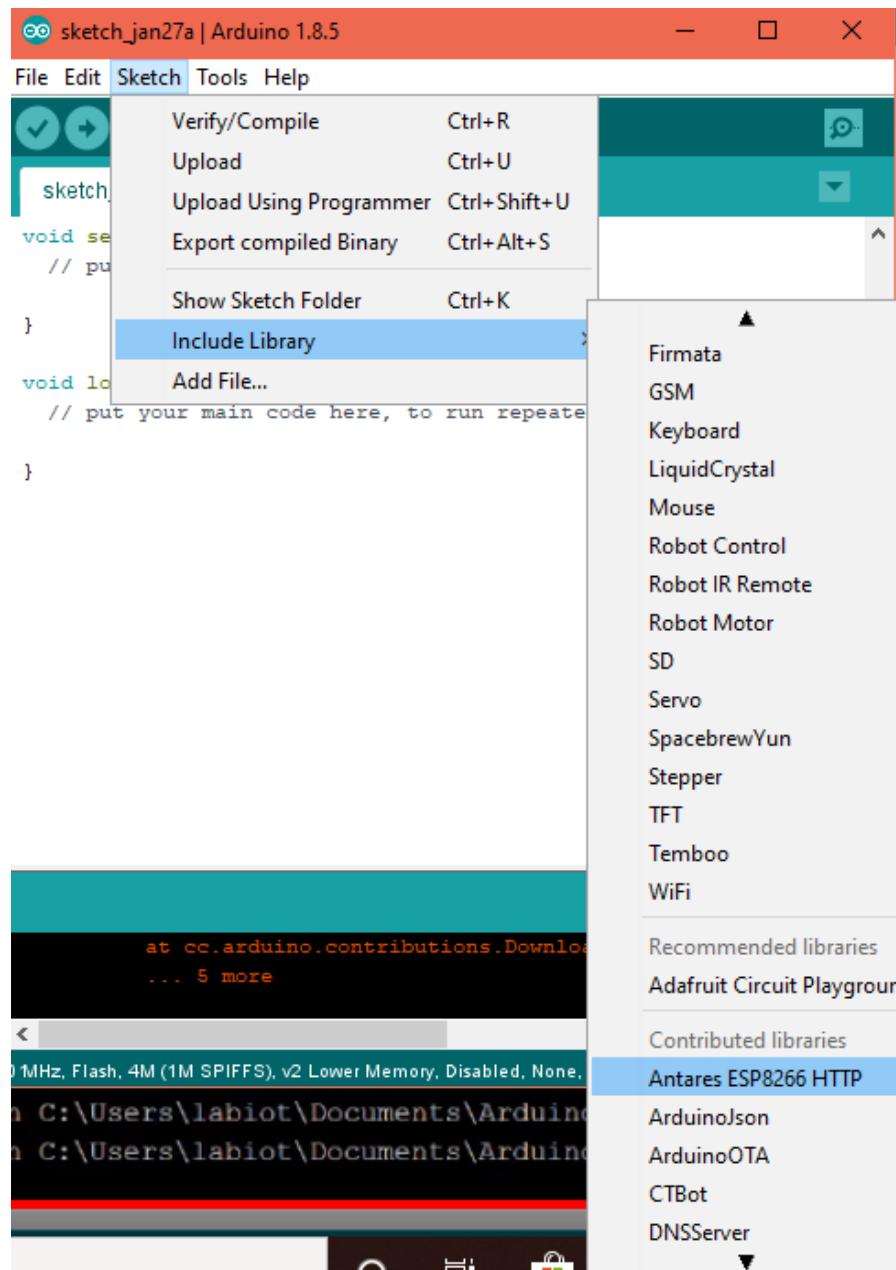
Gambar 3.55 Tampilan Library Manager

Jika kita telah klik manage libraries maka akan muncul tampilan seperti gambar diatas.



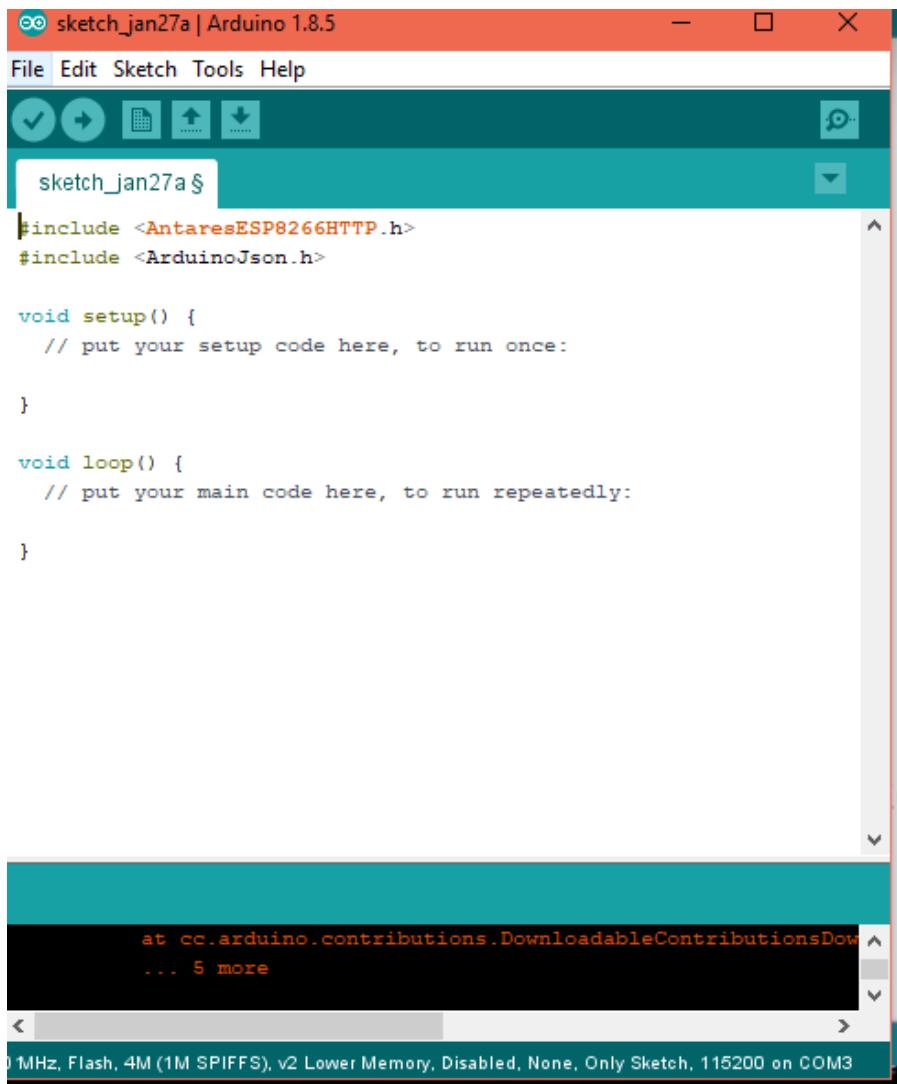
Gambar 3.56 Tampilan Library Manager Antares

Setelah muncul tampilan manage libraries kemudian ketikan antares pada bagian *search*. Ketika telah memasukan kata antares maka akan muncul *library* antares paling atas lalu install.



Gambar 3.57 Tampilan Arduino IDE Manage Library

Jika telah menginstall semua library antares yang dibutuhkan, maka hal yang harus dilakukan selanjutnya yaitu menginclude library dengan cara pilih *sketch* lalu *include library* dan pilih Antares ESP8266HTTP.



Gambar 3.58 Tampilan Arduino IDE Library Antares

Jika kita sudah mengklik library antares yang telah di dwonload maka akan langsung tertampahkan codenya seperti pada gambar diatas.

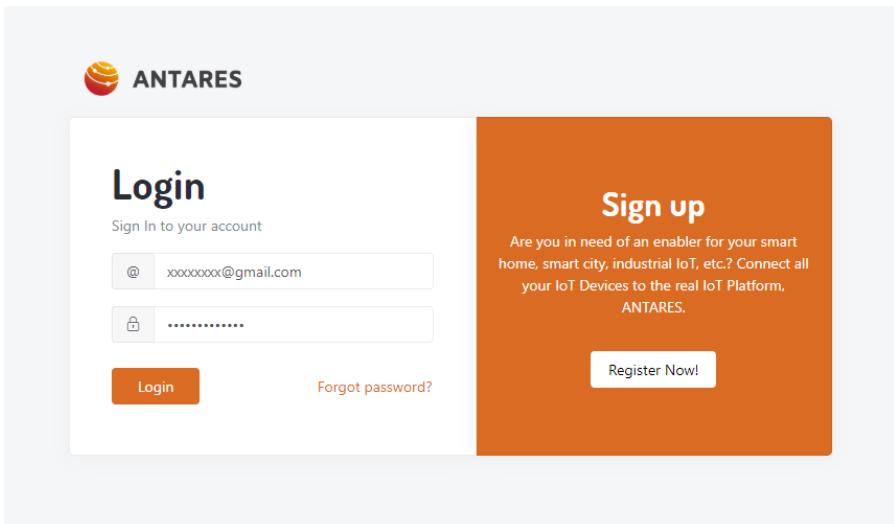
3. Membuat Project Di Antares

Sebelum membuat sebuah program yang isinya gabungan dari beberapa komponen untuk membuat *prototype* ini , diperlukan terlebih dahulu membuat sebuah *project* atau *application* di antares yang bertujuan untuk menyimpan data yang dibaca oleh sensor jarak. Berikut langkah-langkahnya :



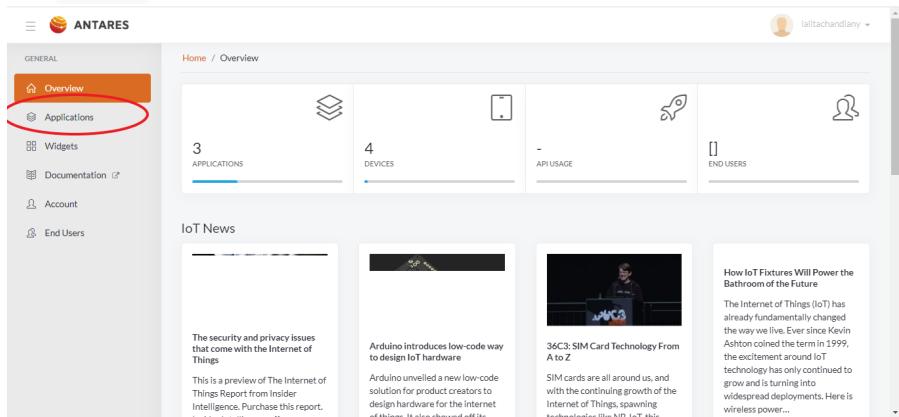
Gambar 3.59 Tampilan Antares

Hal pertama yang harus dilakukan yaitu masuk ke halaman antares hanya dengan mengetikkan urlnya pada browser. Url dari antares yaitu antares.id , ketika kita ketikan url tersebut maka akan muncul tampilan awal antares seperti pada gambar diatas.



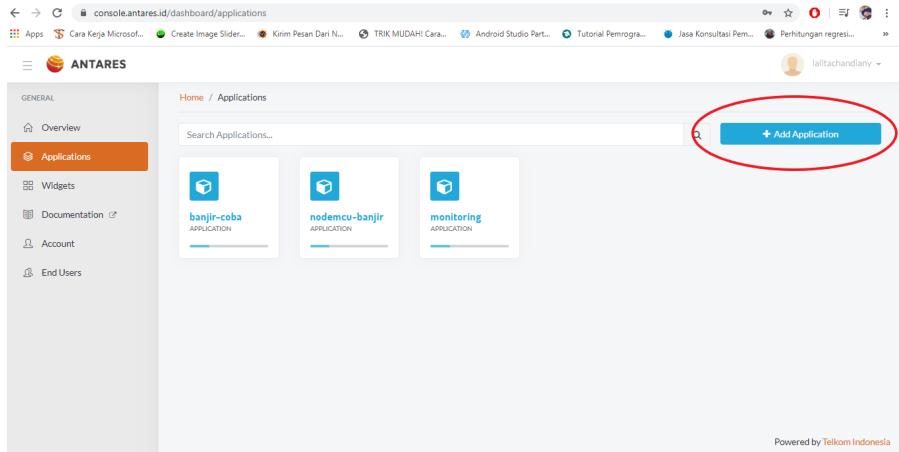
Gambar 3.60 Tampilan Login Antares

Kemudian pilih menu login dan akan muncul tampilan login. Masukan gmail dan password yang sudah terdaftar sebelumnya.



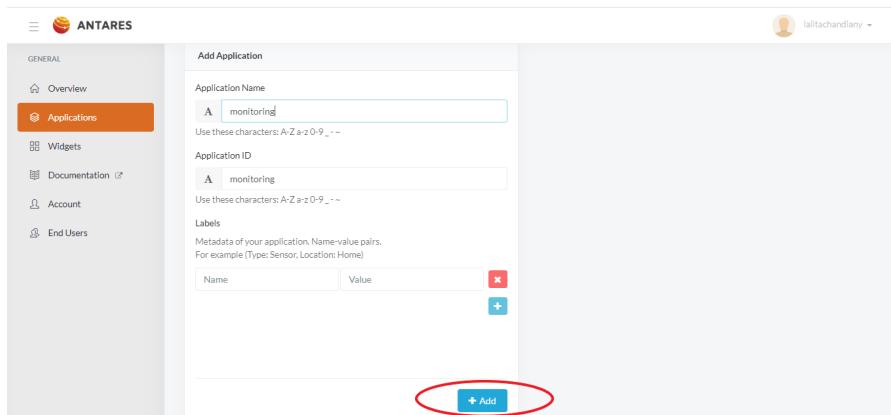
Gambar 3.61 Tampilan Awal Antares

Setelah kita melakukan login akan muncul tampilan awal antares yang disediakan dengan beberapa menu. Untuk membuat sebuah *project* baru atau *application* pilih menu *application*.



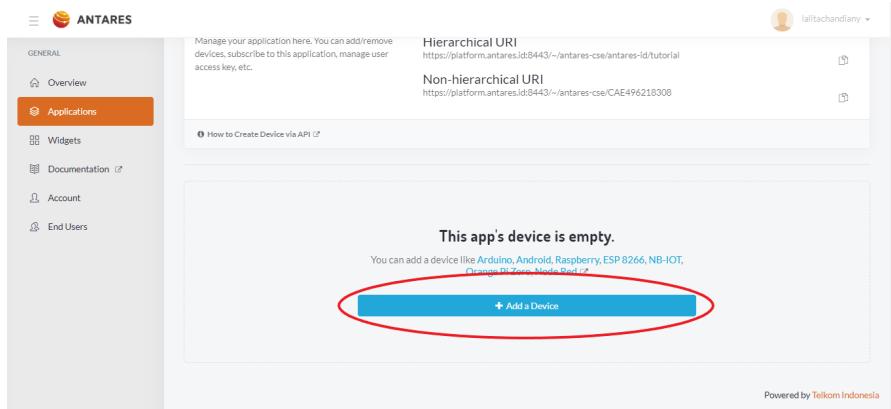
Gambar 3.62 Tampilan Application Antares

Ketika kita mengklik atau pilih menu *application*, maka kita akan dihadapkan dengan tampilan seperti pada gambar diatas. Setelah itu klik *add application* untuk membuat sebuah *project* baru.



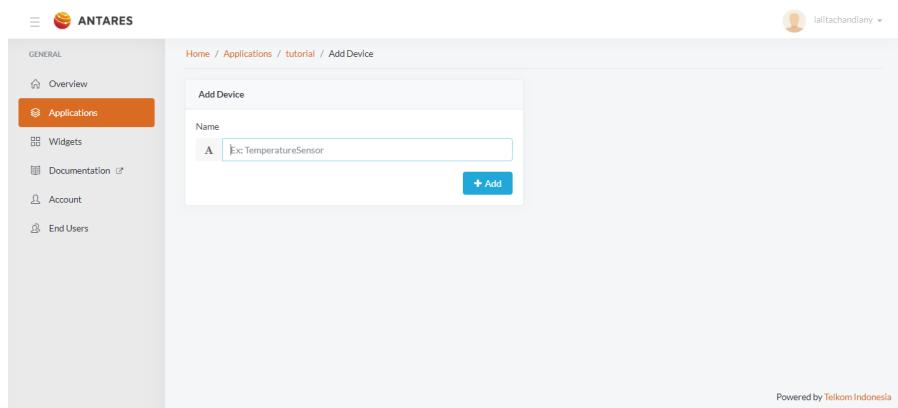
Gambar 3.63 Tampilan Membuat Application Antares

Setelah meng klik *addapplication*, maka ditampilkan selanjutnya harus mengisi *application name* dan *application ID* dengan nama yang bebas. Disini diberikan sebuah *application name* dan *application ID*nya monitoring. Karena nantinya data ini akan digunakan sebagai monitoring ketinggian air. Kemudian setelah mengisi semua klik *add*.



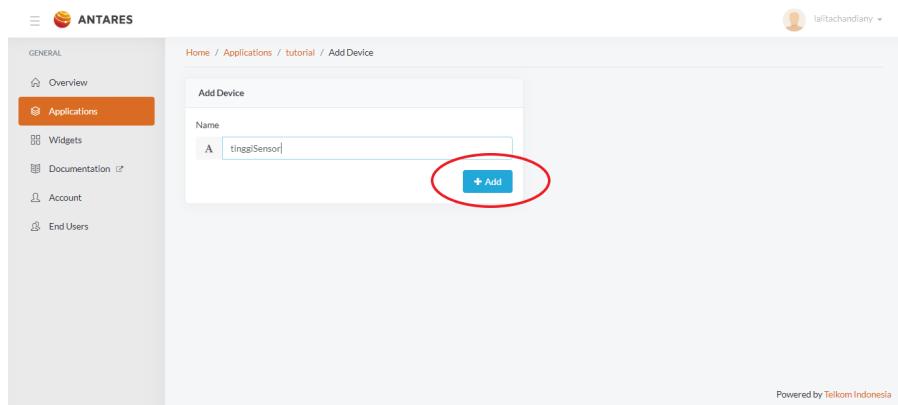
Gambar 3.64 Tampilan Application Yang Dibuat

Jika kita sudah membuat sebuah *application* atau *project* , maka akan muncul tampilan seperti gambar diatas. Kemudian klik *AddDevice* untuk membuat sebuah *device* baru .



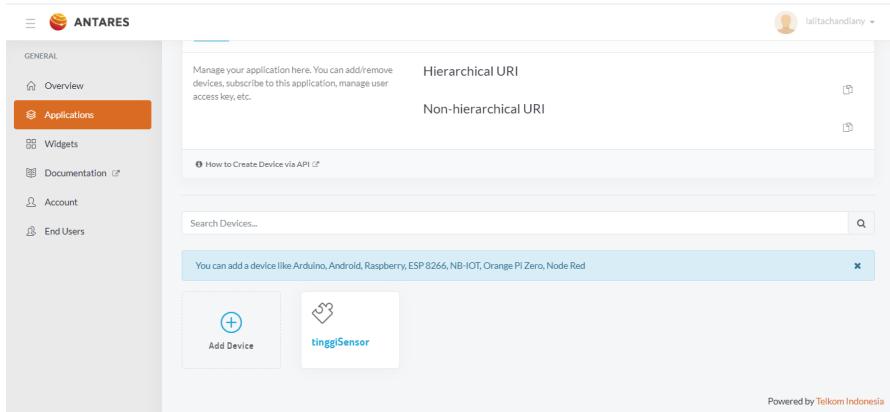
Gambar 3.65 Tampilan Membuat Device Antares

Gambar 3.65 merupakan sebuah tampilan untuk membuat *device* baru.



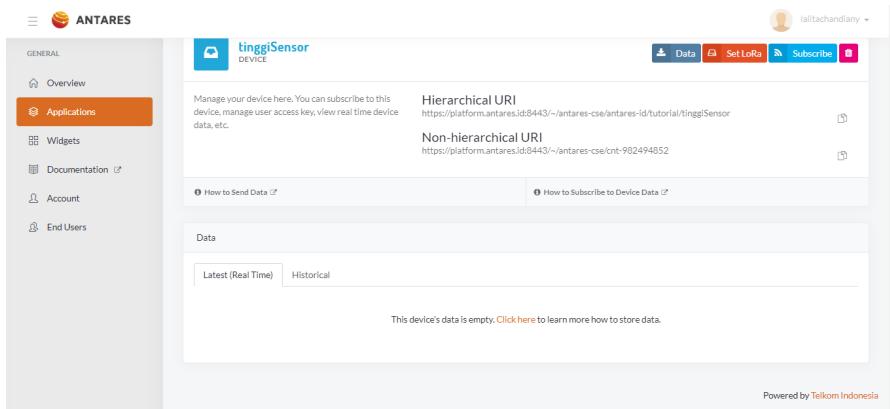
Gambar 3.66 Tampilan Membuat Device Antares

Setelah itu masukan nama *devicename* disini menggunakan nama "tinggisen-sor". Dimana untuk nama *device* ini bisa bebas sesuai dengan kebutuhan. Jika sudah mengisi nama *device* klik *add*.



Gambar 3.67 Tampilan Device Di Application Antares

Sesudah membuat *device* pada *application* maka tampilannya seperti pada gambar diatas. Apabila kita ingin mengetahui isi apa saja yang ada *device* , maka klik nama *devicenya*.



Gambar 3.68 Tampilan Halaman Device Antares

Pada gambar tersebut merupakan sebuah tampilan *device* tinggisensor. Dimana pada *device* ini datanya masih kosong karna belum disambungkan dengan project.

4. Memprogram Komponen Dengan Mengkoneksikan Pada Antares

Setelah kita membuat sebuah *project* atau *application* di antares, maka sekarang membuat sebuah program yang datanya bisa disimpan di antares. Dimana data tersebut dapat terkoneksi dengan antares. Dengan cara menambahkan code seperti berikut :

- (a) Pertama kita harus mengimport library dan menginisiasi pin yang digunakan , dengan cara menambahkan *code* berikut :

```

1 #include <AntaresESP8266HTTP.h>
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

```

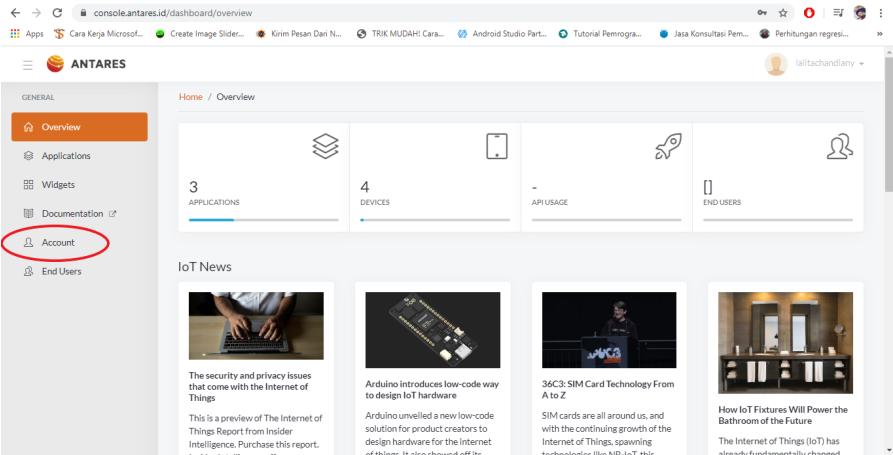
- (b) Kedua setelah kita menginisiasi pin yang digunakan yaitu mendefinisikan variabel, dengan menambahkan *code* berikut :

```

1 // defines pins numbers
2 #define ACCESSKEY "xxx"
3 #define WIFISSID "xxx"
4 #define PASSWORD "1231alita"
5
6 #define projectName "nama project diantares"
7 #define deviceName "nama device diantares"
8 AntaresESP8266HTTP antares(ACCESSKEY);
9
10 // defines variables
11 int duration;
12 int distance;
13 int safetyDistance;

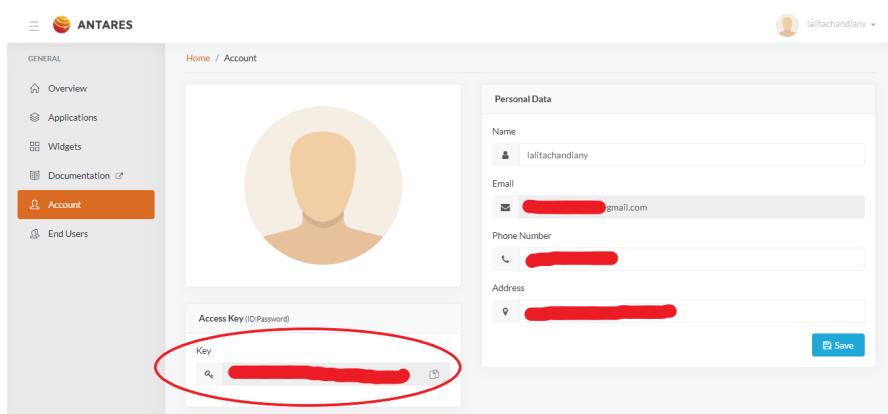
```

Pada saat pendefinisian variabel harus memasukan *acces key* ,WIFI SSID (nama wifi yang akan dikoneksikan), *password* (kata sandi dari wifi SSID) wifi ini digunakan untuk koneksi nodemcu. Sedangkan untuk *acces key* didapatkan dari antares dengan cara sebagai berikut :



Gambar 3.69 Tampilan Antares

Pertama *login* antares ketika muncul halaman utama pilih *users*.



Gambar 3.70 Tampilan Users Antares

Setelah masuk ke halaman *users copy key* tersebut, kemudian di *paste* dibagian pendefinisian variabel.

- (c) Ketika telah mendefinisikan varial , saatnya masuk bagian void setup, dimana void setup ini dieksekusi pertamakali dan hanya sekali dieksekusi hanya pada saat program dijalankan. Selain itu juga pada bagian void setup ini mengatur *inputan* dan *output* berikut codenya :

```

1 void setup() {
2   Serial.begin(115200);
3   antares.setDebug(true);
4   antares.wifiConnection(WIFISSID,PASSWORD);
5   pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
6   pinMode(echoPin, INPUT); // Sets the echoPin as an Input
7   pinMode(buzzerPin, OUTPUT);
8   pinMode(ledPin, OUTPUT);
9 }
10 }
```

- (d) Jika kita telah membuat voidsetup, saatnya membuat void loop. Dimana void loop ini akan dieksekusi terus menerus ketika program dijalankan . Berikut code daei looping sensro ultrasonic , buzzer dan led.

```

1 void loop() {
2
3   // Clears the trigPin
4   digitalWrite(trigPin, LOW);
5   delayMicroseconds(2);
6
7   // Sets the trigPin on HIGH state for 10 micro seconds
8   digitalWrite(trigPin, HIGH);
9   delayMicroseconds(10);
10  digitalWrite(trigPin, LOW);
11
12 // Reads the echoPin, returns the sound wave travel time in
13 // microseconds
```

```

13 duration = pulseIn(echoPin , HIGH);
14
15 // Calculating the distance
16 distance= duration*0.034/2;
17
18
19
20 safetyDistance = distance;
21 if (safetyDistance <= 10){
22     digitalWrite(buzzerPin , HIGH);
23     digitalWrite(ledPin , HIGH);
24 }
25 else {
26     digitalWrite(buzzerPin , LOW);
27     digitalWrite(ledPin , LOW);
28 }
```

- (e) Setelah membuat looping mengenai sensor, buzzer dan led, kemudian membuat code untuk ke antaresnya sebagai berikut :

```

1 antares.add("distance" , distance);
2 antares.add("duration" , duration);
3 antares.send(projectName , deviceName);
```

- (f) Terakhir yaitu membuat sebuah code untuk menampilkannya di serial monitor dengan cara membuat code berikut ini :

```

1 // Prints the distance on the Serial Monitor
2 Serial.print("Distance: ");
3 Serial.println(distance);
4 Serial.print("pulseIn ");
5 Serial.print(duration);
6 }
```

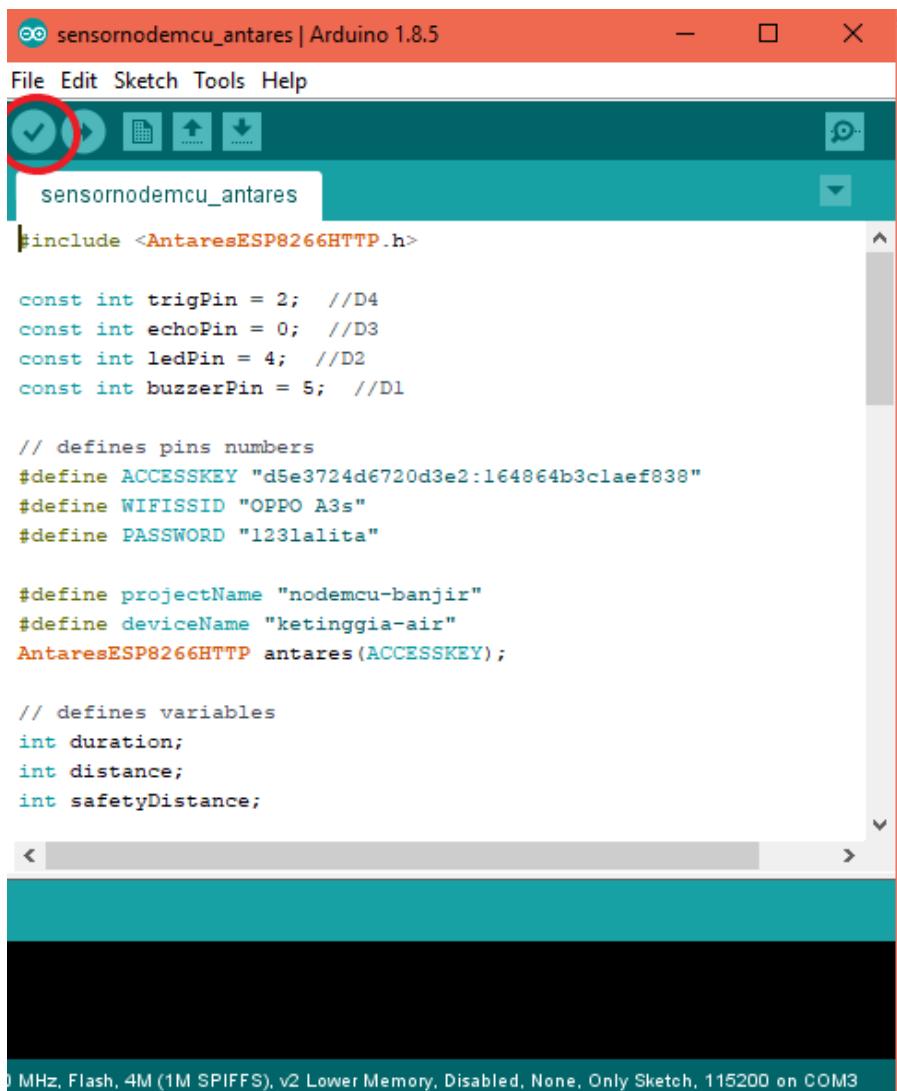
Jika kita gabungkan code dari awal yaitu mengimport library sampedengan menampilkan di serial monitor akan seperti berikut ini :

```

1 #include <AntaresESP8266HTTP.h>
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 // defines pins numbers
9 #define ACCESSKEY "xxx"
10 #define WIFISSID "xxx"
11 #define PASSWORD "xxx"
12
13 #define projectName "nama project"
14 #define deviceName "nama device"
15 AntaresESP8266HTTP antares(ACCESSKEY);
16
17 // defines variables
18 int duration;
19 int distance;
```

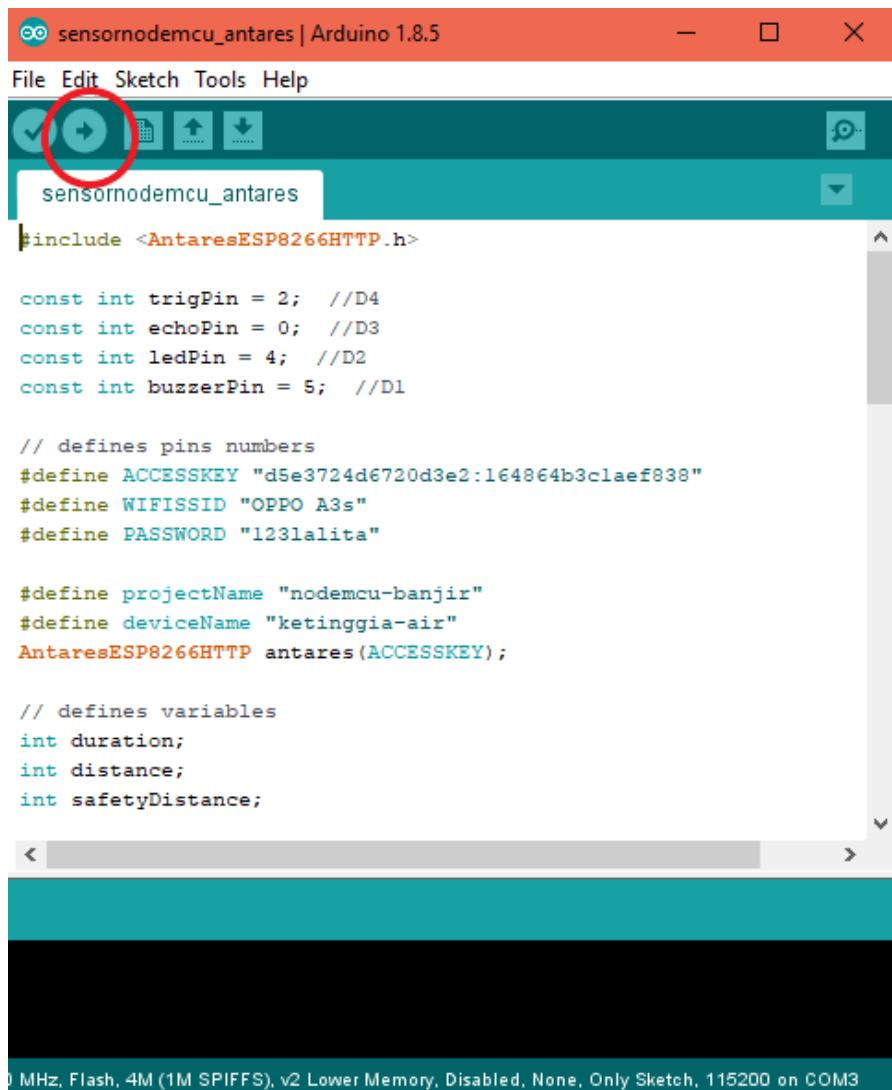
```
20 int safetyDistance;
21
22 void setup() {
23     Serial.begin(115200);
24     antares.setDebug(true);
25     antares.wifiConnection(WIFISSID,PASSWORD);
26     pinMode(trigPin , OUTPUT); // Sets the trigPin as an Output
27     pinMode(echoPin , INPUT); // Sets the echoPin as an Input
28     pinMode(buzzerPin , OUTPUT);
29     pinMode(ledPin , OUTPUT);
30 }
31
32 void loop() {
33
34 // Clears the trigPin
35 digitalWrite(trigPin , LOW);
36 delayMicroseconds(2);
37
38 // Sets the trigPin on HIGH state for 10 micro seconds
39 digitalWrite(trigPin , HIGH);
40 delayMicroseconds(10);
41 digitalWrite(trigPin , LOW);
42
43 // Reads the echoPin , returns the sound wave travel time in
44 // microseconds
45 duration = pulseIn(echoPin , HIGH);
46
47 // Calculating the distance
48 distance= duration*0.034/2;
49
50 safetyDistance = distance;
51 if (safetyDistance <= 10){
52     digitalWrite(buzzerPin , HIGH);
53     digitalWrite(ledPin , HIGH);
54 }
55 else{
56     digitalWrite(buzzerPin , LOW);
57     digitalWrite(ledPin , LOW);
58 }
59
60
61 antares.add("distance" , distance);
62 antares.add("duration" , duration);
63 antares.send(projectName , deviceName);
64
65 // Prints the distance on the Serial Monitor
66 Serial.print("Distance: ");
67 Serial.println(distance);
68 Serial.print("pulseIn ");
69 Serial.print(duration);
70 }
```

Setelah semuanya komponen di program dan sudah dikoneksi dengan antares saatnya mengupload file dengan cara berikut :



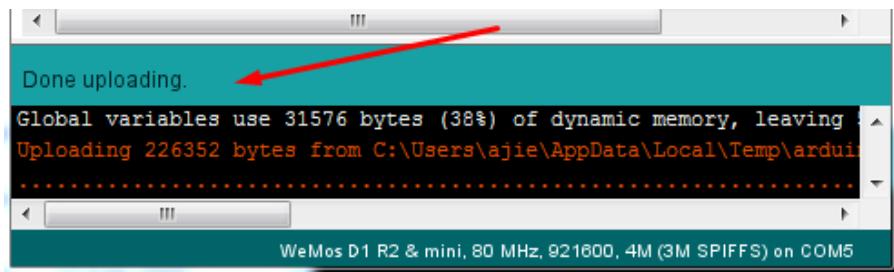
Gambar 3.71 Verifikasi Arduiono IDE

Sebelum mengupload ada baiknya kita memverifikasi terlebih dahulu . Verifikasi ini bertujuan untuk mengecek apakah ada error atau tidaknya. Cara untuk memverifikasi dengan cara klik centang seperti pada gambar diatas.



Gambar 3.72 Upload Arduiono IDE

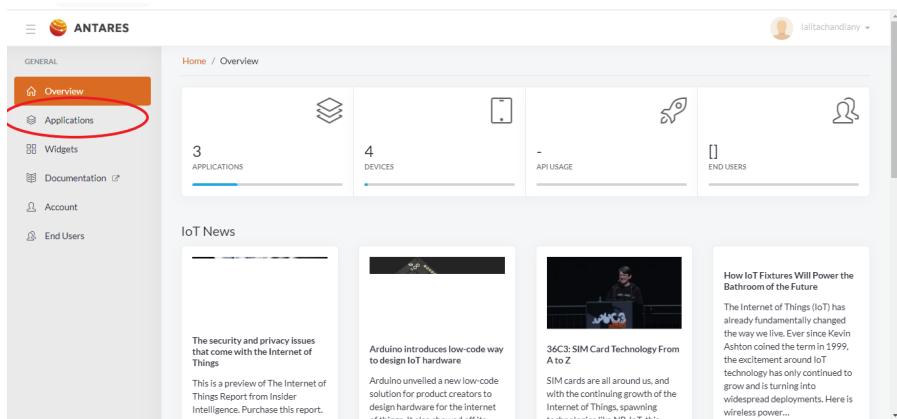
Kemudian setelah melakukan verifikasi dan apabila tidak ada error maka langsung untuk mengupload code yang sebelumnya telah dibuat klik gambar panah pada arduino IDE seperti pada gambar diatas.



Gambar 3.73 Upload Arduiono IDE

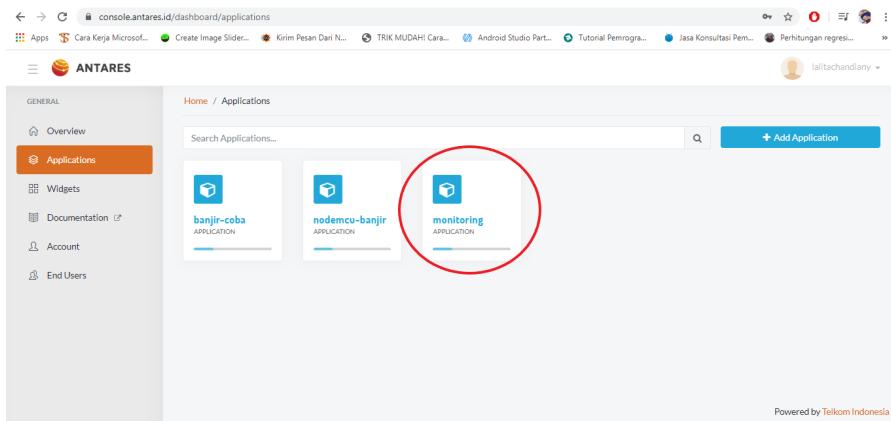
Jika proses *upload* sudah selesai , maka akan adanya tulisan *done upload* dibagian pojok kanan bawah.

Setelah upload maka semua komponen akan menjalankan perintahnya sesuai code yang telah dibuat sebelumnya. Kemudian untuk melihat data yang dibaca oleh sensor ultrasonik kemudian dikirimkan ke antares sebagai berikut :



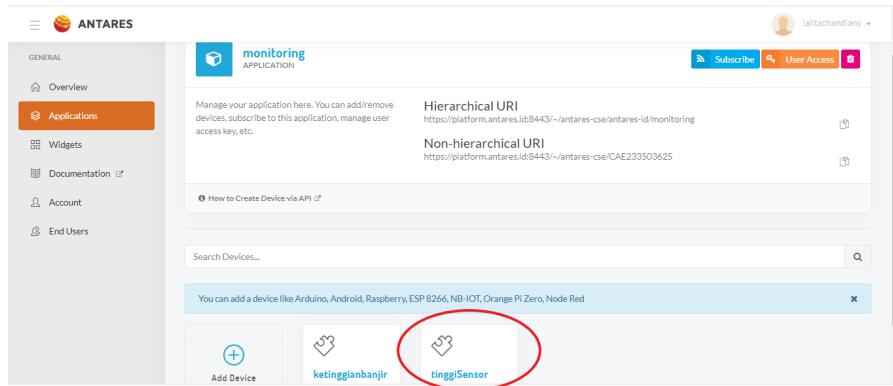
Gambar 3.74 Halaman Utama Antares

Hal pertama yang harus dilakukan yaitu *login* antares, kemudian akan muncul tampilan seperti gambar diatas dan pilih menu *application*.



Gambar 3.75 Tampilan Application

Setelah pilih menu *application* maka akan tampil halaman seperti diatas kemudian pilih *project* yang telah kita buat serta dikoneksikan dengan *project prototype* ini yaitu pilih yang bernama monitoring.



Gambar 3.76 Tampilan Project

Ketika kita telah memilih *projectnya* maka akan masuk ke tampilan yang kita harus memilih *device* yang telah dikoneksikan sebelumnya yaitu nama *device* *cennya* tinggi sensor.

The screenshot shows the ANTARES application interface. On the left, there's a sidebar with 'GENERAL' and several menu items: 'Overview', 'Applications' (which is highlighted in orange), 'Widgets', 'Documentation', 'Account', and 'End Users'. The main area displays a table of data with three rows. Each row contains a timestamp, a resource index, and a JSON-formatted data payload.

Time	Resource Index (ri)	Data
2019-12-13 18:29:36	/antares-cse/cin-699331610	{ "distance": 9, "duration": 543 }
2019-12-13 18:29:53	/antares-cse/cin-821588716	{ "distance": 9, "duration": 543 }
2019-12-13 18:30:09	/antares-cse/cin-695670066	{ "distance": 0, "duration": 5 }
2019-12-13 18:30:32	/antares-cse/cin-662747050	

Gambar 3.77 Tampilan Data Sensor Ultrasonic

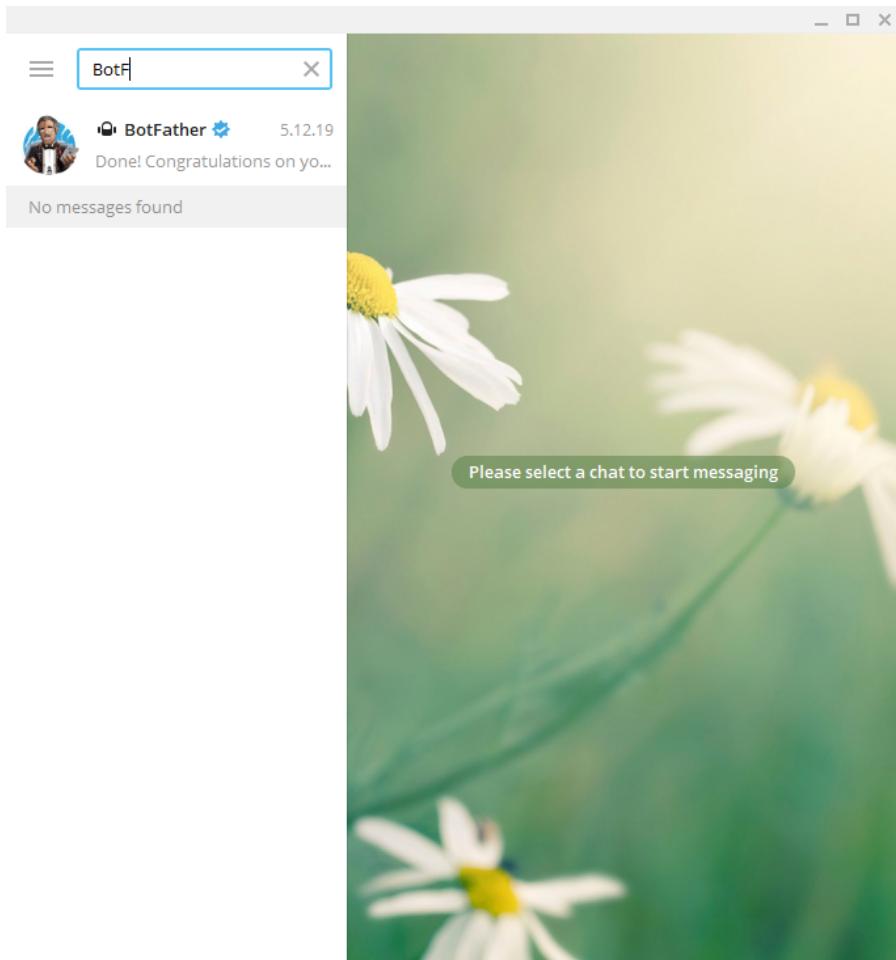
Gambar diatas merupakan sebuah data yang dibaca oleh sensor ultrasonic yang dikirimkan ke antares secara *realtime*.

5. Membuat Notifikasi Melalui Telegram.

Pada *prototype* prediksi ketingian air (PKA) untuk mendeteksi banjir peringatan dini ini menggunakan notifikasi melalui telegram. Notifikasi ini akan dikirimkan ketika sensor membaca jarak ketinggian air kurang dari 5 meter.

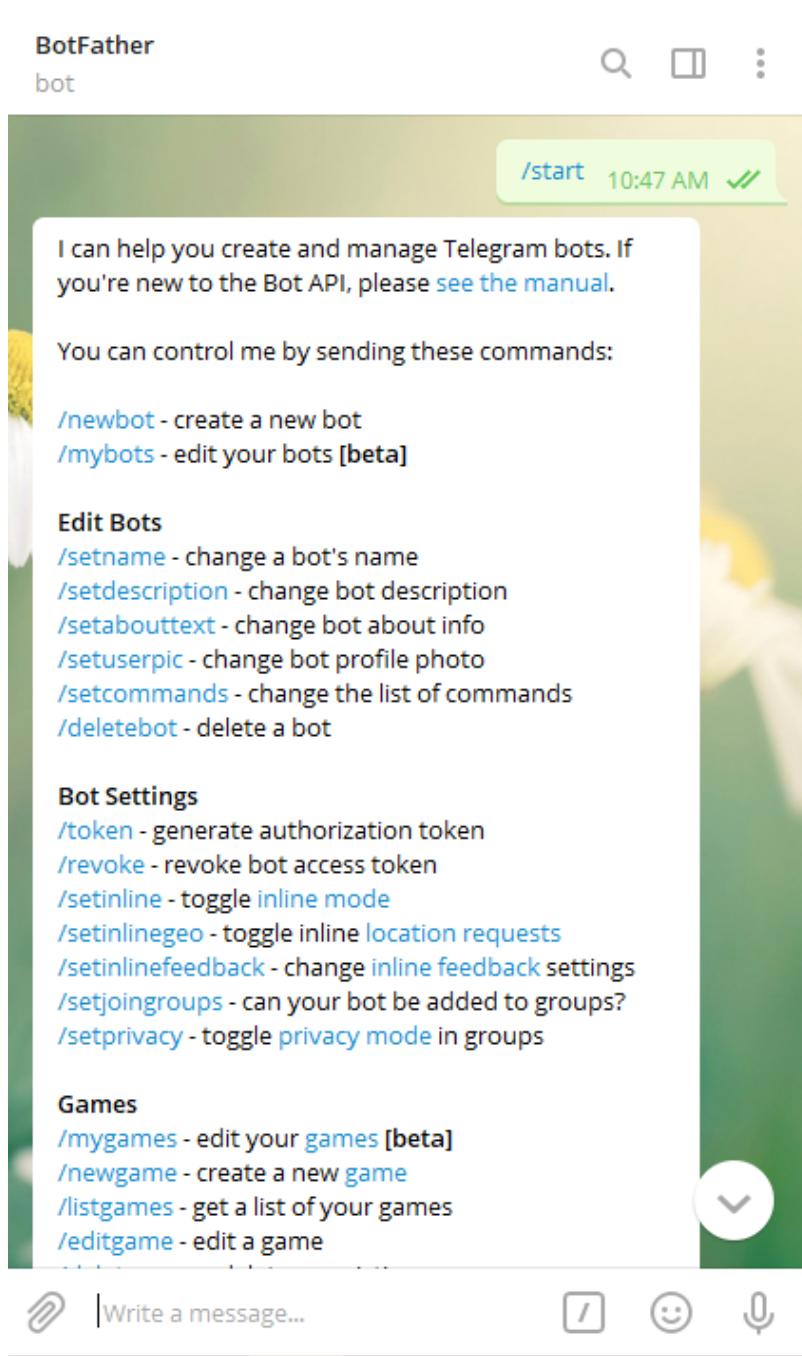
Untuk membuat sebuah notifikasi melalui telegram ini kita perlu memprogramnya melalui arduino IDE. Sebelum memprogramnya kita harus membuat sebuah *bot* ditelegram. Dengan cara berikut :

- (a) Buka aplikasi telegram kemudikan ketikan BotFather.



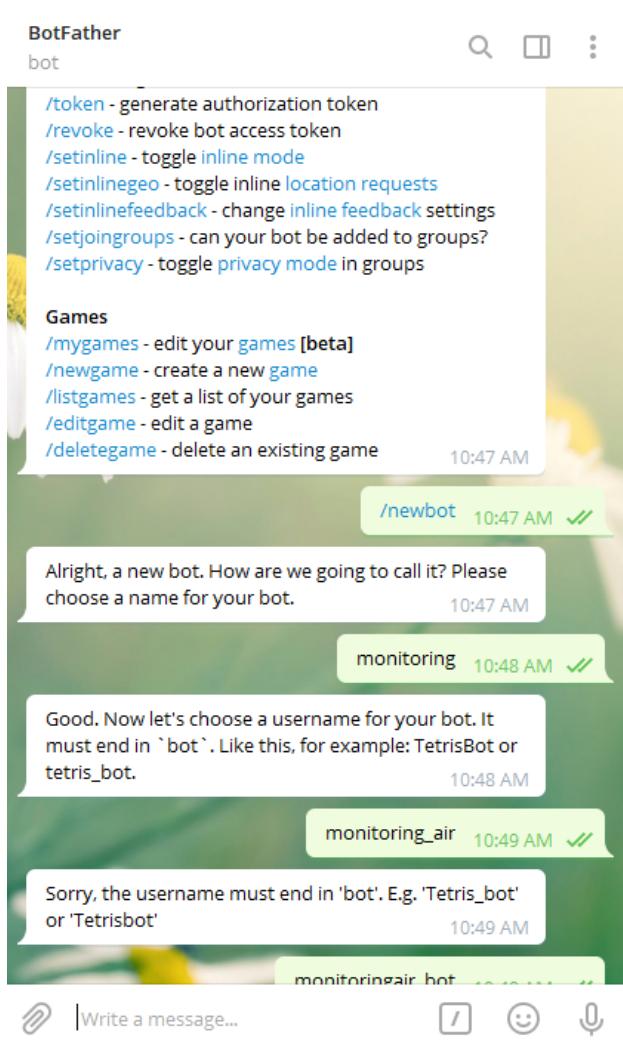
Gambar 3.78 Tampilan Telegram

- (b) Lalu buka BotFather, ketik START.

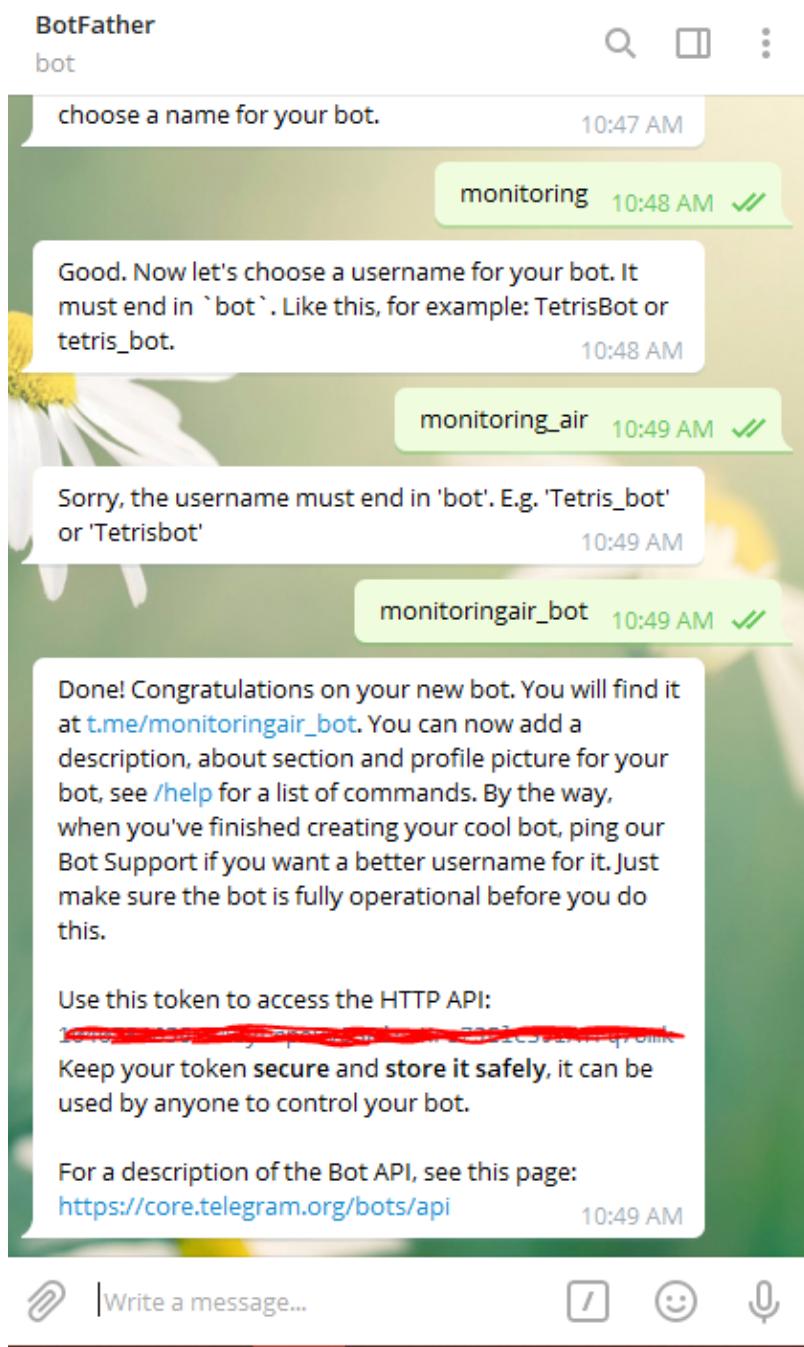


Gambar 3.79 Tampilan Bot Telegram

- (c) Lalu ketik /newbot , selanjutnya akan diminta memberikan nama bot dan username bot. Jika sudah makan akan muncul Token, seperti yang dilingkari dibawah ini. Simpan Token tersebut.

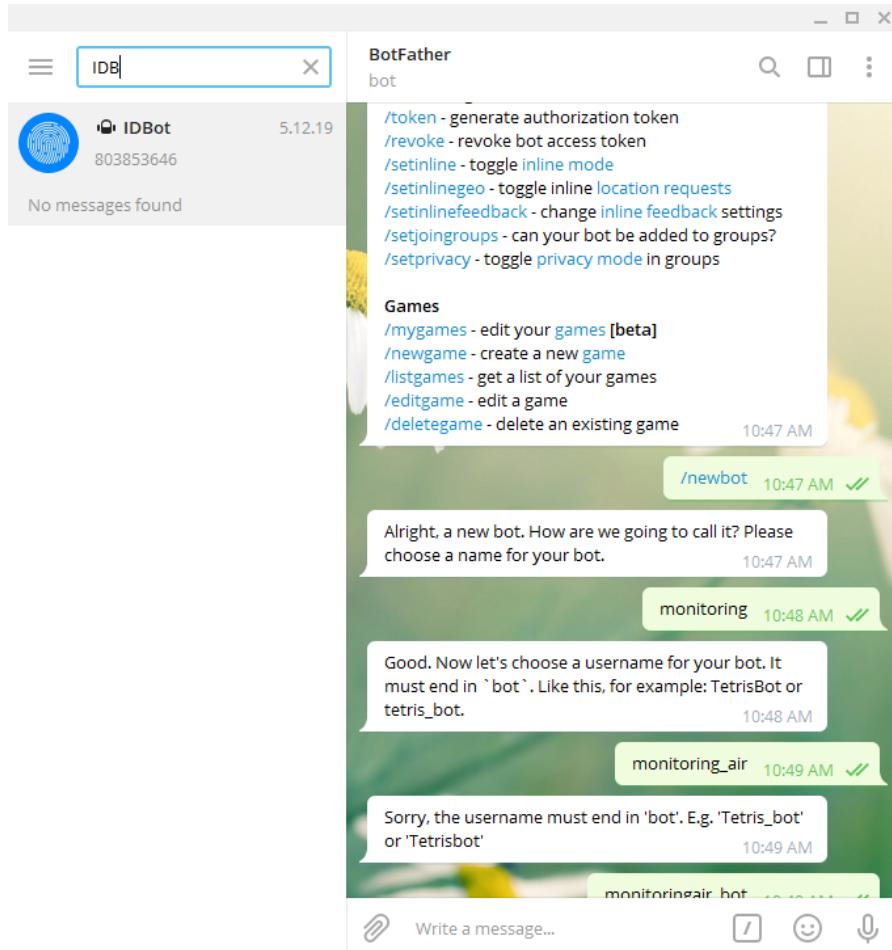


Gambar 3.80 Tampilan Bot Telegram



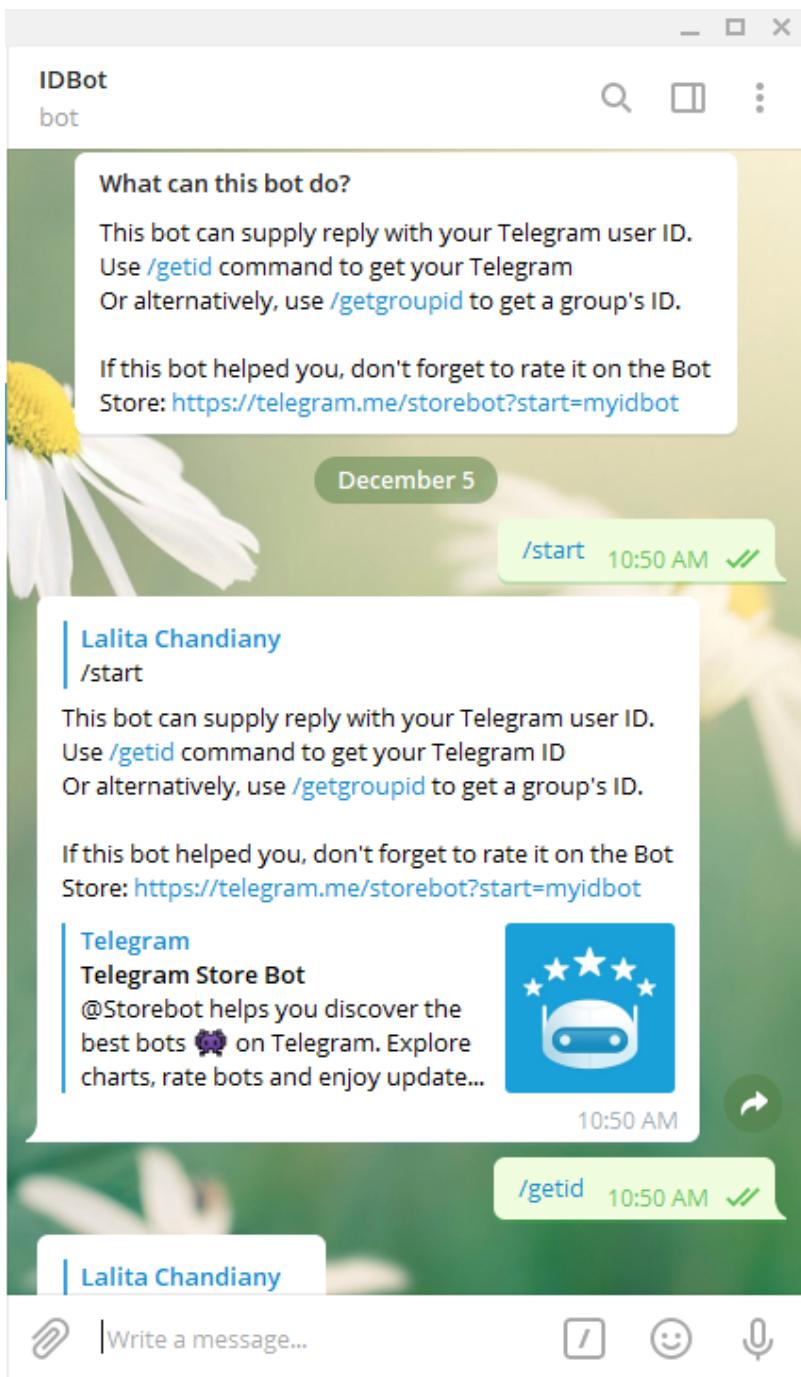
Gambar 3.81 Tampilan Bot Telegram

- (d) Selain BotFather kita juga harus mengetahui idnya yaitu cari IDBot.

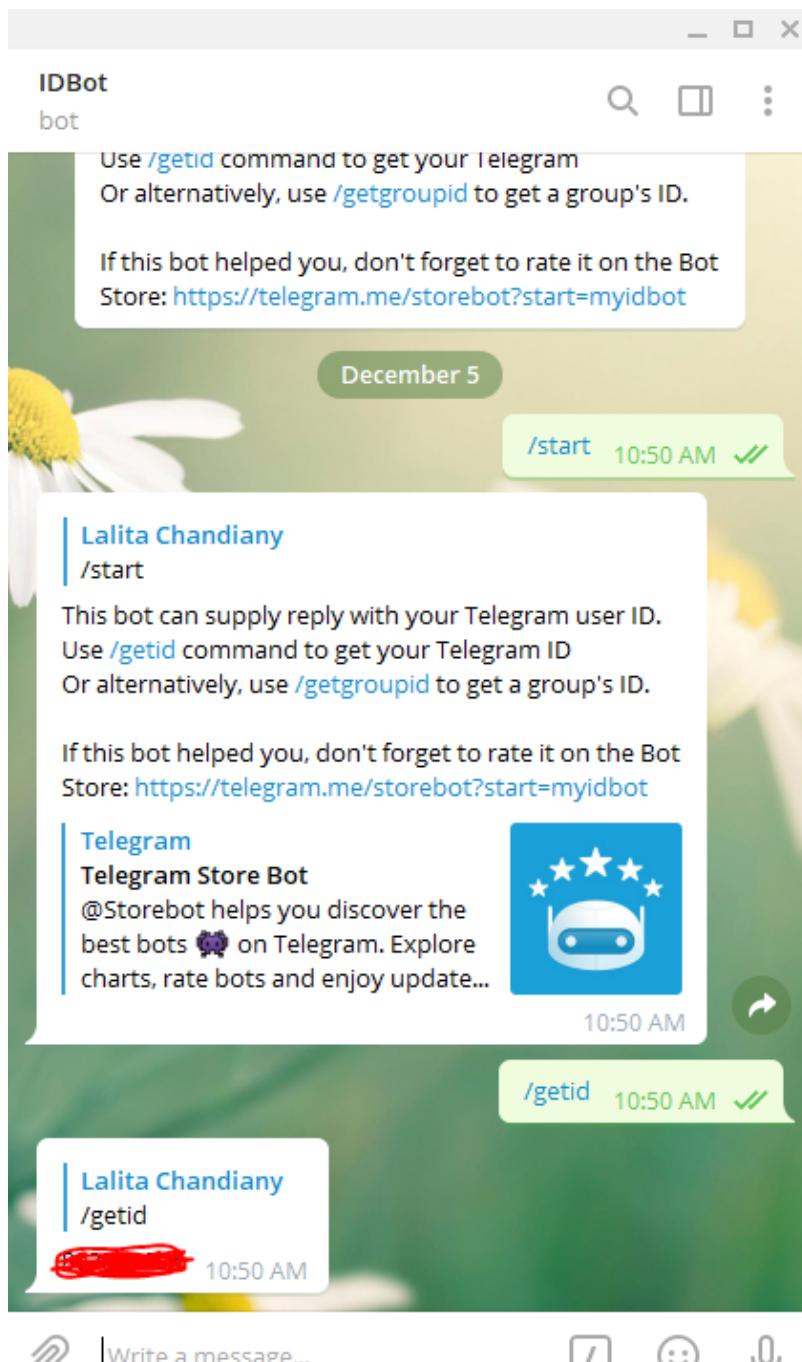


Gambar 3.82 Tampilan IDBot Telegram

- (e) Klik /start , lalu ketik /getid. Nanti akan muncul id telegram kamu seperti yang dibawah ini:

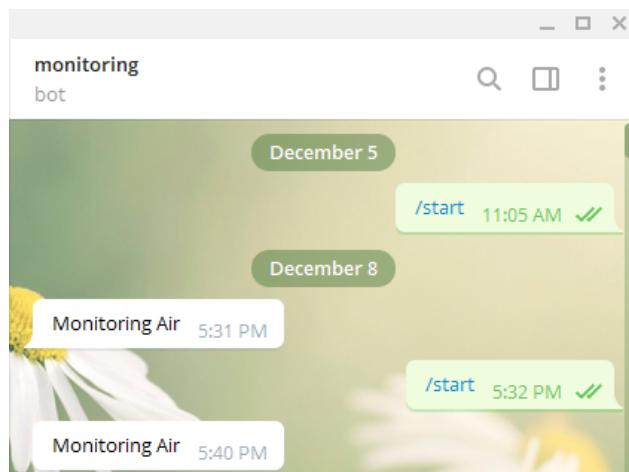


Gambar 3.83 Tampilan IDBot Telegram



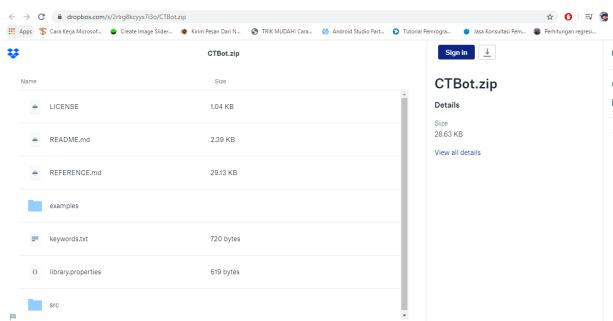
Gambar 3.84 Tampilan IDBot Telegram

Setelah membuat bot dan juga sudah mengetahui Token dan id telegram agan. Selanjutnya yang tidak kalah penting perlu masuk ke Bot yang sudah buat dan klik /start.

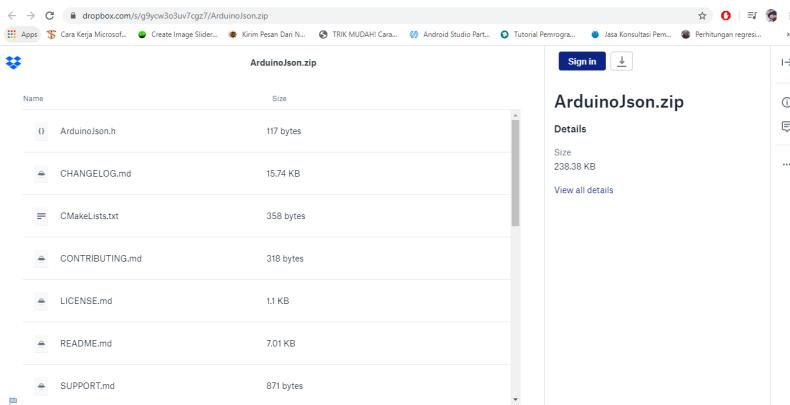


Gambar 3.85 Tampilan Bot Monitoring Di Telegram

Setelah bot dibuat maka selanjutnya kita harus mendownload libriray untuk telegram. Kita dapat mendownloadnya di url <https://www.dropbox.com/s/2rlrg8kcyyx7i3o/CTBot.zip> untuk ctbot dan <https://www.dropbox.com/s/g9ycw3o3uv7cgz7/ArduinoJson.zip> untuk arduino json.



Gambar 3.86 Tampilan Download CTbot



Gambar 3.87 Tampilan Dwonload Arduino Json

Setelah mendownload maka saatnya memprogram melalui arduino IDE yaitu sebagai berikut :

- (a) Langkah pertama yaitu *import* library yang telah didownload. Sehingga akan muncul secara otomatis code berikut :

```
1 #include <CTBot.h>
```

- (b) Kemudian menginisiasi sebagai berikut kodennya.

```
1 CTBot myBot;
2 String token = "xxx";
3 const int id = xxx;
```

- (c) Kemudian mendefinisikan variabel.

```
1 int duration;
2 int distance;
3 int safetyDistance;
4 boolean kirimTBot = true;
5 String jarak = "";
6
7 String jsonString = "{}";
```

- (d) Membuat sebuah inputan maupun outputan dibagian void setup.

```
1 void setup() {
2
3
4     Serial.begin(115200);
5     myBot.wifiConnect(WIFISSID, PASSWORD);
6     myBot.setTelegramToken(token);
7     delay(1000);
8
9     antares.setDebug(true);
10    antares.wifiConnection(WIFISSID, PASSWORD);
```

```

11  pinMode(trigPin , OUTPUT); // Sets the trigPin as an Output
12  pinMode(echoPin , INPUT); // Sets the echoPin as an Input
13  pinMode(buzzerPin , OUTPUT);
14  pinMode(ledPin , OUTPUT);
15
16 }
17 }
```

- (e) Membuat Json nya agar dapat mengirimkan notifikasi melalui telegram.

```

1  isiJson("distance" , distance);
2  isiJson("duration" , duration);
3
4  // Serial.println(jsonString);
5  antares.isi(jsonString);
6  antares.send(projectName , deviceName);
```

- (f) Membuat void json nya dengan code sebagai berikut :

```

1 void isiJson(String key , int value) {
2   DynamicJsonBuffer jsonBuffer;
3   JsonObject& object = jsonBuffer.parseObject(jsonString);
4   object[key] = value;
5   String newInsert;
6   object.printTo(newInsert);
7   jsonString = newInsert;
8 }
```

- (g) Jika codenya digabungkan dari awal sampai menggunakan bot telegram yaitu sebagai berikut :

```

1 #include <CTBot.h>
2 #include <ArduinoJson.h>
3 #include <AntaresESP8266HTTP.h>
4
5 #define WIFISSID "xxx"
6 #define PASSWORD "xxx"
7
8 #define ACCESSKEY "xxx"
9 #define projectName "nama project"
10 #define deviceName "nama device"
11
12 AntaresESP8266HTTP antares(ACCESSKEY);
13
14
15 CTTBot myBot;
16 String token = "xxx";
17 const int id =xxx;
18
19 // defines variables
20 int duration;
21 int distance;
22 int safetyDistance;
23 boolean kirimTBot = true;
24 String jarak = "";
25
26 String jsonString = "{}";
```

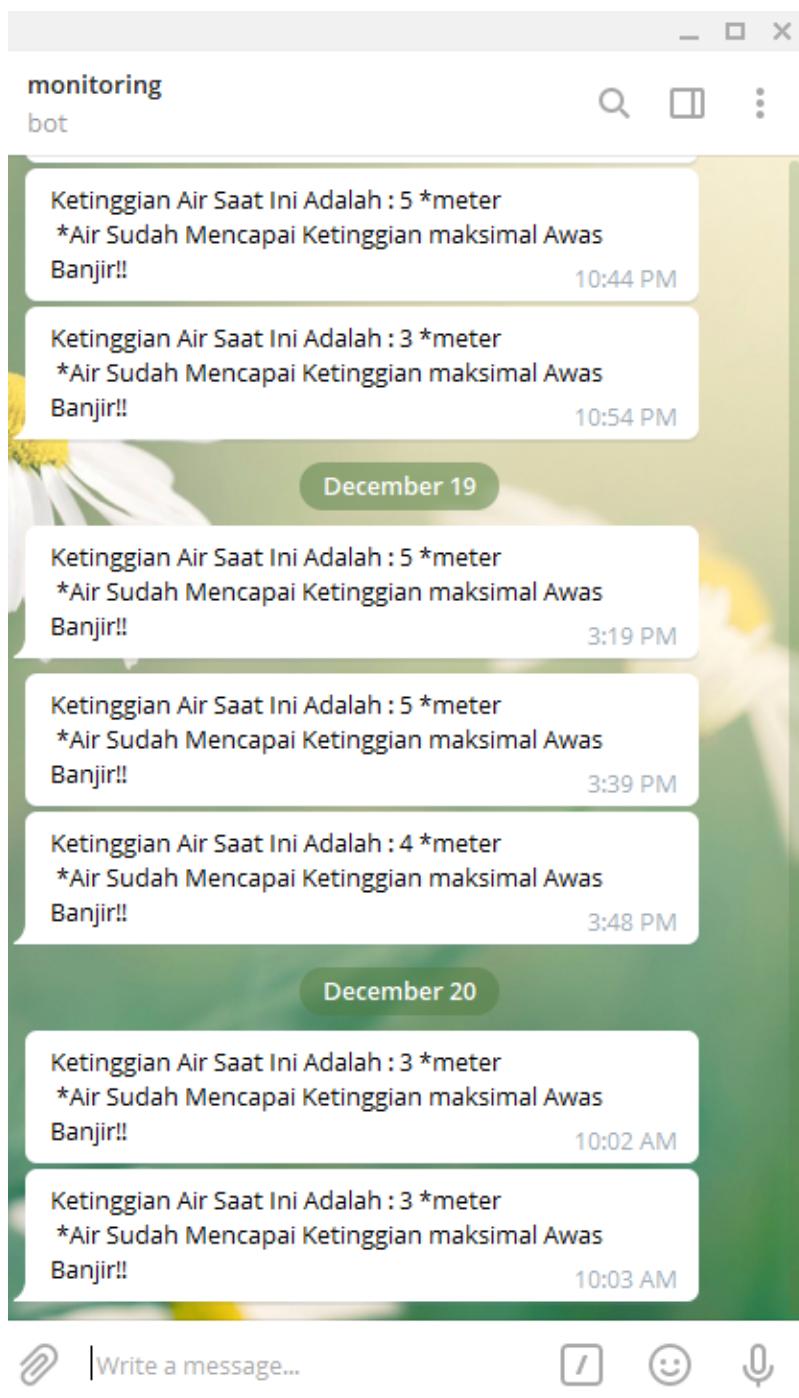
```
27 // defines pins numbers
28 const int trigPin = 2; //D4
29 const int echoPin = 0; //D3
30 const int ledPin = 4; //D2
31 const int buzzerPin = 5; //D1
32
33
34 void setup() {
35
36
37 Serial.begin(115200);
38 myBot.wifiConnect(WIFISSID, PASSWORD);
39 myBot.setTelegramToken(token);
40 delay(1000);
41
42 antares.setDebug(true);
43 antares.wifiConnection(WIFISSID, PASSWORD);
44
45 pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
46 pinMode(echoPin, INPUT); // Sets the echoPin as an Input
47 pinMode(buzzerPin, OUTPUT);
48 pinMode(ledPin, OUTPUT);
49
50 }
51
52 // the loop function runs over and over again forever
53 void loop() {
54     // Clears the trigPin
55     digitalWrite(trigPin, LOW);
56     delayMicroseconds(2);

57     // Sets the trigPin on HIGH state for 10 micro seconds
58     digitalWrite(trigPin, HIGH);
59     delayMicroseconds(10);
60     digitalWrite(trigPin, LOW);

61     // Reads the echoPin, returns the sound wave travel time in
62     // microseconds
63     duration = pulseIn(echoPin, HIGH);

64     // Calculating the distance
65     distance = duration * 0.034 / 2;
66
67
68
69
70
71     isiJson("distance", distance);
72     isiJson("duration", duration);
73
74     // Serial.println(jsonString);
75     antares.isi(jsonString);
76     antares.send(projectName, deviceName);
77
78
79     safetyDistance = distance;
80     if (safetyDistance <= 5) {
```

```
82 jarak = "Ketinggian Air Saat Ini Adalah : ";
83 jarak += int(distance);
84 jarak += " *meter\n";
85 jarak += " *Air Sudah Mencapai Ketinggian maksimal Awas
Banjir!! \n";
86 if (kirimTBot) {
87     myBot.sendMessage(id, jarak, "");
88     kirimTBot = false;
89 }
90 digitalWrite(buzzerPin, HIGH);
91 digitalWrite(ledPin, HIGH);
92 Serial.println("Banjir");
93 }
94 else {
95     digitalWrite(buzzerPin, LOW);
96     digitalWrite(ledPin, LOW);
97     kirimTBot = true;
98     Serial.println("Aman");
99 }
100 }
101
102 }
103 }
104
105 void isiJson(String key, int value) {
106     DynamicJsonBuffer jsonBuffer;
107     JsonObject& object = jsonBuffer.parseObject(jsonString);
108     object[key] = value;
109     String newInsert;
110     object.printTo(newInsert);
111     jsonString = newInsert;
112 }
```



Gambar 3.88 Tampilan Notifikasi DI Telegram

Setelah membuat code untuk notifikasi di telegram. Selanjutnya upload program tersebut untuk mengetahui tidak error jika tidak error maka mengirimkan notifikasi awas banjir. Apabila sensor ultrasonik membaca jarak ketinggian air kurang dari 5 meter. Gambar diatas merupakan isi notifikasi dari bot telegram yang sebelumnya sudah dibuat.

BAB 4

MEMBUAT APLIKASI MONITORING

4.1 Pengenalan Android Dan Android Studio

4.1.1 Android



Gambar 4.1 Logo Android

Android adalah sistem operasi yang dirancang oleh Google dengan basis kernel Linux untuk mendukung kinerja perangkat elektronik layar sentuh, seperti tablet atau smartphone. Jadi, android digunakan dengan sentuhan, gesekan ataupun ketukan pada layar gadget anda.

Android salah satu sistem operasi atau *operating system* berbasis *mobile* yang sangat banyak di gunakan sekarang ini. Utamanya pada telepon pintar (*smartphone*) ataupun tablet.

Android bersifat *open source* atau bebas digunakan, dimodifikasi, diperbaiki dan didistribusikan oleh para pembuat ataupun pengembang perangkat lunak. Dengan sifat *open source* perusahaan teknologi bebas menggunakan OS ini diperangkatnya tanpa lisensi alias gratis.

Kelebihan Android yaitu sebagai berikut :

1. Merupakan Sistem Operasi Open Source

Siapa saja bisa menggunakannya secara gratis. Para *developer* atau pengembang dimudahkan untuk mengoptimalkan dan mengembangkan OS ini untuk *smartphone* yang dibuatnya.

2. Memiliki Banyak Dukungan Aplikasi

Hal ini juga tidak lepas dari sifat Android yang merupakan sistem operasi *Open Source*. Pengembang pun diizinkan untuk mengembangkan aplikasi berbasis source code dari Android.

Oleh karena itu, jika Anda masuk ke *Play Store*, akan ditemukan banyak sekali ribuan aplikasi yang sesuai dengan kebutuhan pengguna.

3. Mudah dimodifikasi

Banyak komponen yang bisa Anda atur ulang atau dimodifikasi, mulai dari ROM hingga custom *overclock* pada sistem operasi. Hal ini bisa berpengaruh terhadap performa ponsel pintar berbasis Android agar bisa bekerja lebih cepat dan sesuai dengan keinginan.

Kekurangan Android yaitu sebagai berikut :

1. Kerja sistemnya cukup berat

Hal ini menyebabkan banyak memori yang dibutuhkan, baik RAM maupun ROM. Bagi *smartphone* yang memiliki RAM dan ROM berkapasitas kecil, tentunya akan menghambat performanya.

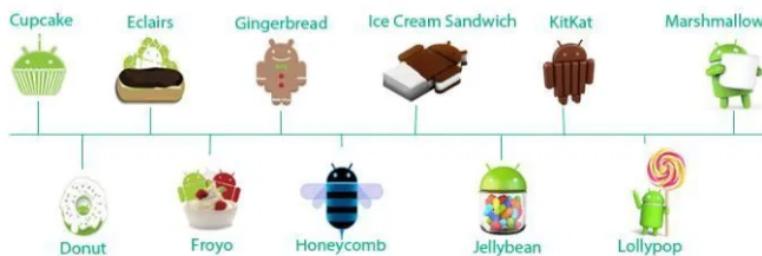
2. Hasil modifikasi sering menyebabkan sistem bekerja tidak stabil dan kurang optimal

Adakalanya hasil modifikasi mengakibatkan OS menjadi sedikit lelet dan kurang responsif. Nantinya, bisa berpengaruh pada hardware sehingga menjadi cepat panas dan kapasitas memori lebih mudah bocor.

3. Kurang responsif jika disandingkan dengan spesifikasi hardware yang tidak baik

Hal tersebut berkaitan dengan kapasitas RAM, ROM, dan kecepatan prosesor yang digunakan pada *smartphone*.

4.1.2 Perkembangan Android



History of Android Operating System

Gambar 4.2 Perkembangan Android

Sejak tahun 2009, Android mulai dikembangkan dengan kode yang dinamai berdasarkan makanan pencuci mulut. Tiap versi dirilis sesuai dengan urutan abjad.

1. Astro 1.0

Versi ini pertama kali dirilis pada 23 September 2008 yang awalnya akan dinamai dengan nama Astro saja. Namun karena alasan hak cipta dan trademark, nama ini tidak jadi disematkan pada versi pertama ini. Versi Astro 1.0 pertama kali digunakan oleh smartphone HTC Dream.

2. Bender 1.1

Bender 1.1 dirilis pada 9 Februari 2009. Lagi-lagi, versi dari OS ini mengalami masalah penamaan yang serupa dengan versi sebelumnya. Awalnya, versi ini diberi nama Bender dan dirilis untuk perangkat T-Mobile G1 saja.

3. Cupcake 1.5

Cupcake 1.5 dirilis pada 30 April 2009. Dimulai dari versi ini, penamaan menggunakan nama makanan pencuci mulut. Karena merupakan versi ketiga, makan penamaannya dimulai dengan huruf C dan Cupcake menjadi nama resminya.

4. Donut 1.6

Versi yang dirilis pada 15 September 2009 ini memiliki peningkatan pada fitur pencarian dan UI yang lebih user friendly. Donut 1.6 sudah mendukung teknologi CDMA/EVDO, 802.1 x, VPNs.

5. Eclair 2.0 2.1

Eclair 2.0 2.1 dirilis pada 3 Desember 2009 dan untuk pertama kalinya membawa fitur baru, yaitu Google Maps yang dapat membantu pengguna dalam bepergian.

6. Froyo 2.2

Froyo atau disingkat dari frozen yoghurt merupakan versi Android yang rilis pada 20 Mei 2010.

Perubahan umumnya antara lain adalah adanya dukungan Adobe Flash 10.1, kecepatan kinerja, integrasi V8 JavaScript engine, pemasangan aplikasi dalam SD Card, kemampuan Wi-Fi Hotspot portable, dan kemampuan auto update dalam aplikasi Android Market.

7. Gingerbread 2.3

Versi ini dirilis pada 6 Desember 2010 dan terdapat perubahan dalam peningkatan kemampuan gaming, peningkatan fungsi copy paste, User Interface, dukungan format video VP8 dan WebM, hingga dukungan jumlah kamera lebih dari satu.

8. Honeycomb 3.0/3.1

Versi yang diluncurkan pada 22 Februari 2011 ini merupakan OS yang didesain khusus untuk pengoptimalan penggunaan pada tablet PC. Versi Honeycomb ini juga mendukung multi prosesor dan akselerasi hardware untuk grafis.

9. Ice Cream Sandwich 4.0

Ice Cream Sandwich 4.0 diluncurkan tanggal 19 Oktober 2011 dan membawa fitur Honeycomb untuk smartphone dengan membawa fitur baru, seperti membuka kunci dengan pengenala wajah, perangkat tambahan fotografi, hingga berbagi informasi menggunakan NFC.

10. Jelly Bean 4.1/4.2/4.3

Di tahun 2012, android mengeluarkan versi Jelly Bean. Lewat versi Jelly Bean (4.1) Google mulai menerapkan teknologi asisten digital Google Now yang bisa diakses langsung dari homescreen.

Pada versi 4.2 terdapat fitur photo sphere untuk panorama, daydream sebagai screensaver, power control, dsb. Sedangkan versi 4.3 merupakan pembaharuan dari versi sebelumnya.

11. KitKat 4.4

KitKat 4.4 diluncurkan pada 3 September 2013. Versi yang sebelumnya bernama Key Lime Pie ini membawa peningkatan yang cukup signifikan karena Google lebih fokus meningkatkan user experience.

Versi ini dioptimalkan untuk berjalan pada rentang yang lebih besar dari versi Android sebelumnya. Disarankan perangkat harus memiliki minimal RAM 512 MB.

12. Lollipop 5.0

Versi yang diluncurkan pada 12 November 2014 ini tersedia secara resmi melalui over the air (OTA). Perubahan yang paling menonjol dalam versi ini adalah User Interface yang didesain ulang dan dibangun dengan material design.

13. Marshmallow 6.0

Sistem operasi ini membawa banyak fitur canggih, mulai dari Doze untuk menghemat baterai, dukungan USB tipe C, percobaan multi window, sensor sidik jari untuk buka kunci layar, hingga pengguna bisa memakai dua aplikasi berbeda dalam satu layar.

14. Nougat 7.0

Versi ini merupakan salah satu upgrade terbesar dalam sistem operasi Android. Nougat 7.0 merupakan pengembangan dari Marshmallow yang meningkatkan performa dan interface yang lebih intuitif.

15. Oreo 8.0

Oreo 8.0 dirilis pada 2017 dengan menambah lebih banyak fitur multi tasking dan perombakan bagian notifikasi. Pengguna bisa mengatur mana saja notifikasi yang ingin ditampilkan.

Tampilan UI-nya juga lebih rapi dan segar, serta difokuskan untuk memudahkan pengguna mengakses aplikasi dan mencari informasi.

16. Pie 9.0

Versi yang diluncurkan pada Agustus 2018 ini mengganti tiga tombol navigasi dengan tombol tunggal berbentuk elips. Android Pie disokong dengan kemampuan kecerdasan buatan (AI) yang menjadikannya bisa mempelajari pola penggunaan secara otomatis.

4.1.3 Android Studio



Gambar 4.3 Logo Android Studio

Android Studio adalah Lingkungan Pengembangan Terpadu (*Integrated Development Environment/IDE*) resmi untuk pengembangan aplikasi Android, yang didasarkan pada IntelliJ IDEA. Selain sebagai editor kode dan fitur developer IntelliJ yang andal, Android Studio menawarkan banyak fitur yang meningkatkan produktivitas Anda dalam membuat aplikasi Android, seperti:

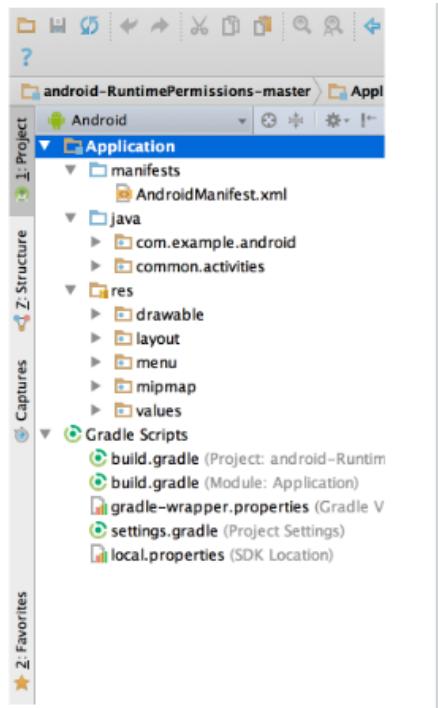
1. Sistem build berbasis Gradle yang fleksibel
2. Emulator yang cepat dan kaya fitur
3. Lingkungan terpadu tempat Anda bisa mengembangkan aplikasi untuk semua perangkat Android.
4. Terapkan Perubahan untuk melakukan push pada perubahan kode dan resource ke aplikasi yang sedang berjalan tanpa memulai ulang aplikasi.

5. Template kode dan integrasi GitHub untuk membantu Anda membuat fitur aplikasi umum dan mengimpor kode sampel.
6. Framework dan fitur pengujian yang lengkap.
7. Fitur lint untuk merekam performa, kegunaan, kompatibilitas versi, dan masalah lainnya.
8. Dukungan C++ dan NDK.
9. Dukungan bawaan untuk Google Cloud Platform, yang memudahkan integrasi Google Cloud Messaging dan App Engine.

4.1.4 Struktur Project

Setiap *project* di Android Studio berisi satu atau beberapa modul dengan file kode sumber dan *file resource*. Jenis modul meliputi:

1. Modul aplikasi Android.
2. Modul library.
3. Modul Google App Engine.



Gambar 4.4 File project Dalam Tampilan Android

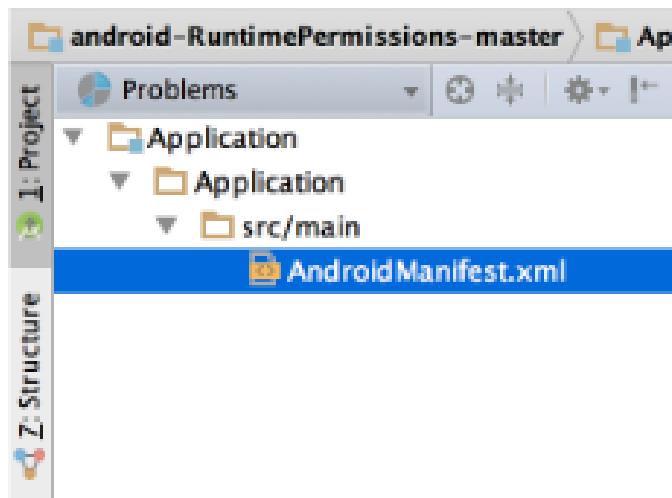
Secara *default*, Android Studio menampilkan *file project* Anda dalam tampilan *project* Android, seperti yang ditunjukkan pada gambar. Tampilan ini disusun menu-rut modul untuk memberikan akses cepat ke file sumber utama *project*.

Semua *file build* terlihat di tingkat teratas di bagian *Gradle Script* dan setiap modul aplikasi berisi folder berikut:

1. manifes: Berisi file *AndroidManifest.xml*.
2. java: Berisi file kode sumber Java, termasuk kode pengujian JUnit.
3. res: Berisi semua resource non-kode, seperti tata letak XML, string UI, dan gambar bitmap.

Struktur *project* Android pada disk berbeda dengan representasi tersatukan ini. Untuk melihat struktur *file project* sebenarnya, pilih *Project* dari menu drop-down *Project*.

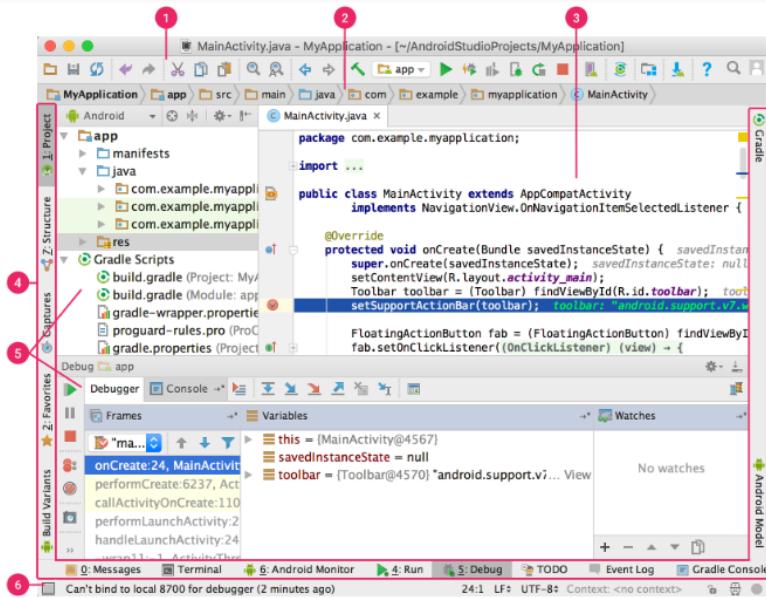
Menyesuaikan tampilan *file project* untuk berfokus pada aspek spesifik dari pengembangan aplikasi. Misalnya, memilih tampilan *Problems* pada *project* akan menampilkan link ke *file* sumber yang berisi *error coding* dan sintaks yang dikenali, seperti tag penutup elemen XML yang tidak ada dalam file tata letak.



Gambar 4.5 File Project Dalam Tampilan Problems

4.1.5 Antarmuka Pengguna

endela utama Android Studio terdiri dari beberapa area logis yang diidentifikasi dalam gambar berikut :



atau mengklik kaca pembesar di sudut kanan atas jendela Android Studio. Tips ini sangat berguna jika, misalnya, mencoba menemukan tindakan IDE tertentu yang lupa cara memicunya.

4.1.6 Jendela Fitur

Sebagai ganti menggunakan perspektif preset, Android Studio mengikuti konteks dan otomatis menampilkan jendela fitur yang relevan saat bekerja. Secara default, jendela fitur yang paling umum digunakan disematkan ke panel jendela fitur di tepi jendela aplikasi.

1. Untuk memperluas atau menciutkan jendela fitur, klik nama fitur di panel jendela fitur. Dapat menarik, menyematkan, melepaskan sematan, memasang, dan melepas jendela fitur.
2. Untuk kembali ke tata letak jendela *fitur default* saat ini, klik Window kemudian *Restore Default Layout* atau sesuaikan tata letak default dengan mengklik Window kemudian *Store Current Layout as Default*.
3. Untuk menampilkan atau menyembunyikan seluruh panel jendela alat, klik ikon jendela di pojok kiri bawah jendela Android Studio.
4. Untuk menemukan jendela alat tertentu, arahkan cursor ke atas ikon jendela dan pilih jendela alat tersebut dari menu. Kita juga bisa menggunakan pintasan keyboard untuk membuka jendela alat dengan mencantumkan pintasan jendela paling umum.

Jendela alat	Windows dan Linux	Mac
Project	Alt+1	Command+1
Kontrol Versi	Alt+9	Command+9
Run	Shift+F10	Control+R
Debug	Shift+F9	Control+D
Logcat	Alt+6	Command+6
Kembali ke Editor	Esc	Esc
Menyembunyikan Semua Jendela Alat	Control+Shift+F12	Command+Shift+F12

Gambar 4.7 Pintasan keyboard Ke Beberapa Jendela Alat yang Berguna.

Jika ingin menyembunyikan semua toolbar, jendela alat, dan tab editor, klik *View* lalu Enter *Distraction Free Mode*. Langkah ini akan mengaktifkan *Distraction Free Mode*. Untuk keluar dari *Distraction Free Mode*, klik *View* kemudian *Exit Distraction Free Mode*.

Kita dapat menggunakan Speed Search untuk menelusuri dan memfilter di dalam sebagian besar jendela fitur pada Android Studio. Untuk menggunakan *Speed Search*, pilih jendela alat, lalu ketik kueri penelusuran Anda.

4.1.7 Pelengkapan kode

Android Studio memiliki tiga jenis pelengkapan kode, yang dapat diakses menggunakan pintasan keyboard.

Jenis	Deskripsi	Windows dan Linux	Mac
Basic Completion	Menampilkan saran dasar untuk variabel, jenis, metode, ekspresi, dan sebagainya. Jika memanggil Basic Completion dua kali berturut-turut, Anda akan melihat lebih banyak hasil, termasuk anggota pribadi dan anggota statis yang tidak diimpor.	Control+Spasi	Control+Spasi
Smart Completion	Menampilkan opsi yang relevan berdasarkan konteks. Smart Completion mengetahui jenis dan alur data yang diharapkan. Jika Anda memanggil Smart Completion dua kali berturut-turut, Anda akan melihat lebih banyak hasil, termasuk chain.	Control+Shift+Spasi	Control+Shift+Spasi
Statement Completion	Melengkapi pernyataan saat ini secara otomatis, seperti menambahkan tanda kurung, tanda kurung siku, tanda kurung kurawal, pemformatan, dll. yang tidak lengkap.	Control+Shift+Enter	Shift+Command+Enter

Gambar 4.8 Pintasan keyboard untuk Pelengkapan Kode.

4.1.8 Navigasi

Berikut ini beberapa tips untuk membantu menjelajah di dalam Android Studio.

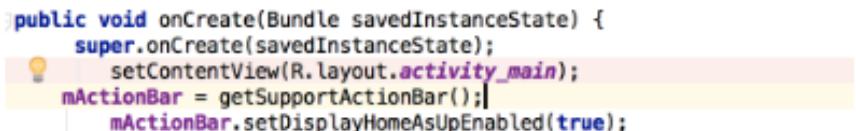
1. Beralih antar *file* yang baru saja diakses menggunakan tindakan *Recent Files*. Tekan Control+E (Command+E) untuk memunculkan tindakan *Recent Files*. Secara default, file yang terakhir diakses akan dipilih. Anda juga dapat mengakses jendela fitur mana saja melalui kolom kiri dalam tindakan ini.
2. Lihat struktur file saat ini menggunakan tindakan *File Structure*. Munculkan tindakan *File Structure* dengan menekan Control+F12 (Command+F12). Dengan tindakan ini, dapat membuka bagian mana pun dari file saat ini dengan cepat.
3. Telusuri dan buka *class* tertentu dalam *project* menggunakan tindakan *Navigate to Class*. Munculkan tindakan ini dengan menekan Control+N (Command+O). *Navigate to Class* mendukung ekspresi canggih, termasuk camel humps, jalur, baris navigasi ke, pencocokan nama tengah, dan banyak lagi. Jika Anda menggilinya dua kali berturut-turut, hasil dari *class project* akan ditampilkan.
4. Buka file atau folder menggunakan tindakan *Navigate to File*. Munculkan tindakan *Navigate to File* dengan menekan Control+Shift+N (Command+Shift+O). Untuk menelusuri folder dan bukan file, tambahkan / (garis miring) di akhir ekspresi Anda.

- Buka metode atau kolom menurut nama menggunakan tindakan *Navigate to Symbol*. Munculkan tindakan *Navigate to Symbol* dengan menekan Control+Shift+Alt+N (Command+Option+O).
- Temukan semua bagian kode yang merujuk ke *class*, metode, kolom, parameter, atau pernyataan di posisi kursor saat ini dengan menekan Alt+F7 (Option+F7).

4.1.9 Gaya dan Pemformatan

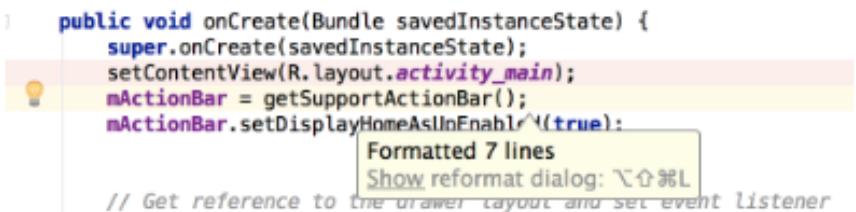
Saat Anda mengedit, Android Studio otomatis menerapkan pemformatan dan gaya seperti yang ditentukan dalam setelan gaya kode Anda. Anda dapat menyesuaikan setelan gaya kode menurut bahasa pemrograman, termasuk menentukan konvensi untuk tab dan indentasi, spasi, penggabungan, tanda kurung kurawal, dan baris kosong. Untuk menyesuaikan setelan gaya kode Anda, klik File \rightarrow Settings \rightarrow Editor \rightarrow Code Style (Preferences \rightarrow Editor \rightarrow Code Style).

Meskipun IDE otomatis menerapkan pemformatan selagi bekerja, bisa memanggil tindakan Reformat *Code* secara eksplisit dengan menekan Control+Alt+L (Opt+Command+L), atau otomatis mengindentasi semua baris dengan menekan Control+Alt+I (Control+Option+I).



```
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    mActionBar = getSupportActionBar();
    mActionBar.setDisplayHomeAsUpEnabled(true);
```

Gambar 4. Kode sebelum pemformatan.



```
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    mActionBar = getSupportActionBar();
    mActionBar.setDisplayHomeAsUpEnabled(true);
}
// Get reference to the drawer layout and set event listener
```

Gambar 4.9 Kode Setelah Pemformatan

4.1.10 Sistem build Gradle

Android Studio menggunakan Gradle sebagai dasar dari sistem build, dengan lebih banyak kemampuan khusus Android yang disediakan oleh plugin Android untuk Gradle. Sistem build ini berjalan sebagai fitur terintegrasi dari menu Android Studio, dan terpisah dari *command line* dapat menggunakan fitur-fitur sistem build untuk:

1. Menyesuaikan, mengonfigurasi, dan memperluas proses pembuatan *build*.
2. Membuat banyak APK untuk aplikasi, dengan berbagai fitur yang menggunakan *project* dan modul yang sama.
3. Menggunakan kembali kode dan *resource* di seluruh set sumber.

Berkat fleksibilitas Gradle, dapat mencapai semua ini tanpa mengubah file sumber inti aplikasi. File *build* Android Studio diberi nama *build.gradle*. File tersebut adalah file teks biasa yang menggunakan sintaks Groovy untuk mengonfigurasi *build* dengan elemen yang disediakan oleh plugin Android untuk Gradle. Setiap *project* memiliki satu file *build* tingkat atas untuk seluruh *project* dan file *build* tingkat modul terpisah untuk setiap modul. Saat Anda mengimpor *project* yang ada, Android Studio akan otomatis menghasilkan file *build* yang diperlukan.

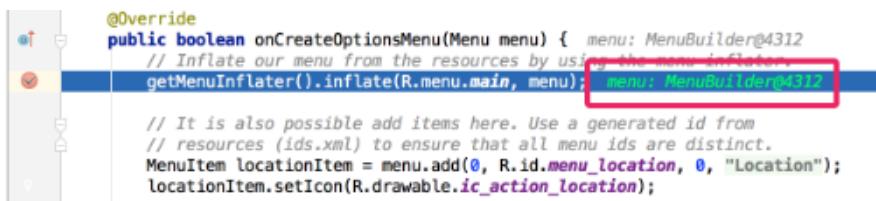
4.1.11 Fitur Profil dan Debug

Android Studio membantu menjalankan proses debug dan meningkatkan performa kode, termasuk proses debug inline dan fitur analisis performa.

1. Proses debug inline

Gunakan proses debug inline untuk menyempurnakan panduan kode dalam tampilan debugger dengan verifikasi inline untuk referensi, ekspresi, dan nilai variabel. Informasi debug inline meliputi:

- (a) Nilai variabel inline
- (b) Objek perujuk yang merujuk ke objek terpilih
- (c) Nilai yang dihasilkan metode
- (d) Ekspresi operator dan lambda
- (e) Nilai tooltip



The screenshot shows the Java code for an onCreateOptionsMenu() method. A red box highlights the line 'menu: MenuBuilder@4312' which is annotated with '@Override'. The code uses the getMenuInflater().inflate() method to inflate a menu from a resource. Below the code, a note says it's also possible to add items directly using a generated ID from resources. A MenuItem object is then added to the menu with an icon from a drawable resource.

```
@Override
public boolean onCreateOptionsMenu(Menu menu) {    menu: MenuBuilder@4312
    // Inflate our menu from the resources by using the menuInflater.
    getMenuInflater().inflate(R.menu.main, menu);    menu: MenuBuilder@4312

    // It is also possible add items here. Use a generated id from
    // resources (ids.xml) to ensure that all menu ids are distinct.
    MenuItem locationItem = menu.add(0, R.id.menu_location, 0, "Location");
    locationItem.setIcon(R.drawable.ic_action_location);}
```

Gambar 4.10 Nilai Variabel Inline

Untuk mengaktifkan proses debug inline, di jendela Debug, klik Settings dan centang kotak *Show Values Inline*.

2. Akses file data

Android SDK Tools, seperti *Systrace* dan *logcat*, menghasilkan data performa dan proses debug untuk analisis aplikasi secara mendetail.

Untuk menampilkan file data yang dihasilkan, buka jendela alat Captures. Pada daftar file yang dihasilkan, klik dua kali file untuk melihat data. Klik kanan sembarang file .hprof untuk mengonversinya ke dalam format file standar Menyeleksi penggunaan RAM.

3. Pemeriksaan kode

Setiap kali mengompilasi program, Android Studio akan otomatis menjalankan Lint yang telah dikonfigurasi dan pemeriksaan IDE lainnya untuk memudahkan mengidentifikasi serta memperbaiki masalah kualitas struktur kode.

Fitur Lint memeriksa file sumber *project* Android untuk menemukan potensi bug dan peluang pengoptimalan guna mencapai ketepatan, keamanan, performa, kegunaan, aksesibilitas, serta internasionalisasi.



Gambar 4.11 Hasil pemeriksaan Lint di Android Studio

Selain pemeriksaan Lint, Android Studio juga menjalankan pemeriksaan kode IntelliJ dan memvalidasi anotasi untuk menyederhanakan alur kerja coding.

4.1.12 Anotasi di Android Studio

Android Studio mendukung anotasi variabel, parameter, dan nilai kembalian untuk membantu merekam bug, seperti pengecualian *pointer null* dan konflik jenis *resource*. Android SDK Manager mengemas library Support-Annotations di Android Support Repository untuk digunakan dengan Android Studio. Android Studio memvalidasi anotasi yang sudah dikonfigurasi selama pemeriksaan kode.

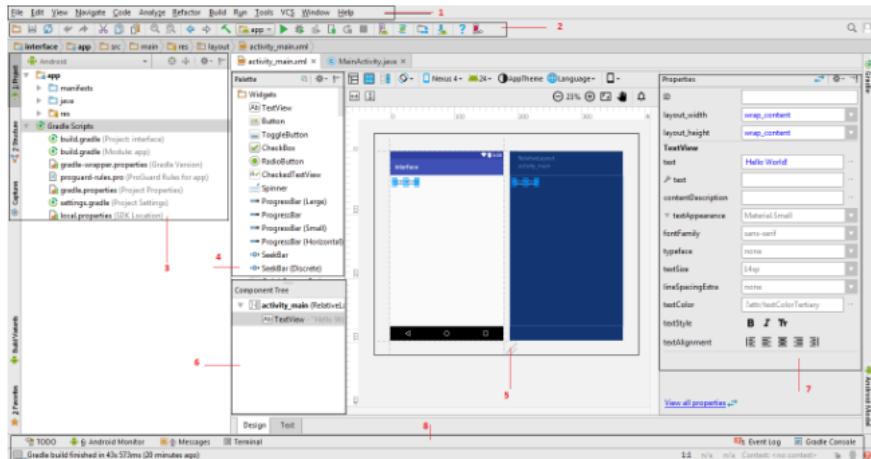
4.1.13 Pesan Log

Saat membuat dan menjalankan aplikasi dengan Android Studio, dapat melihat output adb dan pesan log perangkat di jendela Logcat.

4.1.14 Interfaces dan Struktur Project Android Studio

Interfaces dan struktur project android studio ini sangat penting perlu diketahui. Mengapa? Karena agar pada saat memprogram kita tidak bingung dari berbagai interfaces dan struktur project yang sedia. Beriku penjelesannya:

1. Antar Muka (*Interface*) Android Studio



Gambar 4.12 Interfaces Android Studio

2. Menu Bar

Seperti pada aplikasi lain Menu bar merupakan bagian antar muka (*interface*) pengguna yang berisi perintah dan opsi yang dapat dipilih untuk mengeksekusi suatu perintah.



Gambar 4.13 Menu Bar Android Studio

3. Tool Bar

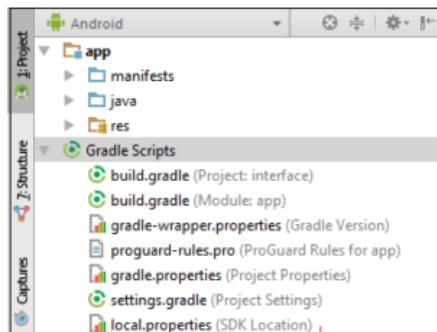
Dengan tool bar kita bisa mempercepat perintah pada sebuah aplikasi.



Gambar 4.14 Tool Bar Android Studio

4. Struktur Project

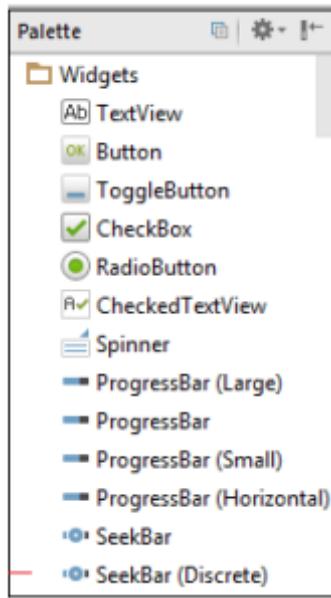
Pada bagian ini akan ditampilkan folder-folder dari sebuah *project* aplikasi android yang dibuat menggunakan android studio.



Gambar 4.15 Struktur *Project* Android Studio

5. Pallette

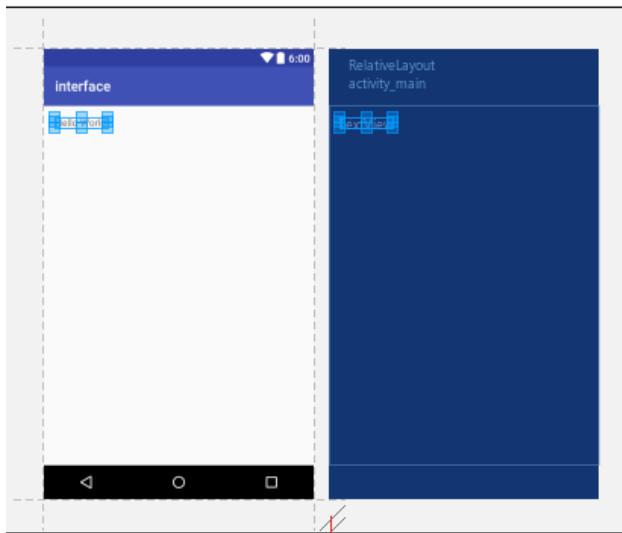
Di *pallette* tersedia semua *tools* untuk membuat aplikasi android dan enaknya lagi untuk menggunakanya cukup dengan mendrag and drop ke design android.



Gambar 4.16 Pallet Android Studio

6. Design Android

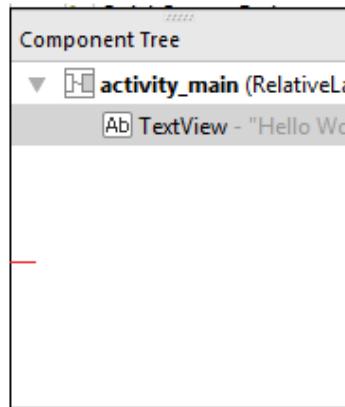
Tempat ini digunakan untuk mendesign layout aplikasi dengan cara drag and drop (tidak mengetikan script xml).



Gambar 4.17 Design Android Studio

7. Tree

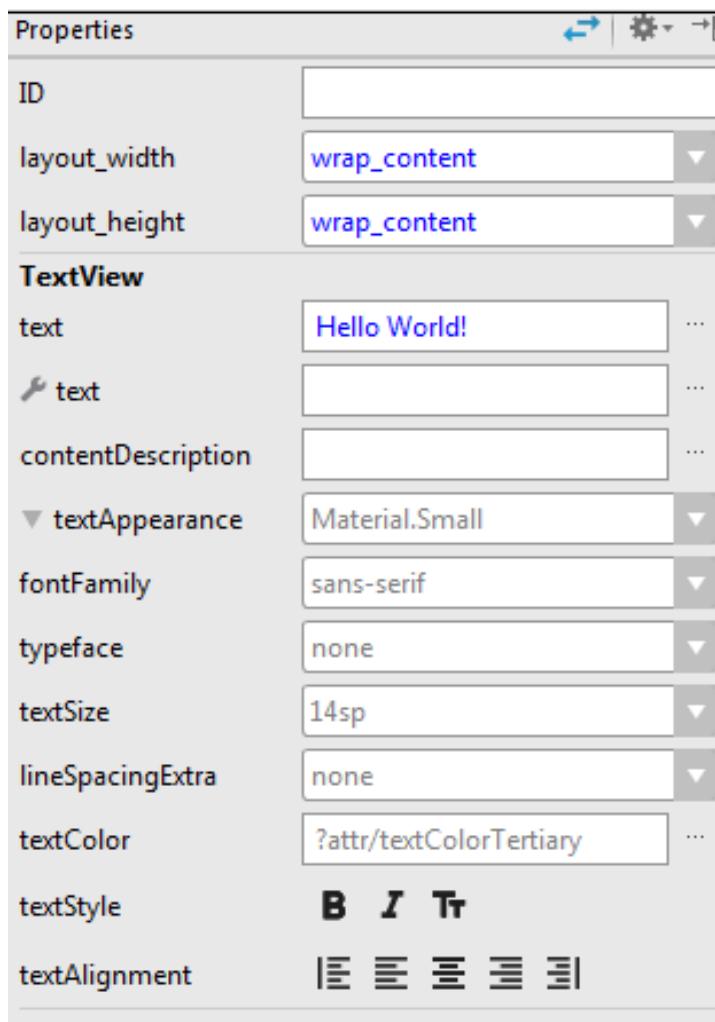
Setelah kita menggunakan *tools* yang ada di pallete maka akan ditampilkan pada component tree misal kita mendrag TextView (widget untuk membuat teks) ke design android. Maka nanti ditampilkan di component tree TextView.



Gambar 4.18 Tree Android Studio

8. Properties

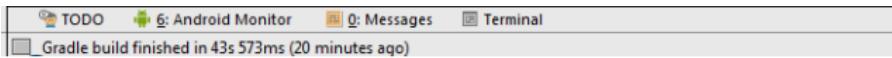
Disini akan ditampilkan pengaturan-pengaturan dari komponen yang digunakan untuk design aplikasi. Jika TextView maka akan ditampilkan untuk mengatur warna, size dan lainnya.



Gambar 4.19 Properties Android Studio

9. Status Bar

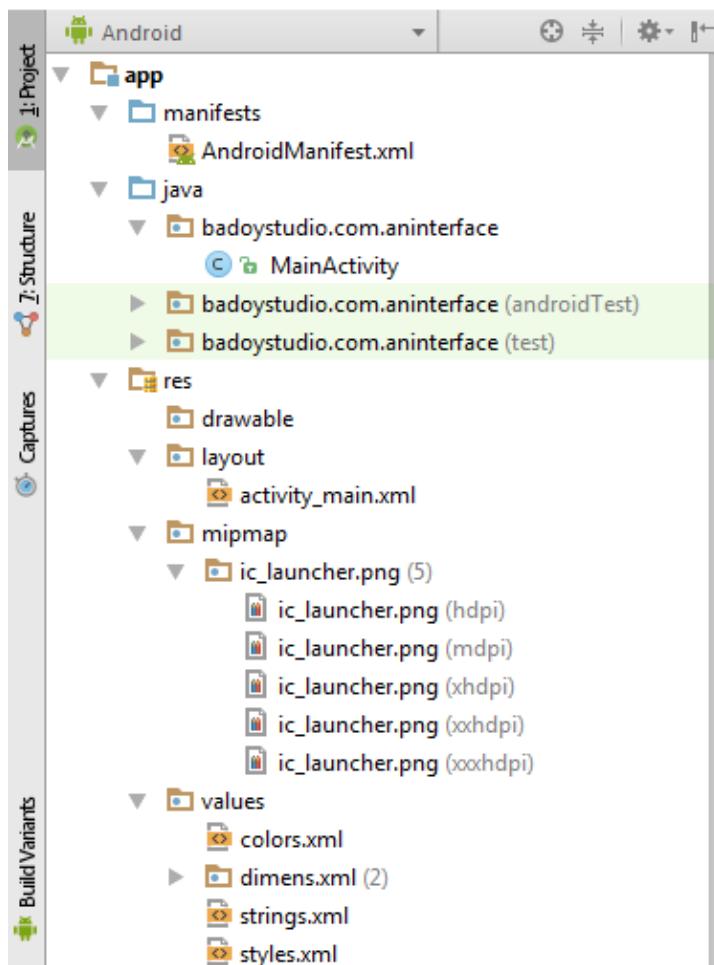
Menampilkan proses pada Android Studio. Proses *Loading*, *Error*, dan lainnya.



Gambar 4.20 Status Bar Android Studio

4.1.15 Struktur Folder Project Android Studio

Berikut adalah struktur folder project pada Android Studio :



Gambar 4.21 Struktur Folder Projectr Android Studio

1. Manifests

AndroidManifest.xml : File ini digunakan untuk melakukan beberapa pengaturan seperti nama aplikasi, icon aplikasi, *theme style* dan *User permission* (jika membuat aplikasi yang membutuhkan akses hardware smartphone ataupun koneksi internet).

2. Java

MainActivity.java : digunakan untuk memberikan perintah agar aplikasi melakukan sesuatu menggunakan bahasa pemrograman java.

3. Res Drawable

Folder yang digunakan untuk memasukan gambar pendukung aplikasi yang kita buat baik berupa icon ataupun lainnya

4. Res Layout

activitymain.xml : digunakan untuk pengaturan layout untuk interface utama pada aplikasi android yang kita buat.

5. mipmap

Folder ini digunakan untuk memasukan gambar berupa icon. Icon default aplikasi yang kita buat juga diambil dari folder ini.

6. values

colors.xml : file ini berisi kode-kode untuk pengaturan warna. Warna status bar, teks, ataupun lainnya.

dimens.xml : digunakan untuk pengaturan margin aplikasi.

strings.xml : digunakan untuk pengaturan teks-teks aplikasi yang kita buat.

styles.xml : digunakan untuk memberikan nama warna setelah kode-kode warna dimasukan atau disetting pada color.xml.

4.1.16 Bahasa Pemograman Java

Java adalah sebuah bahasa pemrograman yang sangat popular yang dikembangkan oleh Sun Microsystems (saat ini dimiliki oleh Oracle). Dikembangkan lama setelah C dan C++, Java menggabungkan banyak fitur-fitur canggih dari bahasa-bahasa canggih tersebut, sambil mengatasi beberapa kelemahan mereka. Walaupun demikian, tingkat kecanggihan bahasa pemrograman bergantung pada *library* mereka. *Library* ini ada untuk membantu para developer untuk membuat aplikasi. Beberapa fitur inti java :

1. Mudah dipelajari dan dimengerti.
2. Didesain untuk tidak bergantung kepada platform dan aman, menggunakan mesin virtual
3. Bersifat *object-oriented* (fokus kepada objek program ketimbang logic)

Android sangat bergantung kepada sifat-sifat dasar dari Java tersebut. Android SDK mengandung banyak *library* Java standar (*library* struktur data, *library* matematika, *library* grafik, *library* networking dan apapun yang dapat Anda inginkan) dan juga *library* special Android yang dapat membantu mengembangkan aplikasi Android yang keren.

4.2 Membuat Aplikasi Monitoring *Prototype*

Pada *prototype* prediksi ketinggian air (PKA) untuk mendeteksi banjir peringatan dini ini dibuat sebuah aplikasi berbasis android.selain notifikasi melalui telegram adanya juga sebuah aplikasi yang digunakan untuk memonitoring ketinggian air agar masyarakat mendapatkan informasi atau data mengenai ketinggian air secara *real time*.

Untuk membuat sebuah aplikasi monitoring ini menggunakan sebuah software yaitu android studio dengan menggunakan bahasa java, dimana bahasa java sudah dibahas pada bab ini di penjelasan sebelumnya.

4.2.1 Membuat *Splash Screen*

Splash screen adalah tampilan pertama program yang muncul sementara sebelum masuk ke menu utama. Berikut langkah-langkahnya:

1. Buka aplikasi android studio kemudian pilih *start a new android studio project*



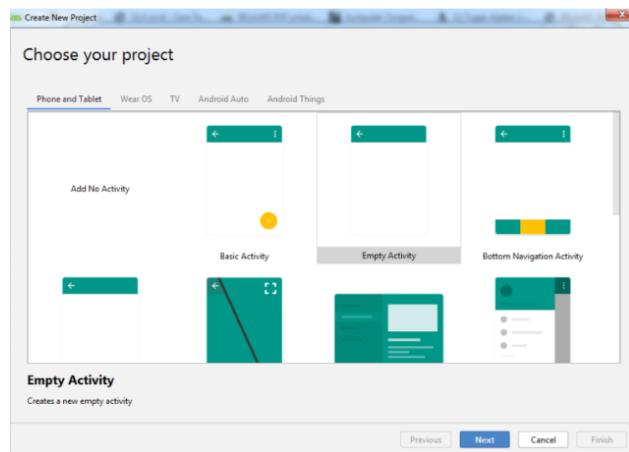
Gambar 4.22 Tampilan Awal Android Studio

2. Pilih *Activity*

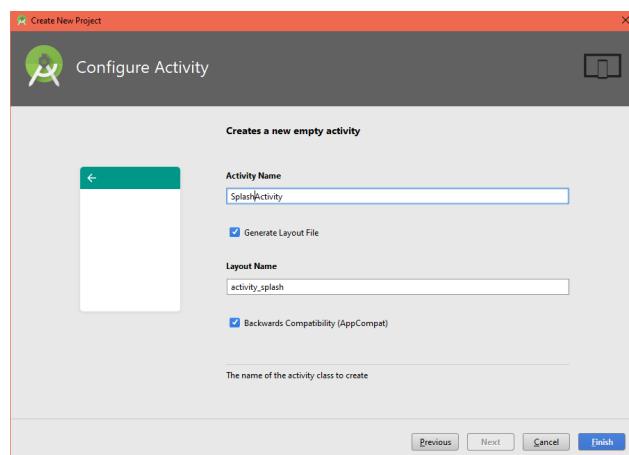
Activity (aktivitas) adalah sebuah komponen aplikasi yang menyediakan layar yang digunakan pengguna untuk berinteraksi guna melakukan sesuatu, misalnya memilih nomor ponsel, mengambil foto, mengirim email, atau menampilkan peta.

Tiap *Activity* (aktivitas) diberi sebuah jendela untuk menggambarkan antarmuka penggunanya. Jendela ini biasanya mengisi layar, namun mungkin lebih kecil daripada layar dan mengambang di atas jendela lain.

Pilih *Empty Activity* kemudian klik Next.

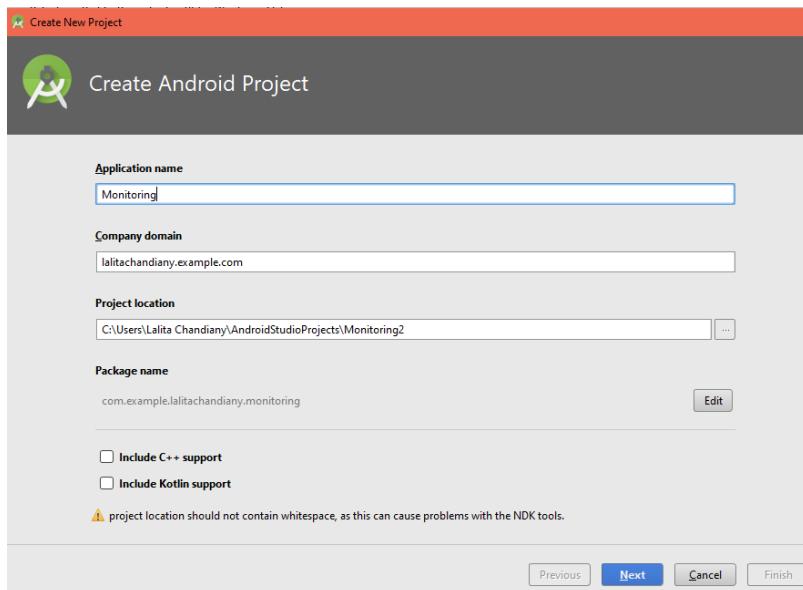


Gambar 4.23 Tampilan Memilih Activity

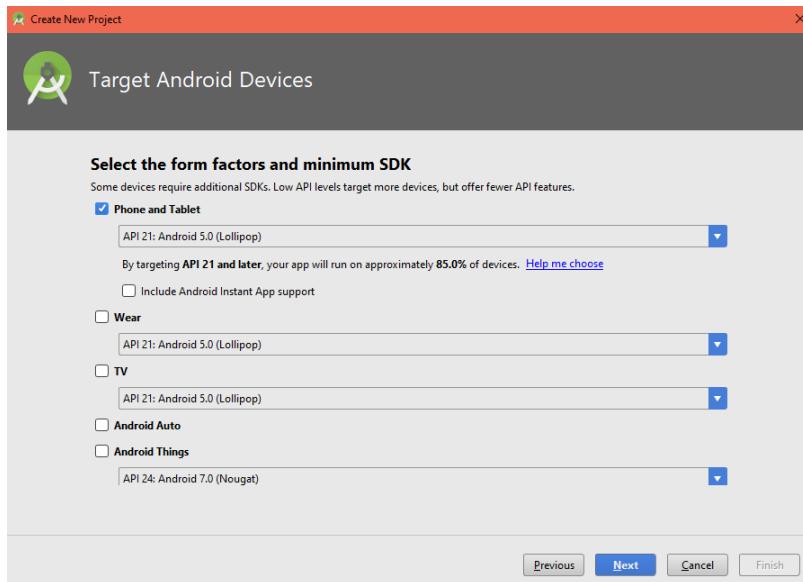


Gambar 4.24 Membuat Activity Spalsh Activity

3. Konfigurasi *Project* Setelah memilih *activity*, langkah selanjutnya yaitu kita harus memberikan konfigurasi pada *project android* yang kita buat. Adapun yang harus di konfigurasi yaitu :
 - (a) Name : isikan nama project
 - (b) Package name : isikan nama perusahaan atau pembuat aplikasi.
 - (c) Save location : tentukan file project aplikasi android ini akan disimpan pada direktori apa.
 - (d) Language : pilih bahasa pemrograman yang akan digunakan, Java atau Kotlin. Pada tutorial ini kita menggunakan Java.
 - (e) Minimum API level : tentukan API berapa yang akan digunakan. Pada bulan April 2017 google merilis daftar versi OS yang banyak digunakan tahun 2017 dan hasilnya Lollipop (OS versi 5.0 dengan SDK 21/22) ada di urutan nomor satu. Jadi untuk minimum SDKnya kita pilih API 15 yaitu untuk versi OS Ice Cream Sandwich ke atas.

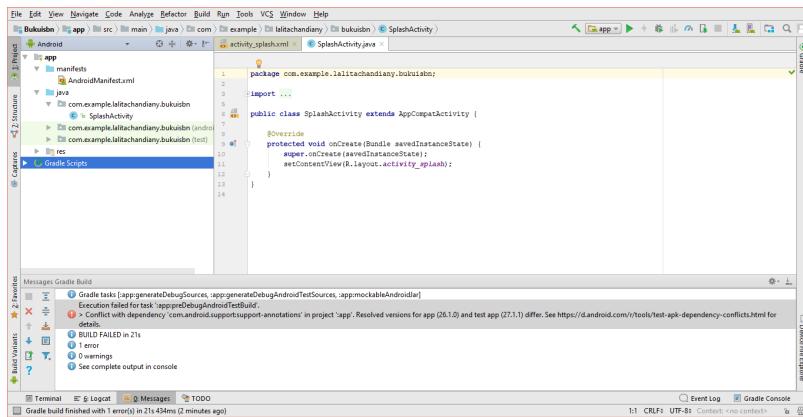


Gambar 4.25 Konfigurasi Project



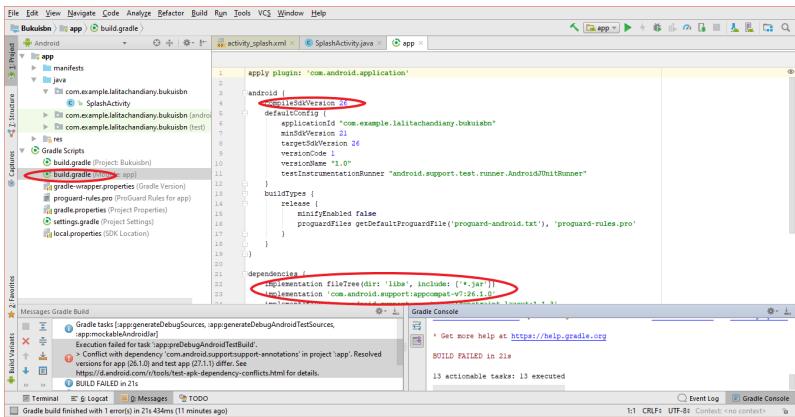
Gambar 4.26 Konfigurasi Project

4. Proses Gradle



Gambar 4.27 Proses Gradle

Jika pertama kali tampilan di android studio error maka yang bermasalah yaitu mengenai gradlenya untuk mengatasinya yaitu harus menyesuaikan versisnya. Dengan cara mengubah versi gradlenya sesuai dengan apa yang diminta oleh android studio. Untuk menggantinya masuk ke GradleScript-build gradle dan ubah versinya dibagian yang telah dilingkari seperti gambar berikut ini :

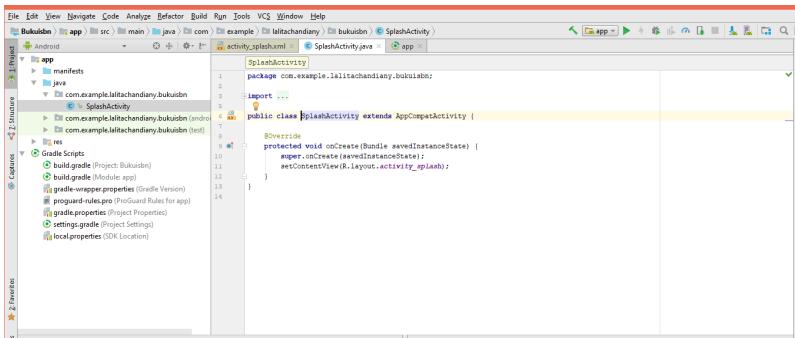


Gambar 4.28 Build Gradle

Tunggu sampai proses gradle selesai. Ini biasanya memakan waktu beberapa menit. Lama atau sebentaranya tergantung spesifikasi ram dan prosesor yang digunakan pada komputer atau laptop.

5. Workspace Android Studio

Jika proses Gradle telah selesai, maka akan terbuka workspace project Android Studio seperti gambar dibawah ini (pertama kali terbuka adalah tab SplashActivity.java. Disinilah nantinya kita memberikan kode-kode java agar aplikasi bisa melakukan sebuah perintah).



Gambar 4.29 Workspace

Masukan kode berikut ini untuk membuat splash screen di SpalshActivity.java

```

1 package com.example.lalitachandiany.monitoring;
2
3 import android.content.Intent;
4 import android.support.v7.app.AppCompatActivity;

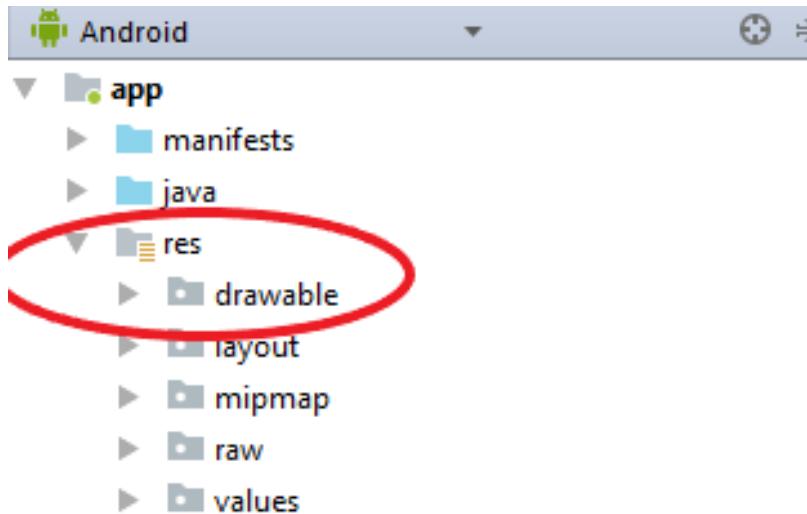
```

```

5 import android.os.Bundle;
6
7 public class SplashActivity extends AppCompatActivity {
8
9     @Override
10    protected void onCreate(Bundle savedInstanceState) {
11        super.onCreate(savedInstanceState);
12        setContentView(R.layout.activity_splash);
13
14        Thread thread = new Thread(){
15            public void run () {
16                try {
17                    sleep(4000);
18                } catch (InterruptedException e) {
19                    e.printStackTrace();
20                } finally {
21                    startActivity(new Intent(SplashActivity.this ,
22 HomeActivity.class));
23                    finish();
24                }
25            };
26            thread.start();
27        }
28    }

```

Kemudian untuk tampilan Splashescreennya kita harus memasukan gambar yang akan digunakan kedaam folder drawable.



Gambar 4.30 Drawable

Kemudian untuk membuat tampilanya diprogram dalam activityplash.xml dengan kode berikut ini :

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <android.widget.LinearLayout xmlns:android="http://schemas.
  android.com/apk/res/android"
3   xmlns:app="http://schemas.android.com/apk/res-auto"
4   xmlns:tools="http://schemas.android.com/tools"
5   android:layout_width="match_parent"
6   android:layout_height="match_parent"
7   android:orientation="vertical"
8   android:background="@drawable/spalsh"
9   tools:context="com.example.lalitachandiany.monitoring.
  SplashActivity">
10
11
12 </android.widget.LinearLayout>
```

4.2.2 Membuat Halaman Menu

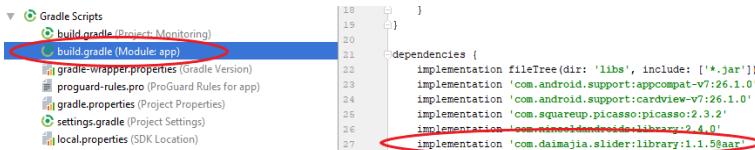
Menu adalah daftar perintah-perintah suatu perangkat lunak (program) yang apabila dieksekusi akan menjalankan suatu perintah tertentu dari aplikasi. Menu digunakan sebagai alternatif dari antarmuka baris perintah. Pilihan yang diberikan oleh menu dapat dipilih dengan menggunakan kursor atau antarmuka pengguna.

Dalam menu aplikasi monitoring ini akan adanya slider dan 4 menu utama yaitu *about*, ketinggian air, keterangan air dan *don't panic*. Untuk membuatnya berikut langkah-langkahnya :

1. Memnbuat Slider

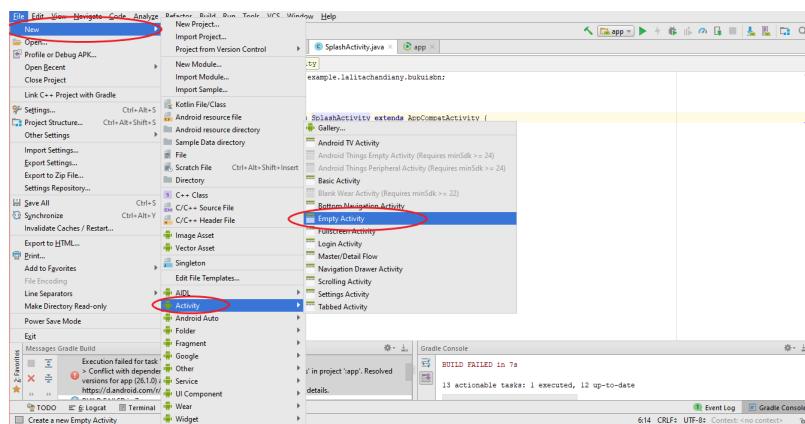
Slider adalah informasi yang berjalan (sliding) di sebuah aplikasi. Untuk membuatnya hal pertama kali yang harus dilakukan yaitu memasukan gambar apa saja yang akan ditampilkan dengan cara memasukanya kedalam folder *drawable*.

Jika telah memasukan gambar kedalam gradle maka kita import library untuk slider dengan cara memasukan perintahnya di gradle.



Gambar 4.31 Library Slider

Setelah memasukan atau *import library* maka kita harus membuat sebuah *activity* baru untuk home. *Activity home* ini akan digunakan untuk 4 menu dan slider yang telah dijelaskan sebelumnya untuk membuat *activity* klik file-new-activity-emtpy activity.



Gambar 4.32 Activity Baru

Jika kita telah membuat *activity* baru maka masukan kode berikut ini kedalam file *home.java*

```

1 public class HomeActivity extends AppCompatActivity {
2
3     @Override
4         protected void onCreate(Bundle savedInstanceState) {
5             super.onCreate(savedInstanceState);
6             setContentView(R.layout.activity_home);
7             // inisialisasi
8             SliderLayout sliderLayout = (SliderLayout) findViewById(R
9 .id.slider);
10            // Load image dari URL
11            // Load Image Dari res/drawable
12            HashMap<String ,Integer> file_maps = new HashMap<String ,
13 Integer >();
14            file_maps.put("Sudahkah Anda Siap",R.drawable.siap);
15            file_maps.put("Bersihkan Sampah",R.drawable.sungai);
16            file_maps.put("Banjir",R.drawable.banjirempat);
17            file_maps.put("Tampilan Prototipe", R.drawable.full);
18            for(String name : file_maps.keySet()){
19                TextSliderView textSliderView = new TextSliderView(
20 this);
21                // initialize a SliderLayout
22                textSliderView
23                    .description(name)
24                    .image(file_maps.get(name))
25                    .setScaleType(BaseSliderView.ScaleType.Fit);
26                // add your extra information
27                textSliderView.bundle(new Bundle());
28                textSliderView.getBundle()
29                    .putString("extra",name);
30                sliderLayout.addSlider(textSliderView);
31            }
32            sliderLayout.setPresetTransformer(SliderLayout.
33 Transformer.Accordion);

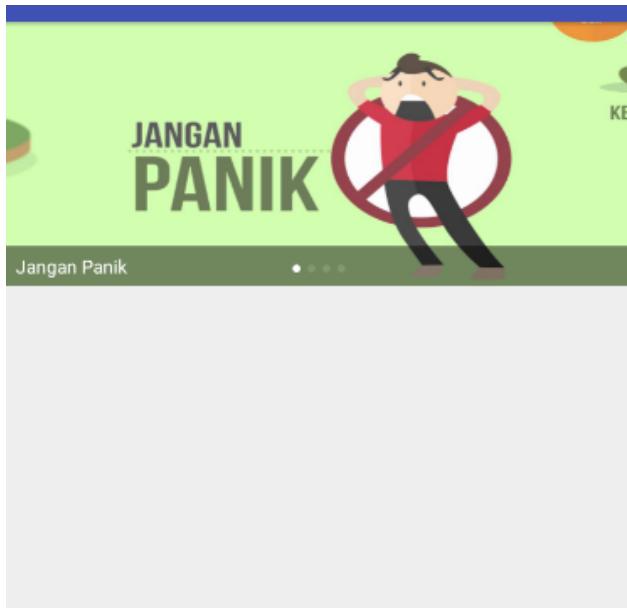
```

```
30     sliderLayout.setPresetIndicator(SliderLayout.  
31     PresetIndicators.Center_Bottom);  
32     sliderLayout.setCustomAnimation(new DescriptionAnimation  
33     ());  
34     sliderLayout.setDuration(4000);
```

Kemudian untuk tampilanya masukan kode berikut ini kedalam file homeactivity.xml.

```
1 <?xml version="1.0" encoding="utf-8"?>  
2 <RelativeLayout  
3     xmlns: android="http://schemas.android.com/apk/res/android"  
4     xmlns: app="http://schemas.android.com/apk/res-auto"  
5     xmlns: tools="http://schemas.android.com/tools"  
6     android: layout_width="match_parent"  
7     android: layout_height="match_parent"  
8     android: id="@+id/ activity_main"  
9     tools: context="com.example.lalitachandiany.sliderdua.  
10    MainActivity">  
11    <com.daimajia.slider.library.SliderLayout  
12        android: id="@+id/ slider"  
13        android: layout_width="match_parent"  
14        android: layout_height="200dp" />  
15  
16 </RelativeLayout>
```

Maka hasil nya seperti berikut ini :



Gambar 4.33 Tampilan Slider

2. Membuat Menu Di Home

Dalam aplikasi monitoring ini terdapat 4 menu yaitu menu *about*, ketinggian air, keterangan air dan *don't panic*. Untuk membuat menu ini kita tidak perlu membuat *activity* baru karna menu ini akan ditampilkan secara bersamaan dengan slider, maka kita memprogramnya di *activity home*.

Hal yang harus pertama dilakukan yaitu mengimport *library cardview* karna untuk membuat menu ini dibutuhkan sebuah *library cardview*. Caranya sama dengan langkah sebelumnya dan masukan perintah seperti dibawah ini :

```
dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    implementation 'com.android.support:appcompat-v7:26.1.0'
    implementation 'com.android.support:cardview-v7:26.1.0'
    implementation 'com.squareup.picasso:picasso:2.3.2'
```

Gambar 4.34 Library Cardview

Jika telah memasukan *library cardview* maka masukan gambar yang akan dijadikan sebagai icon menu ke drawable dengan cara yang sama seperti sebelumnya. Setelah itu masukan kode berikut kedalam file Home.java.

```
1 /inisialisasi
2     ImageView awas = findViewById(R.id.awas);
3     ImageView air = findViewById(R.id.air);
4     ImageView banjir = findViewById(R.id.banjir);
5     ImageView panic = findViewById(R.id.panic);
6
7     awas.setOnClickListener(new View.OnClickListener() {
8         @Override
9         public void onClick(View view) {
10             Intent awas = new Intent(HomeActivity.this,
11             MonitoringActivity.class);
12             startActivity(awas);
13         }
14     });
15
16     banjir.setOnClickListener(new View.OnClickListener() {
17         @Override
18         public void onClick(View view) {
19             Intent banjir = new Intent(HomeActivity.this,
20             AboutActivity.class);
21             startActivity(banjir);
22         }
23     });
24
25     air.setOnClickListener(new View.OnClickListener() {
26         @Override
27         public void onClick(View view) {
28             Intent air = new Intent(HomeActivity.this,
AirActivity.class);
29             startActivity(air);
30         }
31     });
32 }
```

```
29 });
30
31     panic . setOnClickListener (new View . OnClickListener () {
32         @Override
33         public void onClick (View view) {
34             Intent panic = new Intent (HomeActivity . this ,
35             PanicActivity . class );
36             startActivity (panic );
37         }
38     });
39 }
```

Jika digabungkan dengan *code* slider sebelumnya maka seperti berikut:

```
1 package com.example.lalitachandiany.monitoring;
2
3 import android.content.Intent;
4 import android.support.v7.app.AppCompatActivity;
5 import android.os.Bundle;
6 import android.view.View;
7 import android.widget.ImageView;
8
9 import com.daimajia.slider.library.Animations .
10 DescriptionAnimation;
11 import com.daimajia.slider.library.SliderLayout;
12 import com.daimajia.slider.library.SliderTypes.BaseSliderView;
13 import com.daimajia.slider.library.SliderTypes.TextSliderView;
14
15 import java.util.HashMap;
16
17 public class HomeActivity extends AppCompatActivity {
18
19     @Override
20     protected void onCreate (Bundle savedInstanceState) {
21         super.onCreate (savedInstanceState);
22         setContentView (R.layout.activity_home);
23         // inisialisasi
24         SliderLayout sliderLayout = (SliderLayout) findViewById (R
25 . id.slider);
26         // Load image dari URL
27         // Load Image Dari res/drawable
28         HashMap<String , Integer> file_maps = new HashMap<String ,
29 Integer >();
30         file_maps . put ("Sudahkah Anda Siap" , R.drawable . siap);
31         file_maps . put ("Bersihkan Sampah" , R.drawable . sungai);
32         file_maps . put ("Banjir" , R.drawable . banjirempat);
33         file_maps . put ("Tampilan Prototipe" , R.drawable . full);
34         for (String name : file_maps . keySet ()) {
35             TextSliderView textSliderView = new TextSliderView (
36 this );
37             // initialize a SliderLayout
38             textSliderView
39                 . description (name)
40                 . image (file_maps . get (name))
41                 . setScaleType (BaseSliderView . ScaleType . Fit );
42             // add your extra information
```

```
39         textSliderView .bundle (new Bundle ());
40         textSliderView .getBundle ()
41             .putString ("extra" ,name );
42         sliderLayout .addSlider (textSliderView );
43     }
44     sliderLayout .setPresetTransformer (SliderLayout .
45 Transformer .Accordion );
45     sliderLayout .setPresetIndicator (SliderLayout .
46 PresetIndicators .Center_Bottom );
46     sliderLayout .setCustomAnimation (new DescriptionAnimation
47 ());
47     sliderLayout .setDuration (4000 );
48
49
50 // inisialisasi
51 ImageView awas = findViewById (R.id.awas );
52 ImageView air = findViewById (R.id.air );
53 ImageView banjir = findViewById (R.id.banjir );
54 ImageView panic = findViewById (R.id.panic );
55
56 awas .setOnClickListener (new View .OnClickListener () {
57     @Override
58     public void onClick (View view) {
59         Intent awas = new Intent (HomeActivity .this ,
60 MonitoringActivity .class );
61         startActivity (awas );
62     }
63 });
64
65 banjir .setOnClickListener (new View .OnClickListener () {
66     @Override
67     public void onClick (View view) {
68         Intent banjir = new Intent (HomeActivity .this ,
69 AboutActivity .class );
70         startActivity (banjir );
71     }
72 });
73
74 air .setOnClickListener (new View .OnClickListener () {
75     @Override
76     public void onClick (View view) {
77         Intent air = new Intent (HomeActivity .this ,
78 AirActivity .class );
79         startActivity (air );
80     }
81 });
82
83 panic .setOnClickListener (new View .OnClickListener () {
84     @Override
85     public void onClick (View view) {
86         Intent panic = new Intent (HomeActivity .this ,
87 PanicActivity .class );
88         startActivity (panic );
89     }
90 });
91 }
```

88 }

Setelah itu untuk tampilanya tambahkan *code* berikut ini kedalam file activity-home.xml.

```
1 <LinearLayout
2     android:layout_width="match_parent"
3     android:layout_height="wrap_content"
4     android:clipToPadding="false"
5     android:gravity="center"
6     android:orientation="horizontal">
7
8     <android.support.v7.widget.CardView
9         android:id="@+id/bankcardId"
10        android:layout_width="140dp"
11        android:layout_height="160dp"
12        android:layout_margin="10dp"
13        android:clickable="true"
14        android:foreground="?android:attr/
15           selectableItemBackground">
16
17     <LinearLayout
18         android:layout_width="match_parent"
19         android:layout_height="match_parent"
20         android:background="@color/colorPrimaryDark"
21         android:gravity="center"
22         android:orientation="vertical">
23
24     <ImageView
25         android:id="@+id/banjir"
26         android:layout_width="80dp"
27         android:layout_height="80dp"
28         android:padding="10dp"
29         android:src="@drawable/banjirr" />
30
31     <TextView
32         android:layout_width="wrap_content"
33         android:layout_height="wrap_content"
34         android:layout_marginTop="10dp"
35         android:background="@android:color/
36           holo_blue_bright"
37         android:text="About"
38         android:textColor="?attr/colorAccent"
39         android:textStyle="bold" />
40
41     <View
42         android:layout_width="match_parent"
43         android:layout_height="1dp"
44         android:layout_margin="10dp"
45         android:background="@color/lightgray" />
46
47     <TextView
48         android:layout_width="wrap_content"
49         android:layout_height="wrap_content"
50         android:gravity="center"
51         android:padding="5dp"
52         android:text="Apa itu Banjir?"
```

```
51             android:textColor="@android:color/darker_gray"
52         ""
53
54     
```

```
55         </LinearLayout>
56     
```

```
57     <android.support.v7.widget.CardView>
58         android:layout_width="140dp"
59         android:layout_height="160dp"
60         android:layout_margin="10dp"
61         android:clickable="true"
62         android:foreground="?android:attr/
63         selectableItemBackground">
64
65         <LinearLayout
66             android:layout_width="match_parent"
67             android:layout_height="match_parent"
68             android:background="@color/colorPrimaryDark"
69             android:gravity="center"
70             android:orientation="vertical">
71
72             <ImageView
73                 android:id="@+id/awas"
74                 android:layout_width="90dp"
75                 android:layout_height="80dp"
76                 android:padding="10dp"
77                 android:src="@drawable/awas" />
78
79             <TextView
80                 android:layout_width="wrap_content"
81                 android:layout_height="wrap_content"
82                 android:layout_marginTop="10dp"
83                 android:background="@android:color/
84                 holo_blue_bright"
85                 android:text="Ketinggian AIR"
86                 android:textColor="@color/colorAccent"
87                 android:textStyle="bold" />
88
89             <View
90                 android:layout_width="match_parent"
91                 android:layout_height="1dp"
92                 android:layout_margin="10dp"
93                 android:background="@color/lightgray" />
94
95             <TextView
96                 android:layout_width="wrap_content"
97                 android:layout_height="wrap_content"
98                 android:gravity="center"
99                 android:padding="5dp"
100                android:text="Monitoring"
101                android:textColor="@android:color/darker_gray"
102                android:textStyle="bold"/>
103
104         </LinearLayout>
```

```
103     </android.support.v7.widget.CardView>
104
105 </LinearLayout>
106
107 <LinearLayout
108     android:layout_width="match_parent"
109     android:layout_height="wrap_content"
110     android:clipToPadding="false"
111     android:gravity="center"
112     android:orientation="horizontal">
113
114     <android.support.v7.widget.CardView
115         android:layout_width="140dp"
116         android:layout_height="160dp"
117         android:layout_margin="10dp"
118         android:clickable="true"
119         android:foreground="?android:attr/
selectableItemBackground">
120
121         <LinearLayout
122             android:layout_width="match_parent"
123             android:layout_height="match_parent"
124             android:background="@color/colorPrimaryDark"
125             android:gravity="center"
126             android:orientation="vertical">
127
128             <ImageView
129                 android:id="@+id/air"
130                 android:layout_width="80dp"
131                 android:layout_height="80dp"
132                 android:padding="10dp"
133                 android:src="@drawable/air" />
134
135             <TextView
136                 android:layout_width="wrap_content"
137                 android:layout_height="wrap_content"
138                 android:layout_marginTop="10dp"
139                 android:background="@android:color/
holo_blue_bright"
140                 android:text="Keterangan Air"
141                 android:textColor="@color/colorAccent"
142                 android:textStyle="bold" />
143
144             <View
145                 android:layout_width="match_parent"
146                 android:layout_height="1dp"
147                 android:layout_margin="10dp"
148                 android:background="@color/lightgray" />
149
150             <TextView
151                 android:layout_width="wrap_content"
152                 android:layout_height="wrap_content"
153                 android:gravity="center"
154                 android:padding="5dp"
155                 android:text="Kamu Harus Tau"
```

```
156                                     android:textColor="@android:color/darker_gray"
157                                     ""
158
159             </LinearLayout>
160         </android.support.v7.widget.CardView>
161
162     <android.support.v7.widget.CardView
163         android:layout_width="140dp"
164         android:layout_height="160dp"
165         android:layout_margin="10dp"
166         android:clickable="true"
167         android:foreground="?android:attr/
168         selectableItemBackground">
169
170         <LinearLayout
171             android:layout_width="match_parent"
172             android:layout_height="match_parent"
173             android:background="@color/colorPrimaryDark"
174             android:gravity="center"
175             android:orientation="vertical">
176
177             <ImageView
178                 android:id="@+id/panic"
179                 android:layout_width="64dp"
180                 android:layout_height="64dp"
181                 android:padding="10dp"
182                 android:src="@drawable/panic" />
183
184             <TextView
185                 android:layout_width="wrap_content"
186                 android:layout_height="wrap_content"
187                 android:layout_marginTop="10dp"
188                 android:background="@android:color/
189                 holo_blue_bright"
190                 android:text="Don't Panic"
191                 android:textColor="@color/colorAccent"
192                 android:textStyle="bold" />
193
194             <View
195                 android:layout_width="match_parent"
196                 android:layout_height="1dp"
197                 android:layout_margin="10dp"
198                 android:background="@color/lightgray" />
199
200             <TextView
201                 android:layout_width="wrap_content"
202                 android:layout_height="wrap_content"
203                 android:gravity="center"
204                 android:padding="5dp"
205                 android:text="Wajib Baca !!"
206                 android:textColor="@android:color/darker_gray
207                 "
208                 android:textStyle="bold"/>
209
210         </LinearLayout>
```

```
208     </android.support.v7.widget.CardView>
209
210 </LinearLayout>
211
212 <LinearLayout
213     android:layout_width="match_parent"
214     android:layout_height="wrap_content"
215     android:clipToPadding="false"
216     android:gravity="center"
217     android:orientation="horizontal">
218
219
220 </LinearLayout>
221
222 </LinearLayout>
```

Jika digabungkan dengan *code* slider sebelumnya maka seperti berikut:

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/
3     android"
4         xmlns:app="http://schemas.android.com/apk/res-auto"
5         xmlns:tools="http://schemas.android.com/tools"
6         android:layout_width="match_parent"
7         android:layout_height="match_parent"
8         android:background="@color/backgroundcolor"
9         android:gravity="center"
10        android:orientation="vertical"
11        android:padding="10dp">
12
13    <com.daimajia.slider.library.SliderLayout
14        android:id="@+id/slider"
15        android:layout_width="match_parent"
16        android:layout_height="150dp" />
17
18    <LinearLayout
19        android:layout_width="match_parent"
20        android:layout_height="wrap_content"
21        android:clipToPadding="false"
22        android:gravity="center"
23        android:orientation="horizontal">
24
25        <android.support.v7.widget.CardView
26            android:id="@+id/bankcardId"
27            android:layout_width="140dp"
28            android:layout_height="160dp"
29            android:layout_margin="10dp"
30            android:clickable="true"
31            android:foreground="?android:attr/
32            selectableItemBackground">
33
34            <LinearLayout
35                android:layout_width="match_parent"
36                android:layout_height="match_parent"
37                android:background="@color/colorPrimaryDark"
38                android:gravity="center"
39                android:orientation="vertical">
```

```
38
39         <ImageView
40             android:id="@+id/banjir"
41             android:layout_width="80dp"
42             android:layout_height="80dp"
43             android:padding="10dp"
44             android:src="@drawable/banjirr" />
45
46         <TextView
47             android:layout_width="wrap_content"
48             android:layout_height="wrap_content"
49             android:layout_marginTop="10dp"
50             android:background="@android:color/
51             holo_blue_bright"
52             android:text="About"
53             android:textColor="?attr/colorAccent"
54             android:textStyle="bold" />
55
56         <View
57             android:layout_width="match_parent"
58             android:layout_height="1dp"
59             android:layout_margin="10dp"
60             android:background="@color/lightgray" />
61
62         <TextView
63             android:layout_width="wrap_content"
64             android:layout_height="wrap_content"
65             android:gravity="center"
66             android:padding="5dp"
67             android:text="Apa itu Banjir?"
68             android:textColor="@android:color/darker_gray"
69             android:textStyle="bold"/>
70
71     </LinearLayout>
72 </android.support.v7.widget.CardView>
73
74 <android.support.v7.widget.CardView
75     android:layout_width="140dp"
76     android:layout_height="160dp"
77     android:layout_margin="10dp"
78     android:clickable="true"
79     android:foreground="?android:attr/
80     selectableItemBackground">
81
82     <LinearLayout
83         android:layout_width="match_parent"
84         android:layout_height="match_parent"
85         android:background="@color/colorPrimaryDark"
86         android:gravity="center"
87         android:orientation="vertical">
88
89     <ImageView
90         android:id="@+id/awas"
91         android:layout_width="90dp"
92         android:layout_height="80dp"
```

```
91         android:padding="10dp"
92         android:src="@drawable/awas" />
93
94     <TextView
95         android:layout_width="wrap_content"
96         android:layout_height="wrap_content"
97         android:layout_marginTop="10dp"
98         android:background="@android:color/
holo_blue_bright"
99         android:text="Ketinggian AIR"
100        android:textColor="@color/colorAccent"
101        android:textStyle="bold" />
102
103    <View
104        android:layout_width="match_parent"
105        android:layout_height="1dp"
106        android:layout_margin="10dp"
107        android:background="@color/lightgray" />
108
109    <TextView
110        android:layout_width="wrap_content"
111        android:layout_height="wrap_content"
112        android:gravity="center"
113        android:padding="5dp"
114        android:text="Monitoring"
115        android:textColor="@android:color/darker_gray"
116        android:textStyle="bold" />
117
118    </LinearLayout>
119  </android.support.v7.widget.CardView>
120
121 </LinearLayout>
122
123 <LinearLayout
124     android:layout_width="match_parent"
125     android:layout_height="wrap_content"
126     android:clipToPadding="false"
127     android:gravity="center"
128     android:orientation="horizontal">
129
130     <android.support.v7.widget.CardView
131         android:layout_width="140dp"
132         android:layout_height="160dp"
133         android:layout_margin="10dp"
134         android:clickable="true"
135         android:foreground="?android:attr/
selectableItemBackground">
136
137         <LinearLayout
138             android:layout_width="match_parent"
139             android:layout_height="match_parent"
140             android:background="@color/colorPrimaryDark"
141             android:gravity="center"
142             android:orientation="vertical">
```

```
144<ImageView  
145    android:id="@+id/air"  
146    android:layout_width="80dp"  
147    android:layout_height="80dp"  
148    android:padding="10dp"  
149    android:src="@drawable/air" />  
150  
151<TextView  
152    android:layout_width="wrap_content"  
153    android:layout_height="wrap_content"  
154    android:layout_marginTop="10dp"  
155    android:background="@android:color/  
holo_blue_bright"  
156    android:text="Keterangan Air"  
157    android:textColor="@color/colorAccent"  
158    android:textStyle="bold" />  
159  
160<View  
161    android:layout_width="match_parent"  
162    android:layout_height="1dp"  
163    android:layout_margin="10dp"  
164    android:background="@color/lightgray" />  
165  
166<TextView  
167    android:layout_width="wrap_content"  
168    android:layout_height="wrap_content"  
169    android:gravity="center"  
170    android:padding="5dp"  
171    android:text="Kamu Harus Tau"  
172    android:textColor="@android:color/darker_gray"  
173    ""  
174    android:textStyle="bold"/>  
175</LinearLayout>  
176</android.support.v7.widget.CardView>  
177  
178<android.support.v7.widget.CardView  
179    android:layout_width="140dp"  
180    android:layout_height="160dp"  
181    android:layout_margin="10dp"  
182    android:clickable="true"  
183    android:foreground="?android:attr/  
selectableItemBackground">  
184  
185    <LinearLayout  
186        android:layout_width="match_parent"  
187        android:layout_height="match_parent"  
188        android:background="@color/colorPrimaryDark"  
189        android:gravity="center"  
190        android:orientation="vertical">  
191  
192    <ImageView  
193        android:id="@+id/panic"  
194        android:layout_width="64dp"  
195        android:layout_height="64dp"  
196        android:padding="10dp"
```

```

197         android:src="@drawable/panic" />
198
199     <TextView
200         android:layout_width="wrap_content"
201         android:layout_height="wrap_content"
202         android:layout_marginTop="10dp"
203         android:background="@android:color/
204             holo_blue_bright"
205             android:text="Don't Panic"
206             android:textColor="@color/colorAccent"
207             android:textStyle="bold" />
208
209     <View
210         android:layout_width="match_parent"
211         android:layout_height="1dp"
212         android:layout_margin="10dp"
213         android:background="@color/lightgray" />
214
215     <TextView
216         android:layout_width="wrap_content"
217         android:layout_height="wrap_content"
218         android:gravity="center"
219         android:padding="5dp"
220         android:text="Wajib Baca !!"
221         android:textColor="@android:color/darker_gray
222             "
223             android:textStyle="bold"/>
224
225     </LinearLayout>
226 </android.support.v7.widget.CardView>
227
228 </LinearLayout>
229
230 <LinearLayout
231     android:layout_width="match_parent"
232     android:layout_height="wrap_content"
233     android:clipToPadding="false"
234     android:gravity="center"
235     android:orientation="horizontal">
236
237     </LinearLayout>
238 </LinearLayout>
```

4.2.3 Membuat Menu *About*

Menu About yaitu menu yang berisi tentang sebab akibat terjadinya banjir. Dimana informasi tersebut disajikan melalui video dan tulisan.

langkah pertama yaitu harus membuat sebuah *activity* baru yang diberi nama *about*. Kemudian masukan video yang akan digunakan kedalam folder raw.

Kemudian masukan *code* berikut ini kedalam file *about.java*

```
1 package com.example.lalitachandiany.monitoring;
```

```

2 import android.content.Intent;
3 import android.net.Uri;
4 import android.provider.MediaStore;
5 import android.support.v7.app.AppCompatActivity;
6 import android.os.Bundle;
7 import android.view.View;
8 import android.widget.Button;
9 import android.widget.MediaController;
10 import android.widget.VideoView;
11
12 public class AboutActivity extends AppCompatActivity {
13     private Button btnback;
14
15     @Override
16     protected void onCreate(Bundle savedInstanceState) {
17         super.onCreate(savedInstanceState);
18         setContentView(R.layout.activity_about);
19
20         VideoView videoView = findViewById(R.id.video_view);
21
22         Button btnback = (Button) findViewById(R.id.button1);
23         String videoPath = "android.resource://" + getPackageName() +
24         "/" + R.raw.about;
25         Uri uri = Uri.parse(videoPath);
26         videoView.setVideoURI(uri);
27
28         MediaController mediaController = new MediaController(this);
29         videoView.setMediaController(mediaController);
30         mediaController.setAnchorView(videoView);
31
32         btnback.setOnClickListener(new View.OnClickListener() {
33             public void onClick(View v) {
34                 // TODO Auto-generated method stub
35                 startActivity(new Intent(AboutActivity.this,
36                 HomeActivity.class));
37                 finish();
38             }
39         });
40     }
}

```

Untuk tampilanya masukan *code* berikut ini kedalam file activityabout.xml

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     xmlns:app="http://schemas.android.com/apk/res-auto"
5     xmlns:tools="http://schemas.android.com/tools"
6     android:layout_width="match_parent"
7     android:layout_height="match_parent"
8     tools:context="com.example.lalitachandiany.monitoring.
9     AboutActivity">
10
11     <LinearLayout
12         android:layout_width="match_parent"
13         android:layout_height="match_parent"

```

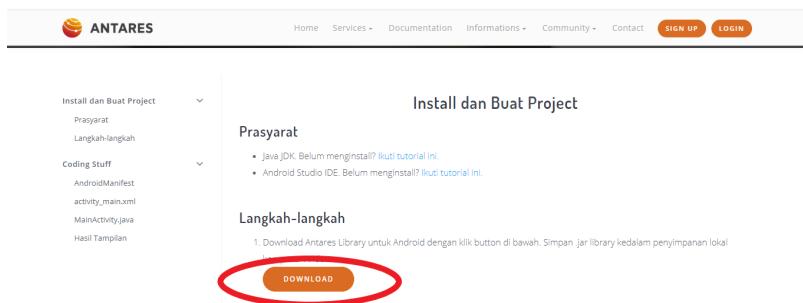
```
13     android:background="@drawable/bcempat"
14     android:gravity="center"
15     android:orientation="vertical">
16
17     <VideoView
18         android:id="@+id/video_view"
19         android:layout_width="550dp"
20         android:layout_height="250dp" />
21
22
23     <TextView
24         android:layout_width="wrap_content"
25         android:layout_height="wrap_content"
26         android:layout_marginTop="10dp"
27         android:text="Sekilas Tentang banjir"
28         android:textStyle="bold" />
29
30     <View
31         android:layout_width="match_parent"
32         android:layout_height="1dp"
33         android:layout_margin="10dp"
34         android:background="@color/lightgray" />
35
36     <TextView
37         android:layout_width="wrap_content"
38         android:layout_height="130dp"
39         android:gravity="fill_vertical"
40         android:padding="5dp"
41         android:text="Banjir merupakan salah satu bencana yang
42             sering terjadi di Indonesia khususnya didaerah kurang serapan air
43             . Selain itu Banjir adalah kondisi dimana suatu wilayah terendam
44             air karena luapan air yang berlebihan.Untuk lebih jelas Silahkan
45             Tonton Video"
46             android:textAppearance="@style/TextAppearance.AppCompat.
47             Body2"
48             android:textColor="@android:color/black"
49             android:textSize="14sp"
50             android:textStyle="bold" />
51
52     <Button
53         android:id="@+id/button1"
54         android:layout_width="134dp"
55         android:layout_height="42dp"
56         android:background="@drawable/bcempat"
57         android:text="Kembali"
58         android:textColor="@color/colorAccent" />
59
60
61     </LinearLayout>
62
63
64 </LinearLayout>
```

4.2.4 Membuat Menu Ketinggian Air

Menu ketinggian air merupakan sebuah menu untuk menampilkan informasi mengenai ketinggian air yang dibaca oleh sensor jarak secara realtime.

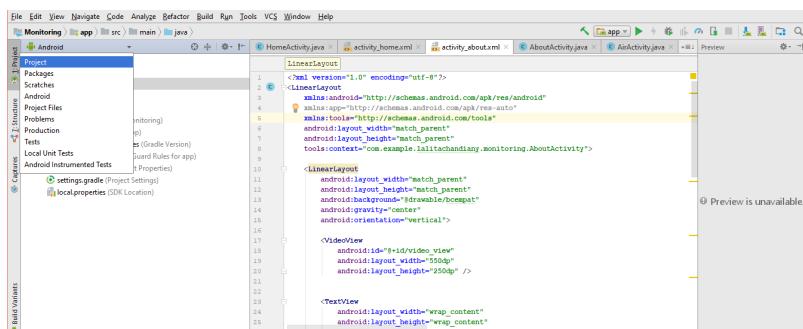
Langkah pertama kita harus mendownload library antares dengan cara berikut :

1. Login ke antares kemudian pilih *documentation* lalu pilih android dan *go to tutorial*
2. Kemudian *download library* dengan cara memilih *download*.



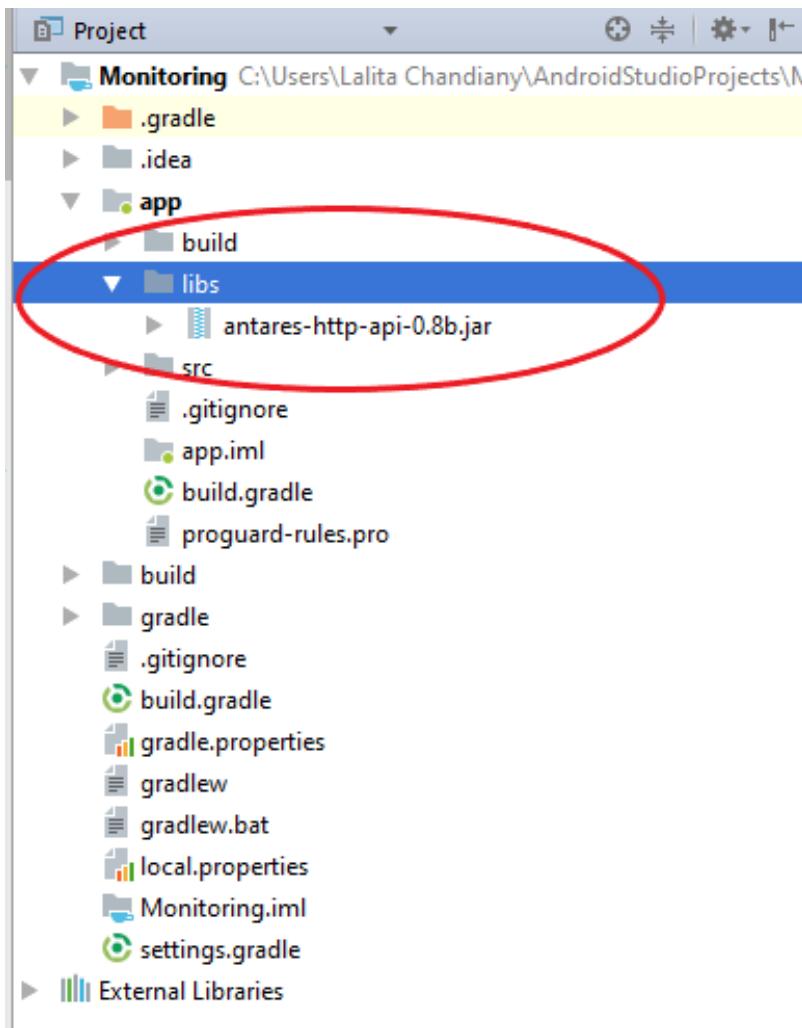
Gambar 4.35 Library Antares

3. Setelah itu masukan *library* yang telah didownload ke folder libs yang ada di android studio.Klik pada bagian android kemudian pilih libs.



Gambar 4.36 Cara Masuk ke Folder Libs

4. Setelah itu *copy-Paste library jar* yang telah didownload ke app -libs. Maka jika sudah *copy - paste* tampilannya seperti berikut.



Gambar 4.37 Tampilan Folder Libs

5. Agar bisa mengeksekusi API Antares, dibutuhkan akses Internet dan agar bisa mengakses Internet di Android, kita harus tambahkan baris berikut di AndroidManifest.xml.

```
1 <!-- IMPORTANT !!!!!!! -->
2 <!-- Tambahkan line berikut di AndroidManifest.xml Anda -->
3 <!-- Setelah manifest -->
4 <uses-permission android:name="android.permission.INTERNET" />
5 <!-- Sebelum application -->
```

6. Kemudian buat *activity* baru dengan nama monitring. *Activity* ini bertujuan untuk menu ketinggian air. Lalu masukan *code* berikut kedalam file monitoring.java

```
1 package com.example.lalitachandiany.monitoring;
2
3 import android.content.Intent;
4 import android.os.Handler;
5 import android.os.Message;
6 import android.support.v7.app.AppCompatActivity;
7 import android.os.Bundle;
8 import android.util.Log;
9 import android.view.View;
10 import android.widget.Button;
11 import android.widget.ProgressBar;
12 import android.widget.TextView;
13 import android.widget.Toast;
14
15 import org.json.JSONException;
16 import org.json.JSONObject;
17
18 import id.co.telkom.iot.AntaresHTTPAPI;
19 import id.co.telkom.iot.AntaresResponse;
20
21 public class MonitoringActivity extends AppCompatActivity
22     implements AntaresHTTPAPI.OnResponseListener {
23
24     // Deklarasi button , text view dll atau deklarasi variabel
25     private Button btnRefresh;
26     private Button btnback;
27
28     private TextView txtData, txtData2;
29     private String TAG = "ANTARES-API";
30     private AntaresHTTPAPI antaresAPIHTTP;
31     private String dataDevice;
32
33     private ProgressBar progressBar;
34     private TextView persentase;
35     private int Value;
36     private int Hasil;
37
38     @Override
39     protected void onCreate(Bundle savedInstanceState) {
40         super.onCreate(savedInstanceState);
41         setContentView(R.layout.activity_monitoring);
42
43         // inisialisasi buton progresbar text view dll
44         progressBar = findViewById(R.id.progress);
45         persentase = findViewById(R.id.persentase);
46         progressBar.setProgress(0); // Set Progress Dimulai Dari 0
47
48         // — Inisialisasi UI yang digunakan di aplikasi — //
49         btnRefresh = (Button) findViewById(R.id.btnRefresh);
50         btnback = (Button) findViewById(R.id.back);
51         txtData = (TextView) findViewById(R.id.txtData);
52         txtData2 = (TextView) findViewById(R.id.txtDatadua);
```



```
142     }
143 }
144 }
```

7. Kemudian untuk tampilan nya masukan *code* berikut kedalam file activitymonitoring.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/
  /android"
3   xmlns:tools="http://schemas.android.com/tools"
4   android:layout_width="match_parent"
5   android:layout_height="match_parent"
6   android:background="@drawable/airbckground"
7   android:orientation="vertical"
8   tools:context="com.example.lalitachandiany.monitoring.
  MonitoringActivity">
9
10  <TextView
11    android:id="@+id/txtData"
12    android:layout_width="wrap_content"
13    android:layout_height="wrap_content"
14    android:layout_above="@+id/persentase"
15    android:layout_marginBottom="16dp"
16    android:layout_toStartOf="@+id/back"
17    android:background="@android:color/holo_blue_bright"
18    android:text="Distance"
19    android:textSize="24sp" />
20
21  <TextView
22    android:id="@+id/txtDatadua"
23    android:layout_width="wrap_content"
24    android:layout_height="wrap_content"
25    android:layout_alignBaseline="@+id/txtData"
26    android:layout_alignBottom="@+id/txtData"
27    android:layout_alignEnd="@+id/progress"
28    android:background="@android:color/holo_blue_bright"
29    android:text="Duration"
30    android:textSize="24sp" />
31
32
33
34
35  <Button
36    android:id="@+id/btnRefresh"
37    android:layout_width="wrap_content"
38    android:layout_height="wrap_content"
39    android:layout_alignParentBottom="true"
40    android:layout_centerHorizontal="true"
41    android:layout_marginBottom="81dp"
42    android:backgroundTint="@color/colorPrimaryDark"
43    android:text="Refresh" />
44
45  <TextView
46    android:id="@+id/textView"
47    android:layout_width="wrap_content"
```

```
48        android:layout_height="wrap_content"
49        android:layout_alignParentTop="true"
50        android:layout_centerHorizontal="true"
51        android:layout_marginTop="34dp"
52        android:shadowColor="@color/cardview_dark_background"
53        android:text="Ketinggian AIR Saat Ini :"
54        android:textColor="@android:color/holo_red_dark"
55        android:textSize="30sp" />
56
57    <TextView
58        android:id="@+id/textView2"
59        android:layout_width="wrap_content"
60        android:layout_height="wrap_content"
61        android:layout_alignParentStart="true"
62        android:layout_below="@+id/textView"
63        android:layout_marginStart="11dp"
64        android:layout_marginTop="41dp"
65        android:text="Distance"
66        android:textColor="@android:color/background_dark"
67        android:textSize="24sp" />
68
69    <TextView
70        android:id="@+id/textView3"
71        android:layout_width="wrap_content"
72        android:layout_height="wrap_content"
73        android:layout_alignBaseline="@+id/textView2"
74        android:layout_alignBottom="@+id/textView2"
75        android:layout_alignStart="@+id/txtDatadua"
76        android:text="Duration"
77        android:textColor="@android:color/background_dark"
78        android:textSize="24sp" />
79
80    <Button
81        android:id="@+id/back"
82        android:layout_width="134dp"
83        android:layout_height="40dp"
84        android:layout_alignParentBottom="true"
85        android:layout_centerHorizontal="true"
86        android:layout_marginBottom="15dp"
87        android:background="@drawable/bcempat"
88        android:text="Kembali" />
89
90    <ProgressBar
91        android:id="@+id/progress"
92        style="?android:attr/progressBarStyleHorizontal"
93        android:layout_width="300dp"
94        android:layout_height="wrap_content"
95        android:layout_below="@+id/persentase"
96        android:layout_centerHorizontal="true"
97        android:layout_gravity="center_vertical"
98        android:background="@android:color/holo_blue_bright"
99        android:max="100"
100       android:progress="0" />
101
102    <TextView
103        android:id="@+id/persentase"
```

```

104     android:layout_width="50dp"
105     android:layout_height="wrap_content"
106     android:layout_centerHorizontal="true"
107     android:layout_centerVertical="true"
108     android:background="@android:color/holo_blue_bright"
109     android:gravity="center"
110     android:text="0 m"
111     android:textColor="@color/colorAccent"
112     android:textSize="15sp"
113     android:textStyle="bold" />
114
115 </RelativeLayout>
```

4.2.5 Membuat Menu Keterangan Air

Menu keterangan air merupakan sebuah menu yang apabila di klik akan menampilkan sebuah halaman yang berisi tentang keterangan level-level ketinggian air.

Untuk membuat menu keterangan air ini hal yang pertama kali harus dibuat yaitu membuat sebuah *activity* baru dengan nama ketinggian air. Kemudian masukan *code* berikut kedalam file ketinggianair.java

```

1 package com.example.lalitachandiany.monitoring;
2
3 import android.content.Intent;
4 import android.support.v7.app.AppCompatActivity;
5 import android.os.Bundle;
6 import android.view.View;
7 import android.widget.Button;
8
9 public class AirActivity extends AppCompatActivity {
10     private Button btnback;
11     @Override
12     protected void onCreate(Bundle savedInstanceState) {
13         super.onCreate(savedInstanceState);
14         setContentView(R.layout.activity_air);
15
16         Button btnback = (Button) findViewById(R.id.button1);
17
18         btnback.setOnClickListener(new View.OnClickListener() {
19             public void onClick(View v) {
20                 // TODO Auto-generated method stub
21                 startActivity(new Intent(AirActivity.this,
22                     HomeActivity.class));
23                 finish();
24             }
25         });
26     }
}
```

Kemudian untuk tampilan menu ketinggian airnya masukan perintah berikut di file activityketinggianair.xml

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout
```

```
3     xmlns: android="http://schemas.android.com/apk/res/android"
4     xmlns: app="http://schemas.android.com/apk/res-auto"
5     xmlns: tools="http://schemas.android.com/tools"
6     android: layout_width="match_parent"
7     android: layout_height="match_parent"
8     tools: context="com.example.lalitachandiany.monitoring.AirActivity"
9     ">
10
11    <LinearLayout
12        android: layout_width="match_parent"
13        android: layout_height="match_parent"
14        android: background="@drawable/bcempat"
15        android: gravity="center"
16        android: orientation="vertical">
17
18        <ImageView
19            android: id="@+id/about"
20            android: layout_width="263dp"
21            android: layout_height="150dp"
22            android: padding="10dp"
23            android: src="@drawable/ombak" />
24
25        <TextView
26            android: layout_width="wrap_content"
27            android: layout_height="wrap_content"
28            android: layout_marginTop="10dp"
29            android: text="Ketinggian Air"
30            android: textStyle="bold" />
31
32        <View
33            android: layout_width="match_parent"
34            android: layout_height="1dp"
35            android: layout_margin="10dp"
36            android: background="@color/lightgray" />
37
38        <TextView
39            android: layout_width="278dp"
40            android: layout_height="162dp"
41            android: gravity="center"
42            android: padding="5dp"
43            android: text="Jika sensor membaca jarak ketinggian air
44            20–10m maka aman dari banjir.
45            Apabila sensor membaca ketinggian air kurang dari 5m dan
46            adanya suara serta lampu led menyala artinya AWAS BANJIR !!""
47            android: textAppearance="@style/TextAppearance.AppCompat.
48            Body2" />
49
50        <Button
51            android: id="@+id/button1"
52            android: layout_width="136dp"
53            android: layout_height="52dp"
54            android: background="@drawable/bcempat"
55            android: text="Kembali" />
56
57    </LinearLayout>
```

55 </LinearLayout>

4.2.6 Membuat Menu *Don't Panic*

Pada menu *don't panic* ini akan menampilkan sebuah video dan text yang berisi tentang langkah-langkah apa saja yang harus dilakukan ketika datangnya banjir. Isi pada menu ini bertujuan agar *user* tidak panik dan tau apa yang harus dilakukan ketika banjir datang.

Untuk membuatnya sama seperti langkah sebelumnya yaitu membuat *activity* baru dengan nama *panic* , kemudian masukan *code* ke file panic.java sebagai berikut :

```
1 package com.example.lalitachandiany.monitoring;
2
3 import android.content.Intent;
4 import android.net.Uri;
5 import android.support.v7.app.AppCompatActivity;
6 import android.os.Bundle;
7 import android.view.View;
8 import android.widget.Button;
9 import android.widget.MediaController;
10 import android.widget.VideoView;
11
12 public class PanicActivity extends AppCompatActivity {
13     private Button btnback;
14
15     @Override
16     protected void onCreate(Bundle savedInstanceState) {
17         super.onCreate(savedInstanceState);
18         setContentView(R.layout.activity_panic);
19
20         VideoView videoView = findViewById(R.id.video_view);
21
22         Button btnback = (Button) findViewById(R.id.button1);
23         String videoPath = "android.resource://" + getPackageName() +
24         "/" + R.raw.panic;
25         Uri uri = Uri.parse(videoPath);
26         videoView.setVideoURI(uri);
27
28         MediaController mediaController = new MediaController(this);
29         videoView.setMediaController(mediaController);
30         mediaController.setAnchorView(videoView);
31
32         btnback.setOnClickListener(new View.OnClickListener() {
33             public void onClick(View v) {
34                 // TODO Auto-generated method stub
35                 startActivity(new Intent(PanicActivity.this,
36                 HomeActivity.class));
37                 finish();
38             }
39         });
40     }
```

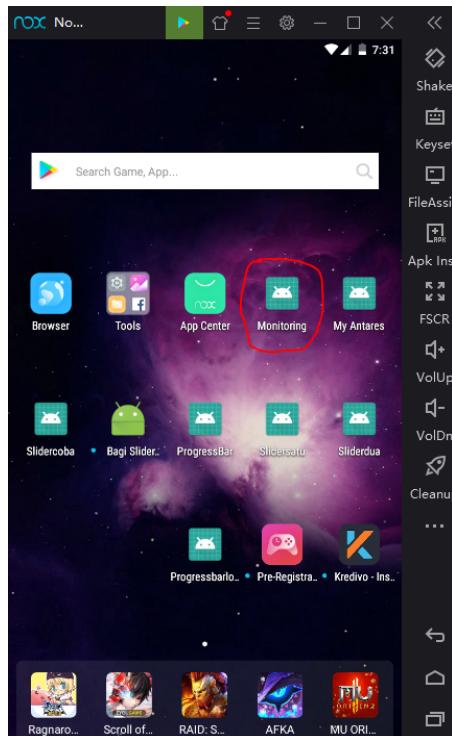
Kemudian untuk tampilan menu *don't pannic* masukan perintah berikut di file panic.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     xmlns:app="http://schemas.android.com/apk/res-auto"
5     xmlns:tools="http://schemas.android.com/tools"
6     android:layout_width="match_parent"
7     android:layout_height="match_parent"
8     tools:context="com.example.lalitachandiany.monitoring.
9         PanicActivity">
10
11     <LinearLayout
12         android:layout_width="match_parent"
13         android:layout_height="match_parent"
14         android:background="@drawable/bcempat"
15         android:gravity="center"
16         android:orientation="vertical">
17
18         <VideoView
19             android:id="@+id/video_view"
20             android:layout_width="550dp"
21             android:layout_height="250dp" />
22
23         <TextView
24             android:layout_width="wrap_content"
25             android:layout_height="wrap_content"
26             android:layout_marginTop="10dp"
27             android:text="Ikuti Petunjuk"
28             android:textStyle="bold" />
29
30         <View
31             android:layout_width="match_parent"
32             android:layout_height="1dp"
33             android:layout_margin="10dp"
34             android:background="@color/lightgray" />
35
36         <TextView
37             android:layout_width="284dp"
38             android:layout_height="103dp"
39             android:gravity="fill_vertical"
40             android:padding="5dp"
41             android:text="Pada saat banjir masyarakat tidak harus
42             panik hal pertama yang harus di lakukan :
43             1. Selmatkan barang-barang berharga .
44             2. Pergi ketempat yang lebih tinggi .
45             3. Selalu pantau tentang informasi Banjir ."
46             android:textAppearance="@style/TextAppearance.AppCompat.
47             Body2" />
48
49         <Button
50             android:id="@+id/button1"
51             android:layout_width="134dp"
52             android:layout_height="40dp"
53             android:background="@drawable/bcempat"
54             android:text="Kembali" />
```

```
52  
53  
54  
55     </LinearLayout>  
56 </LinearLayout>  
57
```

4.3 Tampilan Aplikasi Monitoring

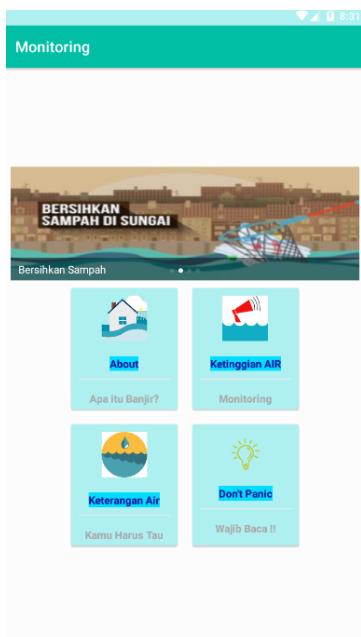
Tampilan dari aplikasi monitoring yang sebelumnya telah dibuat, jika mengikuti semua langkah-langkahnya tanpa adanya step yang terlewatkan maka akan menjadi seperti berikut :



Gambar 4.38 Tampilan Aplikasi DI Emulator



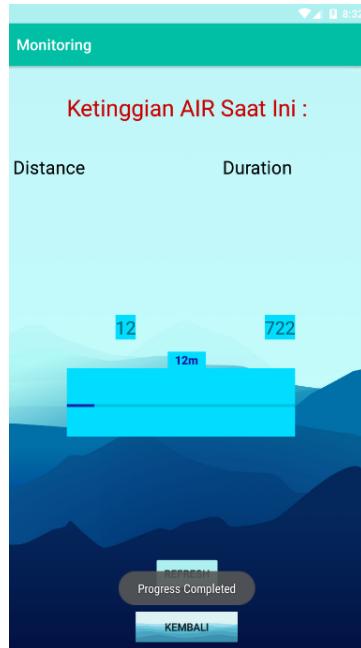
Gambar 4.39 Splash Screen



Gambar 4.40 Tampilan Menu Utama



Gambar 4.41 Tampilan Menu About



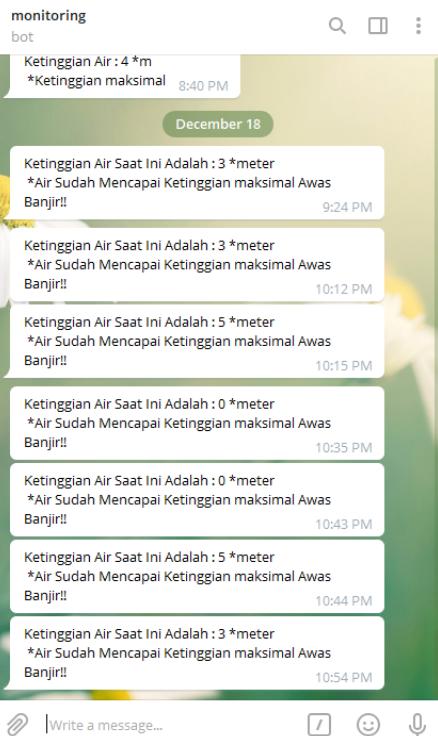
Gambar 4.42 Tampilan Ketinggian Air



Gambar 4.43 Tampilan Menu Keterangan Air



Gambar 4.44 Tampilan Menu *Don't Panic*



Gambar 4.45 Tampilan Notifikasi Telegram

DAFTAR PUSTAKA
