

# **BAGIAN I**

## **PENGUKURAN BERAT DIMENSI**

### **1.1. Definisi**

Pengukuran berat dimensi barang merupakan proses dimana barang diletakkan ditempat tertentu dan dilakukan pengukuran untuk mendapatkan nilai berat volume barang. Proses pengukuran ini dilakukan pada jas pengiriman barang

Ada dua jenis cara perhitungan berat barang yang dikenal luas dalam jasa pengiriman barang, perhitungan berat sesungguhnya dan berat volume. Pengertian dan cara perhitungan tersebut bisa dipahami dengan penjelasan berikut:

- **Berat sesungguhnya (Berat aktual)**

Berat sesungguhnya adalah berat yang diperoleh dari hasil penimbangan. Lazimnya di Indonesia menggunakan satuan kilogram (kg).

Saat melakukan penimbangan barang, biasanya berat barang sering tidak tepat menunjukkan bilangan bulat. Contohnya 4,3 Kg, atau 12,7 Kg. Apabila hal ini terjadi, biasanya pihak ekspedisi membulatkan berat barang ke atas. Misalkan 4,3 kg dibulatkan menjadi 5 kg.

- **Berat Volume (Berat Volumetrik)**

Berat volume adalah berat yang didapat dari hasil perhitungan dengan menggunakan ukuran volume barang yang akan dikirim. Perhitungan ini didasarkan pada kondisi dimana berat aktual barang kecil (ringan) akan tetapi memakan tempat (volume besar).

### **1.2. Cara Perhitungan Pengukuran**

Berat volume adalah berat yang didapat dari hasil perhitungan dengan menggunakan ukuran volume barang yang akan dikirim. Perhitungan ini didasarkan pada kondisi dimana berat aktual barang kecil (ringan) akan tetapi memakan tempat (volume besar).

#### **1. Rumus yang digunakan untuk pengiriman via darat adalah:**

- Panjang (cm) X Lebar (cm) X Tinggi (cm) : 4000.

- Rumus ini sudah dipakai dan diterima secara luas dan di akui oleh ASPERINDO.
- Sebagai contoh, barang dengan panjang 100 cm, lebar 100 cm dan tinggi 100 cm, maka memiliki berat volumetrik :  $100 \times 100 \times 100 / 4000 = 250$  Kg.

**2. Rumus yang digunakan untuk pengiriman barang domestik & internasional via udara adalah:**

- Panjang (cm) X Lebar (cm) X Tinggi (cm) : 6.000.
- Rumus ini sudah dipakai dan diterima secara luas dan di akui oleh ASPERINDO.
- Sebagai contoh, barang dengan panjang 100 cm, lebar 100 cm dan tinggi 100 cm, maka memiliki berat volumetrik :  $100 \times 100 \times 100 / 6.000 = 166.66$  Kg atau 167 kg.

**3. Rumus yang digunakan untuk pengiriman barang domestik & internasional via laut adalah:**

- Panjang (cm) X Lebar (cm) X Tinggi (cm) : 1.000.000 = M3 (kubikasi)
- Rumus ini sudah dipakai dan diterima secara luas di dunia
- Sebagai contoh, barang dengan panjang 100 cm, lebar 100 cm dan tinggi 100 cm, maka memiliki berat volumetrik :  $1000 \times 1000 \times 1000 / 1.000.000 = 1$  M3 atau 1 Kubik

**1.3. Pengukuran Berat Dimensi Barang**

Proses pengukuran berat dimensi barang akan menghasilkan 2 berat yaitu berat volume dan berat asli, sehingga menghasilkan nilai perbandingan kedua berat tersebut. Dengan nilai tersebut, kita dapat menentukan berat mana yang akan pilih dalam penentuan harga pengiriman barang. Berat yang lebih besar akan dipilih untuk menjadi ketentuan harga pengiriman barang.

## **BAGIAN II**

# ARDUINO

## 2.1. Definisi

Arduino adalah mikrokontroler *single-board* yang berbasis *open source* yang menggunakan prosesor Atmel AVR dan memiliki software yang menggunakan Bahasa pemrograman sendiri yang memiliki *syntax* mirip dengan Bahasa pemrograman C. Karena Arduino bersifat *open-source* maka siapa saja dapat menggunakan skema hardware-nya dan dapat memudahkan penggunaan elektronik dalam berbagai bidang untuk membangunnya.

Arduino adalah kit elektronik atau papan rangkaian elektronik open source yang di dalamnya terdapat komponen utama yaitu sebuah chip mikrokontroler dengan jenis AVR dari perusahaan Atmel. Mikrokontroler itu sendiri adalah chip atau IC (integrated circuit) yang bisa diprogram menggunakan komputer. Tujuan menanamkan program pada mikrokontroler adalah agar rangkaian elektronik dapat membaca input, memproses input tersebut dan kemudian menghasilkan output sesuai yang diinginkan. Jadi mikrokontroler bertugas sebagai ‘otak’ yang mengendalikan input, proses dan output sebuah rangkaian elektronik.

Mikrokontroler ada pada perangkat elektronik di sekeliling kita. Misalnya handphone, MP3 player, DVD, televisi, AC, dll. Mikrokontroler juga dipakai untuk keperluan mengendalikan robot. Baik robot mainan, maupun robot industri. Karena komponen utama Arduino adalah mikrokontroler, maka Arduino pun dapat diprogram menggunakan komputer sesuai kebutuhan kita.

## 2.2. Macam – Macam Arduino

Arduino juga memiliki berbagai jenis-jenis boards microcontroller :

### 1. Arduino Uno

Arduino Uno adalah board mikrokontroler berdasarkan ATmega328P (datasheet). Ini memiliki 14 pin input / output digital dimana 6 pin input tersebut dapat digunakan sebagai output PWM, 6 input analog, kristal kuarsa 16 MHz, koneksi USB, colokan listrik, header ICSP, dan tombol reset. Ini berisi semua yang diperlukan untuk mendukung mikrokontroler. cukup sambungkan ke komputer dengan kabel USB atau daya dengan adaptor AC-ke-DC atau baterai untuk memulai. Untuk pemrograman cukup menggunakan koneksi USB type A to type B.

Uno berbeda dengan semua board sebelumnya dalam hal koneksi USB-to-serial yaitu menggunakan fitur Atmega8U2 yang diprogram sebagai

konverter USB-to-serial berbeda dengan board sebelumnya yang menggunakan chip FTDI driver USB-to-serial.



Gambar 2.1 Arduino Uno

Spesifikasi	
<b>Microcontroller</b>	ATmega328P
<b>Operating Voltage</b>	5V
<b>Input Voltage (recommended)</b>	7-12V
<b>Input Voltage (limit)</b>	6-20V
<b>Digital I/O Pin</b>	14 (6 pin input dapat digunakan sebagai output PWM)
<b>PWM Digital I/O Pin</b>	6
<b>Analog Input Pin</b>	6
<b>Arus DC per Pin I/O</b>	20 mA
<b>Arus DC untuk 3.3V Pin</b>	50 mA
<b>Flash Memory</b>	32 KB (ATmega328P) dimana 0,5 KB digunakan oleh bootloader
<b>SRAM</b>	2 KB (ATmega328P)
<b>EEPROM</b>	1 KB (ATmega328P)
<b>Clock Speed</b>	16 MHz
<b>LED_BUILTIN</b>	13
<b>Panjang</b>	68,6 mm
<b>Lebar</b>	53,4 mm
<b>Berat</b>	25 g

Arduino Uno merupakan board microcontroller yang paling banyak digunakan, karna merupakan board yang digunakan untuk belajar atau baru pertama kali menggunakan platform ini.

## 2. Arduino Leonardo

Arduino Leonardo adalah board mikrokontroler berdasarkan ATmega32u4 (datasheet). Ia memiliki 20 pin input / output digital (7 di antaranya dapat digunakan sebagai output PWM dan 12 sebagai input analog), osilator kristal 16 MHz, koneksi micro USB, colokan listrik, header ICSP, dan tombol reset. Ini berisi semua yang diperlukan untuk mendukung mikrokontroler; cukup sambungkan ke komputer dengan kabel USB atau daya dengan adaptor AC-ke-DC atau baterai untuk memulai.



Gambar 2.2 Arduino Leonardo

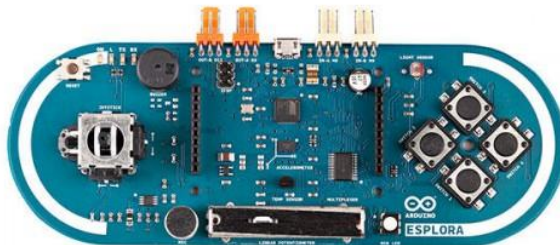
### Spesifikasi

<b>Microcontroller</b>	ATmega32u4
<b>Operating Voltage</b>	5V
<b>Input Voltage (recommended)</b>	7-12V
<b>Input Voltage (limit)</b>	6-20V
<b>Digital I/O Pin</b>	20
<b>PWM Pin</b>	7
<b>Analog Input Pin</b>	12
<b>Arus DC per Pin I/O</b>	40 mA
<b>Arus DC untuk 3.3V Pin</b>	50 mA
<b>Flash Memory</b>	32 KB (ATmega328P) dimana 4 KB digunakan oleh bootloader
<b>SRAM</b>	2,5 KB (ATmega32u4)
<b>EEPROM</b>	1 KB (ATmega32u4)
<b>Clock Speed</b>	16 MHz
<b>LED_BUILTIN</b>	13

<b>Panjang</b>	68,6 mm
<b>Lebar</b>	53,3 mm
<b>Berat</b>	20 g

### 3. Arduino Esplora

Arduino Esplora adalah board berbasis Arduino Leonardo dengan sensor dan aktuator terintegrasi. Esplora berbeda dari semua board sebelumnya karena Esplora menyediakan sejumlah sensor terintegrasi yang siap pakai untuk interaksi. Ini dirancang untuk orang-orang yang ingin bangun dan berjalan dengan Arduino tanpa harus belajar tentang elektronik terlebih dahulu. Esplora menggunakan mikrokontroler AVR Atmega32U4 sama seperti Leonardo dengan osilator kristal 16 MHz dan koneksi micro USB yang dapat bertindak sebagai perangkat klien USB, seperti mouse atau keyboard. Di sudut kiri atas board ada tombol reset, yang dapat digunakan untuk me-restart papan tersebut.



Gambar 2.3 Arduino Esplora

#### Spesifikasi

<b>Microcontroller</b>	ATmega32u4
<b>Operating Voltage</b>	5V
<b>Flash Memory</b>	32 KB (ATmega328P) dimana 4 KB digunakan oleh bootloader
<b>SRAM</b>	2,5 KB (ATmega32u4)
<b>EEPROM</b>	1 KB (ATmega32u4)
<b>Clock Speed</b>	16 MHz
<b>LED_BUILTIN</b>	13
<b>Panjang</b>	164,04 mm
<b>Lebar</b>	60 mm
<b>Berat</b>	53 gr

#### 4. Arduino Micro

Arduino Micro adalah board terkecil dari keluarga arduino, mudah diintegrasikan dalam benda sehari-hari untuk menjadikannya interaktif.

Micro menggunakan mikrokontroler ATmega32u4, yang dikembangkan bersama dengan Adafruit. Micro memiliki 20 pin input / output digital (7 di antaranya dapat digunakan sebagai output PWM dan 12 sebagai input analog), osilator kristal 16 MHz, koneksi micro USB, header ICSP, dan tombol reset. Cukup sambungkan ke komputer dengan kabel micro USB untuk memulai. Memiliki USB bawaan yang membuat Micro dikenali sebagai mouse atau keyboard.



Gambar 2.4 Arduino Micro

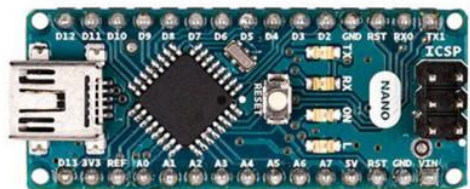
##### Spesifikasi

<b>Microcontroller</b>	ATmega32u4
<b>Operating Voltage</b>	5V
<b>Input Voltage (recommended)</b>	7-12V
<b>Input Voltage (limit)</b>	6-20V
<b>Digital I/O Pin</b>	20
<b>PWM Pin</b>	7
<b>Analog Input Pin</b>	12
<b>Arus DC per Pin I/O</b>	40 mA
<b>Arus DC untuk 3.3V Pin</b>	50 mA
<b>Flash Memory</b>	32 KB (ATmega328P) dimana 4 KB digunakan oleh bootloader
<b>SRAM</b>	2,5 KB (ATmega32u4)
<b>EEPROM</b>	1 KB (ATmega32u4)
<b>Clock Speed</b>	16 MHz
<b>LED_BUILTIN</b>	13

<b>Panjang</b>	68,6 mm
<b>Lebar</b>	53,3 mm
<b>Berat</b>	20 g

### 5. Arduino Nano

Arduino Nano adalah board kecil yang compact mirip dengan Arduino UNO. Walaupun bentuk boardnya kecil, tetapi lengkap, dan ramah dengan menggunakan microcontroler ATmega328P (Arduino Nano 3.x). Arduino Nano memiliki fungsi yang kurang lebih sama dari Arduino Duemilanove, tetapi dalam paket yang berbeda.



Gambar 2.5 Arduino Nano

#### Spesifikasi

<b>Microcontroller</b>	ATmega328
<b>Architerture</b>	AVR
<b>Operating Voltage</b>	5V
<b>Input Voltage (recommended)</b>	7-12V
<b>Digital I/O Pin</b>	22
<b>PWM Output</b>	6
<b>Analog Input Pin</b>	8
<b>Arus DC per Pin I/O</b>	40 mA
<b>Flash Memory</b>	32 KB (ATmega328P) dimana 2 KB digunakan oleh bootloader
<b>SRAM</b>	2 KB (ATmega328)
<b>EEPROM</b>	1 KB (ATmega328)
<b>Clock Speed</b>	16 MHz
<b>Konsumsi daya</b>	19 mA
<b>Panjang</b>	45 mm
<b>Lebar</b>	18 mm
<b>Berat</b>	7 g



## 6. Arduino Mega 2560

Arduino Mega 2560 adalah board mikrokontroler yang menggunakan ATmega2560. Mega 2560 memiliki 54 pin input / output digital (15 di antaranya dapat digunakan sebagai output PWM), 16 input analog, 4 UART (port serial perangkat keras), osilator kristal 16 MHz, koneksi USB, colokan listrik, header ICSP, dan tombol reset. Ini berisi semua yang diperlukan untuk mendukung mikrokontroler; cukup sambungkan ke komputer dengan kabel USB atau daya dengan adaptor AC-ke-DC atau baterai untuk memulai.

Arduino Mega 2560 dibuat untuk proyek/project yang lebih kompleks dan memiliki ruang penyimpanan yang lebih besar untuk sketsa, sehingga board Arduino Mega 2560 biasa digunakan untuk pembuatan robot dan printer 3D.



Gambar 2.6 Arduino Mega 2560

Spesifikasi	
Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pin	54 (15 pin input dapat digunakan sebagai output PWM)
Analog Input Pin	16
Arus DC per Pin I/O	20 mA
Arus DC untuk 3.3V Pin	50 mA
Flash Memory	256 KB (ATmega328P) dimana 8 KB digunakan oleh bootloader

<b>SRAM</b>	8 KB
<b>EEPROM</b>	4 KB
<b>Clock Speed</b>	16 MHz
<b>LED_BUILTIN</b>	13
<b>Panjang</b>	101,52 mm
<b>Lebar</b>	53,3 mm
<b>Berat</b>	37 g

## 7. Arduino Due

Arduino Due adalah board Arduino pertama yang tidak menggunakan ATmega, melainkan sudah menggunakan chip yang lebih tinggi pada mikrokontroler inti yaitu ARM 32-bit (Atmel SAM3X8E ARM Cortex-M3 CPU). Dengan 54 pin input / output digital (12 di antaranya dapat digunakan sebagai output PWM), 12 input analog, 4 UART (port serial perangkat keras), jam 84 MHz, koneksi berkemampuan USB OTG, 2 DAC (digital ke analog), 2 TWI, colokan listrik, header SPI, header JTAG, tombol reset dan tombol hapus., ini merupakan board yang cocok untuk proyek Arduino skala besar yang kuat.



Gambar 2.7 Arduino Due

<b>Spesifikasi</b>	
<b>Microcontroller</b>	AT91SAM3X8E
<b>Operating Voltage</b>	3,3V
<b>Input Voltage (recommended)</b>	7-12V
<b>Input Voltage (limit)</b>	6-26V
<b>Digital I/O Pin</b>	54 (12 pin input dapat digunakan sebagai output PWM)
<b>Analog Input Pin</b>	12

<b>Analog Output Pin</b>	2 (DAC)
<b>Total DC Output pada semua jalur I / O</b>	130 mA
<b>Arus DC per Pin 5V</b>	800 mA
<b>Arus DC untuk 3.3V Pin</b>	800 mA
<b>Flash Memory</b>	512 KB
<b>SRAM</b>	96 KB (64 KB dan 32 KB)
<b>EEPROM</b>	84 KB
<b>Clock Speed</b>	84 MHz
<b>Panjang</b>	101,52 mm
<b>Lebar</b>	53,3 mm
<b>Berat</b>	36 g

## 8. Arduino Yun

Arduino Yún adalah board mikrokontroler yang menggunakan ATmega32u4 dan Atheros AR9331. Prosesor Atheros digunakan untuk mendukung distribusi Linux berdasarkan OpenWrt bernama Linino OS. Papan ini memiliki dukungan Ethernet dan WiFi built-in, port USB-A, slot kartu micro-SD, 20 pin input / output digital (7 di antaranya dapat digunakan sebagai output PWM dan 12 sebagai input analog), kristal 16 MHz osilator, koneksi micro USB, header ICSP, dan 3 tombol reset.



Gambar 2.8 Arduino Yun

Arduino Yum adalah board yang cocok untuk membangun proyek yang terhubung seperti proyek Internet of Things. Yum ini memiliki kemampuan yang berbeda dengan Arduino lainnya, karena dapat berkomunikasi dengan Linux onboard.

### Spesifikasi

- AVR Arduino microcontroller

<b>Microcontroller</b>	ATmega32u4
<b>Operating Voltage</b>	5V
<b>Input Voltage (recommended)</b>	5V
<b>Digital I/O Pin</b>	20
<b>PWM Output</b>	7
<b>Analog I/O Pin</b>	12
<b>Arus DC per Pin I/O</b>	40 mA
<b>Arus DC untuk 3.3V Pin</b>	50 mA
<b>Flash Memory</b>	32 KB (dimana 4KB digunakan oleh bootloader)
<b>SRAM</b>	2,5 KB
<b>EEPROM</b>	1 KB
<b>Clock Speed</b>	16 MHz

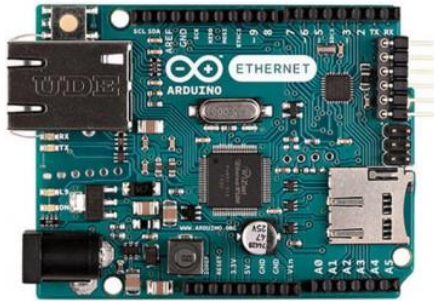
- Arduino Microprocessor

<b>Microcontroller</b>	Atheros AR9331
<b>Architecture</b>	MIPS
<b>Operating Voltage</b>	3,3V
<b>Ethernet</b>	802,3 10/100Mbit/s
<b>Wifi</b>	802,11b/g/n 2,4 GHz
<b>USB Type</b>	2,0 Host
<b>Card Reader</b>	Micro-SD
<b>RAM</b>	64 MB DDR2
<b>Flash Memory</b>	16 KB
<b>SRAM</b>	2,5 KB
<b>EEPROM</b>	1 KB
<b>Clock Speed</b>	400 MHz

## 9. Arduino Ethernet

Arduino Ethernet merupakan gabungan Arduino Uno dengan WizNet W5100 TCP / IP Ethernet Controller, sehingga kita bisa langsung menghubungkan melalui jaringan LAN pada computer.

Arduino Ethernet merupakan board mikrokontroler yang menggunakan ATmega328. Memiliki 14 pin input / output digital, 6 input analog, osilator kristal 16 MHz, koneksi RJ45, colokan listrik, header ICSP, dan tombol reset.



Gambar 2.9 Arduino Ethernet

Arduino Ethernet berbeda karena tidak memiliki chip driver USB-to-Serial pada board, tetapi Ethernet memiliki interface Wiznet Ethernet dan juga memiliki card reader microSD yang dapat kita gunakan sebagai tempat menyimpan file untuk disajikan melalui jaringan.

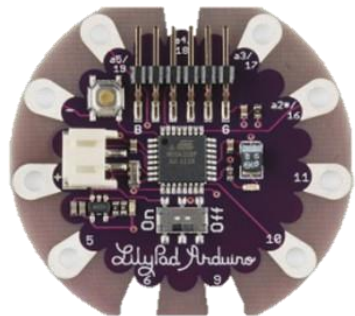
#### Spesifikasi

<b>Microcontroller</b>	ATmega328P
<b>Operating Voltage</b>	5V
<b>Input Voltage Plug (recommended)</b>	7-12V
<b>Input Voltage Plug (limit)</b>	6-20V
<b>Input Voltage PoE (limit)</b>	36-57V
<b>Digital I/O Pin</b>	14 (4 pin input dapat digunakan sebagai output PWM)
<b>Arduino Pins reserved :</b>	
	10 – 13 digunakan untuk SPI
	4 digunakan untuk SD card
	2 W5100 interrupt (saat dijembatani)
<b>Analog Input Pin</b>	6
<b>Arus DC per Pin I/O</b>	40 mA
<b>Arus DC untuk 3.3V Pin</b>	50 mA
<b>Flash Memory</b>	32 KB (ATmega328P) dimana 0,5 KB digunakan oleh bootloader
<b>SRAM</b>	2 KB (ATmega328P)
<b>EEPROM</b>	1 KB (ATmega328P)
<b>Clock Speed</b>	16 MHz
<b>W5100 TCP/IP Embedded Ethernet Controller</b>	
<b>Power Over Ethernet ready Magnetic Jack</b>	

<b>Micro SD card, with active voltage translators</b>	
<b>Panjang</b>	68,6 mm
<b>Lebar</b>	53,3 mm
<b>Berat</b>	28 gr

### 10. Arduino Lilypad Simple

Arduino Lilypad banyak digunakan pada proyek e-textiles dan wearables, karna dapat dipakai pada kain dengan hanya menjahit Arduino tersebut. Sensor dan aktuator yang dipasang serupa dengan benang konduktif. Arduino LilyPad Simple menggunakan ATmega328P dan hanya memiliki 9 pin untuk input / output. Selain itu, juga memiliki konektor JST dan sirkuit pengisian bawaan untuk baterai Lithium Polymer.

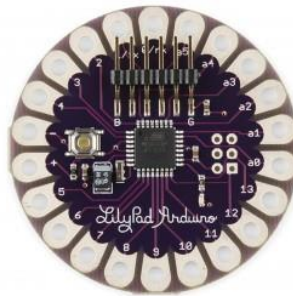


Gambar 2.10 Arduino Lilypad Simple

Spesifikasi	
<b>Microcontroller</b>	ATmega328P
<b>Operating Voltage</b>	2,7-5,5V
<b>Input Voltage</b>	2,7-5,5V
<b>Digital I/O Pin</b>	9
<b>PWM Digital I/O Pin</b>	5
<b>Analog Input Pin</b>	4
<b>Arus DC per Pin I/O</b>	40 mA
<b>Flash Memory</b>	32 KB (dimana 2 KB digunakan oleh bootloader)
<b>SRAM</b>	2 KB
<b>EEPROM</b>	1 KB
<b>Clock Speed</b>	8 MHz

## 11. Arduino Lilypad Main Board

Arduino Lilypad banyak digunakan pada proyek e-textiles dan wearables, karna dapat dipakai pada kain dengan hanya menjahit Arduino tersebut. Sensor dan aktuator yang dipasang serupa dengan benang konduktif. LilyPad Arduino Main Board menggunakan ATmega168V (versi daya rendah dari ATmega168) atau ATmega328V dan Arduino Lilypad Main Board berbeda dengan Arduino Lilypad Simple karena memiliki lebih banyak pin input dan tidak memiliki konektor JST dan sirkuit pengisian bawan untuk batrei Lithium Polymer.



Gambar 2.11 Arduino Lilypad Main Board

Spesifikasi	
<b>Microcontroller</b>	ATmega168 atau ATmega328V
<b>Operating Voltage</b>	2,7-5,5V
<b>Input Voltage</b>	2,7-5,5V
<b>Digital I/O Pin</b>	14
<b>PWM Digital I/O Pin</b>	6
<b>Analog Input Pin</b>	6
<b>Arus DC per Pin I/O</b>	40 mA
<b>Flash Memory</b>	16 KB (dimana 2 KB digunakan oleh bootloader)
<b>SRAM</b>	1 KB
<b>EEPROM</b>	512 bytes
<b>Clock Speed</b>	8 MHz

## 12. Arduino Mini

Arduino Mini adalah versi yang sangat ringkas dari Arduino Nano tanpa koneksi USB-to-Serial, Arduino Mini 05 adalah board mikrokontroler kecil yang awalnya menggunakan ATmega168, tetapi sekarang dilengkapi dengan 328. Ini memiliki 14 pin input / output digital (6 di antaranya dapat digunakan

sebagai output PWM), 8 input analog, dan osilator kristal 16 MHz. Ini dapat diprogram dengan adaptor Serial USB atau adaptor serial USB atau RS232 ke TTL lainnya.



Gambar 2.12 Arduino Mini

#### Spesifikasi

<b>Microcontroller</b>	ATmega328P
<b>Operating Voltage</b>	5V
<b>Input Voltage</b>	7-9V
<b>Digital I/O Pin</b>	14 (6 pin input dapat digunakan sebagai output PWM)
<b>Analog Input Pin</b>	8
<b>Arus DC per Pin I/O</b>	40 mA
<b>Flash Memory</b>	32 KB (dimana 2 KB digunakan oleh bootloader)
<b>SRAM</b>	2 KB
<b>EEPROM</b>	1 KB
<b>Clock Speed</b>	16 MHz
<b>Panjang</b>	30 mm
<b>Lebar</b>	18 mm

### 13. Arduino Robot

Arduino Robot merupakan Arduino yang memiliki roda atau bisa dibilang arduino pertama yang berada di atas roda, banyak digunakan untuk pembuatan robot, karna sudah dilengkapi dengan LCD, roda, speaker, Sensor Infrared, Sd Card Reader, Digital Compas, keypad,dll.

Arduino Robot memiliki 2 prosesor,yang berada pada setiap board. board Motor mengendalikan motor, sedangkan board Kontrol membaca



sensor dan memutuskan cara beroperasi. Board Arduino yang dapat diprogram menggunakan Arduino IDE. Kedua board Motor dan Kontrol adalah board mikrokontroler yang menggunakan ATmega32u4. Arduino Robot ini memiliki banyak pin yang dipetakan ke sensor dan aktuator. Kedua prosesor memiliki komunikasi USB bawaan. Ini memungkinkan Robot untuk muncul ke komputer yang terhubung sebagai port serial / COM virtual (CDC). Robot Arduino adalah hasil dari upaya kolektif dari tim internasional yang melihat bagaimana sains dapat dibuat menyenangkan untuk dipelajari.



Gambar 2.13 Arduino Robot

#### Spesifikasi

##### - Control Board

<b>Microcontroller</b>	ATmega32u4
<b>Operating Voltage</b>	5V
<b>Input Voltage</b>	5V
<b>Digital I/O Pin</b>	5
<b>PWM Channels</b>	6
<b>Analog Input Channels</b>	4 (dari pin I/O Digital)
<b>Analog Input Channels (Multiplexed)</b>	8
<b>Arus DC per Pin I/O</b>	40 mA
<b>Flash Memory</b>	32 KB (ATmega32u4) dimana 4 KB digunakan oleh bootloader
<b>SRAM</b>	2,5 KB (ATmega32u4)
<b>EEPROM (internal)</b>	1 KB (ATmega32u4)
<b>EEPROM (external)</b>	512 Kbit (12C)
<b>Clock Speed</b>	16 MHz
<b>Keypad</b>	5 keys
<b>Knob</b>	potensiometer yang terpasang pada pin analog
<b>Full Color LCD</b>	melalui komunikasi SPI

<b>SD card reader</b>	untuk kartu FAT16 yang diformat
<b>Speaker</b>	8 Ohm
<b>Digital Compass</b>	memberikan penyimpangan dari utara geografis dalam derajat Radius
<b>I2C soldering ports</b>	3
<b>Prototyping areas</b>	4
<b>Radius</b>	185 mm
<b>Tinggi</b>	85 mm

- Motor Board

<b>Microcontroller</b>	ATmega32u4
<b>Operating Voltage</b>	5V
<b>Input Voltage</b>	9V untuk pengisian batrai
<b>AA baterai slot</b>	4 alkaline atau Baterai isi ulang NiMh
<b>Digital I/O Pin</b>	4
<b>PWM Channels</b>	1
<b>Analog Input Channels</b>	4 (dari pin I/O Digital)
<b>Arus DC per Pin I/O</b>	40 mA
<b>DC-DC converter</b>	menghasilkan 5V untuk menyalakan seluruh robot
<b>Flash Memory</b>	32 KB (ATmega32u4) dimana 4 KB digunakan oleh bootloader
<b>SRAM</b>	2,5 KB (ATmega32u4)
<b>EEPROM</b>	1 KB (ATmega32u4)
<b>Clock Speed</b>	16 MHz
<b>Trimmer</b>	Untuk kalibrasi kecepatan
<b>IR line following sensor</b>	5
<b>I2C soldering ports</b>	1
<b>Prototyping areas</b>	2

### 2.3. Komponen Arduino

Arduino bukan hanya sebagai suatu alat yang terbentuk sebagai microcontroller saja, namun memiliki berbagai komponen yang dapat digunakan pada boards microcontroller arduino :

1. USB type A to B

USB type A to B Digunakan untuk menghubungkan Board Arduino yang memiliki Port USB B *Female* dengan computer kita, untuk memberi daya maupun dalam mengupload sketch ke Arduino.



Gambar 2.14 USB Type A to B

## 2. *Breadboard*

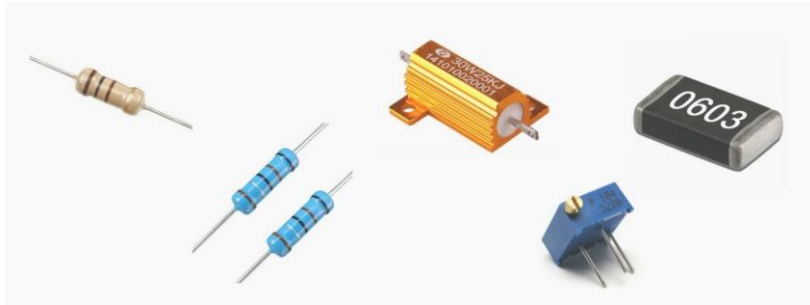
*Breadboard* adalah board yang digunakan untuk membuat rangkaian elektronika sementara tanpa harus menyolder sehingga komponen yang digunakan tidak rusak dan dapat digunakan lagi dalam membuat rangkain yang lain.



Gambar 2.15 Breadboard

## 3. Resistor

Resistor merupakan komponen elctronika yang selalu digunakan dalam setiap rangkaian elektronika pada Arduino, yang berfungsi sebagai pengatur arus yang mengalir dalam suatu rangkaian.



Gambar 2.16 Resistor

4. Led

LED (Light-Emitting Diode) merupakan komponen elektronika yang dapat memancarkan cahaya apabila dialirkan arus listrik. LED biasa digunakan sebagai lampu indicator.



Gambar 2.17 Led

5. Push Button

Push Button merupakan komponen yang digunakan sebagai sakelar sederhana untuk mengontrol beberapa aspek mesin maupun proses.



Gambar 2.18 Push Button

6. Kabel Jumper

Kabel Jumper digunakan untuk menghubungkan Arduino dengan sensor atau *breadboard* dalam membangun sebuah proyek. Kabel Jumper memiliki tiga jenis :

- Male to Male



Gambar 2.19 Male to Male

- Male to Female



Gambar 2.20 Male to Female

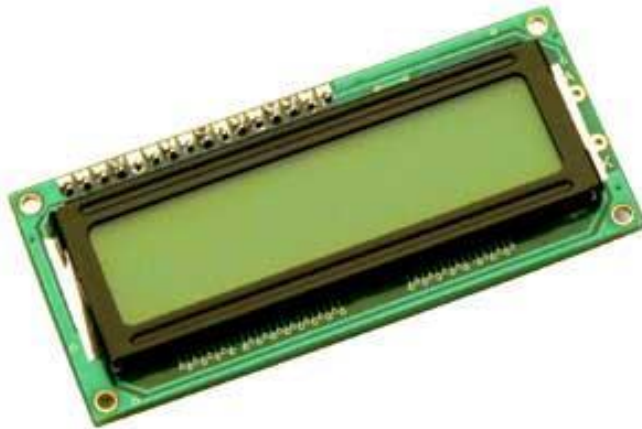
- Female to Female



Gambar 2.21 Female to Female

## 7. LCD Display

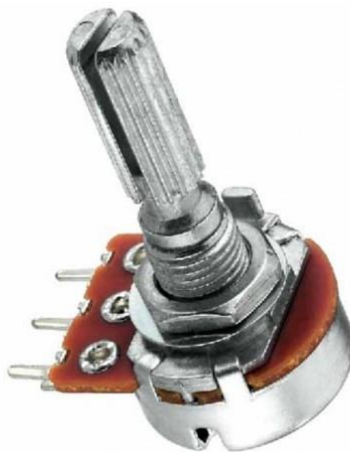
LCD Display merupakan media tampilan yang menggunakan kristal cair untuk menghasilkan gambar yang terlihat, biasa dalam Arduino digunakan untuk memampikan karakter dengan menggunakan library LiquidCrystal.h.



Gambar 2.22 LCD Display

8. Potensiometer

Potensiometer adalah resistor tiga terminal dengan sambungan geser yang membentuk pembagi tegangan dapat kita setel sesuai keinginan kita. Pada Arduino sering digunakan untuk mendapatkan nilai potensiometer pada posisi tertentu, yang kemudian bisa kita manipulasi sesuai kebutuhan kita.

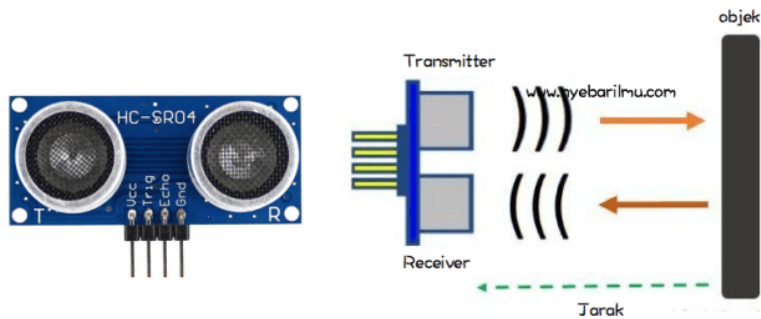


Gambar 2.23 Potensimeter

9. Sensor Ultrasonik hc-sr04

Sensor Ultrasonik merupakan sebuah sensor yang memanfaatkan pantulan gelombang suara dengan frekuensi 40 KHz hingga 400 KHz sehingga dapat digunakan untuk menafsirkan jarak suatu objek di depannya. Sensor Ultrasonik bekerja menggunakan dua unit, yaitu pemancar dan penerima, dimana pemancar akan mengeluarkan

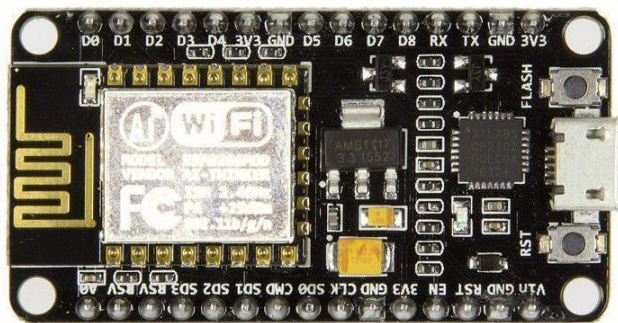
gelombang ke permukaan objek, kemudian objek akan memantulkan dan akan diterima oleh unit penerima kemudian sensor akan menghitung selisih antara waktu pengiriman dan penerimaan gelombang.



Gambar 2.24 Sensor Ultrasonic HC-SR04

#### 10. NodeMCU

Node MCU adalah Firmware yang bersifat opensource yang menggunakan Sytem On Chip ESP8266 dari ESP8266 buatan Expressif system, yang banyak digunakan untuk membuat proyek/produk IoT. Board Node MCU memiliki Beberapa pin GPIO yang dapat kita gunakan untuk menghubungkan perangkat lain dan mampu menghasilkan komunikasi serial PWM, I2C, SPI, dan UART.

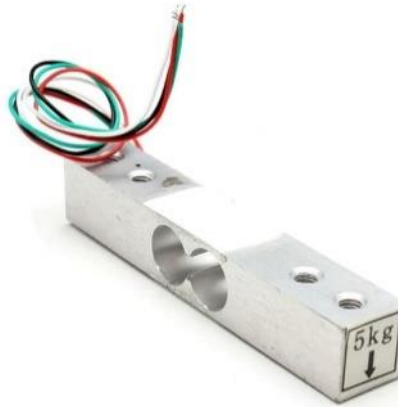


Gambar 2.25 NodeMCU



### 11. Load Cell

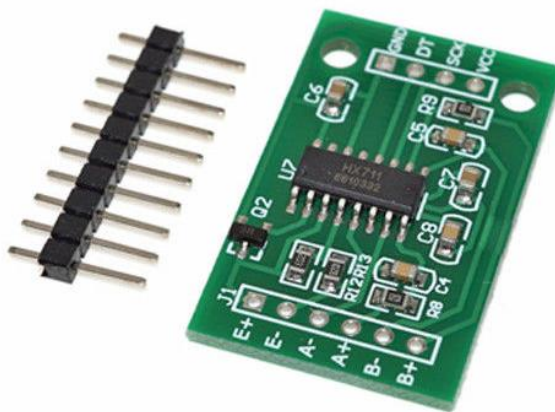
Loadcell adalah sensor berat yang menghasilkan sinyal listrik berdasarkan besarnya sinyal yang sebanding dengan berat yang diukur.



Gambar 2.26 Load Cell

### 12. HX711

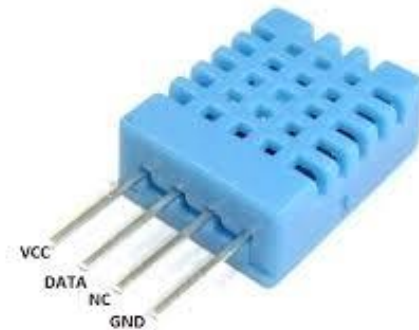
HX711 adalah module yang menerima resistensi listrik dari loadcell kemudian sinyal diperkuat sehingga dapat dibaca oleh Arduino.



Gambar 2.27 HX711

### 13. Sensor DHT11

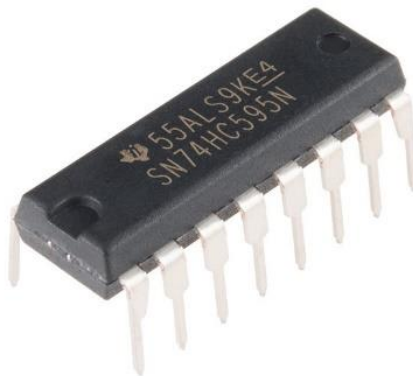
Sensor DHT11 adalah module sensor yang digunakan untuk mengetahui sensing suhu objek dan kelembaban. Dalam pemrograman Arduino Sensor DHT11 menggunakan library DHT.h agar dapat digunakan pada microcontroller Arduino.



Gambar 2.28 DHT11

#### 14. IC Shift Register

Shift Register adalah susunan flip-flop yang terbagi dalam waktu yang bersamaan dan output yang dihasilkan dari masing-masing flip-flop terhubung ke inputan data dari flip-flop selanjutnya. Shift register biasanya digunakan untuk memori sementara dan untuk pergeseran data ke kiri atau ke kanan.



Gambar 2.29 IC Shift Register

#### 15. Buzzer

Buzzer adalah suatu komponen elektronika yang dapat mengubah sinyal listrik menjadi getaran suara. Dalam project Arduino biasa digunakan untuk membuat alarm, bel atau alat peringatan, dll. Buzzer terdiri dari 2 jenis antara lain :

- Passive Buzzer adalah jenis buzzer yang tidak memiliki suara sendiri. Sehingga harus diprogram tinggi rendah nadanya.
- Active Buzzer adalah jenis buzzer yang memiliki suara sendiri atau dapat berdiri sendiri tanpa harus deprogram lagi.

Passive Buzzer



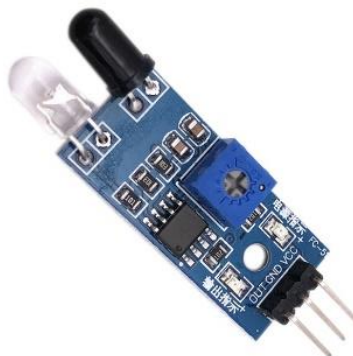
Active Buzzer



Gambar 2.30 Buzzer

#### 16. Sensor Infra Merah

Sensor Infra Merah adalah sebuah sensor yang dapat mendeteksi menggunakan cahaya inframerah yang dipantulkan ke sebuah objek. Dalam sensor inframerah terdapat dua komponen utama yaitu IR emitter yang berfungsi untuk memantulkan cahaya inframerah kemudian, IR receiver yang merupakan komponen yang menerima hasil pantulan cahaya inframerah.



Gambar 2.30 Sensor Infra Merah

## 17. Micro Servo

Micro servo adalah komponen elektronika aktuator putar atau motor yang dirancang dengan sistem umpan balik loop tertutup atau servo. Dalam Arduino apabila ingin menggunakan Micro Servo kita harus menginputkan library *Servo.h*.



Gambar 2.31 Micro Servo

Komponen diatas merupakan hanya sebagian kecil dari komponen-komponen yang dapat digunakan, karna masih banyak lagi komponen lainnya lainnya.

## 2.4. Arduino IDE

### 2.4.1. Definisi

Merupakan sebuah software untuk memprogram arduino. Pada software inilah arduino dilakukan pemrograman untuk melakukan fungsi-fungsi yang dinamakan melalui sintaks pemrograman. Arduino menggunakan bahasa pemrograman C yang dimodifikasi. Kita sebut saja dengan bahasa pemrograman C for Arduino.

Bahasa pemrograman arduino sudah dirubah untuk memudahkan pemula dalam melakukan pemrograman dari bahasa aslinya. Didalam arduino sendiri sudah terdapat IC mikrokontroler yang sudah ditanam program yang bernama

Bootloader. Fungsi dari bootloader tersebut adalah untuk menjadi penengah antara compiler arduino dan mikrokontroler.

Arduino IDE dibuat dari bahasa pemrograman JAVA yang dilengkapi dengan library C/C++ (wiring), yang membuat operasi input/output lebih mudah.

#### **2.4.2. Istilah pada Arduino IDE**

##### **1. Uploading**

Uploading adalah proses untuk menyalin file **.hex** atau file hasil kompilasi kedalam IC mikrokontroler Arduino. Tombol uploading terdapat di bagian atas pada icon yang berbentuk anak panah menghadap ke kanan. Sebelum melakukan uploading, yang perlu diperhatikan adalah jenis board dan COM port yang digunakan. Untuk mengetahui keduanya bisa melalui menu **Tools>Board** dan **Tools>Port**.

##### **2. Library**

Library atau pustaka adalah file yang memberikan fungsi ekstra dari sketch yang dibuat. Didalam arduino sendiri sudah include beberapa library yang berfungsi untuk melakukan proses tertentu. Selain itu, pengguna juga bisa menambahkan library eksternal untuk memperkaya library yang ada di dalam Arduino IDE. Untuk menambahkan library eksternal cukup mudah. Bisa melalui fitur Library Manager, Import file **.zip**, atau menyalin secara manual di folder libraries pada directory arduino.

##### **3. Serial Monitor**

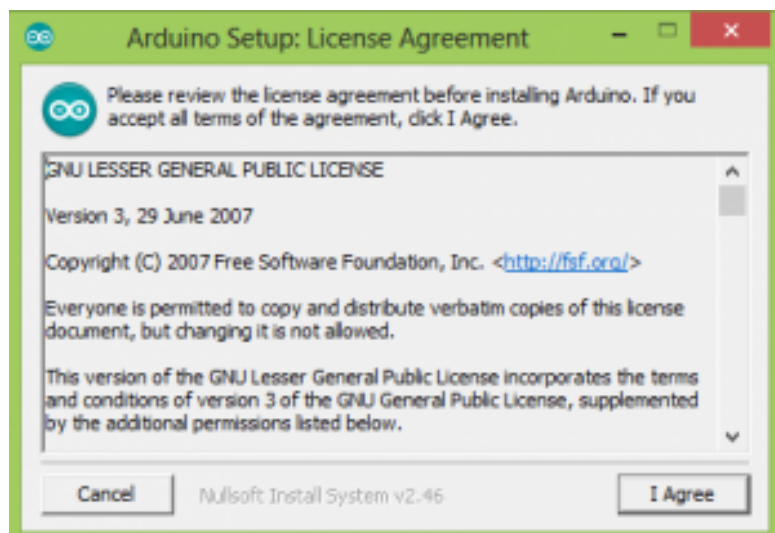
Serial Monitor merupakan suatu kotak dialog yang menunjukkan proses pertukaran data antara arduino dan komputer selama beroperasi. Sehingga serial monitor bisa digunakan untuk menampilkan hasil operasi atau pesan error debugging. Serial monitor juga digunakan untuk mengirim data dari

komputer ke arduini dengan cara menuliskan pesan pada text box dan menekan tombol Send.

### 2.4.3. Cara Menginstall Arduino IDE

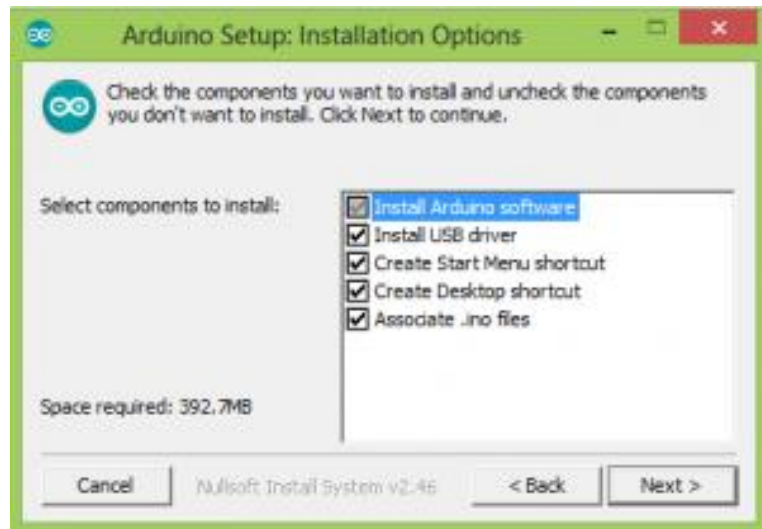
Berikut adalah bagaimana cara menginstall Arduino pada windows :

1. Mendownload file software Arduino ide pada web <https://www.arduino.cc/en/Main/Software>
2. Buka file installer yang sudah di download
3. Klik “*I Agree*” untuk menyetujui lisensi dan melanjutkan installasi.



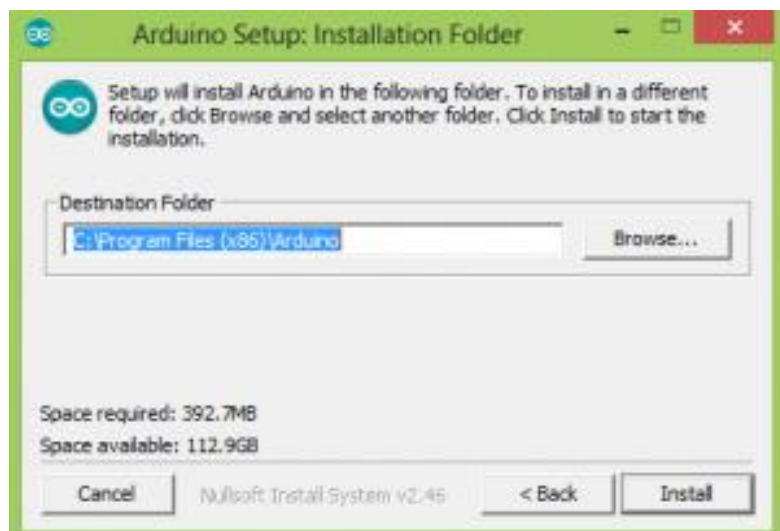
Sdada

4. Install semua komponen yang ada.



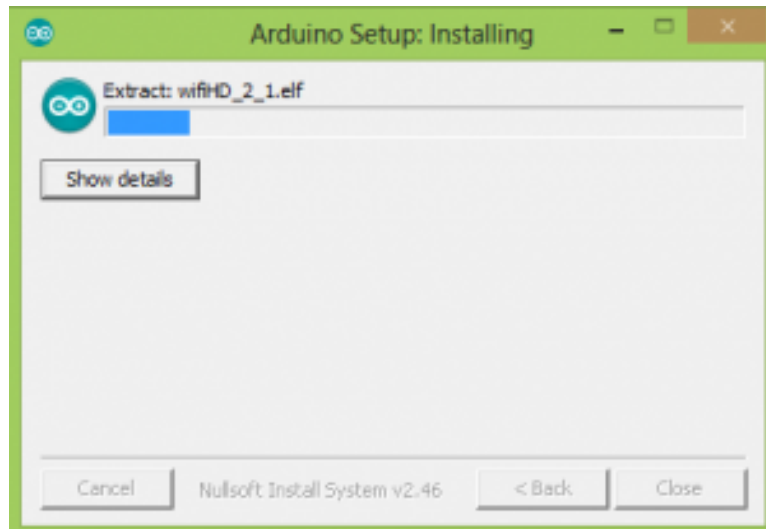
Sdasdasda

5. Pilih lokasi folder installasi Arduino Software atau gunakan *default destination folder*, dan klik *Install*.



Asdasda

6. Tunggu hingga proses installasi selesai.



Asdasda

7. Kemudian pada proses instalasi, akan muncul dialog window untuk menginstall Arduino USB Driver, check *“Always trust software from Arduini srl”* dan klik *Install*.



Asdasda

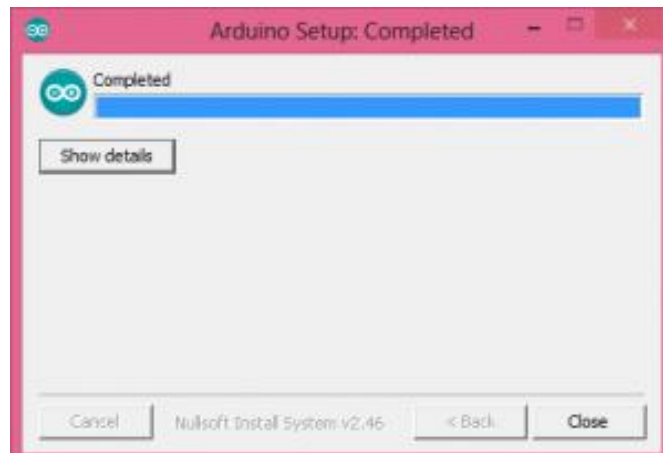
8. Kemudian pada proses instalasi, akan muncul dialog window ke-dua untuk menginstall Arduino USB Driver, check *“Always trust software from Arduini LLC”* dan klik *Install*.





Sadasda

9. Setelah semua proses di atas kamu ikuti, maka Software Arduino (IDE) telah berhasil terinstall. Klik *Close* untuk menutup dialog window.

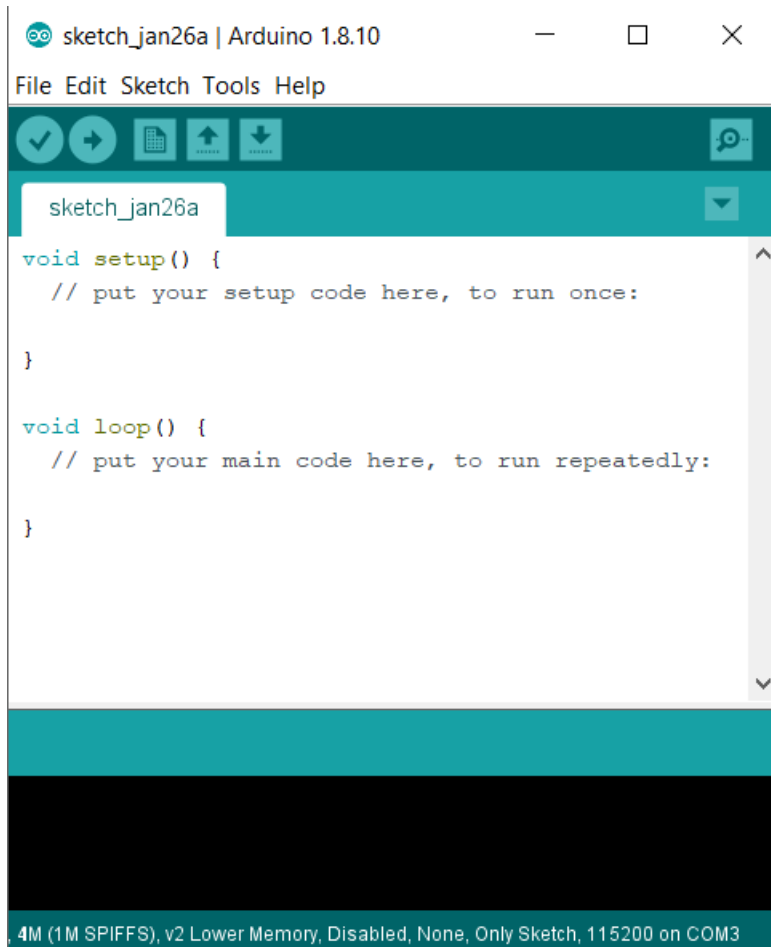


Asdasd

#### 2.4.4. Cara Memprogram Arduino

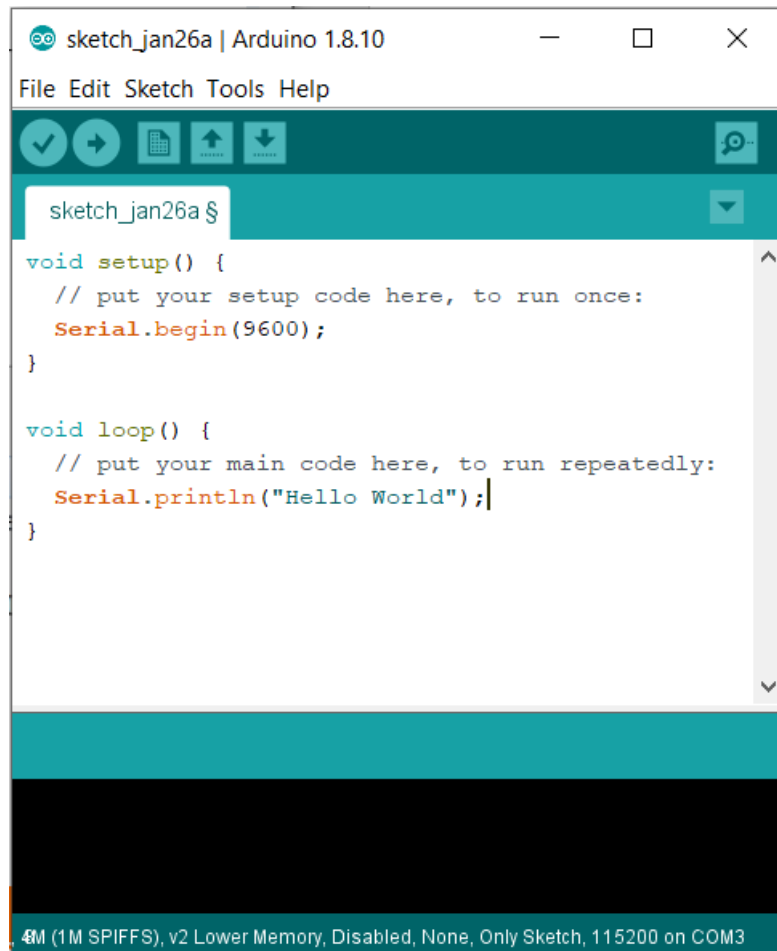
Berikut ini adalah cara memprogram Arduino :

1. Membuka Aplikasi Arduino IDE



## 2. Membuat code program

Pada buku ini saya hanya mencontohkan dengan code program sederhana yaitu Hello world, codenya sebagai berikut:

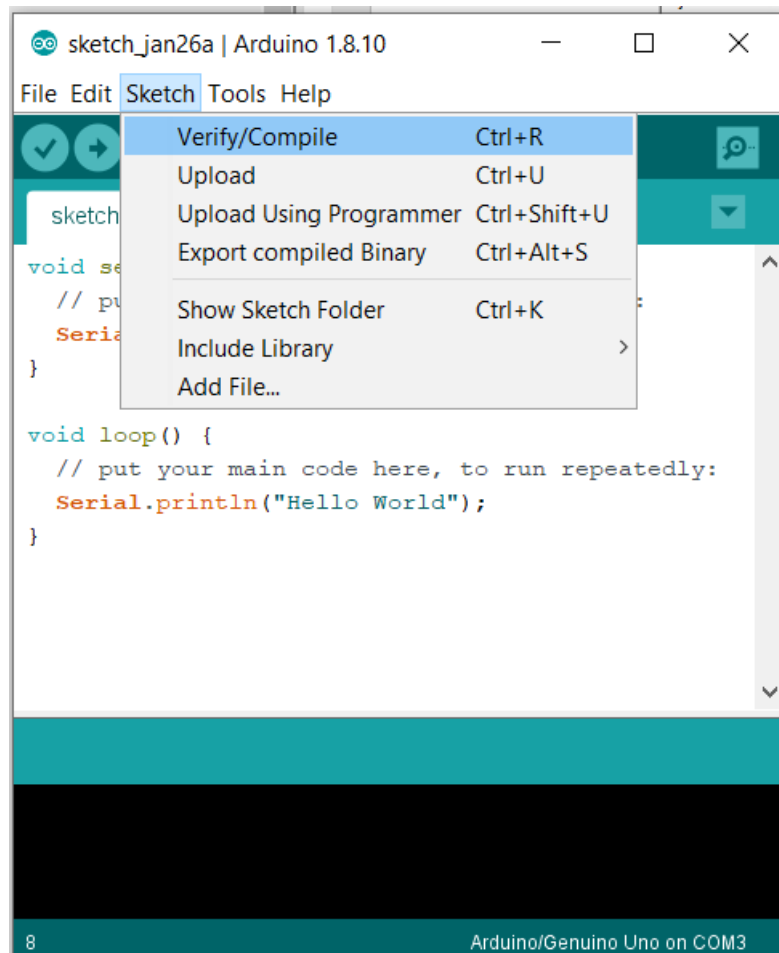


asdasda

### 3. Compile Program Arduino

Setelah kita mengunduh Arduino IDE dan memasukkan code program kita perlu melakukan compile sebelum upload program pada Arduino, langkahnya sebagai berikut:

- a. Klik Sketch
- b. Lalu klik Verify/Compile
- c. Kemudian tunggu sampai proses selesai

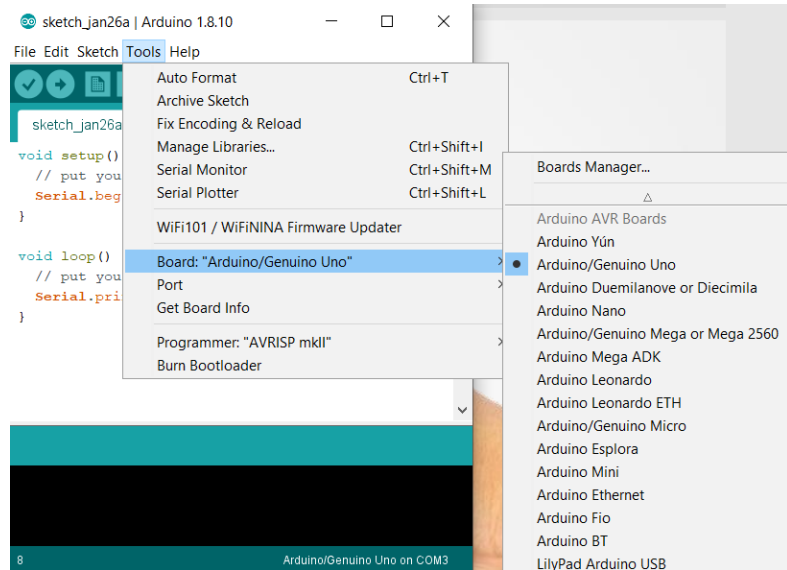


asdasd

#### 4. Upload Program Arduino

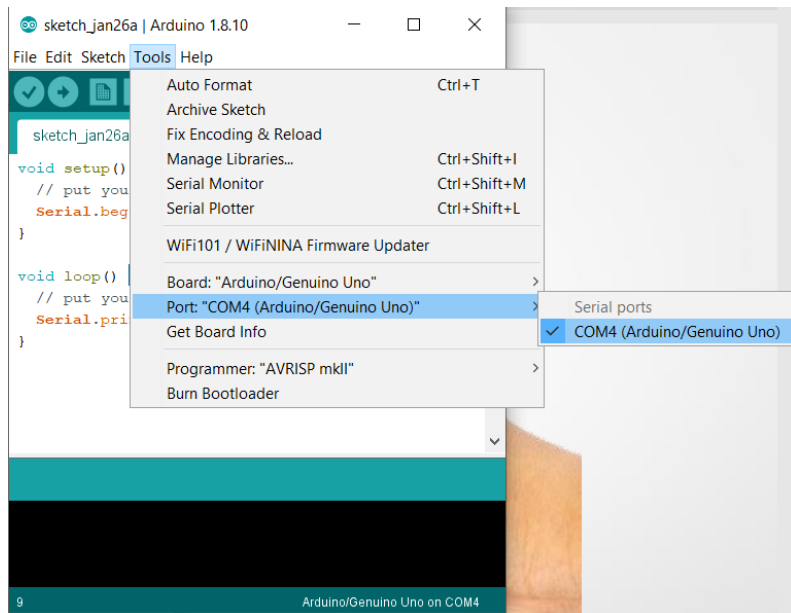
Setelah kita meng-compile programnya tidak ada terjadi error, maka selanjutnya kita akan mengupload code program terse but kedalam board Arduino. Langkahnya sebagai berikut :

- a. Sambungkan Arduino melalui port USB pada laptop/computer
- b. Setelah disambungkan, pilih board sesuai yang digunakan



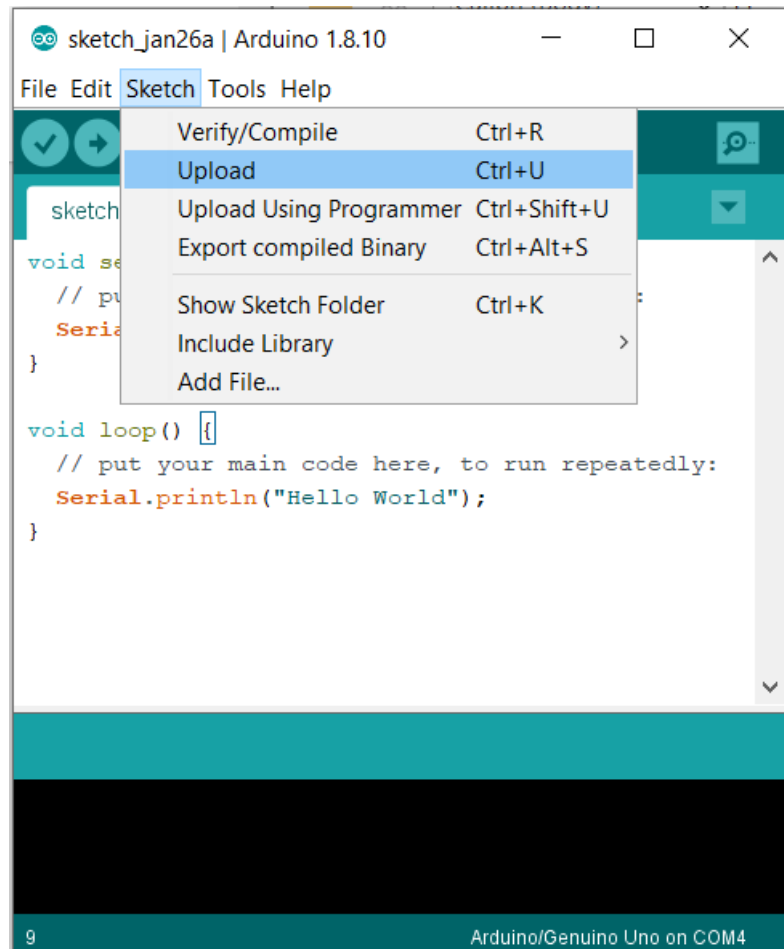
dasdads

c. Lalu cek port yang terdeteksi



asdadsas

- Klik Sketch
- Pilih Upload
- Tunggu hingga proses selesai.



sadasdsa

## **BAGIAN III**

### **ANDROID**

#### **3.1. Sejarah**

Android, Inc. didirikan di Palo Alto, California, pada bulan Oktober 2003 oleh Andy Rubin (pendiri Danger), Rich Miner (pendiri Wildfire Communications, Inc.), Nick Sears (mantan VP T-Mobile), dan Chris White (kepala desain dan pengembangan antarmuka WebTV) untuk mengembangkan "perangkat seluler pintar yang lebih sadar akan lokasi dan preferensi penggunaannya". Tujuan awal pengembangan Android adalah untuk mengembangkan sebuah sistem operasi canggih yang diperuntukkan bagi kamera digital. Namun, disadari bahwa pasar untuk perangkat tersebut tidak cukup besar, dan pengembangan Android lalu dialihkan bagi pasar telepon pintar untuk menyaingi Symbian dan Windows Mobile (iPhone Apple belum dirilis pada saat itu). Meskipun para pengembang Android adalah pakar-pakar teknologi yang berpengalaman, Android Inc. dioperasikan secara diam-diam, hanya diungkapkan bahwa para pengembang sedang menciptakan sebuah perangkat lunak yang diperuntukkan bagi telepon seluler. Masih pada tahun yang sama, Rubin kehabisan uang. Steve Perlman, seorang teman dekat Rubin meminjamkan \$10.000 tunai dan menolak tawaran saham di perusahaan.

Google mengakuisisi Android Inc. pada tanggal 17 Agustus 2005, menjadikannya sebagai anak perusahaan yang sepenuhnya dimiliki oleh Google. Pendiri Android Inc. seperti Rubin, Miner dan White tetap bekerja di perusahaan setelah diakuisisi oleh Google. Setelah itu, tidak banyak yang diketahui tentang perkembangan Android Inc., tetapi banyak anggapan yang menyatakan bahwa Google berencana untuk memasuki pasar telepon seluler dengan tindakannya ini. Di Google, tim yang dipimpin oleh Rubin mulai mengembangkan platform perangkat seluler dengan menggunakan kernel Linux. Google memasarkan platform tersebut kepada produsen perangkat seluler dan operator nirkabel, dengan janji bahwa mereka menyediakan sistem yang fleksibel dan bisa diperbarui. Google telah memilih beberapa mitra perusahaan perangkat lunak dan perangkat keras, serta mengisyaratkan kepada operator seluler bahwa kerja sama ini terbuka bagi siapapun yang ingin berpartisipasi.

Spekulasi tentang niat Google untuk memasuki pasar komunikasi seluler terus berkembang hingga bulan Desember 2006. BBC dan *Wall Street Journal* melaporkan bahwa Google sedang bekerja keras untuk menyertakan aplikasi dan mesin pencariannya di perangkat seluler. Berbagai media cetak dan media daring mengabarkan bahwa Google sedang mengembangkan perangkat seluler dengan merek Google. Beberapa di antaranya berspekulasi bahwa Google telah menentukan spesifikasi teknisnya, termasuk produsen telepon seluler dan operator jaringan. Pada bulan Desember 2007, *InformationWeek* melaporkan

bahwa Google telah mengajukan beberapa aplikasi paten di bidang telepon seluler.

Pada tanggal 5 November 2007, Open Handset Alliance (OHA) didirikan. OHA adalah konsorsium dari perusahaan-perusahaan teknologi seperti Google, produsen perangkat seluler seperti HTC, Sony dan Samsung, operator nirkabel seperti Sprint Nextel dan T-Mobile, serta produsen chipset seperti Qualcomm dan Texas Instruments. OHA sendiri bertujuan untuk mengembangkan standar terbuka bagi perangkat seluler. Saat itu, Android diresmikan sebagai produk pertamanya; sebuah platform perangkat seluler yang menggunakan kernel Linux versi 2.6. Telepon seluler komersial pertama yang menggunakan sistem operasi Android adalah HTC Dream, yang diluncurkan pada 22 Oktober 2008.

Pada tahun 2010, Google merilis seri Nexus; perangkat telepon pintar dan tablet dengan sistem operasi Android yang diproduksi oleh mitra produsen telepon seluler seperti HTC, LG, dan Samsung. HTC bekerja sama dengan Google dalam merilis produk telepon pintar Nexus pertama, yakni Nexus One. Seri ini telah diperbarui dengan perangkat yang lebih baru, misalnya telepon pintar Nexus 4 dan tablet Nexus 10 yang diproduksi oleh LG dan Samsung. Pada 15 Oktober 2014, Google mengumumkan Nexus 6 dan Nexus 9 yang diproduksi oleh Motorola dan HTC. Pada 13 Maret 2013, Larry Page mengumumkan dalam postingan blognya bahwa Andy Rubin telah pindah dari divisi Android untuk mengerjakan proyek-proyek baru di Google. Ia digantikan oleh Sundar Pichai, yang sebelumnya menjabat sebagai kepala divisi Google Chrome, yang mengembangkan Chrome OS.

Sejak tahun 2008, Android secara bertahap telah melakukan sejumlah pembaruan untuk meningkatkan kinerja sistem operasi, menambahkan fitur baru, dan memperbaiki bug yang terdapat pada versi sebelumnya. Setiap versi utama yang dirilis dinamakan secara alfabetis berdasarkan nama-nama makanan pencuci mulut atau camilan bergula; misalnya, versi 1.5 bernama *Cupcake*, yang kemudian diikuti oleh versi 1.6 *Donut*. Versi terbaru adalah 9.0 *Pie*, yang dirilis pada 7 maret 2018.

### **3.2. Android**

Android adalah software untuk perangkat mobile yang mencakup sistem operasi, middleware dan aplikasi kunci. Pengembangan aplikasi pada platform Android menggunakan bahasa pemrograman Java. Serangkaian aplikasi inti Android antara lain klien email, program SMS, kalender, peta, browser, kontak, dan lain-lain.

Dengan menyediakan sebuah platform pengembangan yang terbuka, pengembang Android menawarkan kemampuan untuk membangun aplikasi yang sangat kaya dan inovatif. Pengembang bebas untuk mengambil keuntungan



dari perangkat keras, akses informasi lokasi, menjalankan background services, mengatur alarm, tambahkan pemberitahuan ke status bar, dan banyak lagi.



Gambar 3.1 Logo Android

Android bergantung pada versi Linux 2.6 untuk layanan sistem inti seperti keamanan, manajemen memori, manajemen proses, network stack, dan model driver. Kernel juga bertindak sebagai lapisan abstraksi antara hardware dan seluruh software stack.

### 3.3. Versi Android

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

#### 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.



Gambar 3.2 Android Astro

#### 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.



Gambar 2.3 Android Bender

### 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.



Gambar 3.4 Android Cupcake

### 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.



Gambar 3.5 Android Donut

### 5. **Éclair 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.



Gambar 3.6 Android Eclair

### 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.



Gambar 3.7 Android Froyo

### 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.



Gambar 3.8 Android GingerBread

#### 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.



Gambar 3.9 Android Honeycomb

#### 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 brau, seperti membuka kunci dengan pengenala wajah, perangkat tambahan fotografi, hingga berbagi informasi menggunakan NFC.



Gambar 3.10 Android Ice Cream Sandwich

#### 10. Jelly Bean 4.1 – 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.



Gambar 3.11 Android Jelly Bean

#### 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.



Gambar 3.11 Android Kitkat

## 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*”.



Gambar 3.13 Android Lollipop

## 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.



Gambar 3.14 Android Marshmallow

## 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.



Gambar 3.15 Nougat

### 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.



Gambar 3.16 Android Oreo

### 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.



Gambar 3.17 Android Pie

### 3.4. Android Studio

Android Studio adalah Integrated Development Enviroment (IDE) untuk sistem operasi Android, yang dibangun diatas perangkat lunak JetBrains IntelliJ IDEA dan didesain khusus untuk pengembangan Android. IDE ini merupakan pengganti dari Eclipse Android Development Tools (ADT) yang sebelumnya merupakan IDE utama untuk pengembangan aplikasi android.

Android studio sendiri pertama kali diumumkan di Google I/O conference pada tanggal 16 Mei 2013. Ini merupakan tahap preview dari versi 0.1 pada Mei 2013, dan memasuki tahap beta sejak versi 0.8 dan mulai diliris pada Juni 2014.



Gambar 3.18 Android Studio

#### 3.4.1. Sejarah Android Studio

Pada awalnya mulanya, Android Inc merupakan sebuah perusahaan software kecil yang didirikan pada bulan Oktober 2003 di Palo Alto, California, USA. Didirikan oleh beberapa senior di beberapa perusahaan yang berbasis IT & Communication, Andy Rubin, Rich Miner, Nick Sears dan Chris White.

Menurut Rubin, Android Inc Didirikan untuk mewujudkan mobile device yang lebih peka terhadap lokasi dan preferensi pemilik. Dengan kata lain, Android Inc ingin mewujudkan mobile device yang lebih mengerti pemiliknya. Konsep yang dimiliki Android Inc ternyata menggugah minat Google untuk memilikinya. Pada bulan Agustus 2005, Akhirnya Android Inc diakuisisi oleh Google Inc. seluruh sahamnya dibeli oleh Google. Banyak yang memperkirakan nilai pembelian Android Inc Oleh Google adalah sebesar USD 50 juta. saat itu banyak yang berspekulasi bahwa akuisisi ini adalah langkah awal yang dilakukan Google untuk masuk kepasar mobile phone.

Andy Rubin, Rich Miner, Nick Sears dan Chris White tetap di Android Inc yang dibeli Google, sehingga akhirnya mereka semua menjadi bagian dari raksasa Google dan sejarah Android. Saat itulah mereka mulai menggunakan platform Linux untuk membuat sistem operasi bagi mobile phone.



Maka dari situlah banyak pengembang sistem maupun software berlomba lomba untuk membuat atau merancang sistem Android menggunakan software – software yang support dengan Android, dan sebagai contoh disini kita pengenalan **Android Studio**.

Di atas alat editor dan pengembang hebat IntelliJ, Android Studio menawarkan lebih banyak fitur yang meningkatkan produktivitas Anda saat membangun aplikasi Android, seperti:

- *Sistem build berbasis Gradle yang fleksibel*
- *Emulator cepat dan kaya fitur*
- *Lingkungan terpadu yang dapat Anda kembangkan untuk semua perangkat Android*
- *Instan Berjalan untuk mendorong perubahan pada aplikasi yang sedang berjalan tanpa membangun APK baru*
- *Template kode dan integrasi GitHub untuk membantu Anda membuat fitur aplikasi umum dan kode contoh impor*
- *Alat pengujian ekstensif dan kerangka kerja*
- *Alat lint untuk menangkap kinerja, kegunaan, kompatibilitas versi, dan masalah lainnya*
- *Dukungan C ++ dan NDK*
- *Dukungan terintegrasi untuk Google Cloud Platform, sehingga mudah mengintegrasikan Google Cloud Messaging dan App Engine*

### 3.4.2. Struktuk Proyek Android Studio

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

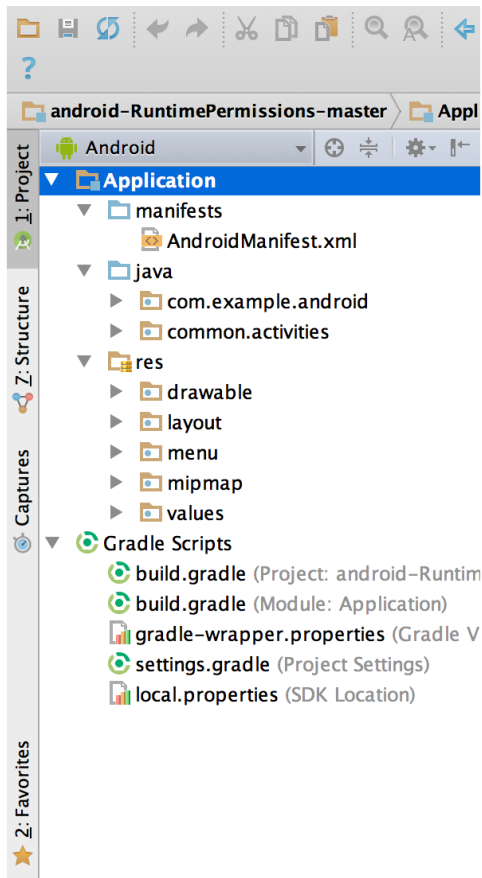
- Modul aplikasi Android
- Modul library
- Modul Google App Engine

Secara default, Android Studio menampilkan file project Anda dalam tampilan project Android, seperti yang ditunjukkan pada gambar 1. Tampilan ini disusun menurut modul untuk memberikan akses cepat ke file sumber utama project Anda.

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

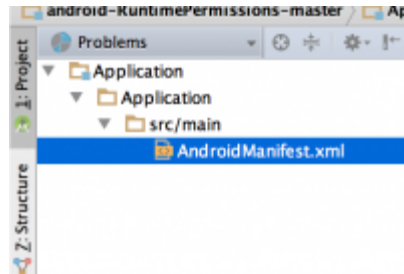
- **manifes:** Berisi file `AndroidManifest.xml`.
- **java:** Berisi file kode sumber Java, termasuk kode pengujian JUnit.
- **res:** Berisi semua resource non-kode, seperti tata letak XML, string UI, dan gambar bitmap.

Struktur project Android pada disk berbeda dengan representasi tersatukan ini. Untuk melihat struktur file project sebenarnya, pilih **Project** dari menu drop-down **Project** (pada gambar 3.19, ditampilkan sebagai **Android**).



Gambar 3.19 Project

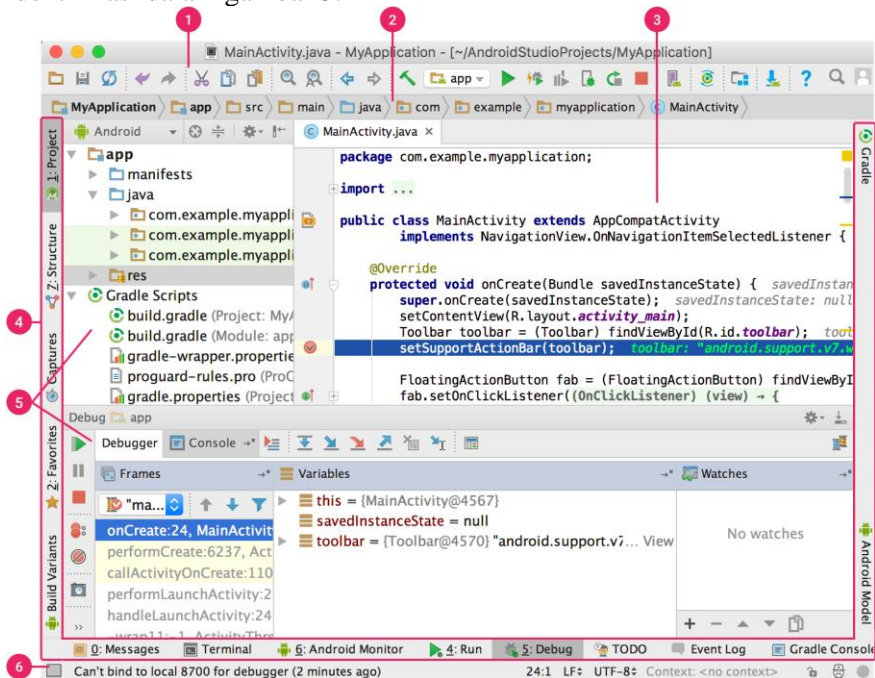
Anda juga dapat menyesuaikan tampilan file project untuk berfokus pada aspek spesifik dari pengembangan aplikasi Anda. Misalnya, memilih tampilan **Problems** pada project Anda akan menampilkan link ke file sumber yang berisi error coding dan sintaks yang dikenali, seperti tag penutup elemen XML yang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



Gambar 3.20 File.xml

### 3.4.3. Antarmuka Pengguna

Jendela utama Android Studio terdiri dari beberapa area logis yang diidentifikasi dalam gambar 3.




Gambar 3.20 Antarmuka Pengguna

1. **Toolbar** memungkinkan Anda melakukan berbagai tindakan, termasuk menjalankan aplikasi dan meluncurkan fitur Android.
2. **Menu navigasi** membantu Anda menjelajah project dan membuka file untuk diedit. Menu ini memberikan tampilan struktur yang lebih ringkas yang terlihat di jendela **Project**.
3. **Jendela editor** adalah tempat Anda membuat dan memodifikasi kode. Tergantung jenis file yang ada, editor ini dapat berubah. Misalnya, saat menampilkan file tata letak, editor akan menampilkan Layout Editor.

4. **Panel jendela fitur** berada di sisi luar jendela IDE dan berisi tombol-tombol yang memungkinkan Anda memperluas atau menciutkan setiap jendela fitur.
5. **Jendela fitur** memberi Anda akses ke tugas tertentu seperti pengelolaan project, penelusuran, kontrol versi, dan banyak lagi. Anda dapat memperluas dan menciutkan jendela ini.
6. **Status bar** menampilkan status project Anda dan IDE itu sendiri, serta semua peringatan atau pesan.

#### 3.4.4. Jendela Fitur

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

- Untuk memperluas atau menciutkan jendela fitur, klik nama fitur di panel jendela fitur. Anda juga dapat menarik, menyematkan, melepaskan sematan, memasang, dan melepas jendela fitur.
- Untuk kembali ke tata letak jendela fitur default saat ini, klik **Window > Restore Default Layout** atau sesuaikan tata letak default dengan mengklik **Window > Store Current Layout as Default**.
- Untuk menampilkan atau menyembunyikan seluruh panel jendela alat, klik ikon jendela  di pojok kiri bawah jendela Android Studio.
- Untuk menemukan jendela alat tertentu, arahkan kursor ke atas ikon jendela dan pilih jendela alat tersebut dari menu.

Anda juga bisa menggunakan pintasan keyboard untuk membuka jendela alat.

Jendela	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

Tabel 3.1 mencantumkan pintasan jendela paling umum

Jika Anda ingin menyembunyikan semua toolbar, jendela alat, dan tab editor, klik **View > Enter Distraction Free Mode**. Langkah ini akan

mengaktifkan *Distraction Free Mode*. Untuk keluar dari *Distraction Free Mode*, klik **View > Exit Distraction Free Mode**.

Anda 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.

### 3.4.5. Perlengkapan Kode

Android Studio memiliki tiga jenis perlengkapan kode, yang dapat Anda akses 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+Spasi	Shift+Command+Enter
----------------------	--	---------------------	---------------------

Tabel 3.2 Pintasan keyboard untuk pelengkapan kode.

Anda juga dapat melakukan perbaikan cepat dan menampilkan tindakan maksud dengan menekan **Alt+Enter**.

### 3.4.6. Navigasi

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

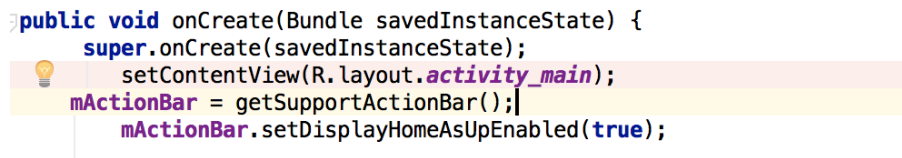
- Beralih antar file yang baru saja diakses menggunakan tindakan *Recent Files*. Tekan **Control+E** (**Command+E** pada Mac) untuk memunculkan tindakan Recent Files. Secara default, file yang terakhir diakses akan dipilih. Anda juga dapat mengakses jendela fitur mana saja melalui kolom kiri dalam tindakan ini.
- Lihat struktur file saat ini menggunakan tindakan *File Structure*. Munculkan tindakan File Structure dengan menekan **Control+F12** (**Command+F12** pada Mac). Dengan tindakan ini, Anda dapat membuka bagian mana pun dari file saat ini dengan cepat.
- Telusuri dan buka class tertentu dalam project menggunakan tindakan *Navigate to Class*. Munculkan tindakan ini dengan menekan **Control+N** (**Command+O** pada Mac). Navigate to Class mendukung ekspresi canggih, termasuk camel humps, jalur, baris navigasi ke, pencocokan nama tengah, dan banyak lagi. Jika Anda memanggilnya dua kali berturut-turut, hasil dari class project akan ditampilkan.
- Buka file atau folder menggunakan tindakan *Navigate to File*. Munculkan tindakan Navigate to File dengan menekan **Control+Shift+N** (**Command+Shift+O** pada Mac). 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** pada Mac).
- Temukan semua bagian kode yang merujuk ke class, metode, kolom, parameter, atau pernyataan di posisi kursor saat ini dengan menekan **Alt+F7** (**Option+F7** pada Mac).

### 3.4.7. Gaya dan Pemformatan

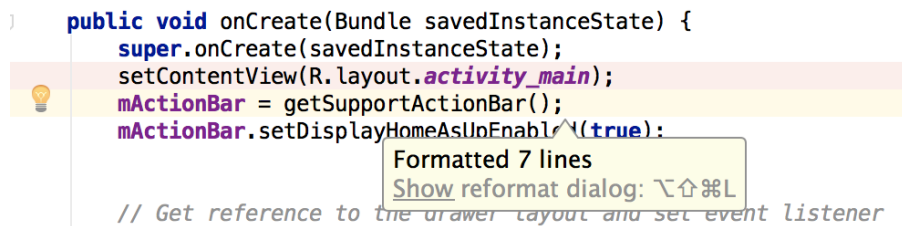
Saat Anda mengedit, Android Studio otomatis menerapkan pemformatan dan gaya seperti yang ditentukan dalam setelan gaya kode Anda. Anda dapat menyesuaikan setelan gaya kode menurut bahasa pemrograman, termasuk menentukan konvensi untuk tab dan indentasi, spasi, penggabungan, tanda kurung kurawal, dan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

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



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

Gambar 3.21 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 3.21 Sesudah Pemformatan

### 3.4.8. Cara Menginstall Android Studio

Penyiapan Android Studio hanya memerlukan beberapa klik.

Pertama, pastikan Anda mendownload Android Studio versi terbaru di <https://developer.android.com/studio/?hl=id>

#### 1. Windows

Untuk menginstal Android Studio di Windows, lakukan langkah berikut:

1. Jika Anda sudah mendownload file .exe (direkomendasikan), klik dua kali untuk meluncurkannya.

Jika Anda sudah mendownload file .zip, ekstrak ZIP, salin folder **android-studio** ke folder **Program Files**, lalu buka folder **studio-android** > **bin** dan jalankan studio64.exe (untuk komputer 64 bit) atau studio.exe (untuk komputer 32 bit).

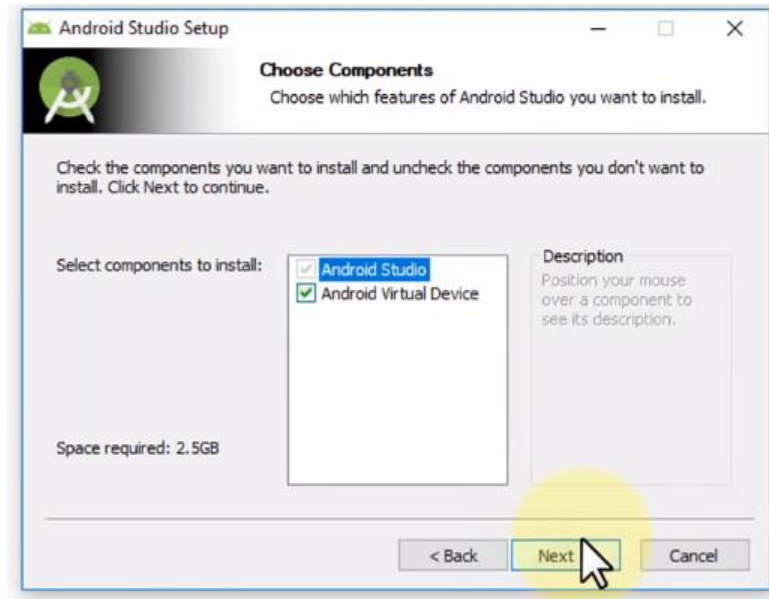
2. Kemudian buka file .exe
3. **klik next** setelah tampilan Install Android Studio muncul



Gambar 3.23 Langkah 1

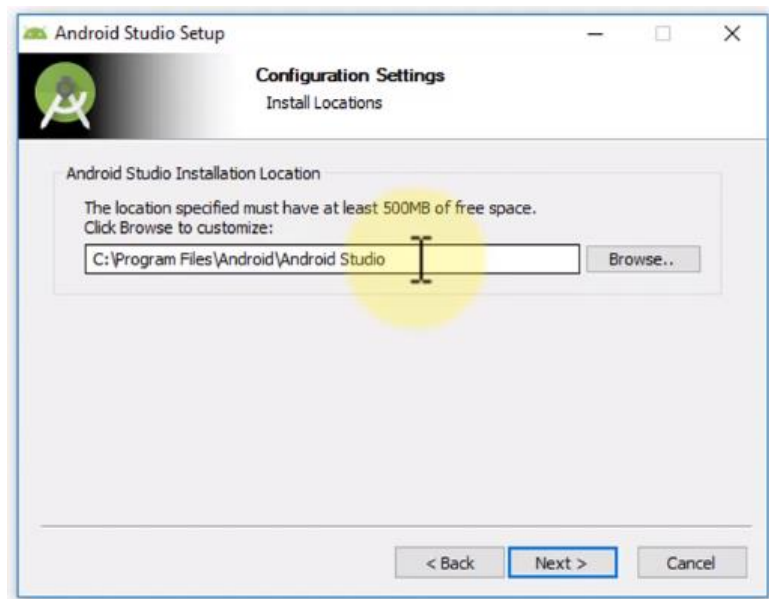
4. **klik next** untuk memasang semua komponen dari Android Studio





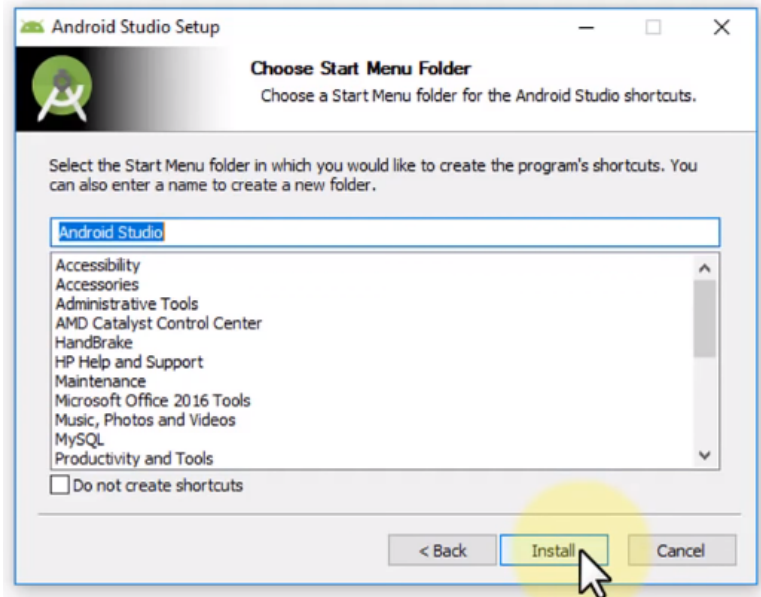
Gambar 3.24 Langkah 2

5. Tentukan lokasi penyimpanan Android Studio dan SDKnya. **Defaultnya tersimpan di Local Disk (C) lalu klik Next**



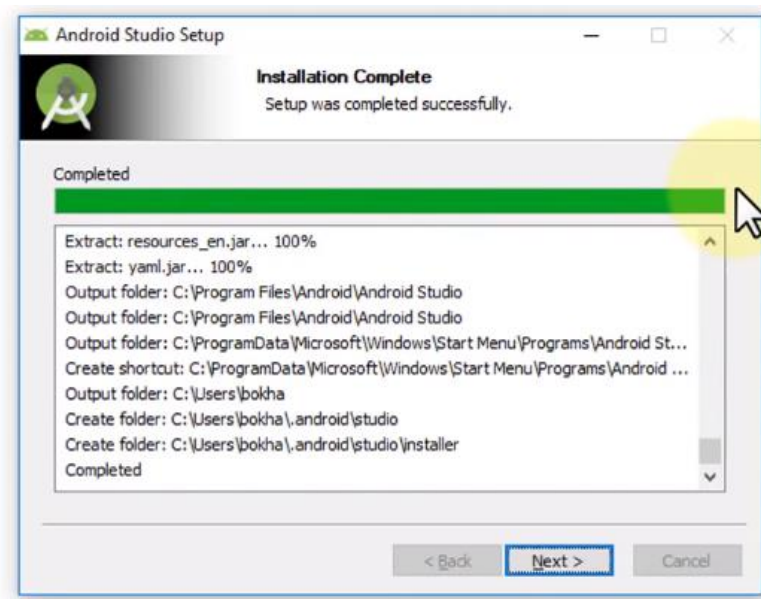
Gambar 3.25 Langkah 3

6. **Klik Install**



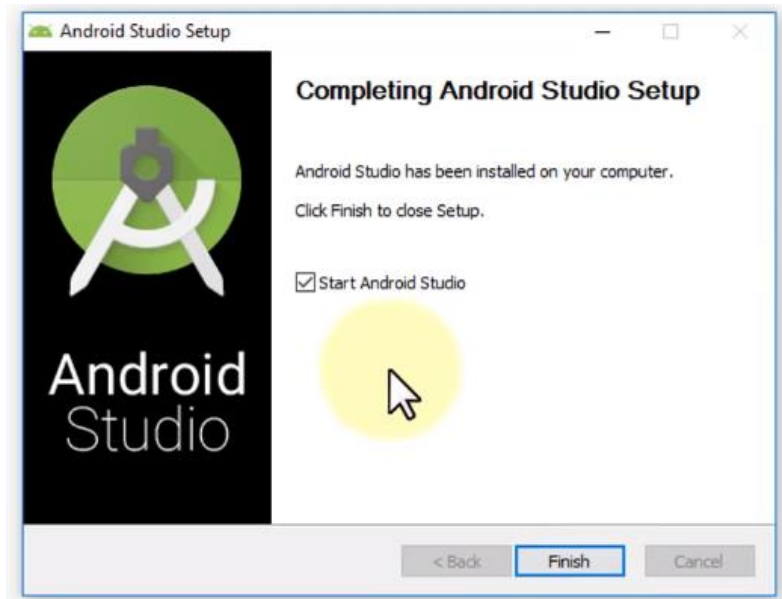
Gambar 3.26 Langkah 4

7. Tunggu sampai proses penginstallan selesai, **klik next**



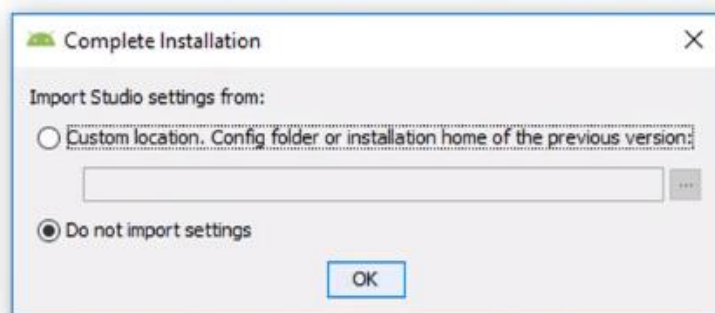
Gambar 3.27 Langkah 5

8. dan **klik finish**



Gambar 3.28 Langkah 6

9. Akan ada dua pilihan seperti gambar dibawah ini, karena baru pertama menginstall IDE Android Studio maka **klik pilihan yang kedua yaitu Do not import settings.**



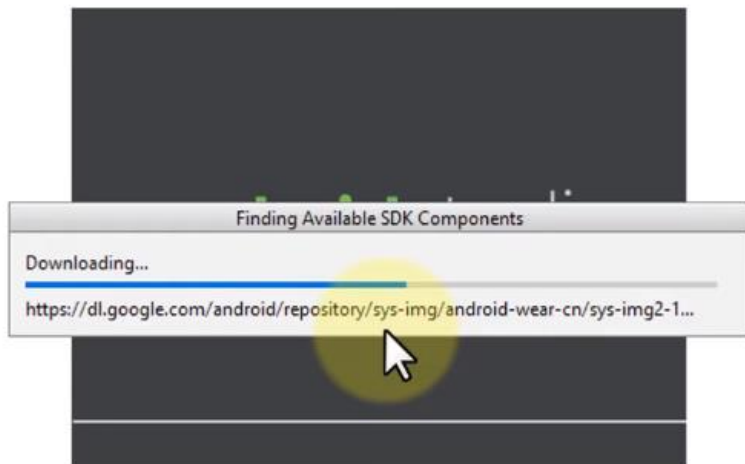
Gambar 3.29 Langkah 7

10. Maka akan muncul launcher aplikasi Android Studio.



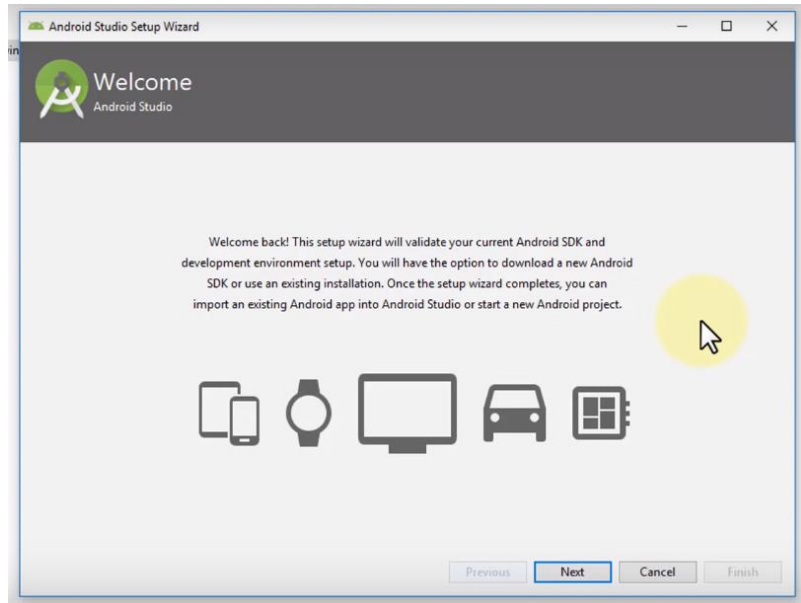
Gambar 3.30 Langkah 8

11. Pastikan laptop atau komputer yang digunakan terhubung dengan internet. Tunggu sampai proses download SDK selesai.



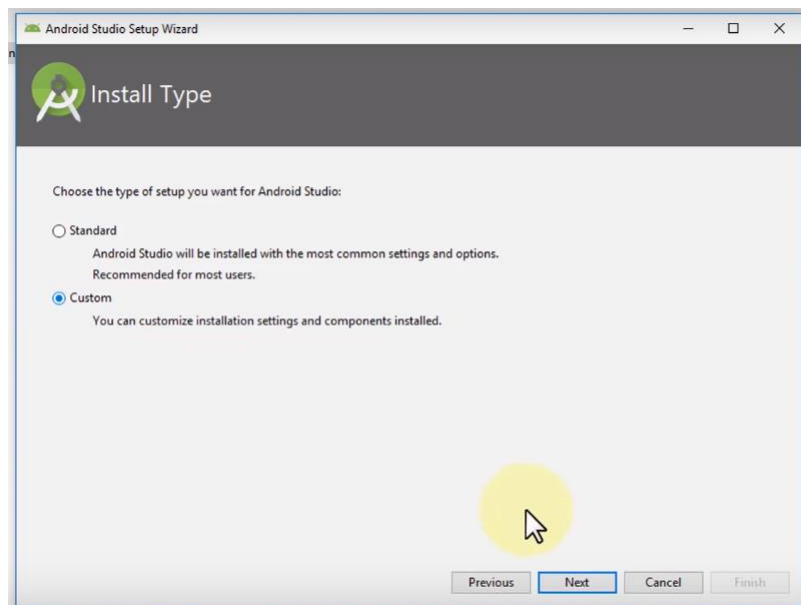
Gambar 3.31 Langkah 9

12. Setelah download SDK selesai maka akan muncul tampilan welcome Android Studio



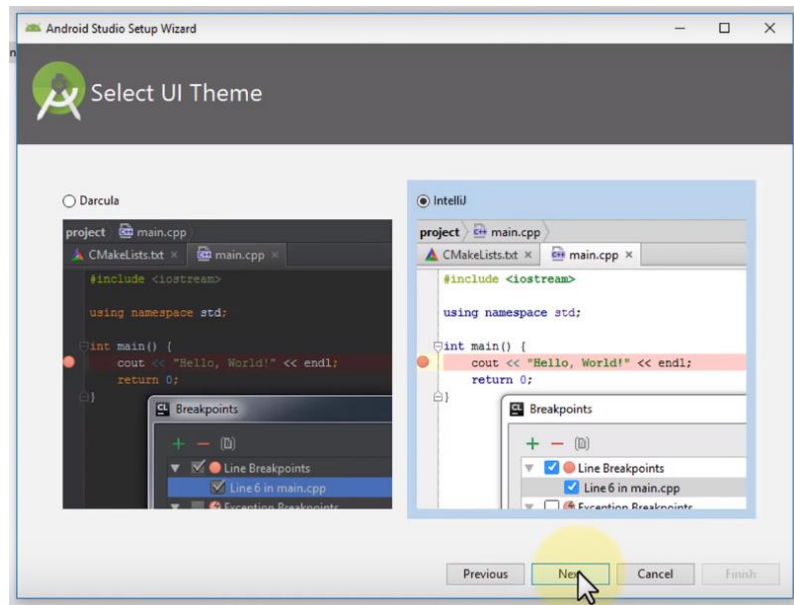
Gambar 3.32 Langkah 10

13. **Klik next** lalu pilih **Mode Install** (disini saya memilih custom), kemudian **klik next**



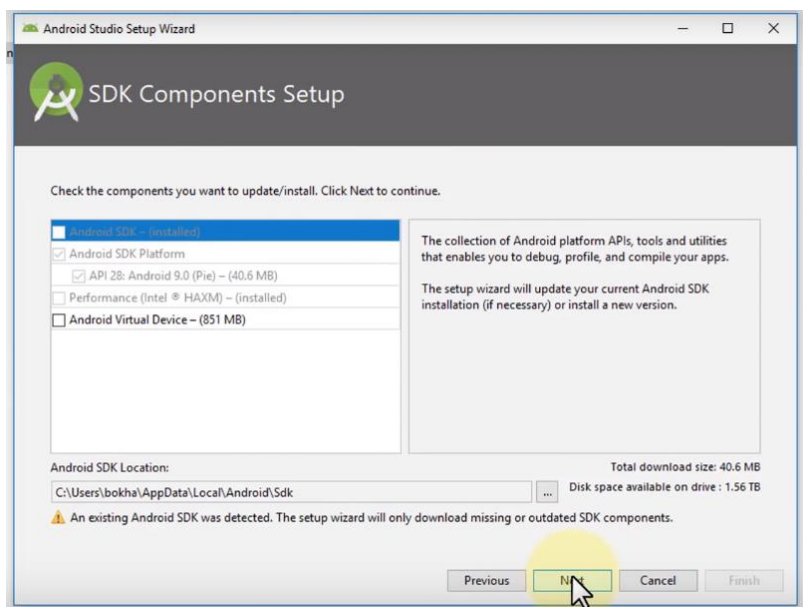
Gambar 3.33 Langkah 11

14. **Pilih theme UI** Android Studionya.



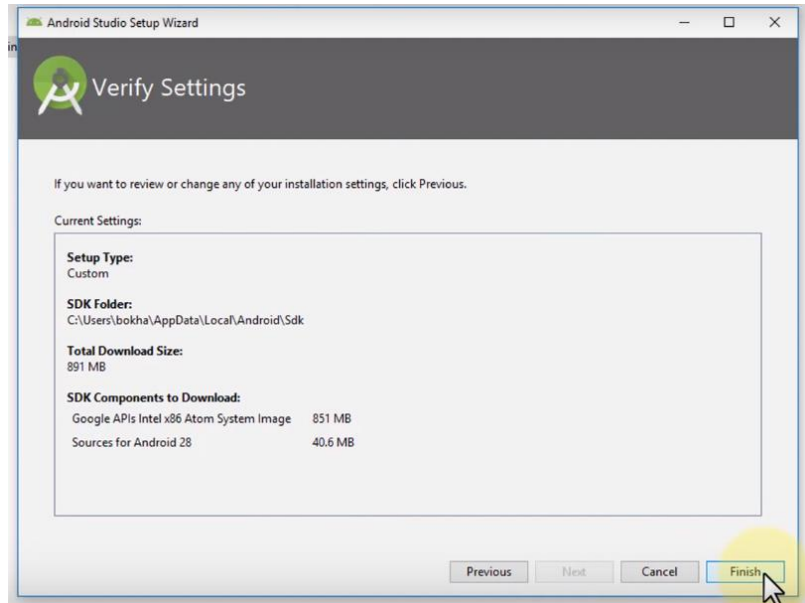
Gambar 3.34 Langkah 12

15. Disini akan dilakukan install komponen SDK, (ceklis android virtual device) lalu **klik next**



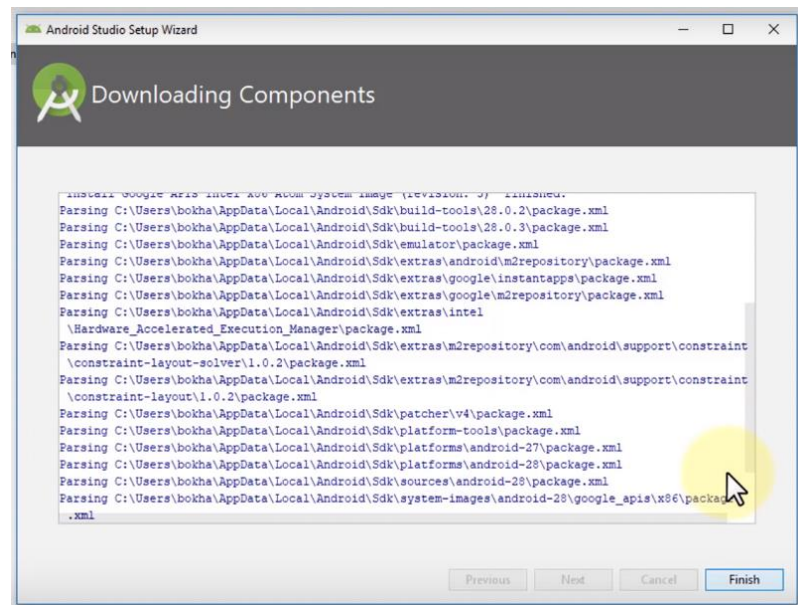
Gambar 3.35 Langkah 13

16. **Klik Finish**, dan tunggu proses download selesai (**pastikan komputer/ laptop terhubung ke internet**)



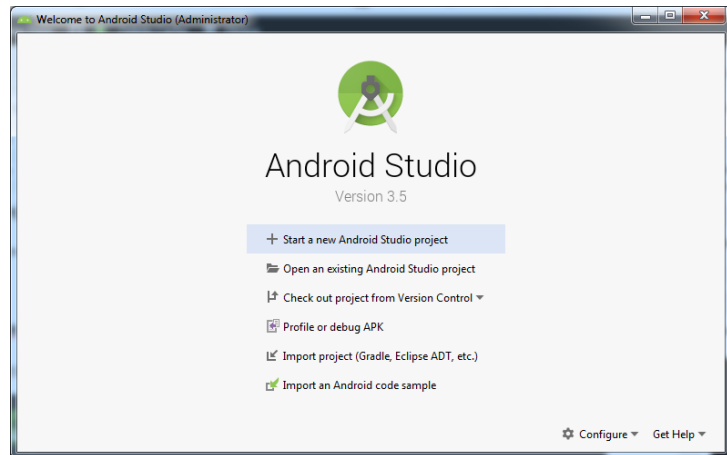
Gambar 3.36 Langkah 14

17. Proses download komponen, jika sudah **klik finish**.



Gambar 3.37 Langkah 15

18. Jika proses download telah selesai maka akan ditampilkan jendela awal Android Studio

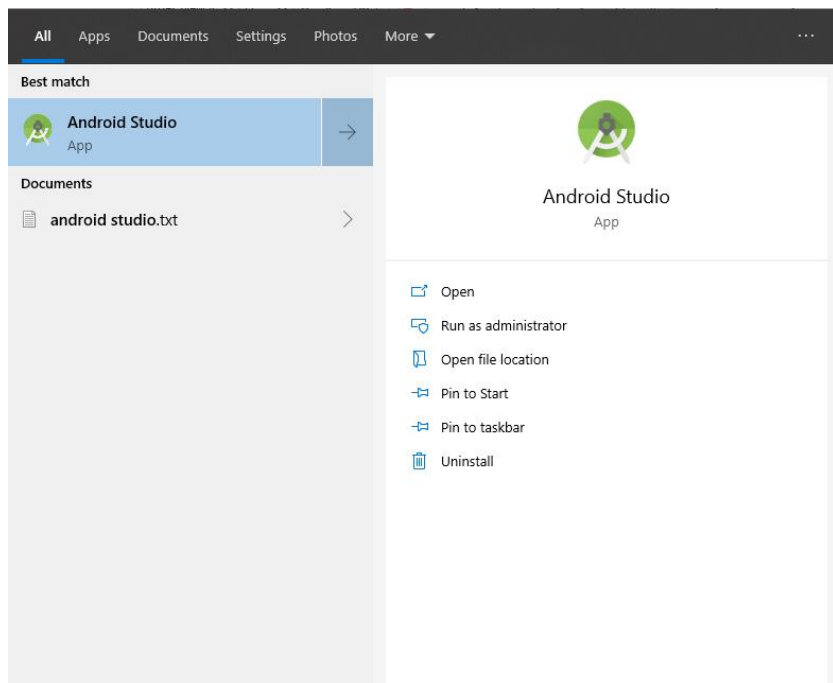


Gambar 3.338 Langkah 16

### 3.4.9. Membuat Aplikasi *Hello Word* di Android Studio

Setelah kita sudah menginstall aplikasi android studio, maka kita disini akan membuat aplikasi sederhana “*Hello Word*” pada aplikasi android studio.

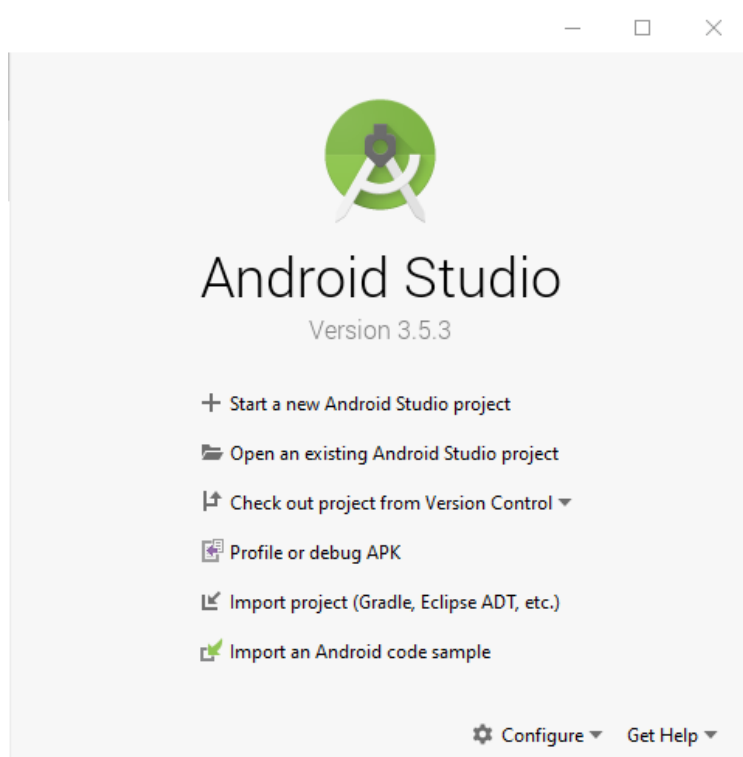
1. Buka aplikasi android studio pada windows search



Gambar 3.39 Membuka Aplikasi Android Studio

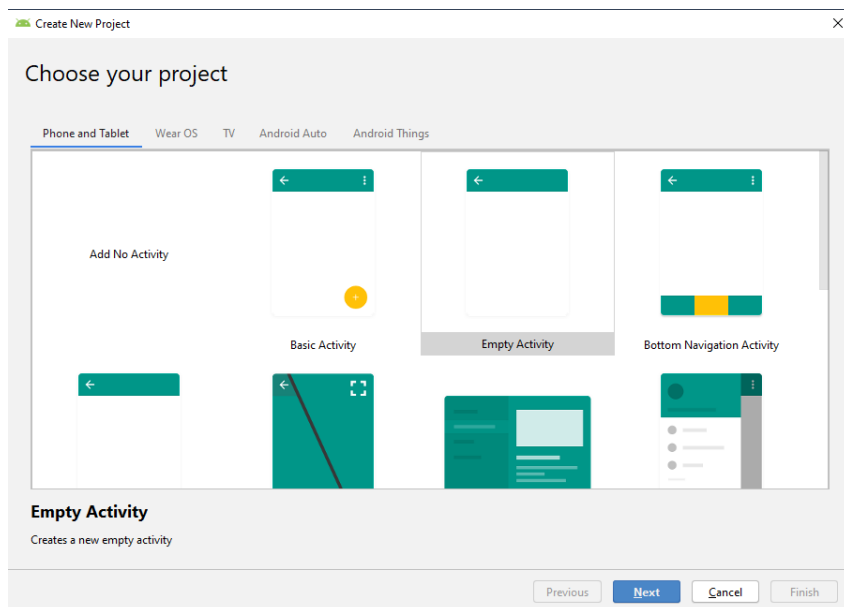
2. Klik start new Android Studio Project





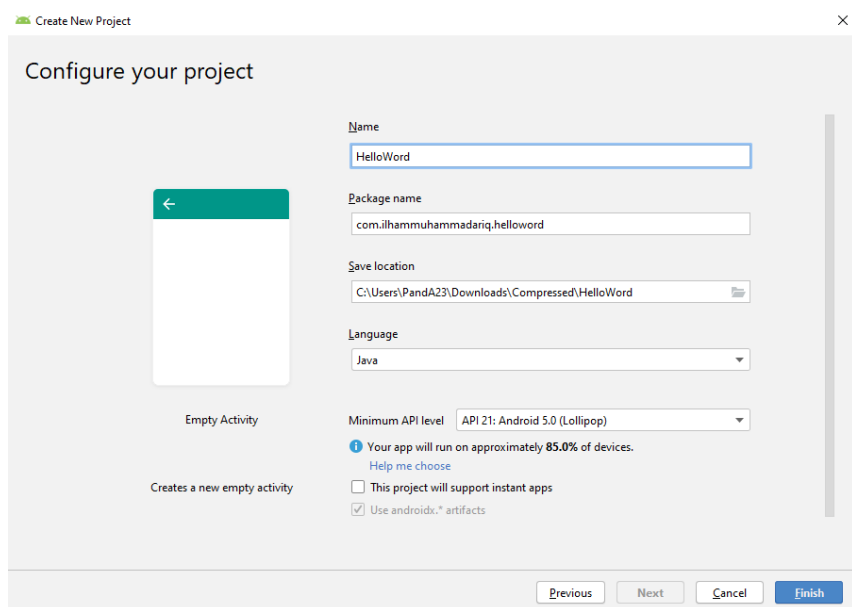
Gambar 3.40 Pilih Menu Project

3. Klik Empty Activity dan next



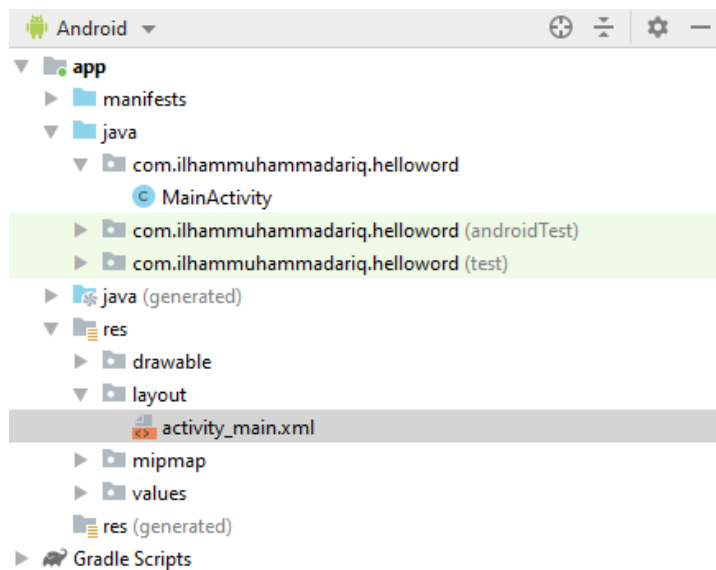
Gambar 3.41 Memilih Project

4. Isi nama aplikasi nya “HelloWord” atau bisa diisi dengan nama yang lain, pilih Bahasa pemrogramannya java dan klik finish




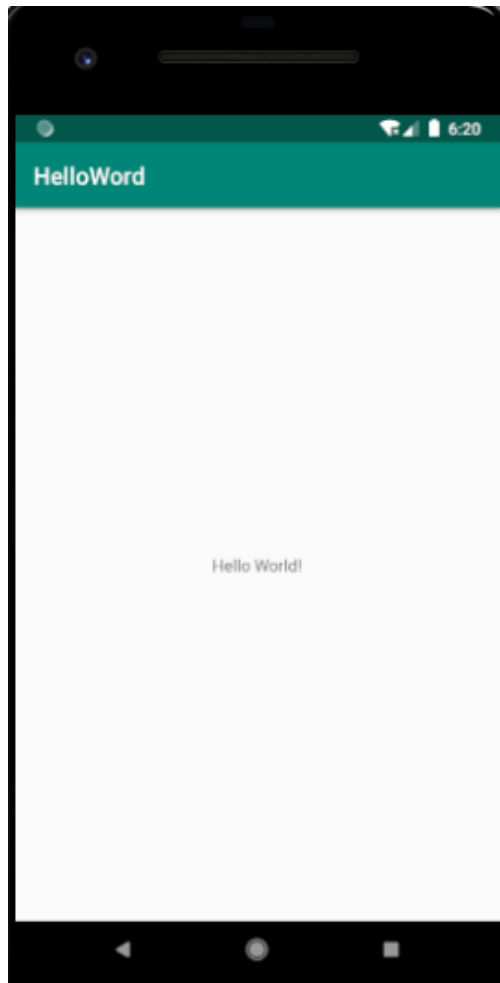
Gambar 3.42 Konfigurasi Project

5. Untuk membuat tampilan “hello word” maka kita hanya mengedit pada bagian .xml yang terletak pada folder res > layout



Gambar 3.43 Mengedit File .xml

6. Karena defaultnya yang telah menampilkan tulisan “hello word” ,maka kita run  aplikasi yang terletak pada pojok atas kanan dengan menggunakan emulator android studio
7. Maka tampilannya akan seperti ini



Gambar 3.44 Tampilan Aplikasi Project

#### **3.4.10. Cara Menjalankan Aplikasi pada Perangkat Sungguhan**

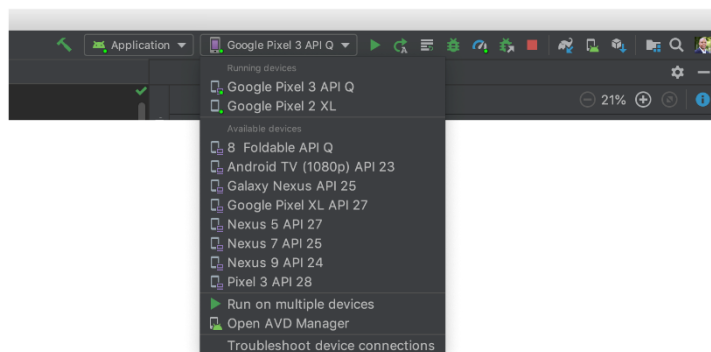
Persiapkan perangkat Anda sebagai berikut:

1. Hubungkan perangkat Anda ke komputer pengembangan dengan kabel USB.


2. Lakukan langkah-langkah berikut untuk mengaktifkan **proses debug USB** di jendela **Opsi developer**:
  - a) Buka aplikasi **Setelan**.
  - b) Jika perangkat Anda menggunakan Android v8.0 atau lebih tinggi, pilih **Sistem**. Jika tidak, lanjutkan ke langkah berikutnya.
  - c) Scroll ke bagian bawah, lalu pilih **Tentang ponsel**.
  - d) Scroll ke bagian bawah, lalu ketuk **Nomor build** tujuh kali.
  - e) Kembali ke layar sebelumnya, scroll ke bagian bawah, dan ketuk **Opsi developer**.
  - f) Di jendela **Opsi developer**, scroll ke bawah untuk menemukan dan mengaktifkan **proses debug USB**.

Jalankan aplikasi di perangkat Anda sebagai berikut:

1. Di Android Studio, pilih aplikasi Anda dari menu drop-down konfigurasi run/debug di toolbar.
2. Di toolbar, pilih perangkat tempat Anda ingin menjalankan aplikasi dari menu drop-down perangkat target.



Gambar 3.45 Pilih Perangkat

1. Klik **Run** .
2. Android Studio akan menginstal aplikasi Anda di perangkat yang terhubung dan menjalankannya. Setelah itu, Anda akan melihat "Hello, World!" ditampilkan dalam aplikasi di perangkat Anda.

## **BAGIAN IV**

### **FIREBASE**

#### **5.1. Definisi dan Sejarah**

Firebase adalah suatu layanan dari Google yang digunakan untuk mempermudah para pengembang aplikasi dalam mengembangkan aplikasi. Dengan adanya Firebase, pengembang aplikasi bisa fokus mengembangkan aplikasi tanpa harus memberikan usaha yang besar. Dua fitur yang menarik dari Firebase yaitu Firebase Remote Config dan Firebase Realtime Database. Selain itu terdapat fitur pendukung untuk aplikasi yang membutuhkan pemberitahuan yaitu Firebase Notification.



Gambar 4.1 Logo Firebase

Firebase berevolusi dari Envolv, startup sebelumnya yang didirikan oleh James Tamplin dan Andrew Lee pada tahun 2011. Envolv menyediakan pengembang Antarmuka pemrograman aplikasi yang memungkinkan integrasi fungsi obrolan daring ke situs web mereka. Setelah merilis layanan obrolan, Tamplin dan Lee menemukan bahwa Envolv digunakan untuk mengirimkan data aplikasi yang bukan pesan obrolan. Pengembang menggunakan Envolv untuk menyinkronkan data aplikasi seperti status permainan secara waktu nyata di seluruh penggunaannya. Tamplin dan Lee memutuskan untuk memisahkan sistem obrolan dan arsitektur waktu nyata yang menjalankannya. Tamplin dan Lee mendirikan Firebase sebagai perusahaan terpisah pada bulan April 2012.

Firebase Inc. mengumpulkan dana awal pada bulan Mei 2012. Perusahaan ini kemudian meningkatkan pendanaan Seri A pada bulan Juni 2013. Pada bulan Oktober 2014, Firebase diakuisisi oleh Google. Pada bulan Oktober 2015, Google mengakuisisi Divshot untuk menggabungkannya dengan Tim Firebase. Sejak akuisisi, Firebase telah berkembang di Google dan memperluas

layanan untuk menjadi platform terpadu bagi pengembang aplikasi. Firebase sekarang terintegrasi dengan berbagai layanan Google lainnya, untuk menawarkan produk dan skala yang lebih luas bagi pengembang aplikasi. Pada bulan Januari 2017, Google mengakuisisi Fabric dan Crashlytics dari Twitter untuk bergabung dengan layanan tersebut ke Tim Firebase. Firebase meluncurkan Cloud Firestore, sebuah database dokumen, pada bulan Oktober 2017.

## **4.2. Fitur Firebase**

### **1. Firebase Cloud Messaging**

Firebase Cloud Messaging (FCM) dulu dikenal sebagai Google Cloud Messaging (GCM), di mana Firebase Cloud Messaging (FCM) dapat mengirim dan menerima pesan lintas platform seperti Android, iOS, dan aplikasi web, yang pada tahun 2016 dapat digunakan tanpa biaya. Firebase Cloud Messaging (FCM), adalah layanan yang diberikan Firebase untuk menggantikan Google Cloud Messaging. Pihak Google menyarankan untuk aplikasi yang masih menggunakan Google Cloud Messaging untuk segera migrasi ke Firebase Cloud Messaging. Fitur-fitur yang diberikan oleh Firebase Cloud Messaging sebenarnya tidak terlalu jauh berbeda dengan Google Cloud Messaging. Dengan Firebase Cloud Messaging kita bisa memberikan pemberitahuan dan membuat komunikasi dua arah antara perangkat. Teknologi yang digunakan terbagi menjadi dua yaitu:

- XMPP (Extensible Messaging and Presence Protocol)
- HTTP (Hypertext Transfer Protocol).

Untuk XMPP harus membangun server XMPP terlebih dahulu, sedangkan untuk HTTP bisa menggunakan console yang disediakan oleh Firebase.

### **2. Firebase Authentication**

Firebase Authentication merupakan layanan sistem otentikasi yang menerapkan kode client-side, sehingga pengguna dapat mendaftar dan login ke aplikasi Facebook, GitHub, Twitter dan Google (Google Play Games). Selain itu, Firebase termasuk sistem manajemen pengguna di mana pengembang dapat mengaktifkan otentikasi pengguna dengan login email dan kata sandi yang disimpan dengan Firebase. Sebagian besar aplikasi perlu mengetahui identitas pengguna. Dengan mengetahui identitas pengguna, aplikasi dapat menyimpan data pengguna secara aman di cloud dan memberikan pengalaman personal yang sama di setiap perangkat pengguna. Firebase Authentication mendukung otentikasi menggunakan sandi, nomor telepon, penyedia identitas gabungan yang populer, seperti Google, Facebook, dan Twitter, dan lain-lain.

### 3. **Firestore Remote Config**

Firestore Remote Config adalah fitur yang memungkinkan digunakan untuk melakukan perubahan konfigurasi di dalam aplikasi Android / iOS, tanpa harus melakukan pembaruan aplikasi di Play Store / App Store. Cara kerja dari Remote Config adalah aplikasi menyimpan terlebih dahulu file XML yang berisi parameter-parameter yang nilainya akan bisa diganti melalui console Firestore. Kemudian objek firestore di dalam aplikasi akan melakukan request data dari server, kemudian memproses data-data tersebut. Secara umum, objek Firestore di dalam aplikasi akan melakukan request 12 jam / 1x, tetapi hal tersebut bisa diubah sesuai yang diinginkan. Salah satu catatan dari Google adalah tidak boleh menggunakan Remote Config untuk melakukan perubahan yang krusial, seperti mengubah izin yang dibutuhkan oleh aplikasi.

### 4. **Firestore Realtime Database**

Firestore menyediakan realtime database dan backend sebagai layanan. Layanan ini menyediakan pengembang Antarmuka pemrograman aplikasi yang memungkinkan data aplikasi disinkronkan di seluruh klien dan disimpan di Firestore cloud. Perusahaan menyediakan pustaka klien yang memungkinkan integrasi dengan Android, iOS, JavaScript, Java, Swift dan Node.js aplikasi. Database juga dapat diakses melalui REST API dan mengikat untuk beberapa JavaScript Frameworks seperti AngularJS, React, Ember.js dan Backbone.js. Developer yang menggunakan realtime database dapat mengamankan data dengan menggunakan aturan keamanan yang diberlakukan oleh server perusahaan. Cloud Firestore yang merupakan generasi mendatang dari Firestore Realtime Database dirilis untuk penggunaan beta. Firestore Realtime Database merupakan database yang di-host pada cloud. Data disimpan sebagai JSON dan disinkronkan secara realtime ke setiap klien yang terhubung.

### 5. **Firestore Storage**

Firestore Storage dirancang untuk pengembang aplikasi yang perlu menyimpan dan menampilkan konten buatan pengguna, seperti foto atau video dan menambahkan keamanan Google pada unggah dan unduh berkas untuk aplikasi Firestore, bagaimana pun kualitas jaringannya. Pengembang dapat menggunakannya untuk menyimpan gambar, audio, video, atau konten lain yang dibuat pengguna secara langsung dari Firestore SDK Klien. Firestore Storage didukung oleh Google Cloud Storage.<sup>[24]</sup>

### 6. **Firestore Hosting**

Firestore Hosting menyediakan hosting yang cepat dan aman untuk aplikasi web serta konten yang statis dan dinamis. Firestore Hosting adalah layanan hosting web statis dan dinamis yang diluncurkan pada 13 Mei 2014.

Mendukung hosting berkas statis seperti CSS, HTML, JavaScript dan berkas lainnya, serta dukungan melalui Cloud Functions. Layanan ini mengirimkan berkas melalui Content Delivery Network (CDN) atau melalui enkripsi HTTP Secure (HTTPS) dan Secure Sockets Layer (SSL). Firebase bermitra dengan Fastly, CDN, untuk menyediakan CDN yang mendukung Firebase Hosting. Perusahaan menyatakan bahwa Firebase Hosting tumbuh dari permintaan pelanggan, pengembang menggunakan Firebase untuk realtime database tetapi membutuhkan tempat untuk meng-host konten mereka.

Dua fitur yang menarik adalah Firebase Remote Config dan Firebase Realtime Database. Secara sederhana, Remote Config adalah fitur yang memungkinkan pengembang mengganti / mengubah beberapa konfigurasi aplikasi Android / iOS tanpa harus memberikan pembaruan aplikasi melalui Play Store / App Store. Salah satu konfigurasi yang bisa dimanipulasi adalah seperti warna / tema aplikasi. Sedangkan Firebase Realtime Database adalah fitur yang memberikan sebuah NoSQL database yang bisa diakses secara waktu nyata oleh pengguna aplikasi. Dan hebatnya adalah aplikasi bisa menyimpan data secara lokal ketika tidak ada akses internet, kemudian melakukan sinkronisasi data segera setelah mendapatkan akses internet.

#### **4.3. Stabilitas**

Untuk menstabilkan aplikasi yang sudah menggunakan layanan Firebase kita juga bisa menggunakan fitur-fitur dibawah ini, antara lain :

##### **1. Firebase Crash Reporting**

Firebase Crash Reporting digunakan untuk membuat laporan rinci tentang kesalahan dalam aplikasi. Kesalahan dikelompokkan ke dalam kelompok tumpukan pelacakan yang serupa dan diurutkan berdasarkan tingkat keparahan dampaknya pada pengguna aplikasi. Selain laporan otomatis, pengembang dapat membuat log peristiwa khusus untuk membantu menangkap langkah-langkah yang mengarah ke kerusakan. Sebelum mendapatkan Crashlytics, Firebase menggunakan Firebase Crash Reporting sendiri. Firebase Crash Reporting merupakan layanan yang diberikan oleh Firebase untuk keperluan merekam setiap pengecualian yang terjadi pada aplikasi. Report yang diberikan cukup detail dengan beberapa filter seperti yang digunakan pada Analytics. Crash Reporting ini juga dibagi menjadi dua bagian yaitu:

- Non Fatal exception, untuk pengecualian yang tidak berdampak fatal (bisa membuat pengecualian khusus).
- Fatal exception, untuk pengecualian yang fatal (aplikasi crash).



## **2. Firebase Performance**

Firebase Performance memberikan wawasan tentang kinerja aplikasi yang dialami pengguna aplikasi. Firebase Performance adalah layanan yang membantu untuk memperoleh data tentang karakteristik performa aplikasi iOS dan Android. Gunakan Firebase Performance SDK untuk mengumpulkan data performa aplikasi, kemudian tinjau dan analisis data tersebut di Firebase console. Dengan Firebase Performance, dapat memahami di mana dan kapan performa aplikasi dapat ditingkatkan, sehingga dapat menggunakan informasi tersebut untuk memperbaiki masalah performa.

## **3. Firebase Tes Lab untuk Android dan iOS**

Firebase Tes Lab untuk Android dan iOS menyediakan infrastruktur berbasis cloud untuk menguji aplikasi Android dan iOS. Dengan satu operasi, pengembang dapat memulai pengujian aplikasi di berbagai macam perangkat dan konfigurasi perangkat. Hasil pengujian termasuk log, video, dan tangkapan layar, tersedia dalam proyek di Firebase console. Bahkan jika pengembang tidak menulis kode pengujian apa pun untuk aplikasi, Tes Lab dapat menjalankan aplikasi secara otomatis dengan mencari gangguan. Tes Lab untuk iOS saat ini dalam versi beta.

## **4.4. Grow**

Setelah aplikasi stabil dan ingin menumbuhkan pengguna atau berlanjut ke sistem bisnis, bisa menggunakan fitur-fitur untuk grow dari firebase yaitu:

### **1. Firebase Notifications**

Firebase Notifications adalah layanan yang memungkinkan pemberitahuan untuk pengguna yang ditargetkan oleh pengembang aplikasi seluler tanpa biaya.

### **2. Firebase App Indexing**

Firebase App Indexing digunakan untuk memasukkan aplikasi ke Google Search. Jika pengguna sudah menginstal aplikasi, pengguna dapat menjalankannya dan dapat langsung membuka konten yang dicari. Firebase App Indexing kembali berinteraksi dengan pengguna aplikasi dengan cara membantu pengguna menemukan konten publik dan pribadi secara langsung di perangkat pengguna.

Firebase App Indexing menawarkan kelengkapan otomatis kueri untuk membantu pengguna menemukan sesuatu yang mereka butuhkan dengan lebih cepat. Jika pengguna belum memiliki aplikasi tersebut, kueri yang relevan akan memicu untuk memasang aplikasi dari hasil Google Search. Firebase App Indexing, sebelumnya yaitu Google App Indexing, mendapatkan aplikasi ke Google Search. Penambahan App Indexing untuk

mempromosikan kedua jenis hasil aplikasi dalam Google Search dan juga menyediakan kelengkapan otomatis kueri.

### **3. Firebase Dynamic Links**

Firebase Dynamic Links adalah URL cerdas yang secara dinamis mengubah perilaku untuk memberikan pengalaman terbaik di berbagai platform (situs web / iOS / Android) serta tautan dalam ke APP. Dynamic Links berfungsi di semua pemasangan aplikasi: jika pengguna membuka Dynamic Links di iOS atau Android dan aplikasi belum dipasang, pengguna akan diminta untuk memasang aplikasi tersebut terlebih dahulu. Setelah dipasang, aplikasi akan mulai berjalan dan dapat mengakses link.

### **4. Firebase Invites**

Firebase Invites adalah solusi siap pakai untuk berbagi aplikasi melalui email atau SMS. Untuk menyesuaikan pengalaman pengguna invites atau untuk menghasilkan link secara terprogram, gunakan Firebase Dynamic Links.

# **BAGIAN V**

## **PENGUKURAN DIMENSI BARANG MENGGUNAKAN ARDUINO**

### **5.1. STUDI KASUS**

Didalam proses logistik yang dimana barang melewati proses pengecekan berat barang yang masih dilakukan secara manual sebelum dilakukan pengiriman . Berat barang asli belum tentu sama dengan berat dimensional. Dengan demikian, diperlukan pengukuran dimensi barang untuk menghasilkan berat dimensional tersebut. Kemudian berat dimensional ini dibandingkan dengan berat asli sehingga menghasilkan nilai selisih sebagai penentu nilai berat mana yang di pakai untuk penentuan harga pengiriman barang. Jadi, dibuat sebuah alat yang dapat memberikan pekerja lebih mudah melihat data terbaru menggunakan aplikasi android untuk menentukan harga pengiriman barang dan proses pengecekan barang menjadi lebih mudah dan cepat.

### **5.2. ALAT YANG DIBUTUHKAN**

Berdasarkan studi kasus diatas alat-alat yang dibutuhkan :

- Leptop
- *Software* Arduino UNO
- Kabel USB Type A/B
- *Jumper Wires*
- *HC-SR04 Ultrasonic Sensor*
- LED
- *Node MCU*
- *HX711*
- *Loadcell*
- Android
- *Software* Android Studio
- Triplek
- Akrilik
- Gergaji
- Palu
- Baut
- Paku

### 5.3. LANGKAH-LANGKAH PENYELESAIAN PROTOTYPE

Langkah-langkah dalam menyelesaikan prototype studi kasus diatas:

1. Kita melakukan pengujian alat-alat yang kita gunakan untuk memastikan apakah alat-alat yang kita gunakan dapat berjalan dengan lancar dan baik.

#### 1.1. Pengujian Ultrasonic Sensor

##### 1.1.1. Tujuan Pengujian

Pengujian *Ultrasonic Sensor* bertujuan untuk memastikan ultrasonic sensor yang kita gunakan bisa berjalan dengan baik. Pada studi kasus ini Ultrasonik sensor sangatlah penting karna digunakan untuk mengukur panjang, tinggi, dan lebar sehingga dapat volume dan berat dimensi barang sebagai data utama yang akan di proses pada Arduino.

##### 1.1.2. Alat yang Digunakan

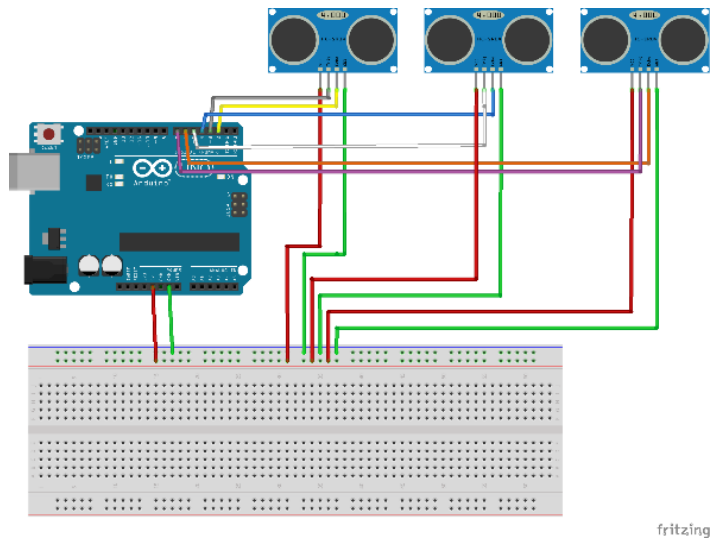
Alat yang digunakan untuk pengujian *ultrasonic sensor* adalah :

1. Arduino Uno R3
2. Kabel USB Arduino
3. PC/Laptop
4. *Software* Arduino IDE
5. *Jumper Wires*
6. *HC-SR04 Ultrasonic Sensor*

##### 1.1.3. Langkah Pengujian

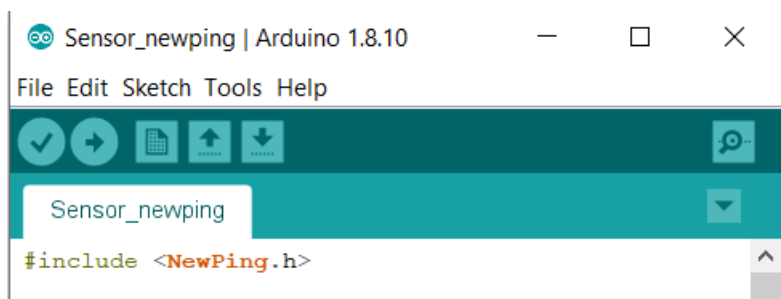
Langkah-langkah pengujian *ultrasonic sensor* adalah sebagai berikut :

1. Merangkai *ultrasonic sensor* dengan *microcontroller* Arduino Uno seperti dibawah ini :



Gambar 5.1 Rangkaian Pengujian *Ultrasonic Sensor*

2. Setelah merangkai *ultrasonic sensor* install *library* NewPing pada *software* Arduino IDE agar ultrasonic sensor dapat dijalankan.



3. Kemudian membuat konstruktor sesuai library NewPing, untuk setiap sensor ultrasonic.

```
NewPing sonar1(3, 2, 11);
NewPing sonar2(5, 4, 17);
NewPing sonar3(7, 6, 17);
```

Note: `NewPing sonar(trigger_pin, echo_pin, [max_cm_distance]);`

4. Membuat variable jarak maksimum yang nanti akan dikurangi dengan jarak yang di dapatkan dari sensor ultrasonic.

```
int jarakAtas = 11;  
int jarakDepan = 17;  
int jarakSamping = 17;
```

5. Pada metode `Setup()` kita tambahkan untuk pembacaan serial monitor.

```
void setup() {  
  Serial.begin(9600);  
}
```

6. Kemudian pada metode `Loop()` kita tambahkan code yang dimana akan mendapatkan nilai tinggi, lebar panjang, dan perhitungan volume.

```

void loop() {
    delay(2000);
    tinggi = jarakAtas - sonar1.ping_cm();
    Serial.print("Tinggi: ");
    Serial.println(tinggi);
    Serial.print("cm");

    delay(2000);
    lebar = jarakDepan - sonar2.ping_cm();
    Serial.print("Lebar: ");
    Serial.println(lebar);
    Serial.print("cm");

    delay(2000);
    panjang = jarakSamping - sonar3.ping_cm();
    Serial.print("Panjang: ");
    Serial.println(panjang);
    Serial.print("cm");

    volume = panjang * lebar * tinggi;
    Serial.print("Volume: ");
    Serial.println(volume);
    Serial.print("cm^3");

}

```

7. Jangan lupa inisiasi variable tinggi, lebar, panjang dan volume

```

int tinggi, lebar, panjang, volume;

```

8. Full codenya sebagai berikut:

```
Sensor_newping | Arduino 1.8.10
File Edit Sketch Tools Help

Sensor_newping $

#include <NewPing.h>
NewPing sonar1(3, 2, 11);
NewPing sonar2(5, 4, 17);
NewPing sonar3(7, 6, 17);

int jarakAtas = 11;
int jarakDepan = 17;
int jarakSamping = 17;
int tinggi, lebar, panjang, volume;

void setup() {
  Serial.begin(9600);
}

void loop() {
  delay(2000);
  tinggi = jarakAtas - sonar1.ping_cm();
  Serial.print("Tinggi: ");
  Serial.print(tinggi);
  Serial.println(" cm");
  lebar = jarakSamping - sonar3.ping_cm();
  Serial.print("Lebar: ");
  Serial.print(lebar);
  Serial.println(" cm");
  panjang = jarakDepan - sonar2.ping_cm();
  Serial.print("Panjang: ");
  Serial.print(panjang);
  Serial.println(" cm");
  volume = panjang * lebar * tinggi;
  Serial.print("Volume: ");
  Serial.print(volume);
  Serial.println(" cm^3");
}
```

Gambar 4.2 Source code Ultrasonic Sensor

## 9. Lalu Compile dan Upload pada Arduino Uno

### 1.1.4. Hasil Pengujian

Dari pengujian *ultrasonic sensor* yang telah dilakukan, membuktikan bahwa *ultrasonic sensor* dapat berfungsi dengan baik. Maka *ultrasonic sensor* dapat membaca jarak antara *ultrasonic sensor* dengan barang.

## 1.2. Pengujian HX711 dan Load Cell

### 1.2.1. Tujuan Pengujian

Pengujian *HX711 dan Load Cell* bertujuan untuk memastikan *HX711 dan Load Cell* bisa berjalan dengan baik. *HX711 dan Load Cell* digunakan untuk mengukur berat asli barang. Code program yang digunakan di sini yaitu saat melakukan kalibrasi load cell.



### 1.2.2. Alat yang Digunakan

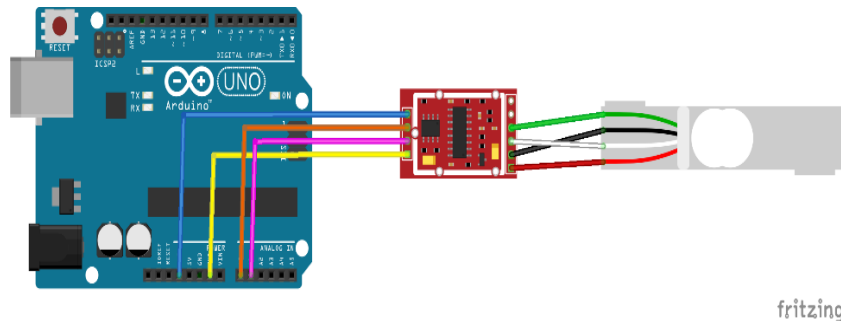
Alat yang digunakan untuk pengujian *ultrasonic sensor* adalah:

1. Arduino Uno R3
2. Kabel USB Arduino
3. PC/Laptop
4. *Software* Arduino IDE
5. *Jumper Wires*
6. *HX711*
7. *Load Cell*

### 1.2.3. Langkah Pengujian

Langkah-langkah pengujian *HX711* dan *Load Cell* adalah sebagai berikut :

1. Merangkai *HX711* dan *Load Cell* dengan *microcontroller* Arduino Uno seperti dibawah ini :



Gambar 4.6 Rangkaian Pengujian HX711 dan LoadCell

2. Setelah merangkai *HX711* dan *Load Cell* install *library* *hx711* pada *software* Arduino IDE agar *HX711* dan *Load Cell* dapat dijalankan.



3. Kemudian kita mendefinisikan pin Arduino yang digunakan untuk HX711

```
#define DOUT A1 //mendefinisikan pin arduino yang  
#define CLK A0 //mendefinisikan pin arduino yang
```

4. Kemudian kita membuat konstruktor sesuai dengan library HX711

```
HX711 scale(DOUT, CLK);
```

5. Membuat inisiasi variable kalibrasi

```
float calibration_factor = -393; //nilai kalibrasi
```

Note: nilai kalibrasi diatas merupakan nilai awal yang dimana setelah melakukan kalibrasi akan diganti.

6. Menambahkan code pada metode setup().

```

void setup() {
  Serial.begin(9600);
  Serial.println("Memulai program kalibrasi pada sensor berat");
  Serial.println("Pastikan tidak ada beban diatas sensor");
  delay(5000);
  scale.set_scale();
  scale.tare(); // auto zero / mengesolkan pembacaan berat

  long zero_factor = scale.read_average(); //membaca nilai output
  Serial.print("Zero factor: ");
  Serial.println(zero_factor);
}

```

7. Dan kemudian menambahkan code pada metode loop().

```

void loop() {

  scale.set_scale(calibration_factor); //sesuaikan hasil pembacaan
  delay(2000);
  Serial.print("Berat: ");
  Serial.print(scale.get_units(), 2);
  Serial.print(" gram");
  Serial.print(" calibration_factor: ");
  Serial.print(calibration_factor);
  Serial.println();

  if(Serial.available())
  {
    char temp = Serial.read();
    if (temp == '+' || temp == 'a')
      calibration_factor += 0.1;
    else if (temp == '-' || temp == 'z')
      calibration_factor -= 0.1;
    else if (temp == 's')
      calibration_factor += 10;
    else if (temp == 'x')
      calibration_factor -= 10;
    else if (temp == 'd')
      calibration_factor += 100;
    else if (temp == 'c')
      calibration_factor -= 100;
    else if (temp == 'f')
      calibration_factor += 1000;
    else if (temp == 'v')
      calibration_factor -= 1000;
    else if (temp == 't')
      scale.tare();
  }
}

```

Note: kode bagian If(serial.available()) merupakan nilai untuk melakukan kalibrasi untuk mendapatkan berat yang tepat.

8. Lalu compile dan upload
9. Full code sebagai berikut

```
sketch_jan26a $  
  
#include "HX711.h" //memasukan library HX711  
  
#define DOUT A1 //mendefinisikan pin arduino yang terhubung dengan sensor ke pin nomor 1 pada pin header  
#define CLK A0 //mendefinisikan pin arduino yang terhubung dengan sensor ke pin nomor 0 pada pin header  
  
HX711 scale(DOUT, CLK);  
  
float calibration_factor = -393; //nilai kalibrasi awal  
  
void setup() {  
  Serial.begin(9600);  
  Serial.println("Memulai program kalibrasi pada sensor berat");  
  Serial.println("Pastikan tidak ada beban diatas sensor");  
  delay(5000);  
  scale.set_scale();  
  scale.tare(); // auto zero / mengesetkan pembacaan berat menjadi 0  
  
  long zero_factor = scale.read_average(); //membaca nilai output dari sensor  
  Serial.print("Zero factor: ");  
  Serial.println(zero_factor);  
}  
  
void loop() {  
  
  scale.set_scale(calibration_factor); //sesuaikan hasil pembacaan dengan nilai kalibrasi  
  delay(2000);  
  Serial.print("Berat: ");  
  Serial.print(scale.get_units(), 2);  
  Serial.print(" gram");  
  Serial.print(" calibration_factor: ");  
  Serial.print(calibration_factor);  
  Serial.println();  
}
```

```
if(Serial.available())
{
  char temp = Serial.read();
  if (temp == '+' || temp == 'a')
    calibration_factor += 0.1;
  else if (temp == '-' || temp == 'z')
    calibration_factor -= 0.1;
  else if (temp == 's')
    calibration_factor += 10;
  else if (temp == 'x')
    calibration_factor -= 10;
  else if (temp == 'd')
    calibration_factor += 100;
  else if (temp == 'c')
    calibration_factor -= 100;
  else if (temp == 'f')
    calibration_factor += 1000;
  else if (temp == 'v')
    calibration_factor -= 1000;
  else if (temp == 't')
    scale.tare();
}
```

10. Setelah itu buka serial monitor, dan melakukan kalibrasi sesuai dengan berat yang digunakan untuk kalibrasi, catat nilai `calibration_factor`.
11. Kemudian masukan nilai `calibration_factor` yang di dapatkan dan hapus code yang digunakan untuk melakukan kalibrasi, seperti gambar di bawah ini:



```

sketch_jan15b | Arduino 1.8.10
File Edit Sketch Tools Help

#include "HX711.h" //memasukan library HX711

#define DOUT A1 //mendefinisikan pin arduino yang terhubung dengan pin DT module HX711
#define CLK A0 //mendefinisikan pin arduino yang terhubung dengan pin SCK module HX711

HX711 scale(DOUT, CLK);

float calibration_factor = -393; //nilai kalibrasi awal
int GRAM;

void setup() {
  Serial.begin(9600);
  scale.set_scale();
  scale.tare();
}

void loop() {
  delay(4000);
  scale.set_scale(calibration_factor);
  GRAM = scale.get_units(), 4;
  Serial.print(GRAM);
  Serial.println(" gram");
}

```

Gambar 4.7 Source HX711 dan LoadCell

Codingan ini menjelaskan bahwa *HX711* membaca *scale* yang didapatkan dari *Load Cell* yang berupa berat dari barang yang berada di atas *load cell* tersebut.

#### 1.2.4. Hasil Pengujian

Dari pengujian *HX711* dan *Load Cell* yang telah dilakukan, membuktikan bahwa *HX711* dan *Load Cell* dapat berfungsi dengan baik. Maka *HX711* dan *Load Cell* dapat membaca berat asli barang yang ditempatkan pada atas *load cell*.

### 1.3. Pengujian Node Mcu

#### 1.3.1. Tujuan Pengujian

Pengujian *Node Mcu* ini bertujuan untuk memastikan *Node Mcu* bisa berjalan dengan baik. *Node Mcu* digunakan untuk menerima data json dari Arduino uno dan mengirimkan data tersebut ke firebase.

### 1.3.2. Alat yang Digunakan

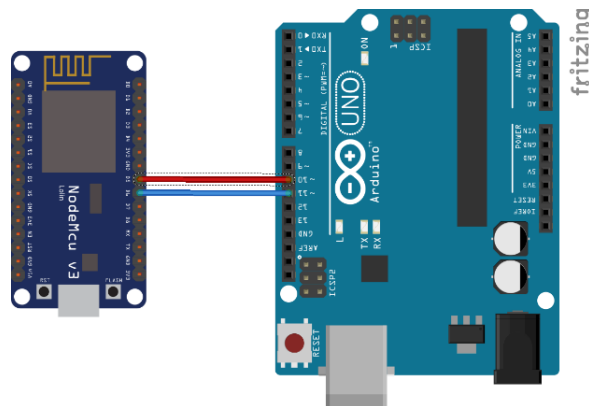
Alat yang digunakan yang digunakan untuk pengujian *Node Mcu* adalah:

1. Arduino Uno R3
2. Kabel USB Arduino
3. Kabel USB
4. PC/Laptop
5. *Software* Arduino IDE
6. *Jumper Wires*
7. *Node Mcu*

### 1.3.3. Langkah Pengujian

Langkah-langkah pengujian sistem adalah sebagai berikut :

1. Merangkai *Node Mcu* dengan *microcontroller* Arduino Uno seperti dibawah ini :



Gambar 4.3 Rangkaian Pengujian *NodeMcu*

2. Setelah merangkai *Node MCU*
3. Sebelumnya pada Arduino Uno sudah diupload program sensor ultrasonic yang telah dimodifikasi menggunakan library Arduinojson, codenya sebagai berikut:

```
sketch_jan26a $
#include <NewPing.h>

#include <SoftwareSerial.h>
#include <ArduinoJson.h>
SoftwareSerial s(10,11);

NewPing sonar1(3, 2, 11);
NewPing sonar2(5, 4, 17);
NewPing sonar3(7, 6, 17);

int jarakAtas = 11;
int jarakSamping = 17;
int jarakDepan = 17;

int tinggi, panjang, lebar, volume, beratdimensi;

void setup() {
  s.begin(9600);
  Serial.begin(9600);
  Serial.print(s);
}

void loop() {
  StaticJsonBuffer<1000> jsonBuffer;
  JsonObject& root = jsonBuffer.createObject();
  delay(1000);

  panjang = jarakDepan - sonar2.ping_cm();
  root["panjang"] = panjang;
  delay(1000);
  lebar = jarakSamping - sonar3.ping_cm();
  root["lebar"] = lebar;
  delay(1000);

  tinggi = jarakAtas - sonar1.ping_cm();
  root["tinggi"] = tinggi;
  delay(1000);
  volume = panjang * lebar * tinggi;
  root["volume"] = volume;
  delay(1000);
  beratdimensi = volume / 4;
  root["beratdimensi"] = beratdimensi;
  delay(1000);
  root.printTo(s);
  delay(1000);
}
```

Note: kode diatas data Panjang, lebar, tinggi, dll. Diubah formatnya menjadi json lalu dikirimkan melalui serial sesuai pin yang digunakan pada Arduino ke node mcu.

4. Kembali pada Nodemcu kita install *library* ESP8266Wifi, FirebaseArduino, ArduinoJson, dan SoftwareSerial pada *software* Arduino IDE agar *Node MCU* dapat dijalankan.



```
sketch_jan26a $
#include <ESP8266WiFi.h>
#include <FirebaseArduino.h>
#include <ArduinoJson.h>
#include <SoftwareSerial.h>
```

5. Kemudian kita melakukan pendefinisian yang dibutuhkan untuk melakukan koneksi ke firebase

```
#define FIREBASE_HOST "proyek3-9c81e.firebaseio.com"
#define FIREBASE_AUTH "TOuGgQ4K4zvviJSKalkJVzR8fk2c1cTyP7RdWm4vw"
#define WIFI_SSID "A1601"
#define WIFI_PASSWORD "123450989"
```

6. Membuat konstruktor untuk library SoftwareSerial agar dapat menerima data yang dikirimkan dari Arduino Uno

```
SoftwareSerial s (D6, D5);
```

7. Menambahkan code pada metode setup ().

```
void setup() {
  // Initialize Serial port
  Serial.begin(9600);
  s.begin(9600);
  while (!Serial) continue;

  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("connecting");
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(500);
  }
  Serial.println();
  Serial.print("connected: ");
  Serial.println(WiFi.localIP());

  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
}
```

8. Menambahkan code pada metode loop().

```
void loop() {  
    StaticJsonBuffer<1000> jsonBuffer;  
    JsonObject& root = jsonBuffer.parseObject(s);  
    if (root == JsonObject::invalid())  
        return;  
  
    Serial.println("JSON received and parsed");  
    root.prettyPrintTo(Serial);  
    delay(1000);  
  
    int volume = root["volume"];  
  
    if (volume > 0){  
        unsigned long currentMillis = millis(); // grab current ti  
        if ((unsigned long) (currentMillis - previousMillis) >= i  
            if(kirimdata){  
                String name = Firebase.push("/Pengukuran", root);  
                // handle error  
                if (Firebase.failed()){  
                    Serial.print("Sent failed");  
                    Serial.println(Firebase.error());  
                    return;  
                }else{  
                    Serial.print(name);  
                    Serial.println("Berhasil di kirim");  
                    delay(100);  
                }  
                kirimdata = false;  
            }  
            previousMillis = millis();  
        }else{  
            Serial.println(currentMillis);  
  
            Serial.println(previousMillis);  
        }  
    }else{  
        kirimdata = true;  
        if(kirimdata){  
            }  
            Serial.println("Waktu Persiapan");  
            Serial.println("Tidak ada barang");  
            delay(2000);  
        }  
    }  
}
```

9. Jangan lupa buat inisiasi variable dibawah konstruktor library  
SoftwareSerial

```

    unsigned long interval=60000; // the time we need to wait
    unsigned long previousMillis=0; // millis() returns the current time
    boolean kirimdata = true;
}

```

10. Lalu compile dan upload

11. Full code sebagai berikut:

```

#include <ESP8266WiFi.h>
#include <FirebaseArduino.h>
#include <ArduinoJson.h>
#include <SoftwareSerial.h>

#define FIREBASE_HOST "proyek3-9c81e.firebaseio.com"
#define FIREBASE_AUTH "TOuGgQ4K4zviJSKa1kJVzR8fk2c1cTyP7RdWm4vw"
#define WIFI_SSID "A1601"
#define WIFI_PASSWORD "123450989"

SoftwareSerial s(D6,D5);

unsigned long interval=60000; // the time we need to wait
unsigned long previousMillis=0; // millis() returns an unsigned long
boolean kirimdata = true;

void setup() {
    // Initialize Serial port
    Serial.begin(9600);
    s.begin(9600);
    while (!Serial) continue;

    WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
    Serial.print("connecting");
    while (WiFi.status() != WL_CONNECTED) {
        Serial.print(".");
        delay(500);
    }
    Serial.println();
    Serial.print("connected: ");
    Serial.println(WiFi.localIP());
}

```

```

    Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
}

void loop() {
    StaticJsonBuffer<1000> jsonBuffer;
    JsonObject& root = jsonBuffer.parseObject(s);
    if (root == JsonObject::invalid())
        return;

    Serial.println("JSON received and parsed");
    root.prettyPrintTo(Serial);
    delay(1000);

    int volume = root["volume"];

    if (volume > 0){
        unsigned long currentMillis = millis(); // grab current ti
        if ((unsigned long)(currentMillis - previousMillis) >= i
            if(kirimdata){
                String name = Firebase.push("/Pengukuran", root);
                // handle error
                if (Firebase.failed()){
                    Serial.print("Sent failed");
                    Serial.println(Firebase.error());
                    return;
                }else{
                    Serial.print(name);
                    Serial.println("Berhasil di kirim");
                    delay(100);
                }
                kirimdata = false;
            }

            previousMillis = millis();
        }else{
            Serial.println(currentMillis);
            Serial.println(previousMillis);
        }
    }else{
        kirimdata = true;
        if(kirimdata){
        }
        Serial.println("Waktu Persiapan");
        Serial.println("Tidak ada barang");
        delay(2000);
    }
}

```

Codingan ini menjelaskan bahwa *Node MCU* membaca data json yang dikirimkan melalui serial oleh Arduino kemudian nodemcu melakukan connect ke wifi apabila sudah terkoneksi maka *Node MCU* akan mengirimkan data json tersebut ke firebase.

#### 1.3.4. Hasil Pengujian

Dari pengujian *Node MCU* yang telah dilakukan, membuktikan bahwa *Node MCU* dapat berfungsi dengan baik. Maka *Node MCU* dapat membaca data json dan mengirimkan data tersebut ke firebase.

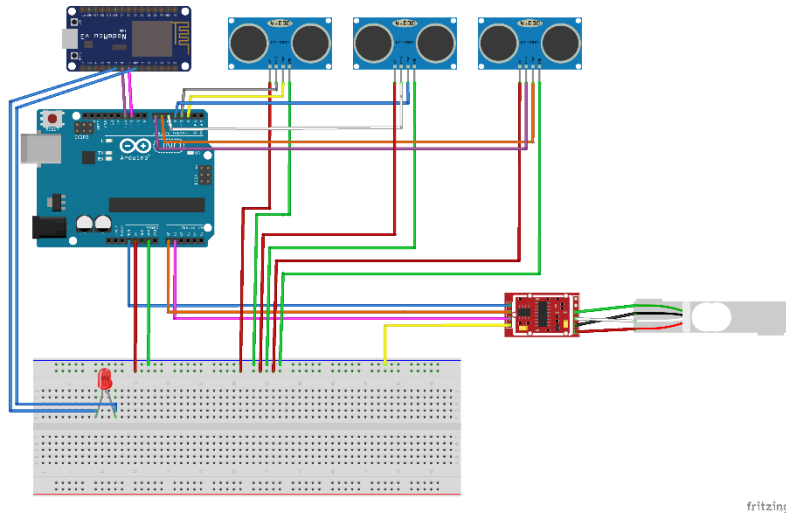
2. Setelah melakukan pengujian pada setiap alat yang kita akan gunakan untuk memabangun prototype ini, selajutnya kita akan menggabungkan alat yang di atas semua menjadi satu dan ditambahkan komponen LED sebagai lampu indicator.

##### 2.1. Alat yang digunakan

1. Leptop
2. *Software* Arduino UNO
3. Kabel USB Type A/B
4. *Jumper Wires*
5. *HC-SR04 Ultrasonic Sensor*
6. LED
7. *Node MCU*
8. *HX711*
9. *Loadcell*

##### 2.2. Langkah-langkah Membangun Prototype

1. Merangkai Arduino Uno, Sensor Ultrasonic, *HX711* dan *Load Cell*, *Node MCU* dan LED seperti gambar berikut :



2. Setelah kita merangkai prototype di atas, selanjutnya kita mengisi program pada Arduino Uno, yang dimana akan mendapatkan nilai panjang, tinggi, lebar, volume, berat dimensi, dan berat yang di dapatkan dari load cell dan mengirimkan data ke nodemcu melalui serial berupa data json, yang terlebih dahulu kita mengimport library yang digunakan, sebagai berikut:

```

Arduino_part
#include <HX711.h>
#include <NewPing.h>
#include <SoftwareSerial.h>
#include <ArduinoJson.h>

```

3. Membuat konstruktor library SoftwareSerial.

```

SoftwareSerial s(10,11);

```

4. Membuat Konstruktor library NewPing.

```

NewPing sonar1(3, 2, 20); //tinggi
NewPing sonar2(5, 4, 30); //panjang
NewPing sonar3(7, 6, 20); //lebar

```

5. Mendefinisikan pin yang terhubung pada module HX711

```

#define DOUT 12 //mendefinisikan pin arduino yang terhubung dengan pin DT module HX711
#define CLK 13 //mendefinisikan pin arduino yang terhubung dengan pin SCK module HX711

```

## 6. Kemudian kita membuat Konstruktor library HX711

```
HX711 scale(DOUT, CLK);
```

## 7. Selanjutnya kita melakukan inisiasi variable yang akan digunakan nanti

```
float calibration_factor = 471.70; //nilai kalibrasi awal
int berat;

int jarakAtas = 20;
int jarakDepan = 30;
int jarakSamping = 20;

int tinggi, panjang, lebar, volume, beratdimensi;
```

## 8. Menambah code pada metode setup().

```
void setup() {
  s.begin(9600);
  Serial.begin(9600);
  Serial.print(s);
  scale.set_scale(calibration_factor);
  scale.tare();
}
```

## 9. Menambah code pada metode loop().

```
void loop() {
  StaticJsonBuffer<1000> jsonBuffer;
  JsonObject& root = jsonBuffer.createObject();
  delay(1000);

  panjang = jarakDepan - sonar2.ping_cm();
  root["panjang"] = panjang;
  delay(1000);
  lebar = jarakSamping - sonar3.ping_cm();
  root["lebar"] = lebar;
  delay(1000);
  tinggi = jarakAtas - sonar1.ping_cm();
  root["tinggi"] = tinggi;
  delay(1000);
  volume = panjang * lebar * tinggi;
  root["volume"] = volume;
  delay(1000);
  beratdimensi = volume / 4;
  root["beratdimensi"] = beratdimensi;
  delay(1000);

  berat = scale.get_units(1);
  if (berat < 0)
  {
    berat = 0.00;
  }
  root["berat"] = berat;
  root.printTo(s);
}
```

10. Lalu compile dan upload ke board Arduino Uno menggunakan port yang terdeteksi

Full code :

```
Arduino_part$
#include <HX711.h>
#include <NewPing.h>
#include <SoftwareSerial.h>
#include <ArduinoJson.h>

SoftwareSerial s(10,11);

NewPing sonar1(3, 2, 20); //tinggi
NewPing sonar2(5, 4, 30); //panjang
NewPing sonar3(7, 6, 20); //lebar

#define DOUT 12 //mendefinisikan pin arduino yang terhubung dengan pin DT module HX711
#define CLK 13 //mendefinisikan pin arduino yang terhubung dengan pin SCK module HX711

HX711 scale(DOUT, CLK);

float calibration_factor = 471.70; //nilai kalibrasi awal
int berat;

int jarakAtas = 20;
int jarakDepan = 30;
int jarakSamping = 20;

int tinggi, panjang, lebar, volume, beratdimensi;

void setup() {
  s.begin(9600);
  Serial.begin(9600);
  Serial.print(s);
  scale.set_scale(calibration_factor);
  scale.tare();
}

void loop() {
  StaticJsonBuffer<1000> jsonBuffer;
  JsonObject& root = jsonBuffer.createObject();
  delay(1000);

  panjang = jarakDepan - sonar2.ping_cm();
  root["panjang"] = panjang;
  delay(1000);
  lebar = jarakSamping - sonar3.ping_cm();
  root["lebar"] = lebar;
  delay(1000);
  tinggi = jarakAtas - sonar1.ping_cm();
  root["tinggi"] = tinggi;
  delay(1000);
  volume = panjang * lebar * tinggi;
  root["volume"] = volume;
  delay(1000);
  beratdimensi = volume / 4;
  root["beratdimensi"] = beratdimensi;
  delay(1000);

  berat = scale.get_units(1);
  if (berat < 0)
  {
    berat = 0.00;
  }
  root["berat"] = berat;
  root.printTo(s);
}
```



11. Setelah mengupload program Arduino Uno, selanjutnya kita akan membuat program yang nanti akan di upload pada board NodeMCU, yang dimana program ini dapat menerima data json yang dikirimkan dari Arduino Uno dan dapat terkoneksi dengan internet dan firebase sehingga bisa meneruskan data json tersebut ke firebase. Pertama seperti biasa kita mengimport library yang kita butuhkan, seperti berikut :

```
NodeMcu_part
#include <ESP8266WiFi.h>
#include <FirebaseArduino.h>
#include <ArduinoJson.h>
#include <SoftwareSerial.h>
```

12. Selanjutnya kita melakukan pendefinisian yang akan digunakan untuk melakukan koneksi ke firebase dan internet

```
#define FIREBASE_HOST "proyek3-9c81e.firebaseio.com"
#define FIREBASE_AUTH "TOuGgQ4K4zvviJSKa1kJVzR8fk2c1cTyP7RdWm4vw"
#define WIFI_SSID "A1601"
#define WIFI_PASSWORD "123450989"
```

13. Membuat konstruktor library SoftwareSerial, yang digunakan untuk berkomunikasi dengan Arduino Uno

```
SoftwareSerial s(D6,D5);
```

14. Membuat inisiasi variable yang dibutuhkan

```
unsigned long interval=60000; // the time we need to wait
unsigned long previousMillis=0; // millis() returns an unsigned
boolean kirimdata = true;
int ledPin = 13;
```

15. Menambah code pada metode setup();

```

void setup() {
  // Initialize Serial port
  Serial.begin(9600);
  s.begin(9600);
  while (!Serial) continue;

  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("connecting");
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(500);
  }
  Serial.println();
  Serial.print("connected: ");
  Serial.println(WiFi.localIP());

  pinMode(ledPin, OUTPUT);
  digitalWrite(ledPin, HIGH);

  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
}

```

16. Menambah code pada metode loop().

```

void loop() {
    StaticJsonBuffer<1000> jsonBuffer;
    JsonObject& root = jsonBuffer.parseObject(s);
    if (root == JsonObject::invalid())
        return;

    Serial.println("JSON received and parsed");
    root.prettyPrintTo(Serial);
    delay(1000);

    int volume = root["volume"];

    if (volume > 0) {
        unsigned long currentMillis = millis(); // grab current time
        if ((unsigned long)(currentMillis - previousMillis) >= interval) {
            if(kirimdata) {
                String name = Firebase.push("/Pengukuran", root);
                // handle error
                if (Firebase.failed()){
                    Serial.print("Sent failed");
                    Serial.println(Firebase.error());
                    ledgagalkirim();
                    return;
                }else{
                    Serial.print(name);
                    Serial.println("Berhasil di kirim");
                    digitalWrite(ledPin,LOW);
                    delay(100);
                    digitalWrite(ledPin,HIGH);
                    delay(100);
                }
                kirimdata = false;
                digitalWrite(ledPin,LOW);
            }
            previousMillis = millis();
        }else{
            Serial.println(currentMillis);
            Serial.println(previousMillis);
        }
    }else{
        kirimdata = true;
        if(kirimdata) {
            digitalWrite(ledPin,HIGH);
        }
        Serial.println("Waktu Persiapan");
        Serial.println("Tidak ada barang");
        delay(2000);
    }
}

```

17. Pada metode loop() terdapat metode ledgagalkirim(), oleh karna itu kita harus membuat metode tersebut, sebagai berikut :

```
void ledgagalkirim(){
  digitalWrite(ledPin,LOW);
  delay(100);
  digitalWrite(ledPin,HIGH);
  delay(100);
  digitalWrite(ledPin,LOW);
  delay(100);
  digitalWrite(ledPin,HIGH);
  delay(100);
  digitalWrite(ledPin,LOW);
  delay(100);
  digitalWrite(ledPin,HIGH);
  delay(100);
  digitalWrite(ledPin,LOW);
  delay(100);
  digitalWrite(ledPin,HIGH);
  delay(100);
  digitalWrite(ledPin,LOW);
  delay(100);
  digitalWrite(ledPin,HIGH);
  delay(100);
  digitalWrite(ledPin,LOW);
  delay(100);
  digitalWrite(ledPin,HIGH);
  delay(100);
  digitalWrite(ledPin,LOW);
  delay(700);
  digitalWrite(ledPin,HIGH);
}
```

18. Lalu compile dan upload ke board NodeMCU menggunakan port yang terdeteksi
3. Selanjutnya kita membuat Hardware dari prototype yang dimana digunakan sebagai wadah atau tempat yang digunakan.