

BAGIAN I

PENGUKURAN BERAT DIMENSI

1.1. Definisi

Pengukuran berat dimensi barang merupakan proses dimana barang diletakkan di tempat 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 bebrabsis *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.

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



Gambar 2.1 Arduino Uno

Spesifikasi

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

Pemrograman

Pemrograman Arduino Uno dapat diprogram dengan (Arduino Software (IDE)). Pilih "Arduino / Genuino Uno dari menu Tools> Board (sesuai dengan mikrokontroler pada board Anda). ATmega328 pada

Arduino Uno telah diprogram sebelumnya dengan bootloader yang memungkinkan Anda mengunggah kode baru ke tanpa menggunakan programmer perangkat keras eksternal. Ia berkomunikasi menggunakan protokol STK500 asli (referensi, file header C). Anda juga dapat membypass bootloader dan memprogram mikrokontroler melalui header ICSP (In-Circuit Serial Programming) menggunakan Arduino ISP atau serupa.

Daya

Board Arduino Uno dapat diaktifkan melalui koneksi USB atau dengan catu daya eksternal. Sumber daya dipilih secara otomatis.

Daya eksternal (non-USB) dapat berasal dari adaptor AC-ke-DC (kutil dinding) atau baterai. Adaptor dapat dihubungkan dengan menancapkan colokan 2.1mm pusat-positif ke colokan listrik board. Petunjuk dari baterai dapat dimasukkan ke dalam header pin GND dan Vin dari konektor POWER.

Papan dapat beroperasi dengan suplai eksternal dari 6 hingga 20 volt. Namun, jika disediakan dengan kurang dari 7V, pin 5V dapat memasok kurang dari lima volt dan papan mungkin menjadi tidak stabil. Jika menggunakan lebih dari 12V, pengatur tegangan bisa menjadi terlalu panas dan merusak papan. Kisaran yang disarankan adalah 7 hingga 12 volt.

Power pin adalah sebagai berikut:

- Vin. Tegangan input ke papan Arduino / Genuino ketika menggunakan sumber daya eksternal (berbeda dengan 5 volt dari koneksi USB atau sumber daya yang diatur lainnya). Anda dapat memasok tegangan melalui pin ini, atau, jika memasok tegangan melalui soket daya, akseslah melalui pin ini.
- 5V. Pin ini menghasilkan 5V yang diatur dari regulator di papan. Papan dapat disuplai dengan daya baik dari colokan listrik DC (7 - 12V), konektor USB (5V), atau pin VIN papan (7-12V). Memasok tegangan melalui pin 5V atau 3.3V melewati regulator, dan dapat merusak papan Anda. Kami tidak menyarankannya.
- 3V3. Suplai 3,3 volt dihasilkan oleh regulator terpasang. Penarikan maksimum saat ini adalah 50 mA.
- GND. Pin ground.

- IOREF. Pin ini pada papan Arduino / Genuino memberikan referensi tegangan yang digunakan mikrokontroler. Perisai yang dikonfigurasi dengan benar dapat membaca tegangan pin IOREF dan memilih sumber daya yang sesuai atau mengaktifkan penerjemah tegangan pada output untuk bekerja dengan 5V atau 3.3V.

Memori

ATmega328 memiliki 32 KB (dengan 0,5 KB ditempati oleh bootloader). Ini juga memiliki 2 KB SRAM dan 1 KB EEPROM (yang dapat dibaca dan ditulis dengan perpustakaan EEPROM).

Input dan Output

Masing-masing dari 14 pin digital pada Uno dapat digunakan sebagai input atau output, menggunakan fungsi pinMode (), digitalWrite (), dan digitalRead (). Mereka beroperasi pada 5 volt. Setiap pin dapat menyediakan atau menerima 20 mA sebagai kondisi operasi yang disarankan dan memiliki resistor pull-up internal (terputus secara default) 20-50k ohm. Maksimum 40mA adalah nilai yang tidak boleh dilampaui pada pin I / O apa pun untuk menghindari kerusakan permanen pada mikrokontroler.

Beberapa pin memiliki fungsi khusus:

- Serial: 0 (RX) dan 1 (TX). Digunakan untuk menerima (RX) dan mengirimkan (TX) data serial TTL. Pin ini terhubung ke pin yang sesuai dari chip Serial ATmega8U2 USB-to-TTL.
- Interupsi Eksternal: 2 dan 3. Pin ini dapat dikonfigurasi untuk memicu interupsi pada nilai rendah, naik atau turun, atau perubahan nilai. Lihat fungsi attachInterrupt () untuk detailnya.
- PWM: 3, 5, 6, 9, 10, dan 11. Menyediakan output PWM 8-bit dengan fungsi analogWrite () .
- SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Pin ini mendukung komunikasi SPI menggunakan perpustakaan SPI.
- LED: 13. Ada LED internal yang digerakkan oleh pin digital 13. Ketika pin bernilai TINGGI, LED menyala, ketika pin RENDAH, mati.
- TWI: pin A4 atau SDA dan pin A5 atau SCL. Mendukung komunikasi TWI menggunakan perpustakaan Wire.

Uno memiliki 6 input analog, berlabel A0 hingga A5, yang masing-masing memberikan 10 bit resolusi (yaitu 1024 nilai yang berbeda). Secara default mereka mengukur dari ground ke 5 volt, meskipun apakah mungkin untuk mengubah ujung atas rentang mereka menggunakan pin AREF dan fungsi analogReference (). Ada beberapa pin lain di board:

- AREF. Tegangan referensi untuk input analog. Digunakan dengan analogReference () .
- Reset. Untuk mengatur ulang mikrokontroler. Biasanya digunakan untuk menambahkan tombol reset ke perisai yang memblokir yang ada di board.

Komunikasi

Arduino / Genuino Uno memiliki sejumlah fasilitas untuk berkomunikasi dengan komputer, papan Arduino / Genuino lain, atau mikrokontroler lainnya. ATmega328 menyediakan komunikasi serial UART TTL (5V), yang tersedia pada pin digital 0 (RX) dan 1 (TX). ATmega16U2 pada board menyalurkan komunikasi serial ini melalui USB dan muncul sebagai port com virtual untuk perangkat lunak di komputer. Firmware 16U2 menggunakan driver USB COM standar, dan tidak diperlukan driver eksternal. Namun, pada Windows, file .inf diperlukan. Perangkat Lunak Arduino (IDE) termasuk monitor serial yang memungkinkan data tekstual sederhana untuk dikirim ke dan dari papan. LED RX dan TX di papan tulis akan berkedip ketika data sedang dikirim melalui chip USB-ke-serial dan koneksi USB ke komputer (tetapi tidak untuk komunikasi serial pada pin 0 dan 1).

Pustaka SoftwareSerial memungkinkan komunikasi serial pada salah satu pin digital Uno. ATmega328 juga mendukung komunikasi I2C (TWI) dan SPI.

Perangkat Lunak Arduino (IDE) mencakup perpustakaan Wire untuk menyederhanakan penggunaan bus I2C; lihat dokumentasi untuk detailnya. Untuk komunikasi SPI, gunakan perpustakaan SPI.

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

Microcontroller	ATmega32u4
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pin	20
PWM Pin	7
Analog Input Pin	12
Arus DC per Pin I/O	40 mA
Arus DC untuk 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328P) dimana 4 KB digunakan oleh bootloader
SRAM	2,5 KB (ATmega32u4)
EEPROM	1 KB (ATmega32u4)
Clock Speed	16 MHz
LED_BUILTIN	13
Panjang	68,6 mm
Lebar	53,3 mm
Berat	20 g

Pemrograman

Leonardo dapat diprogram dengan perangkat lunak Arduino (unduh). Pilih "Arduino Leonardo" dari menu Tools > Board (sesuai dengan mikrokontroler di board Anda). ATmega32U4 pada Arduino Leonardo hadir dengan bootloader yang memungkinkan Anda mengunggah kode baru tanpa penggunaan pemrogram perangkat keras eksternal. Alat ini berkomunikasi menggunakan protokol AVR109. Anda juga dapat mem-bypass bootloader dan memprogram mikrokontroler melalui header ICSP (In-Circuit Serial Programming) menggunakan Arduino ISP atau serupa.

Daya

Arduino Leonardo dapat diaktifkan melalui koneksi micro USB atau dengan catu daya eksternal. Sumber daya dipilih secara otomatis. Daya eksternal (non-USB) dapat berasal dari adaptor AC-ke-DC (kutil dinding) atau baterai. Adaptor dapat dihubungkan dengan menancapkan colokan 2.1mm pusat-positif ke colokan listrik board. Lead dari baterai dapat dimasukkan ke header pin Gnd dan Vin pada konektor POWER.

Power pin adalah sebagai berikut:

- VIN. Tegangan input ke papan Arduino saat menggunakan sumber daya eksternal (berbeda dengan 5 volt dari koneksi USB atau sumber daya yang diatur lainnya). Anda dapat memasok tegangan melalui pin ini, atau, jika memasok tegangan melalui soket daya, akseslah melalui pin ini.
- 5V. Catu daya yang diatur digunakan untuk menyalaikan mikrokontroler dan komponen lainnya di papan tulis. Ini dapat berasal dari VIN melalui regulator terpasang, atau dipasok oleh USB atau pasokan 5V yang diatur lainnya.
- 3V3. Suplai 3,3 volt dihasilkan oleh regulator terpasang. Penarikan maksimum saat ini adalah 50 mA.
- GND. Pin tanah.
- IOREF. Tegangan di mana i / o pin papan beroperasi (yaitu VCC untuk papan). Ini adalah 5V pada Leonardo.

Memori

ATmega32u4 memiliki 32 KB (dengan 4 KB digunakan untuk bootloader). Ini juga memiliki 2,5 KB SRAM dan 1 KB EEPROM (yang dapat dibaca dan ditulis dengan perpustakaan EEPROM).

Input dan Output

Masing-masing dari 20 pin i / o digital pada Leonardo dapat digunakan sebagai input atau output, menggunakan fungsi PinMode (), digitalWrite (), dan digitalRead (). Mereka beroperasi pada 5 volt. Setiap pin dapat memberikan atau menerima maksimum 40 mA dan memiliki resistor pull-up internal (terputus secara default) 20-50 kOhms. Selain itu, beberapa pin memiliki fungsi khusus:

- Serial: 0 (RX) dan 1 (TX). Digunakan untuk menerima (RX) dan mengirimkan (TX) data serial TTL menggunakan kemampuan serial hardware ATmega32U4. Perhatikan bahwa pada Leonardo, kelas Serial mengacu pada komunikasi USB (CDC); untuk serial TTL pada pin 0 dan 1, gunakan kelas Serial1.
- TWI: 2 (SDA) dan 3 (SCL). Mendukung komunikasi TWI menggunakan perpustakaan Wire.
- Interupsi Eksternal: 3 (interupsi 0), 2 (interupsi 1), 0 (interupsi 2), 1 (interupsi 3) dan 7 (interupsi 4). Pin ini dapat dikonfigurasi untuk memicu interupsi pada nilai rendah, naik atau turun, atau perubahan nilai. Lihat fungsi attachInterrupt () untuk detailnya.
- PWM: 3, 5, 6, 9, 10, 11, dan 13. Menyediakan output PWM 8-bit dengan fungsi analogWrite () .
- SPI: pada header ICSP. Pin ini mendukung komunikasi SPI menggunakan perpustakaan SPI. Perhatikan bahwa pin SPI tidak terhubung ke salah satu pin I / O digital karena berada di Uno, mereka hanya tersedia pada konektor ICSP. Ini berarti bahwa jika Anda memiliki perisai yang menggunakan SPI, tetapi TIDAK memiliki konektor ICSP 6-pin yang terhubung ke header ICSP 6-pin milik Leonardo, perisai itu tidak akan berfungsi.
- LED: 13. Ada LED internal yang terhubung ke pin digital 13. Ketika pin bernilai TINGGI, LED menyala, ketika pin RENDAH, mati.

- Input Analog: A0-A5, A6 - A11 (pada pin digital 4, 6, 8, 9, 10, dan 12). Leonardo memiliki 12 input analog, berlabel A0 hingga A11, yang semuanya dapat juga digunakan sebagai digital i / o. Pin A0-A5 muncul di lokasi yang sama dengan di Uno; input A6-A11 masing-masing pada pin i / o digital 4, 6, 8, 9, 10, dan 12. Setiap input analog memberikan resolusi 10 bit (mis. 1024 nilai yang berbeda). Secara default input analog mengukur dari ground ke 5 volt, meskipun apakah mungkin untuk mengubah ujung atas rentang mereka menggunakan pin AREF dan fungsi analogReference () .

Ada beberapa pin lain di board:

- AREF. Tegangan referensi untuk input analog. Digunakan dengan analogReference () .
- Reset. Untuk mengatur ulang mikrokontroler. Biasanya digunakan untuk menambahkan tombol reset ke perisai yang memblokir yang ada di board.

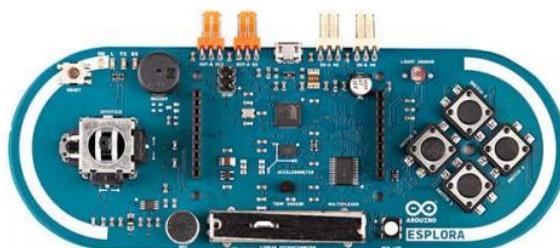
Komunikasi

Leonardo memiliki sejumlah fasilitas untuk berkomunikasi dengan komputer, Arduino lain, atau mikrokontroler lainnya. ATmega32U4 menyediakan komunikasi serial UART TTL (5V), yang tersedia pada pin digital 0 (RX) dan 1 (TX). 32U4 juga memungkinkan untuk komunikasi serial (CDC) melalui USB dan muncul sebagai port com virtual ke perangkat lunak di komputer. Chip ini juga bertindak sebagai perangkat USB 2.0 kecepatan penuh, menggunakan driver USB COM standar. Di Windows, file .inf diperlukan. Perangkat lunak Arduino termasuk monitor serial yang memungkinkan data tekstual sederhana untuk dikirim ke dan dari papan Arduino. RX dan TX LED di papan akan berkedip ketika data sedang dikirim melalui koneksi USB ke komputer (tetapi tidak untuk komunikasi serial pada pin 0 dan 1). Pustaka SoftwareSerial memungkinkan komunikasi serial pada salah satu pin digital Leonardo. ATmega32U4 juga mendukung komunikasi I2C (TWI) dan SPI. Perangkat lunak Arduino mencakup perpustakaan Wire untuk menyederhanakan penggunaan bus I2C; lihat dokumentasi untuk detailnya. Untuk komunikasi SPI, gunakan perpustakaan SPI. Leonardo muncul sebagai keyboard dan mouse generik, dan dapat

diprogram untuk mengontrol perangkat input ini menggunakan kelas Keyboard dan Mouse.

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

Microcontroller	ATmega32u4
Operating Voltage	5V
Flash Memory	32 KB (ATmega328P) dimana 4 KB digunakan oleh bootloader
SRAM	2,5 KB (ATmega32u4)
EEPROM	1 KB (ATmega32u4)
Clock Speed	16 MHz
LED_BUILTIN	13
Panjang	164,04 mm

Lebar	60 mm
Berat	53 gr

Pemrograman

Esplora dapat diprogram dengan perangkat lunak Arduino (unduh). Pilih "Arduino Esplora" dari menu Tools> Board. ATmega32U4 pada Arduino Esplora hadir dengan bootloader yang memungkinkan Anda untuk mengunggah kode baru ke dalamnya tanpa menggunakan programmer perangkat keras eksternal. Ini berkomunikasi menggunakan protokol AVR109. Anda juga dapat mem-bypass bootloader dan memprogram mikrokontroler melalui header ICSP (In-Circuit Serial Programming); lihat instruksi ini untuk detailnya.

Memori

ATmega32u4 memiliki 32 KB (dengan 4 KB digunakan untuk bootloader). Ini juga memiliki 2,5 KB SRAM dan 1 KB EEPROM (yang dapat dibaca dan ditulis dengan perpustakaan EEPROM).

Input dan Output

Desain board Esplora mengingatkan desain gamepad tradisional dengan joystick analog di sebelah kiri dan empat tombol tekan di sebelah kanan. Esplora memiliki input dan output on-board berikut:

- Joystick analog dengan tombol tekan tengah dua sumbu (X dan Y) dan tombol tekan tengah.
- 4 tombol yang diletakkan dalam pola berlian.
- Slider potensiometer linier di dekat bagian bawah papan.
- Mikrofon untuk mendapatkan kenyaringan (amplitudo) dari lingkungan sekitarnya.
- Sensor cahaya untuk mendapatkan kecerahan.
- Sensor suhu membaca suhu sekitar
- Accelerometer tiga sumbu mengukur hubungan papan terhadap gravitasi pada tiga sumbu (X, Y, dan Z)
- Buzzer dapat menghasilkan gelombang persegi.
- RGB memimpin LED terang dengan elemen Merah Hijau dan Biru untuk pencampuran warna.
- 2 Input TinkerKit untuk menghubungkan modul sensor TinkerKit dengan konektor 3-pin.

- 2 TinkerKit Output untuk menghubungkan modul aktuator TinkerKit dengan konektor 3-pin.
- Konektor konektor layar TFT untuk layar LCD warna opsional, kartu SD, atau perangkat lain yang menggunakan protokol SPI.

Untuk memanfaatkan jumlah total sensor yang tersedia, papan menggunakan multiplexer analog. Ini berarti input analog tunggal dari mikrokontroler dibagi di antara semua saluran input (kecuali accelerometer 3-sumbu). Empat pin mikrokontroler tambahan memilih saluran mana yang akan dibaca

Komunikasi

Leonardo Esplora memiliki sejumlah fasilitas untuk berkomunikasi dengan komputer, Arduino lain, atau mikrokontroler lainnya. ATmega32U4 menyediakan komunikasi serial (CDC) melalui USB dan muncul sebagai port com virtual untuk perangkat lunak di komputer. Chip ini juga bertindak sebagai perangkat USB 2.0 kecepatan penuh, menggunakan driver USB COM standar. Di Windows, file .inf diperlukan. Perangkat lunak Arduino termasuk monitor serial yang memungkinkan data teks sederhana untuk dikirim ke dan dari papan Arduino. RX dan TX LED di papan tulis akan berkedip ketika data sedang dikirim melalui koneksi USB ke komputer.

ATmega32U4 juga mendukung komunikasi SPI, yang dapat diakses melalui pustaka SPI.

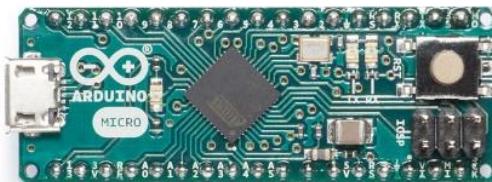
Esplora dapat muncul sebagai keyboard dan mouse generik, dan dapat diprogram untuk mengontrol perangkat input ini menggunakan pustaka Keyboard dan Mouse

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

Microcontroller	ATmega32u4
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pin	20
PWM Pin	7
Analog Input Pin	12
Arus DC per Pin I/O	40 mA
Arus DC untuk 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328P) dimana 4 KB digunakan oleh bootloader
SRAM	2,5 KB (ATmega32u4)
EEPROM	1 KB (ATmega32u4)
Clock Speed	16 MHz
LED_BUILTIN	13
Panjang	68,6 mm
Lebar	53,3 mm
Berat	20 g

Pemrograman

Micro board dapat diprogram dengan Arduino Software (IDE). Pilih "Arduino / Genuino Micro dari menu Tools> Board.

ATmega32U4 pada Micro telah diprogram sebelumnya dengan bootloader yang memungkinkan Anda mengunggah kode baru tanpa menggunakan programmer perangkat keras eksternal. Ini berkomunikasi menggunakan protokol AVR109. Anda juga dapat mem-bypass bootloader dan memprogram mikrokontroler melalui header ICSP (In-Circuit Serial Programming) menggunakan Arduino ISP atau yang serupa.

Daya

Mikro dapat diaktifkan melalui koneksi USB mikro atau dengan catu daya eksternal. Sumber daya dipilih secara otomatis.

Daya eksternal (non-USB) dapat berasal dari catu daya DC atau baterai. Petunjuk dari baterai atau catu daya DC dapat dihubungkan ke pin Gnd dan Vin.

Papan dapat beroperasi dengan suplai eksternal 6 hingga 20 volt. Namun, jika disediakan dengan kurang dari 7V, pin 5V dapat memasok kurang dari lima volt dan papan mungkin menjadi tidak stabil. Jika menggunakan lebih dari 12V, pengatur tegangan bisa menjadi terlalu panas dan merusak papan. Kisaran yang disarankan adalah 7 hingga 12 volt.

Power pin adalah sebagai berikut:

- VI. Tegangan input ke papan MICRO saat menggunakan sumber daya eksternal (berbeda dengan 5 volt dari koneksi USB atau sumber daya yang diatur lainnya). Anda dapat memasok tegangan melalui pin ini.
- 5V. Catu daya yang diatur digunakan untuk menyalaikan mikrokontroler dan komponen lainnya di papan tulis. Ini dapat berasal dari VIN melalui regulator terpasang, atau dipasok oleh USB atau pasokan 5V yang diatur lainnya.
- 3V. Suplai 3,3 volt dihasilkan oleh regulator terpasang. Penarikan maksimum saat ini adalah 50 mA.
- GND. Pin ground

Memori

ATmega32U4 memiliki 32 KB (dengan 4 KB digunakan untuk bootloader). Ini juga memiliki 2,5 KB SRAM dan 1 KB EEPROM (yang dapat dibaca dan ditulis dengan perpustakaan EEPROM).

Input dan Output

Setiap 20 pin i / o digital pada Micro dapat digunakan sebagai input atau output, menggunakan fungsi pinMode (), digitalWrite (), dan digitalRead (). Mereka beroperasi pada 5 volt. Setiap pin dapat menyediakan atau menerima 20 mA sebagai kondisi operasi yang disarankan dan memiliki resistor pull-up internal (terputus secara default) 20-50 k ohm. Maksimum 40mA adalah nilai yang tidak boleh dilampaui untuk menghindari kerusakan permanen pada mikrokontroler.

Selain itu, beberapa pin memiliki fungsi khusus:

- Serial: 0 (RX) dan 1 (TX). Digunakan untuk menerima (RX) dan mengirimkan (TX) data serial TTL menggunakan kemampuan serial perangkat keras ATmega32U4. Perhatikan bahwa pada Mikro, kelas Serial mengacu pada komunikasi USB (CDC); untuk serial TTL pada pin 0 dan 1, gunakan kelas Serial1.
- TWI: 2 (SDA) dan 3 (SCL). Mendukung komunikasi TWI menggunakan perpustakaan Wire.
- Interupsi Eksternal: 0 (RX), 1 (TX), 2, 3 dan 7. Pin ini dapat dikonfigurasikan untuk memicu interupsi pada nilai rendah, tepi naik atau turun, atau perubahan nilai. Lihat fungsi attachInterrupt () untuk detailnya.
- PWM: 3, 5, 6, 9, 10, 11 dan 13. Menyediakan output PWM 8-bit dengan fungsi analogWrite () .
- SPI: pada header ICSP. Pin ini mendukung komunikasi SPI menggunakan perpustakaan SPI. Perhatikan bahwa pin SPI tidak terhubung ke salah satu pin I / O digital karena berada di Uno, mereka hanya tersedia pada konektor ICSP dan pada pin terdekat berlabel MISO, MOSI dan SCK.
- RX_LED / SS Ini adalah pin tambahan dibandingkan dengan Leonardo. Terhubung ke RX_LED yang menunjukkan aktivitas transmisi selama komunikasi USB, tetapi juga dapat digunakan sebagai slave select pin (SS) dalam komunikasi SPI.
- LED: 13. Ada LED internal yang terhubung ke pin digital 13. Ketika pin bernilai TINGGI, LED menyala, ketika pin RENDAH, mati.

- Input Analog: A0-A5, A6 - A11 (pada pin digital 4, 6, 8, 9, 10, dan 12). Mikro memiliki total 12 input analog, pin dari A0 ke A5 diberi label langsung pada pin dan yang lain yang dapat Anda akses dalam kode menggunakan konstanta dari A6 melalui A11 dibagi masing-masing pada pin digital 4, 6, 8, 9, 10, dan 12. Kesemuanya dapat juga digunakan sebagai I / O digital. Setiap input analog memberikan resolusi 10 bit (mis. 1024 nilai yang berbeda). Secara default input analog mengukur dari ground ke 5 volt, meskipun apakah mungkin untuk mengubah ujung atas rentang mereka menggunakan pin AREF dan fungsi analogReference () .

Ada beberapa pin lain di board:

- AREF. Tegangan referensi untuk input analog. Digunakan dengan analogReference () .
- Setel ulang. Bawa baris ini RENDAH untuk mengatur ulang mikrokontroler. Biasanya digunakan untuk menambahkan tombol reset ke perisai yang memblokir yang ada di papan tulis

Komunikasi

Mikro memiliki sejumlah fasilitas untuk berkomunikasi dengan komputer, dewan lain dari keluarga Arduino & Genuino, atau mikrokontroler lainnya. 32U4 menyediakan komunikasi serial UART TTL (5V), yang tersedia pada pin digital 0 (RX) dan 1 (TX). ATmega32U4 juga memungkinkan untuk komunikasi serial (CDC) melalui USB dan muncul sebagai port com virtual ke perangkat lunak di komputer. Chip ini juga bertindak sebagai perangkat USB 2.0 kecepatan penuh, menggunakan driver USB COM standar. Di Windows, file .inf diperlukan. Perangkat Lunak Arduino (IDE) termasuk monitor serial yang memungkinkan data teksual sederhana untuk dikirim ke dan dari papan. RX dan TX LED di papan akan berkedip ketika data sedang dikirim melalui koneksi USB ke komputer (tetapi tidak untuk komunikasi serial pada pin 0 dan 1).

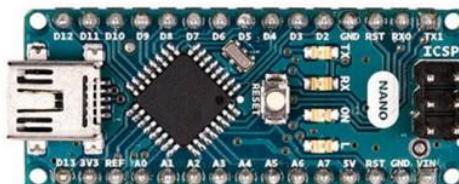
Pustaka SoftwareSerial memungkinkan komunikasi serial pada pin digital Micro lainnya.

ATmega32U4 juga mendukung komunikasi I2C (TWI) dan SPI. Perangkat Lunak Arduino (IDE) mencakup perpustakaan Wire untuk menyederhanakan penggunaan bus I2C; lihat dokumentasi untuk detailnya. Untuk komunikasi SPI, gunakan perpustakaan SPI.

Mikro muncul sebagai keyboard dan mouse generik, dan dapat diprogram untuk mengontrol perangkat input ini menggunakan kelas Keyboard dan Mouse.

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

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

Lebar	18 mm
Berat	7 g

Pemrograman

Arduino Nano dapat diprogram dengan perangkat lunak Arduino (unduh). Pilih "Arduino Duemilanove atau Nano w / ATmega328" dari menu Tools> Board (sesuai dengan mikrokontroler pada board Anda). ATmega328 pada Arduino Nano hadir dengan bootloader yang memungkinkan Anda untuk mengunggah kode baru ke dalamnya tanpa menggunakan programmer perangkat keras eksternal. Ini berkomunikasi menggunakan protokol STK500 asli. Anda juga dapat mem-bypass bootloader dan memprogram mikrokontroler melalui header ICSP (In-Circuit Serial Programming) menggunakan Arduino ISP atau yang serupa.

Daya

Arduino Nano dapat diaktifkan melalui koneksi Mini-B USB, catu daya eksternal 6-20V yang tidak diatur (pin 30), atau catu daya eksternal yang diatur 5V (pin 27). Sumber daya dipilih secara otomatis ke sumber tegangan tertinggi.

Memori

ATmega328 memiliki 32 KB, (juga dengan 2 KB digunakan untuk bootloader. Untuk ATmega328 memiliki 2 KB SRAM dan 1 KB EEPROM.

Input dan Output

Masing-masing dari 14 pin digital pada Nano dapat digunakan sebagai input atau output, menggunakan fungsi pinMode (), digitalWrite (), dan digitalRead (). Mereka beroperasi pada 5 volt. Setiap pin dapat memberikan atau menerima maksimum 40 mA dan memiliki resistor pull-up internal (terputus secara default) 20-50 kOhms. Selain itu, beberapa pin memiliki fungsi khusus:

- Serial: 0 (RX) dan 1 (TX). Digunakan untuk menerima (RX) dan mengirimkan (TX) data serial TTL. Pin ini terhubung ke pin yang sesuai dari chip Serial FTDI USB-to-TTL.

- Interupsi Eksternal: 2 dan 3. Pin ini dapat dikonfigurasi untuk memicu interupsi pada nilai rendah, naik atau turun, atau perubahan nilai. Lihat fungsi attachInterrupt () untuk detailnya.
- PWM: 3, 5, 6, 9, 10, dan 11. Menyediakan output PWM 8-bit dengan fungsi analogWrite () .
- SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Pin ini mendukung komunikasi SPI, yang, meskipun disediakan oleh perangkat keras yang mendasarinya, saat ini tidak termasuk dalam bahasa Arduino.
- LED: 13. Ada LED bawaan yang terhubung ke pin digital 13. Ketika pin bernilai TINGGI, LED menyala, ketika pin RENDAH, mati.

Nano memiliki 8 input analog, yang masing-masing memberikan 10 bit resolusi (yaitu 1024 nilai yang berbeda). Secara default mereka mengukur dari ground ke 5 volt, meskipun apakah mungkin untuk mengubah ujung atas rentang mereka menggunakan fungsi analogReference (). Pin analog 6 dan 7 tidak dapat digunakan sebagai pin digital. Selain itu, beberapa pin memiliki fungsi khusus:

- I2C: A4 (SDA) dan A5 (SCL). Mendukung komunikasi I2C (TWI) menggunakan perpustakaan Wire (dokumentasi di situs web Wiring).

Ada beberapa pin lain di papan tulis:

- AREF. Tegangan referensi untuk input analog. Digunakan dengan analogReference () .
- Setel ulang. Bawa baris ini RENDAH untuk mengatur ulang mikrokontroler. Biasanya digunakan untuk menambahkan tombol reset ke perisai yang memblokir yang ada di papan tulis.

Komunikasi

Arduino Nano memiliki sejumlah fasilitas untuk berkomunikasi dengan komputer, Arduino lain, atau mikrokontroler lainnya. ATmega328 menyediakan komunikasi serial UART TTL (5V), yang tersedia pada pin digital 0 (RX) dan 1 (TX). FTDI FT232RL pada papan saluran komunikasi serial ini melalui USB dan driver FTDI (termasuk dengan perangkat lunak Arduino) menyediakan port virtual untuk perangkat lunak pada komputer. Perangkat lunak Arduino

termasuk monitor serial yang memungkinkan data tekstual sederhana untuk dikirim ke dan dari papan Arduino. RX dan TX LED di papan akan berkedip ketika data sedang dikirim melalui chip FTDI dan koneksi USB ke komputer (tetapi tidak untuk komunikasi serial pada pin 0 dan 1). Pustaka SoftwareSerial memungkinkan komunikasi serial pada salah satu pin digital Nano. ATmega328 juga mendukung komunikasi I2C (TWI) dan SPI. Perangkat lunak Arduino mencakup perpustakaan Wire untuk menyederhanakan penggunaan bus I2C. Untuk menggunakan komunikasi SPI, silakan lihat lembar data ATmega328.

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

Pemrograman

Board Mega 2560 dapat diprogram dengan Arduino Software (IDE). Untuk detailnya, lihat referensi dan tutorialnya. ATmega2560 pada Mega 2560 telah diprogram sebelumnya dengan bootloader yang memungkinkan Anda untuk mengunggah kode baru ke dalamnya tanpa menggunakan programmer perangkat keras eksternal. Ini berkomunikasi menggunakan protokol STK500 asli (referensi, file header C). Anda juga dapat mem-bypass bootloader dan memprogram mikrokontroler melalui header ICSP (In-Circuit Serial Programming) menggunakan Arduino ISP atau yang serupa.

Daya

Mega 2560 dapat diaktifkan melalui koneksi USB atau dengan catu daya eksternal. Sumber daya dipilih secara otomatis. Daya eksternal (non-USB) dapat berasal dari adaptor AC-ke-DC (kutil dinding) atau baterai. Adaptor dapat dihubungkan dengan menancapkan colokan 2.1mm pusat-positif ke colokan listrik board.

Petunjuk dari baterai dapat dimasukkan ke dalam header pin GND dan Vin dari konektor POWER.

Board dapat beroperasi dengan suplai eksternal 6 hingga 20 volt. Namun, jika disediakan dengan kurang dari 7V, pin 5V dapat memasok kurang dari lima volt dan papan mungkin menjadi tidak stabil. Jika menggunakan lebih dari 12V, pengatur tegangan bisa menjadi terlalu panas dan merusak papan. Kisaran yang disarankan adalah 7 hingga 12 volt.

Power pin adalah sebagai berikut:

- Vin. Tegangan input ke papan ketika menggunakan sumber daya eksternal (sebagai lawan 5 volt dari koneksi USB atau sumber daya yang diatur lainnya). Anda dapat memasok tegangan melalui pin ini, atau, jika memasok tegangan melalui soket daya, akseslah melalui pin ini.
- 5V. Pin ini menghasilkan 5V yang diatur dari regulator di papan tulis. Papan dapat disuplai dengan daya baik dari colokan listrik DC (7 - 12V), konektor USB (5V), atau pin VIN papan (7-12V). Memasok tegangan melalui pin 5V atau 3.3V melewati regulator, dan dapat merusak papan Anda. Kami tidak menyarankannya.
- 3V3. Suplai 3,3 volt dihasilkan oleh regulator terpasang. Penarikan maksimum saat ini adalah 50 mA.
- GND. Pin tanah.
- IOREF. Pin ini di papan menyediakan referensi tegangan yang digunakan mikrokontroler. Perisai yang dikonfigurasi dengan benar dapat membaca tegangan pin IOREF dan memilih sumber daya yang sesuai atau mengaktifkan penerjemah tegangan pada output untuk bekerja dengan 5V atau 3.3V.

Memori

ATmega2560 memiliki memori flash 256 KB untuk menyimpan kode (8 MB digunakan untuk bootloader), 8 KB SRAM dan 4 KB EEPROM (yang dapat dibaca dan ditulis dengan perpustakaan EEPROM)

Input dan Output

Masing-masing dari 54 pin digital pada Mega dapat digunakan sebagai input atau output, menggunakan fungsi pinMode (), digitalWrite (), dan digitalRead (). Mereka beroperasi pada 5 volt.

Setiap pin dapat menyediakan atau menerima 20 mA sebagai kondisi operasi yang disarankan dan memiliki resistor pull-up internal (terputus secara default) 20-50 k ohm. Maksimum 40mA adalah nilai yang tidak boleh dilampaui untuk menghindari kerusakan permanen pada mikrokontroler.

Selain itu, beberapa pin memiliki fungsi khusus:

- Serial: 0 (RX) dan 1 (TX); Serial 1: 19 (RX) dan 18 (TX); Serial 2: 17 (RX) dan 16 (TX); Serial 3: 15 (RX) dan 14 (TX). Digunakan untuk menerima (RX) dan mengirimkan (TX) data serial TTL. Pin 0 dan 1 juga terhubung ke pin yang sesuai dari chip Serial USB-to-TTL ATmega16U2.
- Interupsi Eksternal: 2 (interupsi 0), 3 (interupsi 1), 18 (interupsi 5), 19 (interupsi 4), 20 (interupsi 3), dan 21 (interupsi 2). Pin ini dapat dikonfigurasi untuk memicu interupsi pada level rendah, naik atau turun, atau perubahan level. Lihat fungsi attachInterrupt () untuk detailnya.
- PWM: 2 hingga 13 dan 44 hingga 46. Menyediakan output PWM 8-bit dengan fungsi analogWrite ().
- SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS). Pin ini mendukung komunikasi SPI menggunakan pustaka SPI. Pin SPI juga pecah pada header ICSP, yang secara fisik kompatibel dengan Arduino / Genuino Uno dan papan Duemilanove dan Diecimila Arduino yang lama.
- LED: 13. Ada LED internal yang terhubung ke pin digital 13. Ketika pin bernilai TINGGI, LED menyala, ketika pin RENDAH, mati.
- TWI: 20 (SDA) dan 21 (SCL). Mendukung komunikasi TWI menggunakan perpustakaan Wire. Perhatikan bahwa pin ini tidak berada di lokasi yang sama dengan pin TWI pada papan Duemilanove atau Diecimila Arduino yang lama.

Mega 2560 memiliki 16 input analog, yang masing-masing menyediakan 10 bit resolusi (yaitu 1024 nilai yang berbeda). Secara default mereka mengukur dari ground ke 5 volt, meskipun apakah mungkin untuk mengubah ujung atas rentang mereka menggunakan pin AREF dan fungsi analogReference () .

Ada beberapa pin lain di papan tulis:

- AREF. Tegangan referensi untuk input analog. Digunakan dengan analogReference () .
- Reset. Untuk mengatur ulang mikrokontroler. Biasanya digunakan untuk menambahkan tombol reset ke perisai yang memblokir yang ada di papan tulis.

Komunikasi

Papan Mega 2560 memiliki sejumlah fasilitas untuk berkomunikasi dengan komputer, papan lain, atau mikrokontroler lainnya. ATmega2560 menyediakan empat UART perangkat keras untuk komunikasi serial TTL (5V). ATmega16U2 (ATmega 8U2 pada revisi 1 dan revisi 2 papan) pada board menyalurkan salah satunya melalui USB dan menyediakan port com virtual untuk perangkat lunak di komputer (mesin Windows akan memerlukan file .inf, tetapi mesin OSX dan Linux akan mengenali papan sebagai port COM secara otomatis. Perangkat Lunak Arduino (IDE) termasuk monitor serial yang memungkinkan data tekstual sederhana untuk dikirim ke dan dari papan. LED RX dan TX di papan akan berkedip ketika data sedang dikirim melalui ATmega8U2 / ATmega16U2 chip dan koneksi USB ke komputer (tetapi tidak untuk komunikasi serial pada pin 0 dan 1).

Pustaka SoftwareSerial memungkinkan komunikasi serial pada salah satu pin digital Mega 2560.

Mega 2560 juga mendukung komunikasi TWI dan SPI. Arduino Software (IDE) termasuk perpustakaan Wire untuk menyederhanakan penggunaan bus TWI, lihat dokumentasi untuk detailnya. Untuk komunikasi SPI, gunakan perpustakaan SPI.

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

Spesifikasi

Microcontroller	AT91SAM3X8E
Operating Voltage	3,3V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-26V
Digital I/O Pin	54 (12 pin input dapat digunakan sebagai output PWM)
Analog Input Pin	12
Analog Output Pin	2 (DAC)
Total DC Output pada semua jalur I / O	130 mA
Arus DC per Pin 5V	800 mA
Arus DC untuk 3.3V Pin	800 mA
Flash Memory	512 KB
SRAM	96 KB (64 KB dan 32 KB)
EEPROM	84 KB
Clock Speed	84 MHz
Panjang	101,52 mm
Lebar	53,3 mm
Berat	36 g

Pemrograman

Karena ini dapat diprogram dengan Arduino Arduino Software (IDE). Mengunggah sketsa ke SAM3X berbeda dari mikrokontroler AVR yang ditemukan di papan Arduino lain karena memori flash perlu

dihapus sebelum diprogram ulang. Upload ke chip dikelola oleh ROM di SAM3X, yang dijalankan hanya ketika memori flash chip kosong. Salah satu dari port USB dapat digunakan untuk memprogram board, meskipun disarankan untuk menggunakan port Programming karena cara menghapus chip ditangani:

- Port pemrograman: Untuk menggunakan port ini, pilih "Arduino Due (ProgrammingPort)" sebagai papan Anda di Arduino IDE. Hubungkan port pemrograman Due (yang terdekat dengan colokan listrik DC) ke komputer Anda. Port pemrograman menggunakan 16U2 sebagai chip USB-ke-serial yang terhubung ke UART pertama SAM3X (RX0 dan TX0). 16U2 memiliki dua pin yang terhubung ke pin Reset dan Erase dari SAM3X. Membuka dan menutup port Programming yang terhubung pada 1200bps memicu prosedur "hapus keras" dari chip SAM3X, mengaktifkan pin Erase dan Reset pada SAM3X sebelum berkomunikasi dengan UART. Ini adalah port yang direkomendasikan untuk pemrograman Due. Ini lebih dapat diandalkan daripada "penghapusan lunak" yang terjadi pada port Asli, dan itu akan berfungsi bahkan jika MCU utama telah crash.
- Port asli: Untuk menggunakan port ini, pilih "Arduino Due (NativeUSBPort)" sebagai papan Anda di Arduino IDE. Port USB asli terhubung langsung ke SAM3X. Hubungkan port USB Asli Due (yang paling dekat dengan tombol reset) ke komputer Anda. Membuka dan menutup port Native pada 1200bps memicu prosedur 'hapus lunak': memori flash dihapus dan papan restart dengan bootloader. Jika MCU mogok karena alasan tertentu, kemungkinan prosedur penghapusan lunak tidak akan berfungsi karena prosedur ini sepenuhnya terjadi pada perangkat lunak pada SAM3X. Membuka dan menutup port asli pada baudrate yang berbeda tidak akan mengatur ulang SAM3X.

Daya

Arduino Due dapat diaktifkan melalui konektor USB atau dengan catu daya eksternal. Sumber daya dipilih secara otomatis. Daya eksternal (non-USB) dapat berasal dari adaptor AC-ke-DC (kutil dinding) atau baterai. Adaptor dapat dihubungkan dengan menancapkan colokan 2.1mm pusat-positif ke colokan listrik board.

Lead dari baterai dapat dimasukkan ke header pin Gnd dan Vin pada konektor POWER.

Board dapat beroperasi dengan suplai eksternal 6 hingga 20 volt. Namun, jika disediakan dengan kurang dari 7V, pin 5V mungkin memasok kurang dari lima volt dan papan mungkin tidak stabil. Jika menggunakan lebih dari 12V, pengatur tegangan bisa menjadi terlalu panas dan merusak papan. Kisaran yang disarankan adalah 7 hingga 12 volt.

Power pin adalah sebagai berikut:

- Vin. Tegangan input ke papan Arduino ketika menggunakan sumber daya eksternal (berbeda dengan 5 volt dari koneksi USB atau sumber daya yang diatur lainnya). Anda dapat memasok tegangan melalui pin ini, atau jika memasok tegangan melalui colokan listrik, akseslah melalui pin ini.
- 5V. Pin ini menghasilkan 5V yang diatur dari regulator di papan. Papan dapat disuplai dengan daya baik dari colokan listrik DC (7 - 12V), konektor USB (5V), atau pin VIN papan (7-12V). Memasok tegangan melalui pin 5V atau 3.3V melewati regulator, dan dapat merusak papan Anda. Kami tidak menyarankannya.
- 3V3. Suplai 3,3 volt dihasilkan oleh regulator terpasang. Penarikan maksimum saat ini adalah 800 mA. Regulator ini juga menyediakan catu daya ke mikrokontroler SAM3X.
- GND. Pin ground.
- IOREF. Pin ini pada papan Arduino memberikan referensi tegangan yang digunakan mikrokontroler. Perisai yang dikonfigurasi dengan benar dapat membaca tegangan pin IOREF dan memilih sumber daya yang sesuai atau mengaktifkan penerjemah tegangan pada output untuk bekerja dengan 5V atau 3.3V.

Memori

SAM3X memiliki memori flash 512 KB (2 blok 256 KB) untuk menyimpan kode. Bootloader di-preburn di pabrik dari Atmel dan disimpan dalam memori ROM khusus. SRAM yang tersedia adalah 96 KB di dua bank yang bersebelahan yaitu 64 KB dan 32 KB. Semua memori yang tersedia (Flash, RAM dan ROM) dapat diakses secara langsung sebagai ruang pengalamatan datar.

Dimungkinkan untuk menghapus memori Flash SAM3X dengan tombol hapus onboard. Ini akan menghapus sketsa yang saat ini dimuat dari MCU. Untuk menghapus, tekan dan tahan tombol Erase selama beberapa detik saat papan dinyalakan.

Input dan Output

- I / O Digital: pin dari 0 hingga 53
- Masing-masing dari 54 pin digital pada Due dapat digunakan sebagai input atau output, menggunakan fungsi pinMode (), digitalWrite (), dan digitalRead (). Mereka beroperasi pada 3,3 volt. Setiap pin dapat memberikan (sumber) arus 3 mA atau 15 mA, tergantung pada pin, atau menerima (tenggelam) arus 6 mA atau 9 mA, tergantung pada pin. Mereka juga memiliki resistor pull-up internal (terputus secara default) 100 KOhm. Selain itu, beberapa pin memiliki fungsi khusus:
 - Serial: 0 (RX) dan 1 (TX)
 - Serial 1: 19 (RX) dan 18 (TX)
 - Serial 2: 17 (RX) dan 16 (TX)
 - Serial 3: 15 (RX) dan 14 (TX) Digunakan untuk menerima (RX) dan mengirimkan (TX) data serial TTL (dengan level 3,3 V). Pin 0 dan 1 terhubung ke pin yang sesuai dari chip Serial USB-to-TTL ATmega16U2.
 - PWM: Pin 2 hingga 13 Menyediakan output PWM 8-bit dengan fungsi analogWrite (). resolusi PWM dapat diubah dengan fungsi analogWriteResolution () .
 - SPI: SPI header (header ICSP pada papan Arduino lainnya) Pin ini mendukung komunikasi SPI menggunakan pustaka SPI. Pin SPI dipecah pada header 6-pin pusat, yang secara fisik kompatibel dengan Uno, Leonardo dan Mega2560. Header SPI hanya dapat digunakan untuk berkomunikasi dengan perangkat SPI lain, bukan untuk pemrograman SAM3X dengan teknik Pemrograman Sirkuit-Serial. SPI of the Due juga memiliki fitur-fitur canggih yang dapat digunakan dengan metode Extended SPI untuk Due.
 - CAN: CANRX dan CANTX Pin-pin ini mendukung protokol komunikasi CAN tetapi belum didukung oleh Arduino API.
 - LED "L": 13 Ada LED internal yang terhubung ke pin digital 13. Ketika pinnya TINGGI, LED menyala, ketika pinnya

RENDAH, mati. Dimungkinkan juga untuk meredupkan LED karena pin digital 13 juga merupakan output PWM.

- TWI 1: 20 (SDA) dan 21 (SCL)
- TWI 2: SDA1 dan SCL1. Mendukung komunikasi TWI menggunakan perpustakaan Wire. SDA1 dan SCL1 dapat dikontrol menggunakan kelas Wire1 yang disediakan oleh perpustakaan Wire. Sementara SDA dan SCL memiliki resistor pullup internal, SDA1 dan SCL1 tidak. Menambahkan dua resistor pullup pada garis SDA1 dan SCL1 diperlukan untuk menggunakan Wire1.
- Input Analog: pin dari A0 ke A11 Due memiliki 12 input analog, yang masing-masing dapat memberikan resolusi 12 bit (mis. 4096 nilai yang berbeda). Secara default, resolusi bacaan diatur pada 10 bit, untuk kompatibilitas dengan papan Arduino lainnya. Dimungkinkan untuk mengubah resolusi ADC dengan analogReadResolution (). Pin input analog Due mengukur dari ground ke nilai maksimum 3.3V. Menerapkan lebih dari 3.3V pada pin Due akan merusak chip SAM3X. Fungsi analogReference () diabaikan pada Due.

Pin AREF terhubung ke pin referensi analog SAM3X melalui jembatan resistor. Untuk menggunakan pin AREF, resistor BR1 harus disolder dari PCB.

- DAC1 dan DAC2 Pin-pin ini menyediakan output analog sejati dengan resolusi 12-bit (level 4096) dengan fungsi analogWrite (). Pin ini dapat digunakan untuk membuat output audio menggunakan perpustakaan Audio.

Harap dicatat bahwa rentang output DAC sebenarnya dari 0,55 V hingga 2,75 V saja.

Pin lain di papan tulis:

- AREF Referensi tegangan untuk input analog. Digunakan dengan analogReference () .
- Reset. Untuk mengatur ulang mikrokontroler. Biasanya digunakan untuk menambahkan tombol reset ke perisai yang memblokir yang ada di papan tulis.

Komunikasi

Arduino Due memiliki sejumlah fasilitas untuk berkomunikasi dengan komputer, Arduino lain, atau mikrokontroler lainnya, dan

berbagai perangkat seperti ponsel, tablet, kamera, dan sebagainya. SAM3X menyediakan satu UART perangkat keras dan tiga USART perangkat keras untuk komunikasi serial TTL (3.3V).

Port Pemrograman terhubung ke ATmega16U2, yang menyediakan port COM virtual ke perangkat lunak pada komputer yang terhubung (Untuk mengenali perangkat, mesin Windows akan memerlukan file .inf, tetapi mesin OSX dan Linux akan mengenali board sebagai port COM secara otomatis). 16U2 juga terhubung ke UART perangkat keras SAM3X. Serial pada pin RX0 dan TX0 menyediakan komunikasi Serial-to-USB untuk pemrograman board melalui mikrokontroler ATmega16U2. Perangkat lunak Arduino termasuk monitor serial yang memungkinkan data tekstual sederhana untuk dikirim ke dan dari papan. LED RX dan TX di papan tulis akan berkedip ketika data sedang dikirim melalui chip ATmega16U2 dan koneksi USB ke komputer (tetapi tidak untuk komunikasi serial pada pin 0 dan 1).

Port USB asli terhubung ke SAM3X. Ini memungkinkan untuk komunikasi serial (CDC) melalui USB. Ini menyediakan koneksi serial ke Monitor Seri atau aplikasi lain di komputer Anda. Ini juga memungkinkan Due untuk mengemulasi mouse atau keyboard USB ke komputer yang terpasang. Untuk menggunakan fitur ini, lihat halaman referensi perpustakaan Mouse dan Keyboard.

Port USB asli juga dapat bertindak sebagai host USB untuk periferal yang terhubung seperti mouse, keyboard, dan smartphone. Untuk menggunakan fitur ini, lihat halaman referensi USBHost.

SAM3X juga mendukung komunikasi TWI dan SPI. Perangkat lunak Arduino termasuk perpustakaan Wire untuk menyederhanakan penggunaan bus TWI; lihat dokumentasi untuk detailnya. Untuk komunikasi SPI, gunakan perpustakaan SPI.

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

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

- Arduino Microprocessor

Microcontroller	Atheros AR9331
------------------------	----------------

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

Pemrograman

Yún dapat diprogram dengan perangkat lunak Arduino (unduh). Pilih "Arduino Yún" dari menu Tools> Board (sesuai dengan mikrokontroler pada board Anda). ATmega32U4 pada Arduino Yún hadir dengan bootloader yang memungkinkan Anda mengunggah kode baru ke dalamnya tanpa menggunakan programmer perangkat keras eksternal. Ini berkomunikasi menggunakan protokol AVR109.

Anda juga dapat mem-bypass bootloader dan memprogram mikrokontroler melalui header ICSP (In-Circuit Serial Programming) menggunakan Arduino ISP atau yang serupa;

Daya

Dianjurkan untuk memberi daya pada papan melalui koneksi micro-USB dengan 5VDC. Jika Anda memberi daya pada papan melalui pin Vin, Anda harus menyediakan 5VDC yang diatur. Tidak ada regulator tegangan terpasang untuk tegangan yang lebih tinggi, yang akan merusak papan.

Yún juga kompatibel dengan catu daya PoE tetapi untuk menggunakan fitur ini Anda harus memasang modul PoE di papan atau membeli papan yang sudah dipasang.

Power pin adalah sebagai berikut:

- VIN. Tegangan input ke papan Arduino. Tidak seperti papan Arduino lainnya, jika Anda akan memberikan daya ke papan melalui pin ini, Anda harus memberikan 5V yang diatur.

- 5V. Catu daya yang digunakan untuk memberi daya pada mikrokontroler dan komponen lainnya di papan tulis. Ini bisa berasal dari VIN atau disediakan oleh USB.
- 3V3. Suplai 3,3 volt dihasilkan oleh regulator terpasang. Penarikan maksimum saat ini adalah 50 mA.
- GND. Pin ground.
- IOREF. Tegangan di mana i / o pin papan beroperasi (yaitu VCC untuk papan). Ini adalah 5V pada Yún.

Memori

ATmega32u4 memiliki 32 KB (dengan 4 KB digunakan untuk bootloader). Ini juga memiliki 2,5 KB SRAM dan 1 KB EEPROM (yang dapat dibaca dan ditulis dengan perpustakaan EEPROM).

Memori pada AR9331 tidak tertanam di dalam prosesor. RAM dan memori penyimpanan terhubung secara eksternal. Yún memiliki 64 MB RAM DDR2 dan 16 MB memori flash. Memori flash dimuat di pabrik dengan distribusi Linux berbasis OpenWrt yang disebut Linino OS. Anda dapat mengubah konten gambar pabrik, seperti ketika Anda menginstal suatu program atau ketika Anda mengubah file konfigurasi. Anda dapat kembali ke konfigurasi pabrik dengan menekan tombol "WLAN RST" selama 30 detik.

Instalasi Linino OS menempati sekitar 9 MB dari 16 MB yang tersedia dari memori flash internal. Anda dapat menggunakan kartu micro SD jika Anda membutuhkan lebih banyak ruang disk untuk menginstal aplikasi.

Input dan Output

Tidak dimungkinkan untuk mengakses pin I / O dari Atheros AR9331. Semua jalur I / O terikat ke 32U4.

Masing-masing dari 20 pin i / o digital pada Yún dapat digunakan sebagai input atau output, menggunakan fungsi pinMode (), digitalWrite (), dan digitalRead (). Mereka beroperasi pada 5 volt. Setiap pin dapat memberikan atau menerima maksimum 40 mA dan memiliki resistor pull-up internal (terputus secara default) 20-50 kOhms. Selain itu, beberapa pin memiliki fungsi khusus: Serial: 0 (RX) dan 1 (TX). Digunakan untuk menerima (RX) dan mengirimkan (TX) data serial TTL menggunakan kemampuan serial perangkat keras ATmega32U4. Perhatikan bahwa pada Yún, kelas Serial mengacu

pada komunikasi USB (CDC); untuk serial TTL pada pin 0 dan 1, gunakan kelas Serial1.

Serial perangkat keras dari ATmega32U4 dan AR9331 pada Yún terhubung bersama dan digunakan untuk berkomunikasi antara kedua prosesor. Seperti biasa dalam sistem Linux, pada port serial AR9331 terbuka konsol untuk akses ke sistem, ini berarti bahwa Anda dapat mengakses ke program dan alat yang ditawarkan oleh Linux dari sketsa Anda.

- TWI: 2 (SDA) dan 3 (SCL). Mendukung komunikasi TWI menggunakan perpustakaan Wire.
- Interupsi Eksternal: 3 (interupsi 0), 2 (interupsi 1), 0 (interupsi 2), 1 (interupsi 3) dan 7 (interupsi 4). Pin ini dapat dikonfigurasi untuk memicu interupsi pada nilai rendah, naik atau turun, atau perubahan nilai. Lihat fungsi attachInterrupt () untuk detailnya. Tidak disarankan untuk menggunakan pin 0 dan 1 sebagai interupsi karena mereka juga merupakan port serial perangkat keras yang digunakan untuk berbicara dengan prosesor Linux. Pin 7 terhubung ke prosesor AR9331 dan dapat digunakan sebagai sinyal jabat tangan di masa depan. Disarankan untuk berhati-hati terhadap kemungkinan konflik jika Anda bermaksud menggunakannya sebagai interupsi.
- PWM: 3, 5, 6, 9, 10, 11, dan 13. Menyediakan output PWM 8-bit dengan fungsi analogWrite ().
- SPI: pada header ICSP. Pin ini mendukung komunikasi SPI menggunakan perpustakaan SPI. Perhatikan bahwa pin SPI tidak terhubung ke salah satu pin I / O digital karena berada di Uno, mereka hanya tersedia pada konektor ICSP. Ini berarti bahwa jika Anda memiliki perisai yang menggunakan SPI, tetapi TIDAK memiliki konektor ICSP 6-pin yang menghubungkan ke header ICSP 6-pin Yún, perisai itu tidak akan berfungsi. Pin SPI juga terhubung ke pin gpio AR9331, yang telah diimplementasikan dalam perangkat lunak antarmuka SPI. Ini berarti bahwa ATMega32u4 dan AR9331 juga dapat berkomunikasi menggunakan protokol SPI.
- LED: 13. Ada LED internal yang terhubung ke pin digital 13. Ketika pin bernilai TINGGI, LED menyala, ketika pin RENDAH, mati. Ada beberapa LED status lainnya pada Yún, mengindikasikan daya, koneksi WLAN, koneksi WAN, dan USB.

- Input Analog: A0 - A5, A6 - A11 (pada pin digital 4, 6, 8, 9, 10, dan 12). Yún memiliki 12 input analog, berlabel A0 hingga A11, yang semuanya juga dapat digunakan sebagai i / o digital. Pin A0-A5 muncul di lokasi yang sama dengan di Uno; input A6-A11 masing-masing pada pin i / o digital 4, 6, 8, 9, 10, dan 12. Setiap input analog memberikan resolusi 10 bit (mis. 1024 nilai yang berbeda). Secara default input analog mengukur dari ground ke 5 volt, meskipun apakah mungkin untuk mengubah ujung atas rentang mereka menggunakan pin AREF dan fungsi analogReference () .
- AREF. Tegangan referensi untuk input analog. Digunakan dengan analogReference () .

Ada 3 tombol reset dengan fungsi berbeda di board:

- YU RST. Bawa baris ini RENDAH untuk mengatur ulang mikroprosesor AR9331. Menyetel ulang AR9331 akan menyebabkan reboot sistem linux. Semua data yang disimpan dalam RAM akan hilang dan semua program yang sedang berjalan akan dihentikan.
- 32U4 RST. Bawa baris ini RENDAH untuk mengatur ulang mikrokontroler ATmega32U4. Biasanya digunakan untuk menambahkan tombol reset ke perisai yang memblokir yang ada di papan tulis.
- RST WLAN. Tombol ini memiliki fitur ganda. Primary berfungsi untuk mengembalikan WiFi ke konfigurasi pabrik. Konfigurasi pabrik terdiri untuk meletakkan WiFi Yún dalam mode titik akses (AP) dan menetapkannya ke alamat IP default yaitu 192.168.240.1, dalam kondisi ini Anda dapat terhubung dengan komputer Anda ke jaringan WiFi yang muncul dengan Nama SSID "Arduino Yun-XXXXXXXXXXXX", di mana kedua belas 'X' adalah alamat MAC dari Yún Anda. Setelah terhubung, Anda dapat mencapai panel web Yún dengan browser di alamat 192.168.240.1 atau "<http://arduino.local>". Perhatikan bahwa mengembalikan konfigurasi WiFi akan menyebabkan reboot lingkungan linux. Untuk mengembalikan konfigurasi WiFi Anda, Anda harus menekan dan menahan tombol WLAN RST selama 5 detik. Ketika Anda menekan tombol, LED biru WLAN akan mulai berkedip dan akan tetap berkedip ketika Anda melepaskan tombol setelah 5 detik yang mengindikasikan bahwa prosedur pemulihan WiFi telah

direkam. Fungsi kedua tombol WLAN RST adalah untuk mengembalikan citra linux ke gambar default pabrik. Untuk memulihkan lingkungan linux Anda harus menekan tombol selama 30 detik. Perhatikan bahwa mengembalikan gambar pabrik membuat Anda kehilangan semua file yang disimpan dan perangkat lunak yang diinstal pada memori flash terpasang yang terhubung ke AR9331.

Komunikasi

Yún memiliki sejumlah fasilitas untuk berkomunikasi dengan komputer, Arduino lain, atau mikrokontroler lainnya. ATmega32U4 menyediakan komunikasi serial UART TTL (5V) khusus. 32U4 juga memungkinkan untuk komunikasi serial (CDC) melalui USB dan muncul sebagai port com virtual ke perangkat lunak di komputer. Chip ini juga bertindak sebagai perangkat USB 2.0 kecepatan penuh, menggunakan driver USB COM standar. Perangkat lunak Arduino termasuk monitor serial yang memungkinkan data teksual sederhana untuk dikirim ke dan dari papan Arduino. RX dan TX LED di papan tulis akan berkedip ketika data sedang dikirim melalui koneksi USB ke komputer.

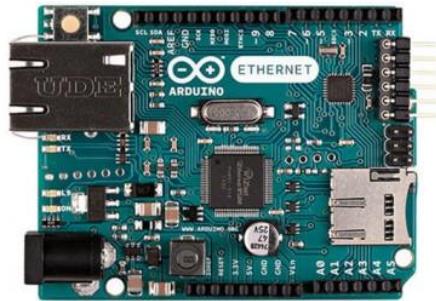
Pin digital 0 dan 1 digunakan untuk komunikasi serial antara 32U4 dan AR9331. Anda dapat menggunakan perpustakaan Ciao untuk komunikasi antara prosesor.

Pustaka SoftwareSerial memungkinkan komunikasi serial pada salah satu pin digital Yún. Pin 0 dan 1 harus dihindari karena digunakan oleh perpustakaan Bridge. ATmega32U4 juga mendukung komunikasi I2C (TWI) dan SPI. Perangkat lunak Arduino mencakup perpustakaan Wire untuk menyederhanakan penggunaan bus I2C ;. Untuk komunikasi SPI, gunakan perpustakaan SPI.

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

Microcontroller	ATmega328P
Operating Voltage	5V
Input Voltage Plug (recommended)	7-12V
Input Voltage Plug (limit)	6-20V
Input Voltage PoE (limit)	36-57V
Digital I/O Pin	14 (4 pin input dapat digunakan sebagai output PWM)

Arduino Pins reserved :

10 – 13	digunakan untuk SPI
4	digunakan untuk SD card
2	W5100 interrupt (saat dijembatani)

Analog Input Pin	6
Arus DC per Pin I/O	40 mA
Arus DC untuk 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328P) dimana 0,5 KB digunakan oleh bootloader
SRAM	2 KB (ATmega328P)

EEPROM	1 KB (ATmega328P)
Clock Speed	16 MHz
W5100 TCP/IP Embedded Ethernet Controller	
Power Over Ethernet ready Magnetic Jack	
Micro SD card, with active voltage translators	
Panjang	68,6 mm
Lebar	53,3 mm
Berat	28 gr

Pemrograman

Dimungkinkan untuk memprogram papan Arduino Ethernet dalam dua cara: melalui header pemrograman serial 6 pin, atau dengan programmer ISP eksternal. Header pemrograman serial 6-pin kompatibel dengan kabel USB FTDI dan papan USB-to-serial dasar gaya-ke-serial Sparkfun dan Adafruit, termasuk konektor USB-Serial Arduino. Ini fitur dukungan untuk reset otomatis, memungkinkan sketsa untuk diunggah tanpa menekan tombol reset di papan tulis. Ketika dicolokkan ke adaptor USB bergaya FTDI, Arduino Ethernet dimatikan dari adaptor.

Anda juga dapat mem-bypass bootloader dan memprogram mikrokontroler melalui header ICSP (In-Circuit Serial Programming) menggunakan Arduino ISP atau yang serupa; lihat instruksi ini untuk detailnya.

Daya

Board juga dapat diaktifkan melalui catu daya eksternal, modul Power over Ethernet (PoE) opsional, atau dengan menggunakan kabel FTDI / konektor Serial USB. Daya eksternal dapat berasal dari adaptor AC-ke-DC (kutil dinding) atau baterai. Adaptor dapat dihubungkan dengan menancapkan colokan 2.1mm pusat-positif ke colokan listrik board. Lead dari baterai dapat dimasukkan ke header pin Gnd dan Vin pada konektor POWER.

Papan dapat beroperasi dengan suplai eksternal 6 hingga 20 volt. Namun, jika disediakan dengan kurang dari 7V, pin 5V mungkin memasok kurang dari lima volt dan papan mungkin tidak stabil. Jika menggunakan lebih dari 12V, pengatur tegangan bisa menjadi terlalu panas dan merusak papan. Kisaran yang disarankan adalah 7 hingga 12 volt.

Power pin adalah sebagai berikut:

- VIN. Tegangan input ke papan Arduino ketika menggunakan sumber daya eksternal (berbeda dengan 5 volt dari koneksi USB atau sumber daya yang diatur lainnya). Anda dapat memasok tegangan melalui pin ini, atau, jika memasok tegangan melalui soket daya, akseslah melalui pin ini.
- 5V. Pin ini menghasilkan 5V yang diatur dari regulator di papan tulis. Papan dapat disuplai dengan daya baik dari colokan listrik DC (7 - 12V), konektor USB (5V), atau pin VIN papan (7-12V). Memasok tegangan melalui pin 5V atau 3.3V melewati regulator, dan dapat merusak papan Anda. Kami tidak menyarankannya.
- 3V3. Suplai 3,3 volt dihasilkan oleh regulator terpasang. Penarikan maksimum saat ini adalah 50 mA.
- GND. Pin tanah.
- IOREF. Pin ini pada papan Arduino memberikan referensi tegangan yang digunakan mikrokontroler. Perisai yang dikonfigurasi dengan benar dapat membaca tegangan pin IOREF dan memilih sumber daya yang sesuai atau mengaktifkan penerjemah tegangan pada output untuk bekerja dengan 5V atau 3.3V.

Modul PoE opsional dirancang untuk mengekstrak daya dari kabel Ethernet Kategori 5 twisted pair konvensional:

- Sesuai IEEE802.3af
- Riak dan derau output rendah (100mVpp)
- Rentang tegangan input 36V hingga 57V
- Overload dan perlindungan hubung singkat
- Output 9V
- Efisiensi tinggi DC / DC converter: ketik 75% @ 50% beban
- Isolasi 1500V (input ke output)

Memori

ATmega328 memiliki 32 KB (dengan 0,5 KB digunakan untuk bootloader). Ini juga memiliki 2 KB SRAM dan 1 KB EEPROM (yang dapat dibaca dan ditulis dengan perpustakaan EEPROM)

Input dan Output

Masing-masing dari 14 pin digital pada papan Ethernet dapat digunakan sebagai input atau output, menggunakan fungsi `pinMode()`, `digitalWrite()`, dan `digitalRead()`. Mereka beroperasi pada 5 volt. Setiap pin dapat memberikan atau menerima maksimum 40 mA dan memiliki resistor pull-up internal (terputus secara default) 20-50 kOhms. Selain itu, beberapa pin memiliki fungsi khusus:

- Serial: 0 (RX) dan 1 (TX). Digunakan untuk menerima (RX) dan mengirimkan (TX) data serial TTL.
- Interupsi Eksternal: 2 dan 3. Pin ini dapat dikonfigurasi untuk memicu interupsi pada nilai rendah, naik atau turun, atau perubahan nilai. Lihat fungsi `attachInterrupt()` untuk detailnya.
- PWM: 3, 5, 6, 9, dan 10. Menyediakan output PWM 8-bit dengan fungsi `analogWrite()`.
- SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Pin ini mendukung komunikasi SPI menggunakan pustaka SPI.
- LED: 9. Ada LED internal yang terhubung ke pin digital 9. Ketika pin bernilai TINGGI, LED menyala, ketika pin RENDAH, mati. Pada sebagian besar papan Arduino, LED ini ditemukan pada pin 13. Itu pada pin 9 pada papan Ethernet karena pin 13 digunakan sebagai bagian dari koneksi SPI.

Board Ethernet memiliki 6 input analog, berlabel A0 hingga A5, yang masing-masing memberikan 10 bit resolusi (yaitu 1024 nilai yang berbeda). Secara default mereka mengukur dari ground ke 5 volt, meskipun apakah mungkin untuk mengubah ujung atas rentang mereka menggunakan pin AREF dan fungsi `analogReference()`. Selain itu, beberapa pin memiliki fungsi khusus:

- TWI: A4 (SDA) dan A5 (SCL). Mendukung komunikasi TWI menggunakan perpustakaan Wire.

Ada beberapa pin lain di papan tulis:

- AREF. Tegangan referensi untuk input analog. Digunakan dengan `analogReference()`.
- Reset. Untuk mengatur ulang mikrokontroler. Biasanya digunakan untuk menambahkan tombol reset ke perisai yang memblokir yang ada di papan tulis.

Komunikasi

Arduino Ethernet memiliki sejumlah fasilitas untuk berkomunikasi dengan komputer, Arduino lain, atau mikrokontroler lainnya. Pustaka SoftwareSerial memungkinkan komunikasi serial pada salah satu pin digital Uno.

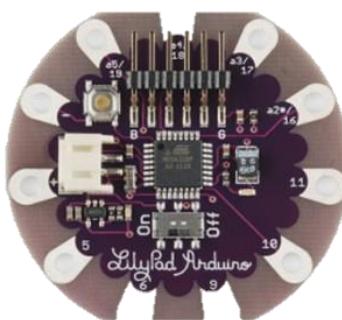
ATmega328 juga mendukung komunikasi TWI dan SPI. Perangkat lunak Arduino termasuk perpustakaan Wire untuk menyederhanakan penggunaan bus TWI; lihat dokumentasi untuk detailnya. Untuk komunikasi SPI, gunakan perpustakaan SPI.

Board juga dapat terhubung ke jaringan kabel melalui ethernet. Saat menghubungkan ke jaringan, Anda harus memberikan alamat IP dan alamat MAC. Perpustakaan Ethernet didukung sepenuhnya.

Pembaca kartu microSD onboard dapat diakses melalui Perpustakaan SD. Saat bekerja dengan perpustakaan ini, SS ada di Pin 4.

10. Arduino Lilypad Simple

Arduino Lilypad banyak digunakan pada proyek e-textiles dan wearables, karena 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

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

Pemrograman

LilyPad Simple dapat diprogram dengan Arduino Arduino Software (IDE). Pilih "LilyPad Arduino" dari menu Tools> Board. Untuk detailnya, lihat Panduan Memulai LilyPad Arduino. ATmega328P pada LilyPad Arduino hadir dengan bootloader yang memungkinkan Anda untuk mengunggah kode baru ke dalamnya dengan perangkat lunak Arduino.

LilyPad Simple tidak memiliki adaptor USB atau konektor USB terpasang. Untuk memprogram board, Anda perlu menggunakan adaptor yang kompatibel dengan FTDI seperti USBSerial Light Adapter.

Daya

LilyPad Arduino Simple dapat diaktifkan melalui catu daya eksternal atau adaptor yang kompatibel dengan FTDI seperti USBSerial Light Adapter.

Board dapat dinyalakan dan dimatikan dengan sakelar on-board. Saat papan dicabut dari adaptor FTDI dan ditenagai melalui baterai, sakelar itu menghidupkan dan mematikan papan; dengan sakelar pada posisi AKTIF, mikrokontroler menerima daya dan papan bekerja dan dengan sakelar pada posisi MATI, mikrokontroler tidak menerima daya. Saat papan diberi daya melalui adaptor FTDI, papan tetap menyala sepanjang waktu; dengan sakelar pada posisi AKTIF, mikrokontroler menerima daya dari baterai (atau papan FTDI melalui sirkuit pengisian baterai jika tidak ada baterai terpasang) dan dengan sakelar pada posisi MATI, mikrokontroler menerima daya dari adaptor FTDI.

Catu daya eksternal harus menyediakan antara 2,7 dan 5,5 volt. The Lilypad Simple dirancang dengan mempertimbangkan penggunaan baterai; Baterai 3,7 volt Lithium Polymer dapat dicolokkan langsung ke konektor JST on-board. Sekali lagi, jangan menyalakan LilyPad Arduino Simple dengan lebih dari 5,5 volt, atau colokkan daya ke belakang: Anda akan membunuhnya.

Board berisi chip pengisian baterai LiPo MCP73831. Jika board terhubung ke koneksi FTDI dan baterai, daya FTDI akan mengisi daya baterai. Ini benar terlepas dari posisi saklar. LED yang berdekatan dengan saklar menyala saat baterai sedang diisi. Pengisian daya akan berhenti secara otomatis ketika baterai terisi penuh. Karena sirkuit pengisian baterai, komponen daya tidak mungkin seperti modem bluetooth melalui konektor FTDI.

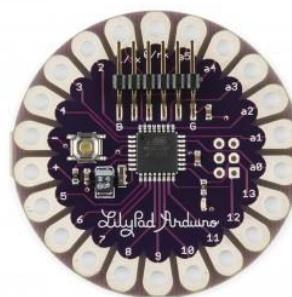
Input dan Output

LilyPad Simple memiliki input dan output yang lebih sedikit daripada Dewan Utama LilyPad Arduino. Ada total 9 pin I / O pada papan Sederhana, satu pin terbuka untuk + 3.3VDC, dan satu pin untuk ground. Masing-masing dari 9 pin I / O digital pada LilyPad Arduino Simple dapat digunakan sebagai input atau output, menggunakan fungsi pinMode (), digitalWrite (), dan digitalRead (). Mereka beroperasi pada volt 5V. Setiap pin dapat memberikan atau menerima maksimum 40 mA dan memiliki resistor pull-up internal (terputus secara default) sebesar 20 kOhms. Selain itu, beberapa pin memiliki fungsi khusus:

- PWM: 5, 6, 9, 10, 11 Menyediakan output PWM 8-bit dengan fungsi analogWrite () .
- Input Analog: A2-A5. LilyPad Simple Arduino memiliki 4 input analog, berlabel A2 hingga A5, yang semuanya juga dapat digunakan sebagai I / O digital. Setiap input analog memberikan resolusi 10 bit (mis. 1024 nilai yang berbeda). Secara default input analog mengukur dari ground ke 5 volt, meskipun apakah mungkin untuk mengubah ujung atas kisaran mereka menggunakan fungsi analogReference () .

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.



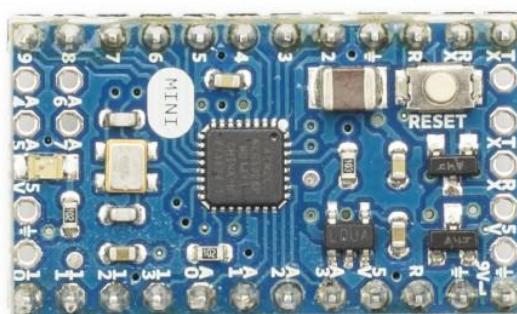
Gambar 2.11 Arduino Lilypad Main Board

Spesifikasi

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

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

Pemrograman

Arduino Mini dapat diprogram dengan perangkat lunak Arduino (unduh). Untuk detailnya, lihat referensi dan tutorialnya. Untuk memprogram Arduino Mini, Anda memerlukan adaptor Serial USB atau USB atau RS232 ke TTL serial lainnya. Lihat halaman tentang memulai dengan Arduino Mini untuk instruksi.

ATmega328P pada Arduino Mini hadir dengan bootloader yang memungkinkan Anda untuk mengunggah kode baru ke dalamnya tanpa menggunakan in-system-programmer. Bootloader berkomunikasi menggunakan protokol STK500 asli (referensi, file header C).

Anda juga dapat mem-bypass bootloader dan memprogram ATmega328P dengan ICSP (In-Circuit Serial Programming).

Input dan Output

Masing-masing dari 14 pin digital pada Mini dapat digunakan sebagai input atau output. Mereka beroperasi pada 5 volt. Setiap pin dapat memberikan atau menerima maksimum 40 mA dan memiliki resistor pull-up internal (terputus secara default) 20-50 kOhms. Pin 3, 5, 6, 9, 10, dan 11 dapat memberikan output PWM; untuk detail lihat fungsi analogWrite (). Jika apa pun selain adaptor Mini USB (atau lainnya) terhubung ke pin 0 dan 1, itu akan mengganggu komunikasi USB, mencegah kode baru diunggah atau komunikasi lainnya dengan komputer.

Mini memiliki 8 input analog, yang masing-masing memberikan 10 bit resolusi (yaitu 1024 nilai yang berbeda). Input 0 hingga 3 dipecah menjadi pin; input 4 hingga 7 membutuhkan penyolderan ke lubang yang disediakan. Secara default input analog mengukur dari ground ke 5 volt, meskipun apakah mungkin untuk mengubah ujung atas rentang mereka menggunakan pin AREF dan beberapa kode level rendah.

13. Arduino Robot

Arduino Robot merupakan Arduino yang memiliki roda atau bisa dibilang arduinio 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

Microcontroller	ATmega32u4
Operating Voltage	5V
Input Voltage	5V
Digital I/O Pin	5
PWM Channels	6
Analog Input Channels	4 (dari pin I/O Digital)
Analog Input Channels (Multiplexed)	8
Arus DC per Pin I/O	40 mA
Flash Memory	32 KB (ATmega32u4) dimana 4 KB digunakan oleh bootloader
SRAM	2,5 KB (ATmega32u4)
EEPROM (internal)	1 KB (ATmega32u4)

EEPROM (external)	512 Kbit (12C)
Clock Speed	16 MHz
Keypad	5 keys
Knob	potensiometer yang terpasang pada pin analog
Full Color LCD	melalui komunikasi SPI
SD card reader	untuk kartu FAT16 yang diformat
Speaker	8 Ohm
Digital Compass	memberikan penyimpangan dari utara geografis dalam derajat Radius
I2C soldering ports	3
Prototyping areas	4
Radius	185 mm
Tinggi	85 mm

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

Pemrograman

Robot dapat diprogram dengan perangkat lunak Arduino (unduh). Pilih "Arduino Robot Control Board" atau "Arduino Robot Motor Board" dari menu Tools> Board. Untuk detailnya, lihat halaman memulai dan tutorial.

Prosesor ATmega32U4 pada Robot Arduino hadir dengan bootloader yang memungkinkan Anda untuk mengunggah kode baru ke dalamnya tanpa menggunakan programmer perangkat keras eksternal. Ini berkomunikasi menggunakan protokol AVR109. Anda dapat mem-bypass bootloader dan memprogram mikrokontroler melalui header ICSP (In-Circuit Serial Programming).

Daya

Robot Arduino dapat diaktifkan melalui koneksi USB atau dengan 4 baterai AA. Sumber daya dipilih secara otomatis. Dudukan baterai memiliki 4 baterai NiMh AA yang dapat diisi ulang.

Untuk tujuan keamanan, motor dinonaktifkan ketika robot diaktifkan dari koneksi USB. Robot ini memiliki pengisi daya baterai terpasang yang membutuhkan daya eksternal 9V yang berasal dari adaptor AC-ke-DC (kutil-dinding). Adaptor dapat dihubungkan dengan mencolokkan steker positif pusat 2.1mm ke colokan listrik Papan Motor. Pengisi daya tidak akan beroperasi jika didukung oleh USB. Papan Kontrol diberi daya oleh catu daya pada Papan Motor.

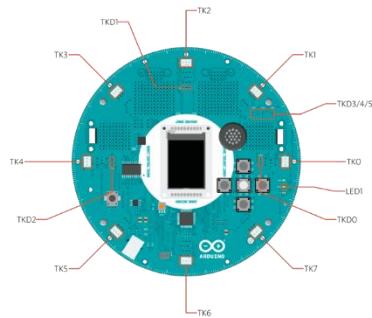
Memori

ATmega32u4 memiliki 32 KB (dengan 4 KB digunakan untuk bootloader). Ini juga memiliki 2,5 KB SRAM dan 1 KB EEPROM (yang dapat dibaca dan ditulis dengan perpustakaan EEPROM). Control Board memiliki EEPROM 512 Kbit tambahan yang dapat diakses melalui I2C. Ada pembaca kartu SD eksternal yang terpasang pada layar GTFT yang dapat diakses oleh prosesor Dewan Kontrol untuk penyimpanan tambahan.

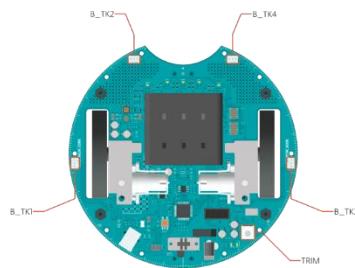
Input dan Output

Robot ini dilengkapi dengan serangkaian konektor pra-solder. Ada sejumlah tempat tambahan bagi Anda untuk memasang komponen tambahan jika diperlukan. Semua konektor diberi label pada papan dan dipetakan ke port bernama melalui perpustakaan Robot yang memungkinkan akses ke fungsi Arduino standar. Setiap pin dapat memberikan atau menerima maksimum 40mA pada 5V. Beberapa pin memiliki fungsi khusus:

- Control Board TK0 to TK7: pin ini multiplexed ke pin analog tunggal pada mikroprosesor Board Kontrol. Mereka dapat digunakan sebagai input analog untuk sensor seperti sensor jarak, sensor ultrasonik analog, atau sakelar mekanis untuk mendeteksi tabrakan.
- Control Board TKD0 ke TKD5: ini adalah pin I / O digital yang terhubung langsung ke prosesor, ditangani menggunakan fungsi Robot.digitalRead () dan Robot.digitalWrite). Pin TKD0 ke TKD3 juga dapat digunakan sebagai input analog dengan Robot.analogRead ()



- Motor Board TK1 ke TK4: pin ini dinamai dalam perangkat lunak sebagai B_TK1 ke B_TK4, mereka dapat berupa pin input digital atau analog, dan mendukung Robot.digitalRead (), Robot.digitalWrite) dan Robot.analogRead () .



- Serial Communication: Papan berkomunikasi satu sama lain menggunakan port serial prosesor. Konektor 10-pin yang menghubungkan kedua papan membawa komunikasi serial, serta informasi daya dan tambahan seperti daya baterai saat ini.
- Control Board SPI: SPI digunakan untuk mengontrol kartu GTFT dan SD. Jika Anda ingin mem-flash prosesor

menggunakan pemrogram eksternal, Anda harus memutuskan koneksi layar terlebih dahulu.

- LED Papan Kontrol: Papan Kontrol memiliki tiga LED on-board. Satu menunjukkan papan dinyalakan (PWR). Dua lainnya menunjukkan komunikasi melalui port USB (LED1 / RX dan TX) .LED1 juga dapat diakses melalui perangkat lunak.
- Kedua papan memiliki konektor I2C yang tersedia: 3 di Papan Kontrol dan 1 di Papan Motor.

Komunikasi

Robot memiliki sejumlah fasilitas untuk berkomunikasi dengan komputer, Arduino lain, atau mikrokontroler lainnya. ATmega32U4 menyediakan komunikasi serial UART TTL (5V), yang tersedia pada digital konektor board-to-board 10-pin. 32U4 juga memungkinkan untuk komunikasi serial (CDC) melalui USB dan muncul sebagai port com virtual ke perangkat lunak di komputer. Chip ini juga bertindak sebagai perangkat USB 2.0 kecepatan penuh, menggunakan driver USB COM standar. Di Windows, file .inf diperlukan. Perangkat lunak Arduino termasuk monitor serial yang memungkinkan data teksual sederhana untuk dikirim ke dan dari papan Robot. RX (LED1) dan TX LED di papan akan berkedip ketika data sedang dikirim melalui koneksi USB ke komputer (tetapi tidak untuk komunikasi serial antar papan). Masing-masing papan memiliki pengenal produk USB yang terpisah dan akan muncul sebagai port yang berbeda pada IDE Anda. Pastikan Anda memilih yang tepat saat pemrograman. ATmega32U4 juga mendukung komunikasi I2C (TWI) dan SPI. Perangkat lunak Arduino mencakup perpustakaan Wire untuk menyederhanakan penggunaan bus I2C; lihat dokumentasi untuk detailnya. Untuk komunikasi SPI, gunakan perpustakaan SPI.

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 elektronika 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 dialirkannya 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

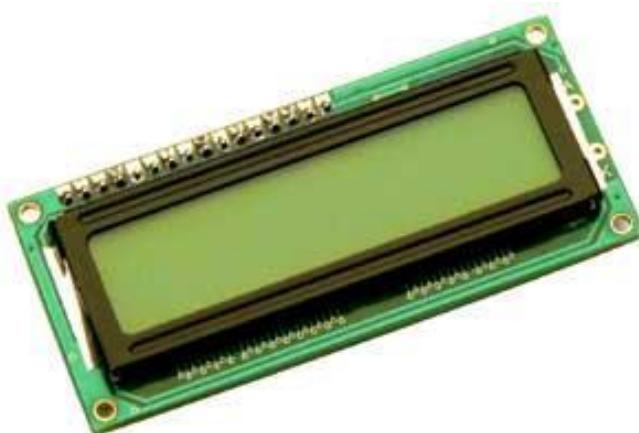
- Female to Female



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 memampulkan 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 manupulasi sesuai kebutuhan kita.

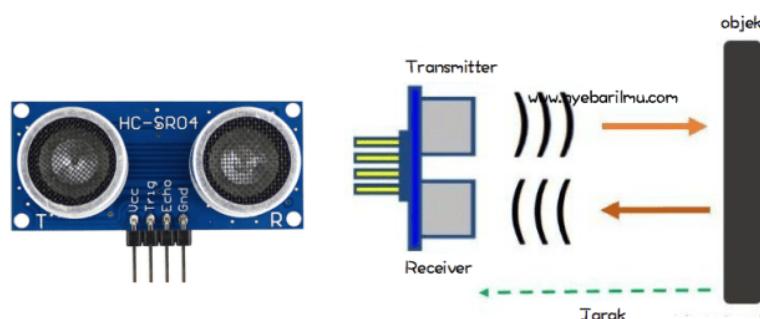


Gambar 2.23 Potensimeter

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

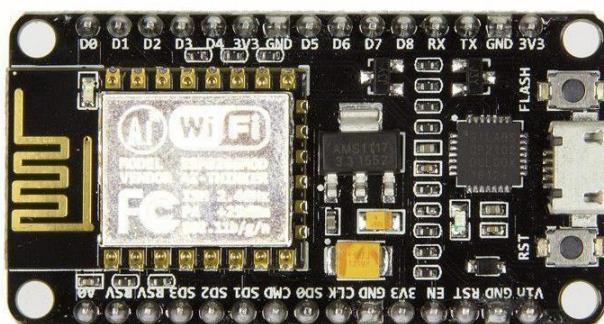
unit, yaitu pemancar dan penerima, dimana pemancar akan mengeluarkan gelombang ke permukaan objek, kemudian objek akan memantulkan dan akan diterima oleh unit penerima kemudian sensor akan menghitung selisih antara waktu pengiriman dan penerimaan gelombang.



Gambar 2.24 Sensor Ultrasonic HC-SR04

10. NodeMCU

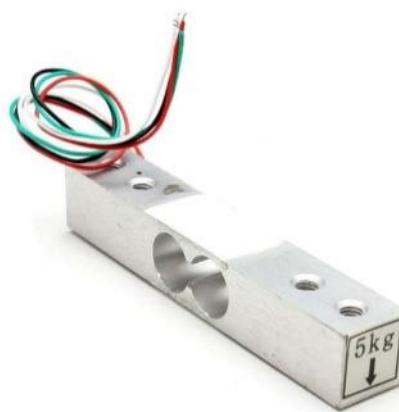
Node MCU adalah Firmware yang bersifat opensource yang menggunakan System On Chip ESP8266 dari ESP8266 buatan Expressif system, yang banyak digunakan untuk membuat projek/produk IoT. Board Node MCU memiliki Beberapa pin GPIO yang dapat kita gunakan untuk menghubungkan perangkat lain dan mampu menghasilkan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



Gambar 2.25 NodeMCU

11. Load Cell

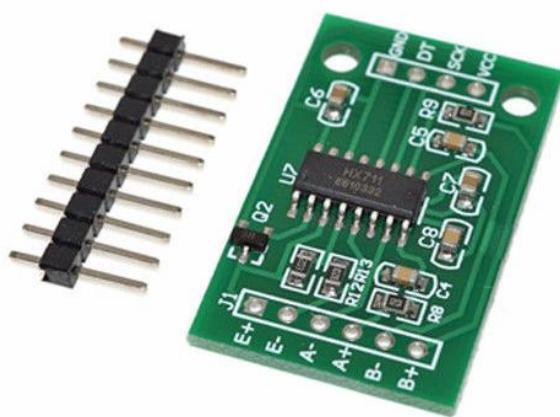
Loadcell adalah sensor berat yang menghasilkan sinyal listrik berdasarkan besarnya sinyal yang sebanding dengan berat yang diukur.



Gambar 2.26 Load Cell

12. HX711

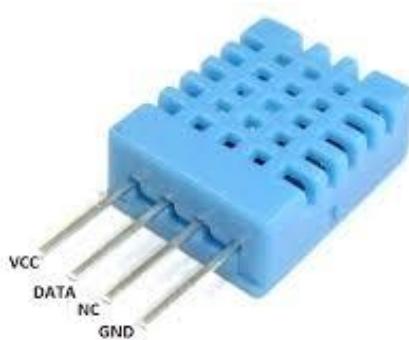
HX711 adalah module yang menerima resistensi listrik dari loadcell kemudian sinyal diperkuat sehingga dapat dibaca oleh Arduino.



Gambar 2.27 HX711

13. Sensor DHT11

Sensor DHT11 adalkah 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 semntara 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 anatara 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 dibenamkan 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. Funi 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.

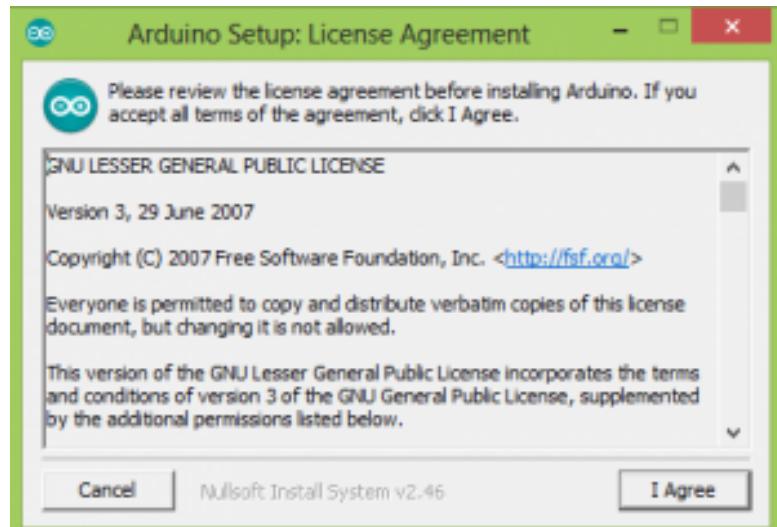
3. Serial Monitor

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.



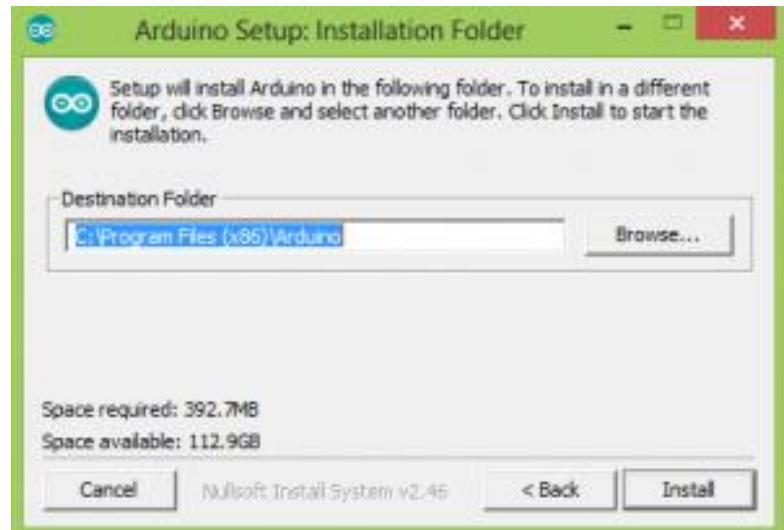
Sdada

4. Install semua komponen yang ada.



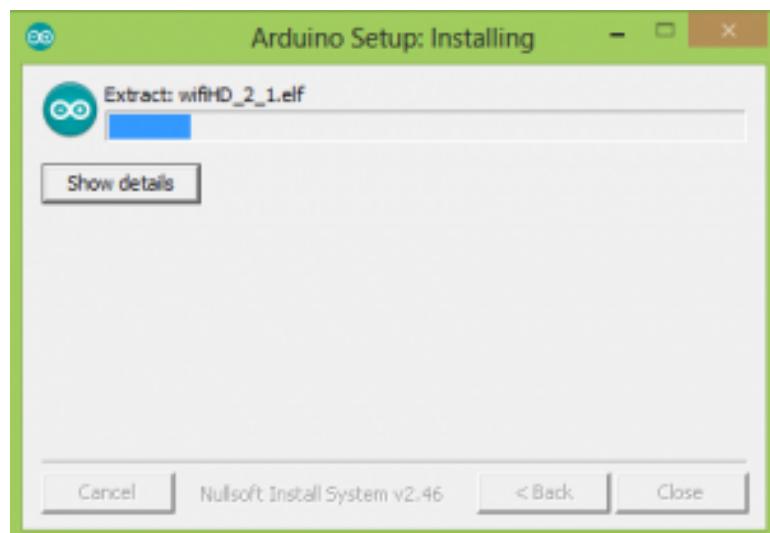
Sdasdasda

5. Pilih lokasi folder instalasi Arduino Software atau gunakan *defaut destination folder*, dan klik *Install*.



Asdasda

6. Tunggu hingga proses instalasi 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*.



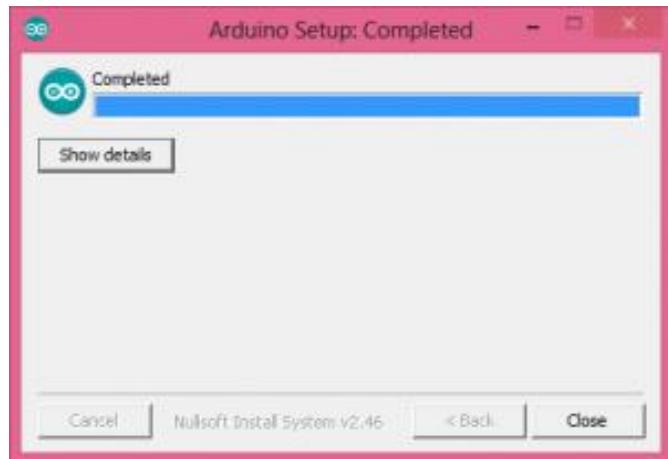
Asdasda

8. Kemudian pada proses installasi, akan muncul dialog window ke-dua untuk menginstall Arduino USB Driver, check “*Always trust software from Arduino 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 :

- Membuka Aplikasi Arduino IDE

The screenshot shows the Arduino IDE interface. The title bar reads "sketch_jan26a | Arduino 1.8.10". The menu bar includes File, Edit, Sketch, Tools, and Help. Below the menu is a toolbar with icons for upload, download, and other functions. The main workspace contains the following code:

```
void setup() {  
    // put your setup code here, to run once:  
  
}  
  
void loop() {  
    // put your main code here, to run repeatedly:  
  
}
```

At the bottom of the IDE, there is a status bar with the text ", 4M (1M SPIFFS), v2 Lower Memory, Disabled, None, Only Sketch, 115200 on COM3" and a cursor placeholder "asdasda".

b) Membuat code program

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

```
void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
}

void loop() {
  // put your main code here, to run repeatedly:
  Serial.println("Hello World");
}
```

4M (1M SPIFFS), v2 Lower Memory, Disabled, None, Only Sketch, 115200 on COM3

asdasda

c) Compile Program Arduino

Setelah kita mengunduh Arduino IDE dan memasukan 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

The screenshot shows the Arduino IDE interface. The title bar reads "sketch_jan26a | Arduino 1.8.10". The menu bar is visible with "File", "Edit", "Sketch" (which is highlighted in blue), "Tools", and "Help". A context menu is open under the "Sketch" menu, listing options: "Verify/Compile" (Ctrl+R), "Upload" (Ctrl+U), "Upload Using Programmer" (Ctrl+Shift+U), "Export compiled Binary" (Ctrl+Alt+S), "Show Sketch Folder" (Ctrl+K), "Include Library" (with a dropdown arrow), and "Add File...". The main code editor window contains the following sketch:

```
void setup() {  
    // put your setup code here, to run once:  
    Serial.println("Hello World");  
}  
  
void loop() {  
    // put your main code here, to run repeatedly:  
    Serial.println("Hello World");  
}
```

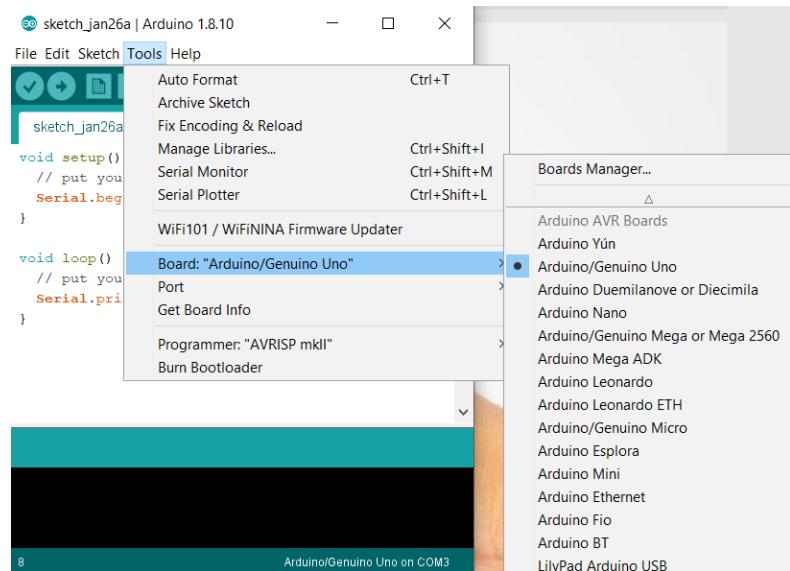
The status bar at the bottom shows "8" on the left and "Arduino/Genuino Uno on COM3" on the right.

asdasd

d) Upload Program Arduino

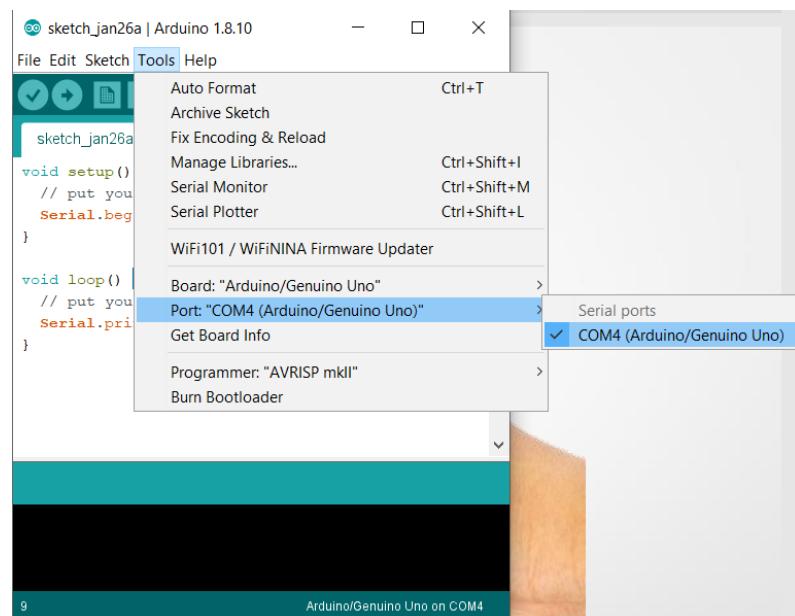
Setelah kita meng-compile programnya tidak ada terjadi error, maka selanjutnya kita akan mengupload code program tersebut ke dalam board Arduino. Langkahnya sebagai berikut :

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



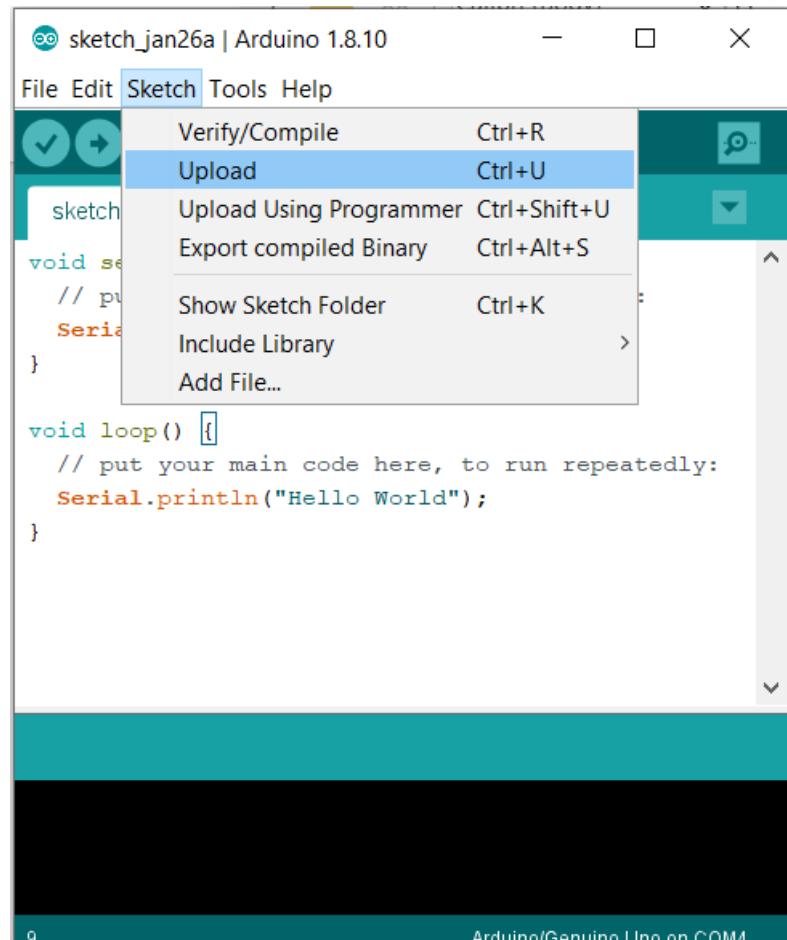
dasdasd

c. Lalu cek port yang terdeteksi



asdadasd

- d. Klik Sketch
- e. Pilih Upload
- f. Tunggu hingga proses selasai.



sadasds

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 penggunanya". 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 pencarinya 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, maka 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 baru, 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 Environment (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 Inch 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. Struktur Proyek Android Studio

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

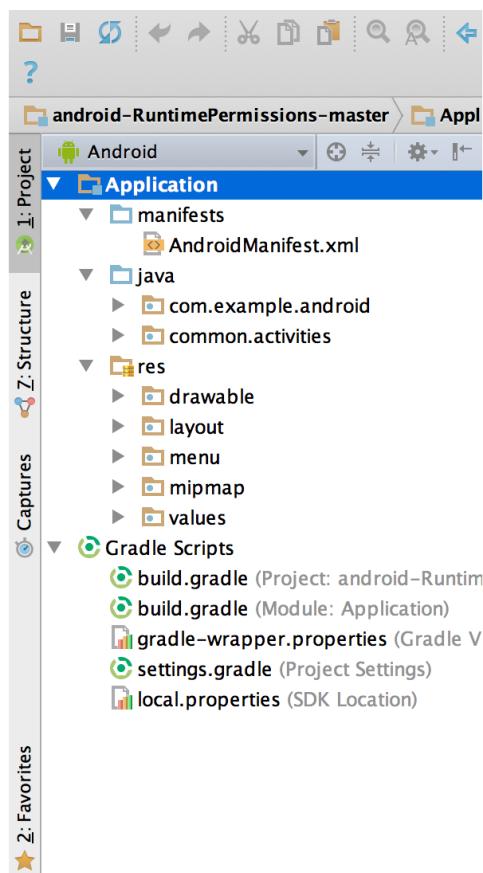
- Modul aplikasi Android
- Modul library
- Modul Google App Engine

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:

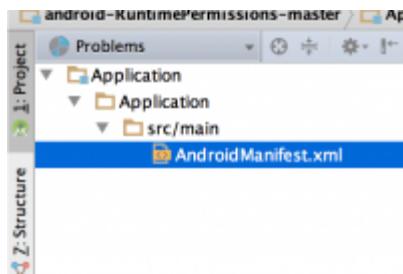
- **manifests**: Berisi file AndroidManifest.xml.
- **java**: Berisi file kode sumber Java, termasuk kode pengujian JUnit.
- **res**: Berisi semua resource non-kode, seperti tata letak XML, string UI, dan gambar bitmap.

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

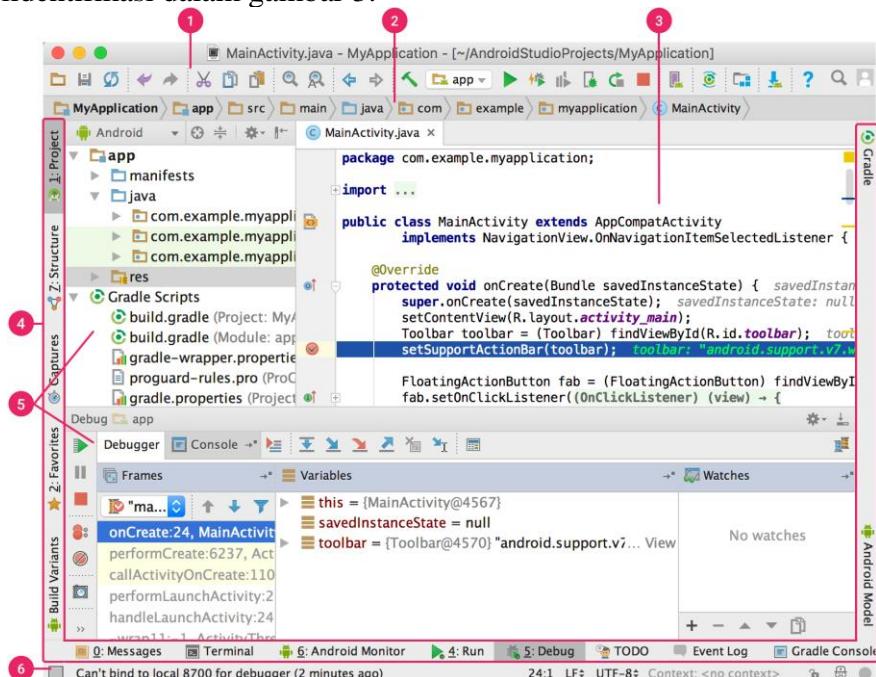
Anda juga dapat menyesuaikan tampilan file project untuk berfokus pada aspek spesifik dari pengembangan aplikasi Anda. Misalnya, memilih tampilan **Problems** pada project Anda akan menampilkan link ke file sumber yang berisi error coding dan sintaks yang dikenali, seperti tag penutup elemen XML yang 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 menciatkan 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 menciatkan 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 menciatkan 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 cursor 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	Control+Shift+F12	Command+Shift+F12
Semua Jendela Alat		

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	Menampilkan saran dasar untuk variabel, jenis, metode, ekspresi, dan sebagainya.	Control+Spasi	Control+Spasi
Completion	Jika		

	<p>memanggil Basic Completion dua kali berturut-turut, Anda akan melihat lebih banyak hasil, termasuk anggota pribadi dan anggota statis yang tidak diimpor.</p>		
Smart Completi on	<p>Menampilkan opsi yang relevan berdasarkan konteks. Smart Completion mengetahui jenis dan alur data yang diharapkan.</p> <p>Jika Anda memanggil Smart Completion dua kali berturut-turut, Anda akan melihat lebih banyak hasil, termasuk chain.</p>	Control+Shift+Spasi	Control+Shift+Spasi
Statement Completi on	<p>Melengkapi pernyataan saat ini secara otomatis, seperti menambahka</p>	Control+Shift+Spasi	Shift+Command+Enter

n tanda
kurung, tanda
kurung siku,
tanda kurung
kurawal,
pemformatan,
dll. yang tidak
lengkap.

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.

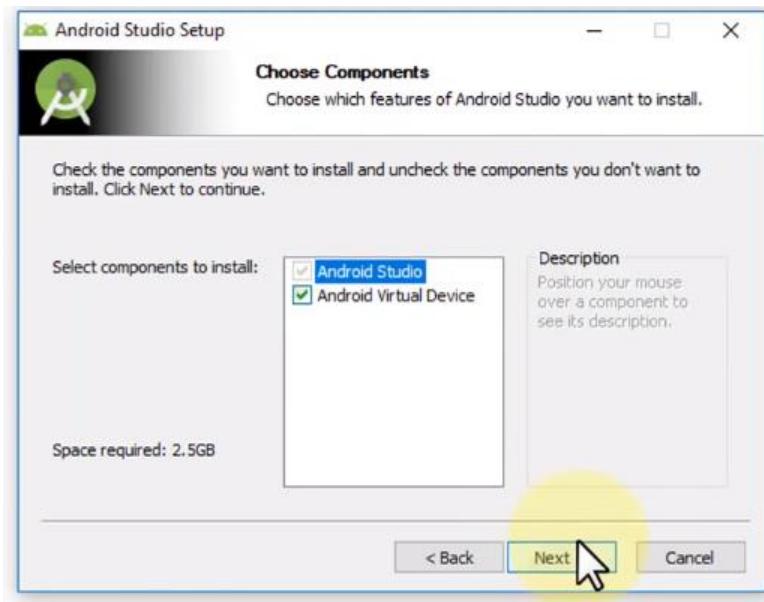
Jika Anda sudah mendownload file .zip, ekstrak ZIP, salin folder **android-studio** ke folder **Program Files**, lalu buka folder **studio-android > bin** dan jalankan studio64.exe (untuk komputer 64 bit) atau studio.exe (untuk komputer 32 bit).

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



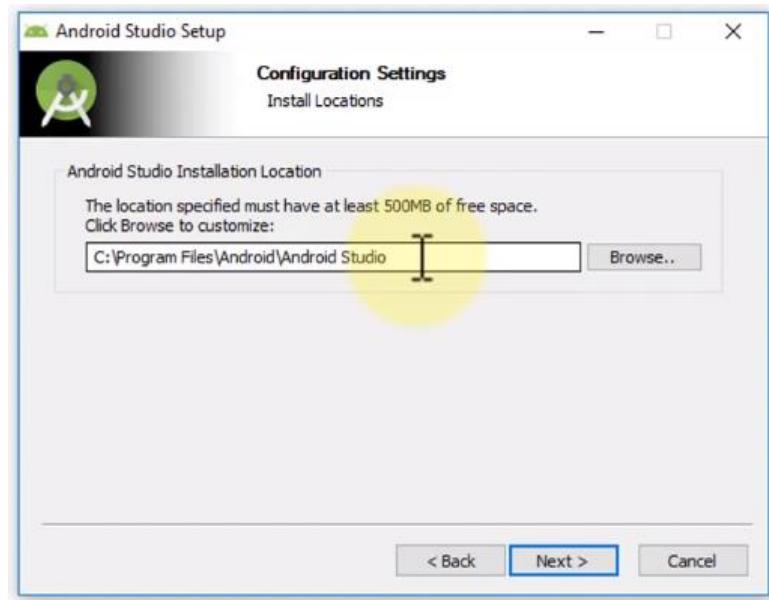
Gambar 3.23 Langkah 1

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



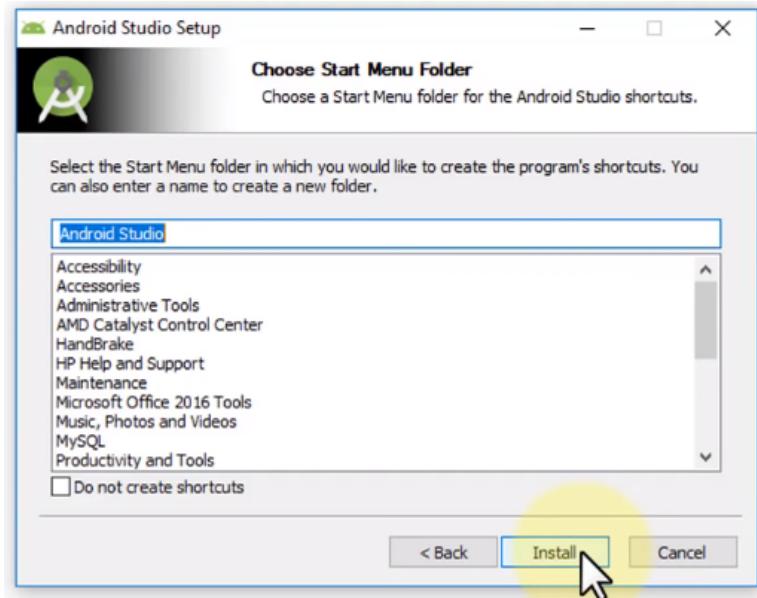
Gambar 3.24 Langkah 2

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



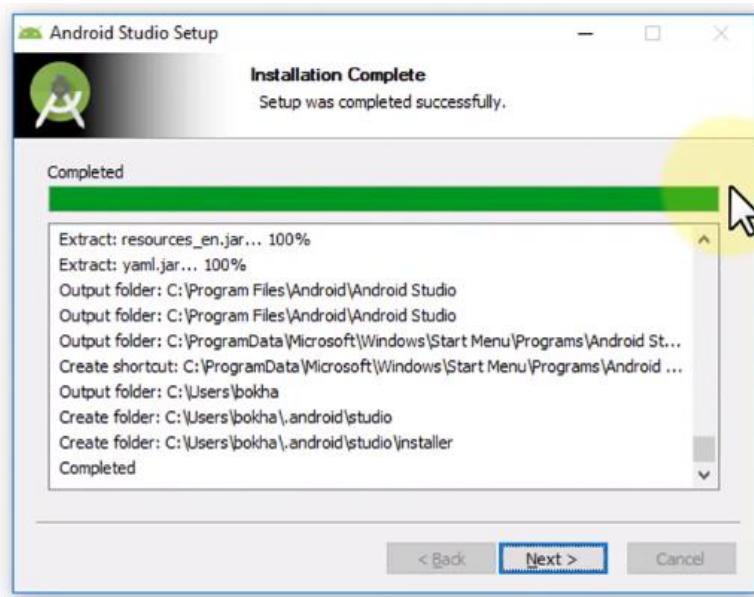
Gambar 3.25 Langkah 3

6. Klik Install



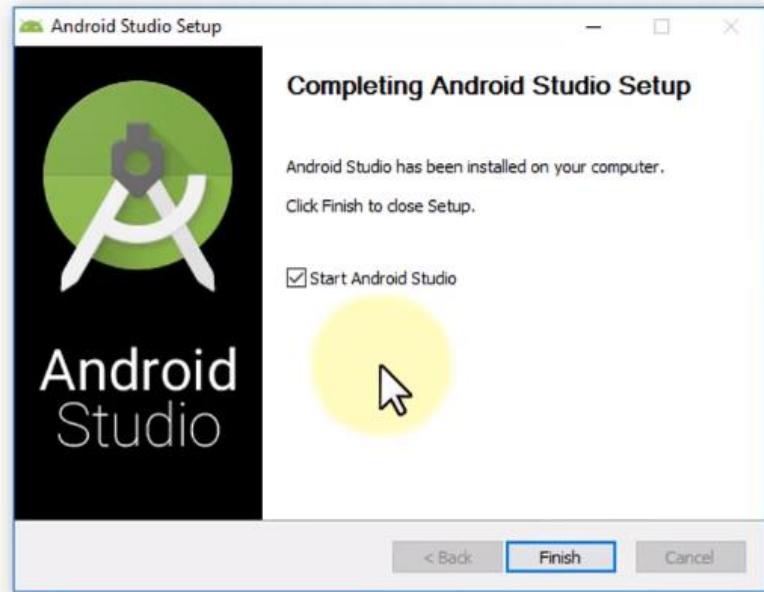
Gambar 3.26 Langkah 4

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



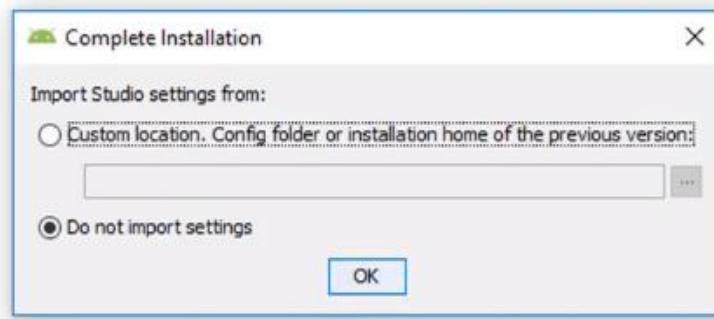
Gambar 3.27 Langkah 5

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**.



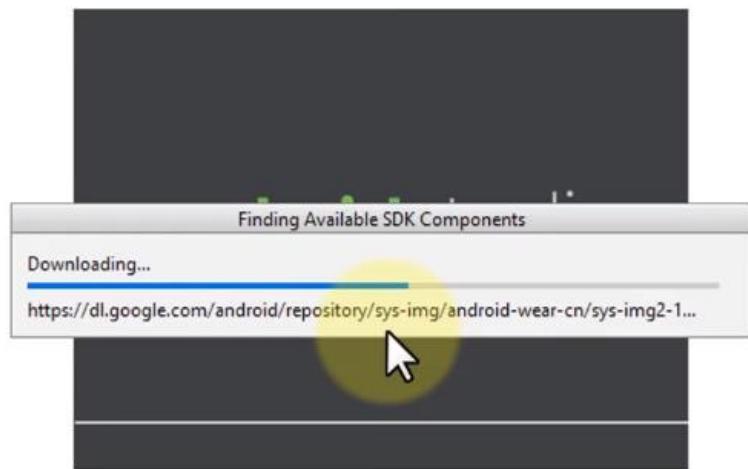
Gambar 3.29 Langkah 7

10. Maka akan muncul launcher aplikasi Android Studio.



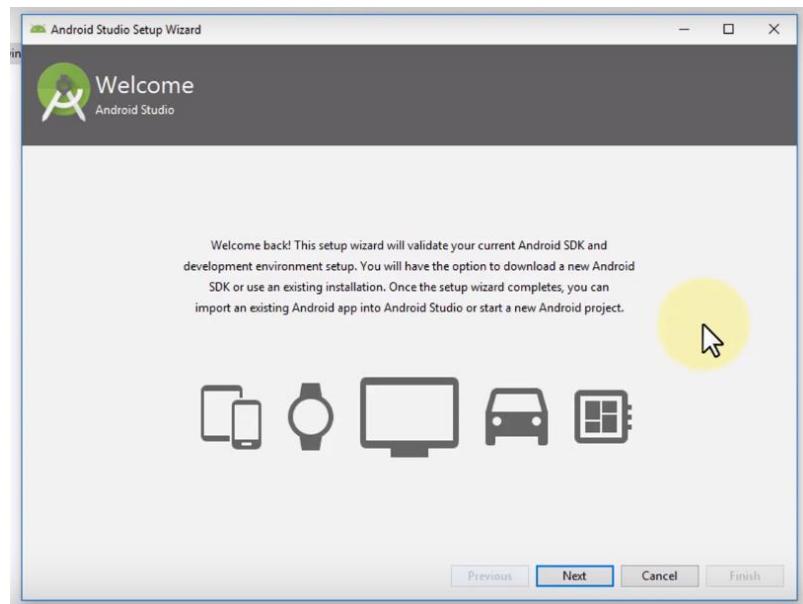
Gambar 3.30 Langkah 8

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



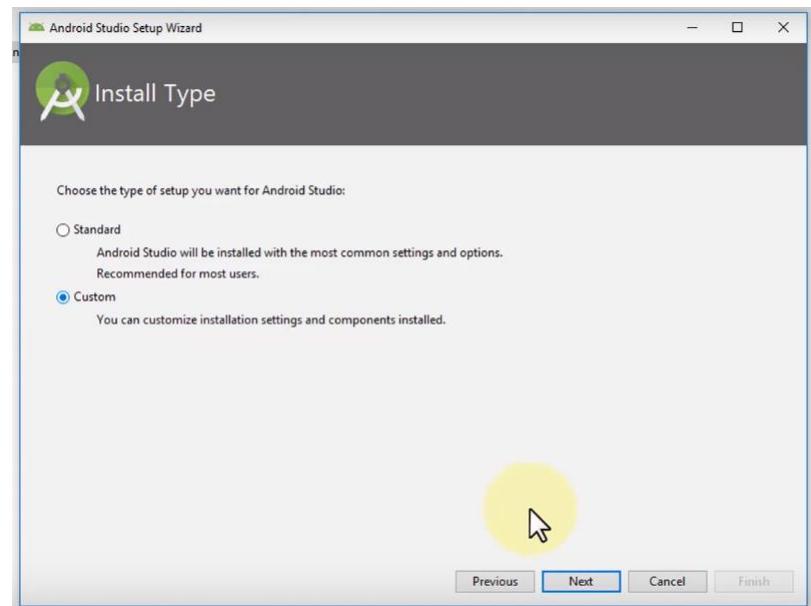
Gambar 3.31 Langkah 9

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



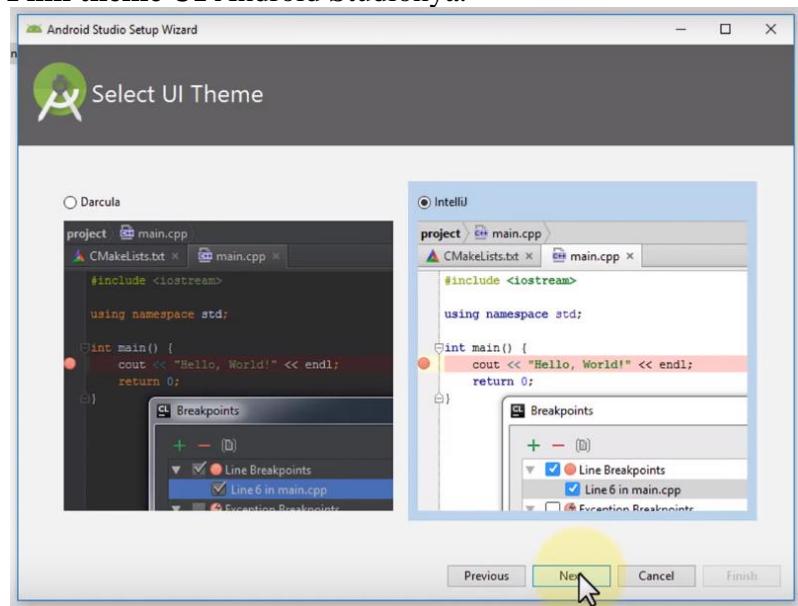
Gambar 3.32 Langkah 10

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



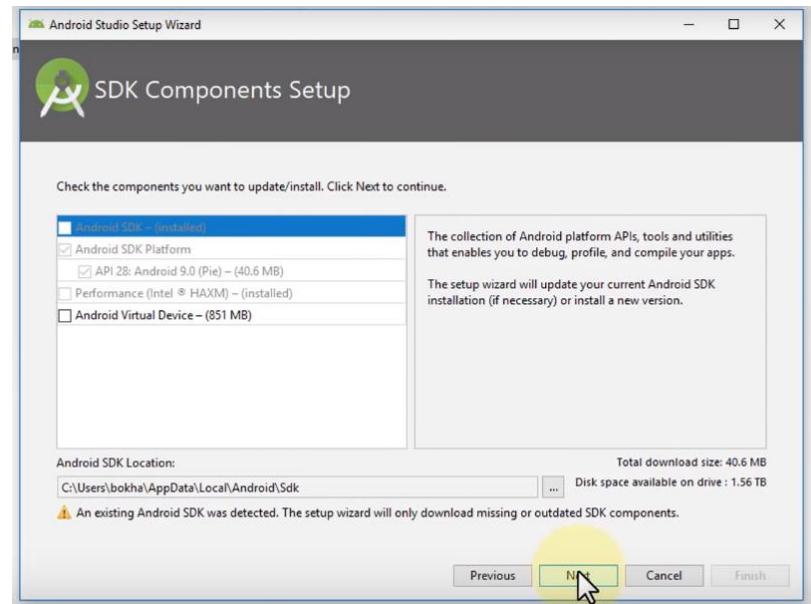
Gambar 3.33 Langkah 11

14. Pilih theme UI Android Studionya.



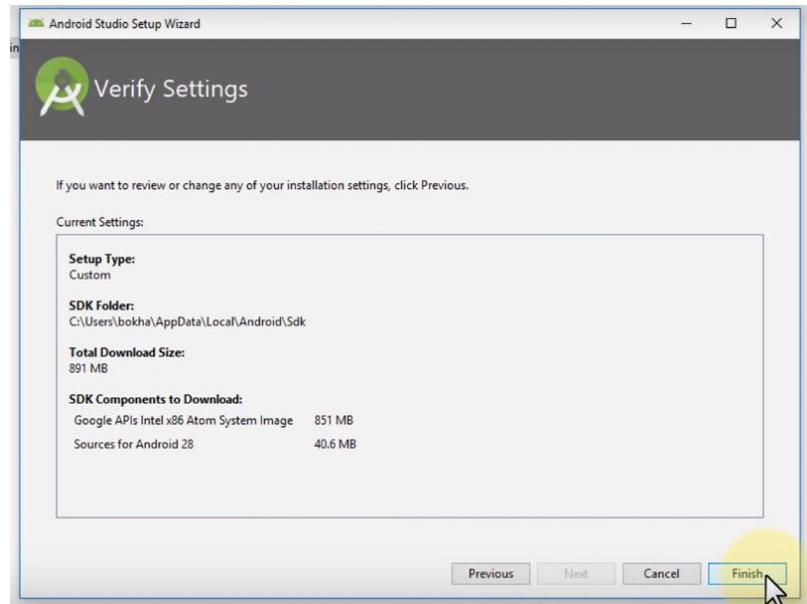
Gambar 3.34 Langkah 12

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



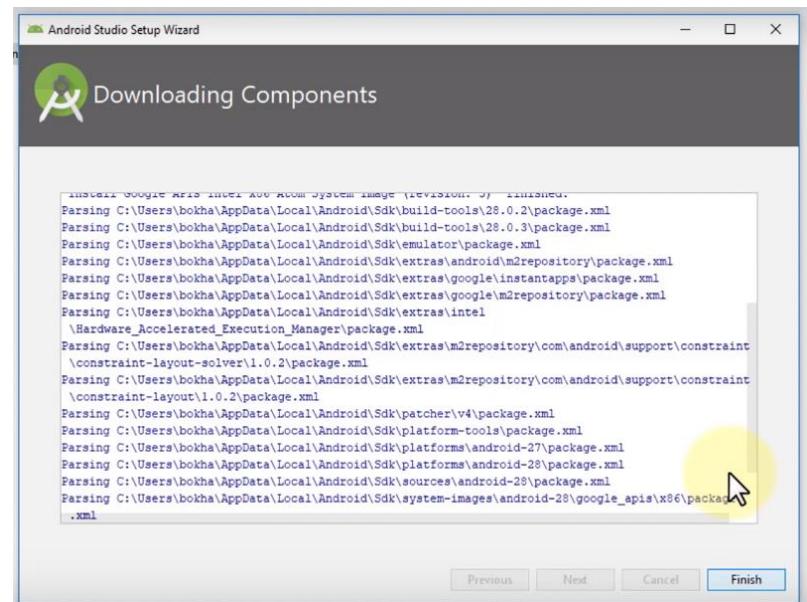
Gambar 3.35 Langkah 13

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



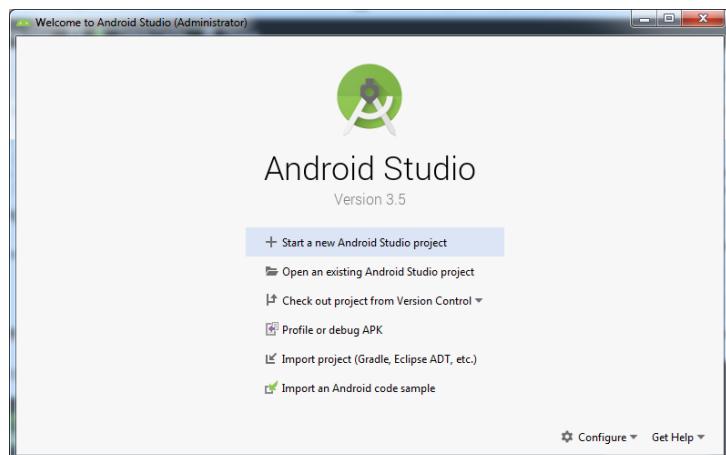
Gambar 3.36 Langkah 14

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

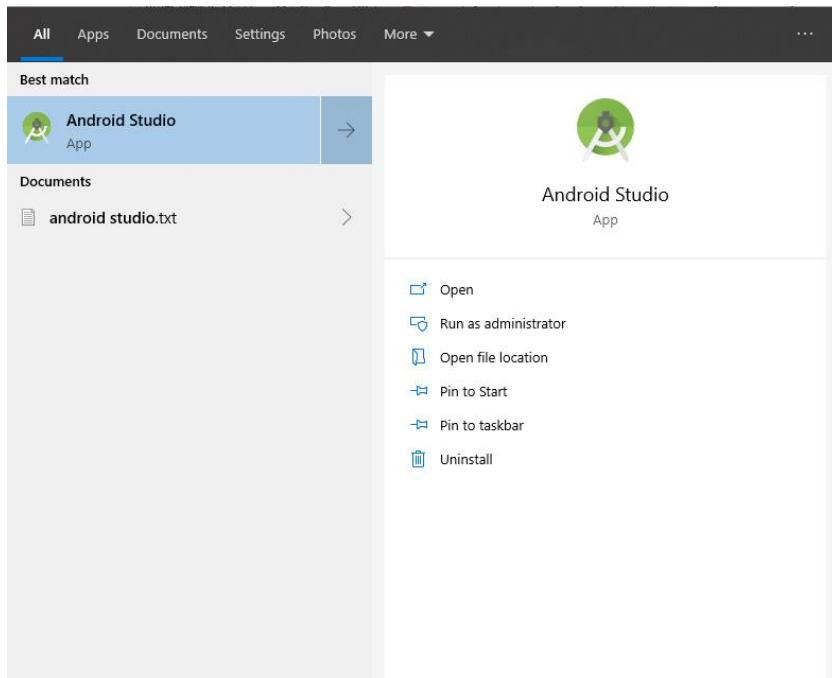


Gambar 3.338 Langkah 16

3.4.9. Membuat Aplikasi *Hello Word* di Android Studio

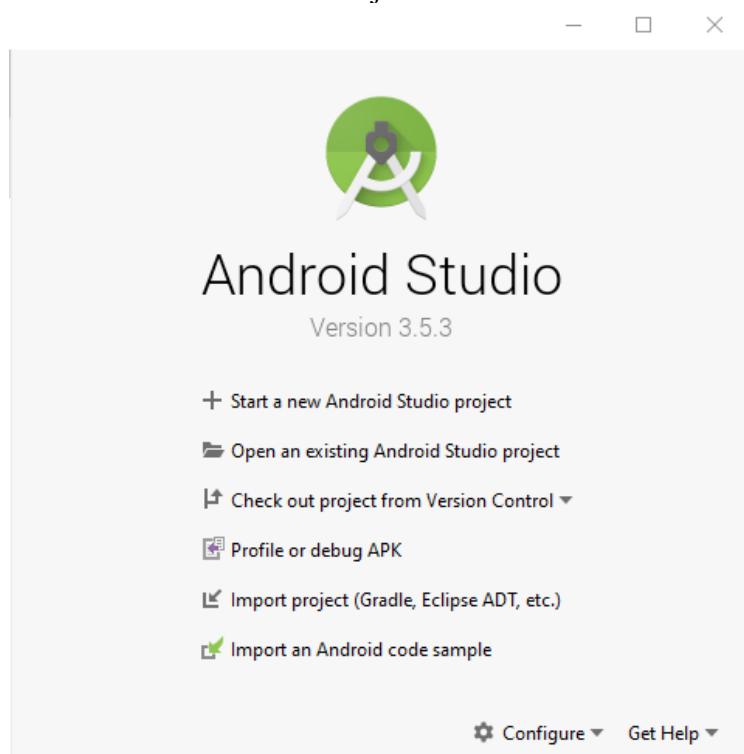
Setelah kita sudah menginstall aplikasi android studio, maka kita disini akan membuat aplikasi sederhana “*Hello Word*” pada aplikasi android studio.

1. Buka aplikasi android studio pada windows search



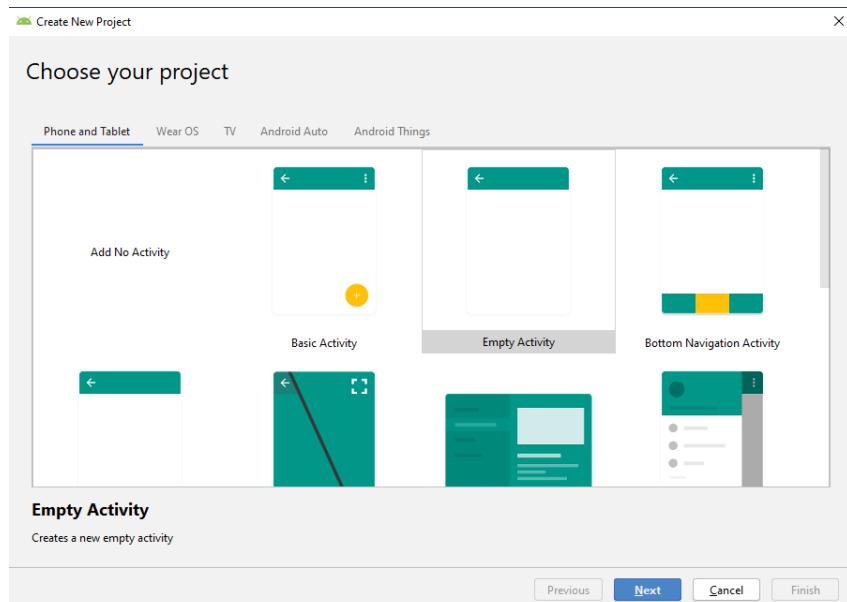
Gambar 3.39 Membuka Aplikasi Android Studio

2. Klik start new Android Studio Project



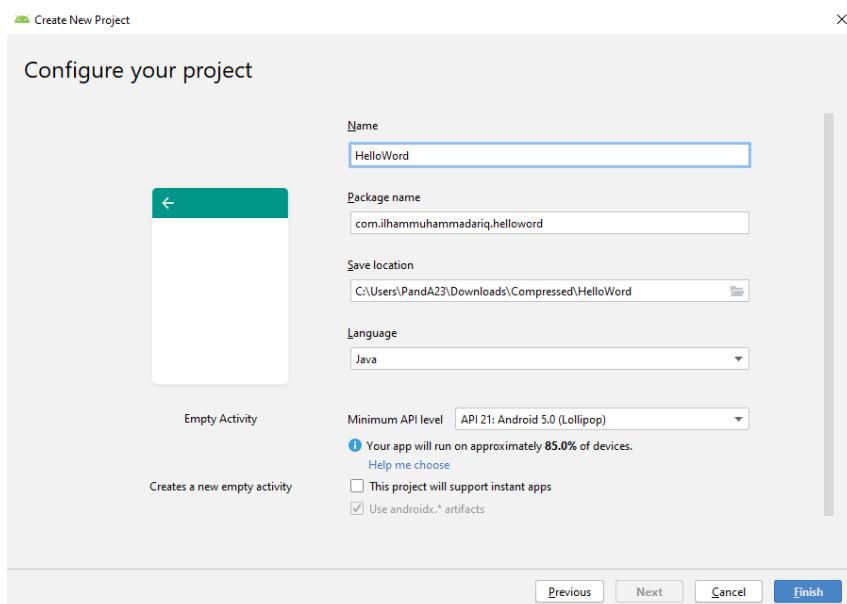
Gambar 3.40 Pilih Menu Project

3. Klik Empty Activity dan next



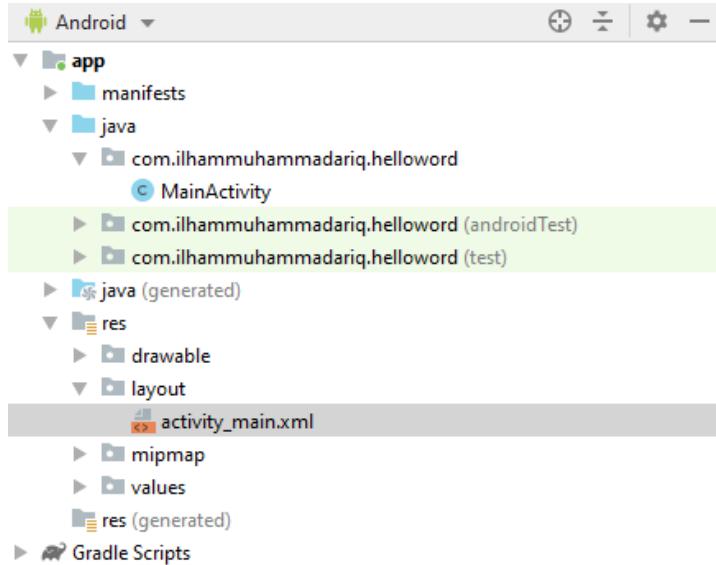
Gambar 3.41 Memilih Project

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

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



Gambar 3.43 Mengedit File .xml

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



Gambar 3.44 Tampilan Aplikasi Project

3.4.10. Cara Menjalankan Aplikasi pada Perangkat Sungguhan

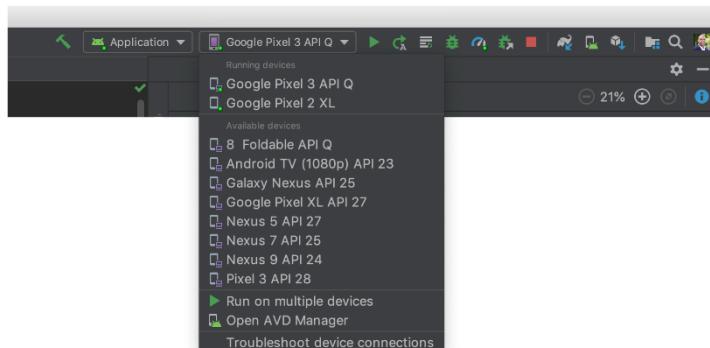
Persiapkan perangkat Anda sebagai berikut:

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

4.1. 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 Envolve, startup sebelumnya yang didirikan oleh James Tamplin dan Andrew Lee pada tahun 2011. Envolve menyediakan pengembang Antarmuka pemrograman aplikasi yang memungkinkan integrasi fungsi obrolan daring ke situs web mereka. Setelah merilis layanan obrolan, Tamplin dan Lee menemukan bahwa Envolve digunakan untuk mengirimkan data aplikasi yang bukan pesan obrolan. Pengembang menggunakan Envolve untuk menyinkronkan data aplikasi seperti status permainan secara waktu nyata di seluruh penggunanya. 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 Firestore

Gunakan database cloud NoSQL kami yang fleksibel dan skalabel guna menyimpan dan menyinkronkan data untuk pengembangan sisi klien dan server.

Cloud Firestore adalah database yang fleksibel dan skalabel untuk pengembangan seluler, web, dan server di Firebase dan Google Cloud Platform. Seperti Firebase Realtime Database, Cloud Firestore membuat data Anda tetap terhubung di aplikasi klien melalui listener realtime dan menawarkan dukungan secara offline untuk seluler dan web. Dengan begitu, Anda dapat membuat aplikasi yang responsif dan mampu bekerja tanpa harus bergantung pada latensi jaringan atau koneksi Internet. Cloud Firestore juga menawarkan integrasi yang lancar dengan produk Firebase dan Google Cloud Platform lainnya, termasuk Cloud Functions.

Kemampuan Utama

1. Fleksibelitas : Model data Cloud Firestore mendukung struktur data yang hierarkis dan fleksibel. Simpan data Anda ke dalam dokumen yang tersusun dalam koleksi. Selain subkoleksi, dokumen dapat berisi objek bertingkat yang kompleks.
2. Pembuatan kueri yang ekspresif : Di Cloud Firestore, Anda dapat menggunakan kueri untuk mengambil masing-masing dokumen tertentu atau semua dokumen dalam koleksi yang sesuai dengan parameter kueri Anda. Kueri Anda dapat meliputi beberapa filter berantai dan menggabungkan penyaringan dan pengurutan. Kueri juga diindeks secara default, sehingga performa kueri sebanding dengan ukuran kumpulan hasil, bukan kumpulan data.
3. Update realtime : Seperti Realtime Database, Cloud Firestore menggunakan sinkronisasi data untuk mengupdate data pada perangkat yang terhubung. Namun, Cloud Firestore juga dirancang untuk membuat kueri pengambilan 1 kali yang sederhana secara efisien.
4. Dukungan offline : Cloud Firestore menyimpan data yang digunakan secara aktif oleh aplikasi Anda, sehingga aplikasi dapat menulis, membaca, mendeteksi, dan melakukan kueri data meskipun perangkat sedang offline. Saat perangkat kembali online, Cloud Firestore akan menyinkronkan semua perubahan lokal kembali ke Cloud Firestore.

5. Dirancang sesuai paket : Cloud Firestore menghadirkan infrastruktur Google Cloud Platform yang terbaik kepada Anda: replikasi data multi-region secara otomatis, jaminan konsistensi yang kuat, batch operasi yang bersifat menyeluruh, dan dukungan transaksi nyata. Kami merancang Cloud Firestore untuk menangani beban kerja database yang tersulit dari aplikasi terbesar di dunia.

Cara Kerja Cloud Firestore

Cloud Firestore adalah database NoSQL yang dihosting di cloud dan dapat diakses langsung melalui SDK asli oleh iOS, Android, dan aplikasi web Anda. Selain REST dan RPC API, Cloud Firestore juga tersedia di Node.js, Java, Python, dan Go SDK yang asli.

Setelah model data NoSQL Cloud Firestore, simpan data Anda dalam dokumen yang berisi pemetaan kolom terhadap nilai. Dokumen ini disimpan dalam koleksi yang berisi container untuk dokumen Anda, yang dapat digunakan untuk mengatur data dan membuat kueri. Dokumen ini mendukung berbagai jenis data, mulai dari string dan angka sederhana, hingga objek yang kompleks dan bertingkat. Anda juga dapat membuat subkoleksi dalam dokumen dan membangun struktur data hierarkis yang berskala sesuai dengan database. Model data Cloud Firestore mendukung struktur data yang paling sesuai untuk aplikasi Anda.

Selain itu, pembuatan kueri di Cloud Firestore bersifat ekspresif, efisien, dan fleksibel. Buatlah kueri dangkal untuk mengambil data pada tingkat dokumen tanpa perlu mengambil keseluruhan koleksi atau subkoleksi bertingkat. Tambahkan pengurutan, penyaringan, dan batasan pada kueri atau cursor untuk memberi nomor pada hasil Anda. Tambahkan listener realtime untuk menjaga data di aplikasi Anda tetap terkini, tanpa harus mengambil keseluruhan database setiap kali ada update. Dengan menambahkan listener realtime ke aplikasi, Anda akan mendapatkan pemberitahuan dengan snapshot data setiap kali data yang dideteksi oleh aplikasi klien Anda berubah. Dengan begitu, hanya perubahan baru yang akan diambil.

Lindungi akses data Anda di Cloud Firestore dengan Firebase Authentication dan Aturan Keamanan Cloud Firestore untuk Android, iOS, dan JavaScript, atau Pengelolaan Akses dan Identitas (IAM) untuk bahasa sisi server.

2. Firebase Authentication

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 menyediakan layanan backend, SDK yang mudah digunakan, dan library UI yang siap pakai untuk mengautentikasi pengguna ke aplikasi Anda. Firebase Authentication mendukung autentikasi menggunakan sandi, nomor telepon, penyedia identitas gabungan yang populer, seperti Google, Facebook, dan Twitter, dan lain-lain.

Firebase Authentication terintegrasi erat dengan layanan Firebase lainnya dan sistem ini memanfaatkan berbagai standar industri, seperti OAuth 2.0 dan OpenID Connect, sehingga dapat dengan mudah diintegrasikan dengan backend kustom Anda.

Kemampuan Utama

Anda dapat membuat pengguna login ke aplikasi Firebase dengan menggunakan FirebaseUI sebagai solusi autentikasi drop-in lengkap atau dengan menggunakan Firebase Authentication SDK untuk mengintegrasikan salah satu atau beberapa metode login ke aplikasi Anda secara manual.

1. FirebaseUI Auth

- Solusi autentikasi drop-in : Cara yang direkomendasikan untuk menambahkan sistem login yang lengkap ke aplikasi Anda.

FirebaseUI menyediakan solusi autentikasi drop-in yang menangani alur UI untuk membuat pengguna login dengan alamat email dan sandi, nomor telepon, dan penyedia identitas gabungan yang populer, seperti Login dengan Google dan Login dengan Facebook.

Komponen FirebaseUI Auth mengimplementasikan praktik terbaik untuk proses autentikasi di perangkat seluler dan situs, yang akan memaksimalkan konversi login dan pendaftaran

untuk aplikasi Anda. FirebaseUI Auth juga menangani kasus khusus, seperti pemulihan akun dan penyambungan akun, yang dapat dianggap sensitif bagi keamanan dan rawan kesalahan untuk ditangani dengan benar.

FirebaseUI dapat disesuaikan dengan mudah agar cocok dengan gaya visual aplikasi Anda. FirebaseUI juga open source, sehingga Anda tidak dibatasi dalam mewujudkan pengalaman pengguna yang Anda inginkan.

2. Firebase SDK Authentication

- Autentikasi berbasis email dan sandi : Mengautentikasi pengguna dengan alamat email dan sandi mereka. Firebase Authentication SDK menyediakan metode untuk membuat dan mengelola pengguna yang login menggunakan alamat email dan sandi. Firebase Authentication juga menangani pengiriman email untuk menyetel ulang sandi.
- Integrasi penyedia identitas tergabung : Mengautentikasi pengguna dengan mengintegrasikan penyedia identitas gabungan. Firebase Authentication SDK menyediakan metode yang memungkinkan pengguna untuk login dengan akun Google, Facebook, Twitter, dan GitHub mereka.
- Autentikasi nomor telefon : Mengautentikasi pengguna dengan mengirim pesan SMS ke ponsel mereka.
- Integrasi sistem autentikasi khusus : Menghubungkan sistem login aplikasi yang ada ke Firebase Authentication SDK, serta memperoleh akses ke Firebase Realtime Database dan layanan Firebase lainnya.
- Autentikasi anonym : Menggunakan fitur yang memerlukan autentikasi tanpa mewajibkan pengguna untuk login terlebih dahulu dengan membuat akun anonim sementara. Jika pengguna kemudian memilih untuk login, Anda dapat mengupgrade akun anonim menjadi akun reguler agar pengguna dapat melanjutkan pekerjaannya.

Cara Kerja Firebase Authentication

Untuk membuat pengguna login ke aplikasi, dapatkan kredensial autentikasi dari pengguna terlebih dahulu. Kredensial ini dapat berupa alamat email dan sandi pengguna, atau token OAuth dari penyedia identitas gabungan. Kemudian, teruskan kredensial ini ke Firebase Authentication SDK. Layanan backend kami selanjutnya akan memverifikasi kredensial tersebut dan menampilkan respons ke klien.

Setelah berhasil login, Anda dapat mengakses informasi profil dasar pengguna dan mengontrol akses pengguna ke data yang disimpan di

produk Firebase lainnya. Anda juga dapat menggunakan token autentikasi yang disediakan untuk memverifikasi identitas pengguna di layanan backend yang dimiliki.

Catatan: Secara default, pengguna yang terautentikasi dapat membaca dan menulis data ke Firebase Realtime Database dan Cloud Storage. Anda dapat mengontrol akses pengguna tersebut dengan mengubah Firebase Realtime Database dan Aturan Keamanan Cloud Storage.

3. Firebase Remote Config

Firebase 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 Firebase. Kemudian objek firebase di dalam aplikasi akan melakukan request data dari server, kemudian memproses data-data tersebut. Secara umum, objek Firebase 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 ijin yang dibutuhkan oleh aplikasi.

Kemampuan Utama

1. Menerapkan perubahan pada basis pengguna aplikasi dengan cepat : Anda dapat membuat perubahan pada perilaku dan tampilan default aplikasi dengan mengubah parameter value sisi server. Misalnya, Anda dapat mengubah tata letak aplikasi atau tema warna untuk menjalankan promosi musiman, tanpa perlu mempublikasikan update aplikasi.
2. Menyesuaikan aplikasi untuk segmen basis pengguna : Anda dapat menggunakan Remote Config untuk memberikan variasi pengalaman pengguna aplikasi ke berbagai segmen basis pengguna berdasarkan versi aplikasi, audience Google Analytics, bahasa, dan lainnya.
3. Menjalankan pengujian A/B untuk menyempurnakan aplikasi : Anda dapat menggunakan penargetan persentil acak Remote Config dengan Google Analytics untuk penyempurnaan pengujian A/B pada aplikasi

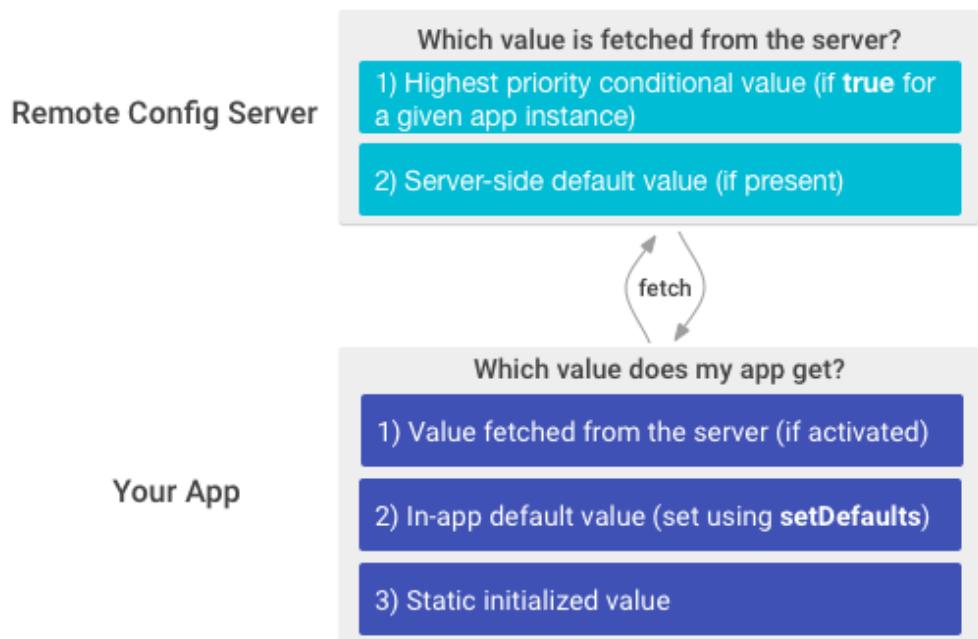
di berbagai segmen basis pengguna, sehingga Anda bisa memvalidasi penyempurnaan sebelum meluncurkannya ke seluruh basis pengguna.

Cara Kerja Firebase Remote Config

Remote Config menyertakan library klien yang menangani tugas penting seperti mengambil parameter value dan menyimpannya dalam cache, sambil tetap memberikan kendali atas waktu *pengaktifan* nilai baru, sehingga dapat memengaruhi pengalaman pengguna aplikasi. Dengan demikian, Anda dapat menjaga desain aplikasi dengan menentukan waktu untuk setiap perubahan.

Metode get library klien Remote Config menyediakan titik akses tunggal untuk parameter value. Aplikasi Anda mendapatkan nilai sisi server menggunakan logika yang sama yang digunakannya untuk mendapatkan nilai default dalam aplikasi, sehingga Anda bisa menambahkan kemampuan Remote Config ke aplikasi tanpa harus menuliskan banyak kode.

Untuk mengganti nilai default dalam aplikasi, gunakan Firebase console atau REST API Remote Config untuk membuat parameter dengan nama yang sama seperti parameter dalam aplikasi Anda. Untuk setiap parameter, Anda dapat menetapkan nilai default sisi server untuk mengganti nilai default dalam aplikasi. Anda juga dapat membuat nilai kondisional untuk mengganti nilai default dalam aplikasi untuk instance aplikasi yang memenuhi kondisi tertentu. Gambar ini menunjukkan cara memprioritaskan parameter value pada backend Remote Config dan aplikasi Anda:



4. Firebase Realtime Database

Firebase 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 Firebase cloud. Perusahaan menyediakan pustaka klien yang memungkinkan integrase 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 Firebase Realtime Database dirilis untuk penggunaan beta. Firebase Realtime Database merupakan database yang di-host pada cloud. Data disimpan sebagai JSON dan disinkronkan secara realtime ke setiap klien yang terhubung.

Kemampuan utama

1. Realtime: Sebagai ganti permintaan HTTP biasa, Firebase Realtime Database menggunakan sinkronisasi data—setiap kali data berubah, semua perangkat yang terhubung akan menerima update dalam waktu

- milidetik. Memberikan pengalaman yang kolaboratif dan imersif tanpa perlu memikirkan kode jaringan.
- 2. Offline : Aplikasi Firebase tetap responsif bahkan saat offline karena SDK Firebase Realtime Database menyimpan data ke disk. Setelah koneksi pulih, perangkat klien akan menerima setiap perubahan yang terlewat dan melakukan sinkronisasi dengan status server saat ini.
 - 3. Dapat diakses dari Perangkat Klien : Firebase Realtime Database dapat diakses secara langsung dari perangkat seluler atau browser web; server aplikasi tidak diperlukan. Keamanan dan validasi data dapat diakses melalui Aturan Keamanan Firebase Realtime Database yang merupakan kumpulan aturan berbasis ekspresi dan dijalankan ketika data dibaca atau ditulis.
 - 4. Menskalakan di beberapa database : Dengan Firebase Realtime Database pada paket harga Blaze, Anda dapat mendukung kebutuhan data aplikasi Anda pada skala tertentu dengan membagi data Anda di beberapa instance database di project Firebase yang sama. Menyederhanakan autentikasi dengan Firebase Authentication pada project Anda dan mengautentikasi pengguna di instance database Anda. Mengontrol akses ke data di tiap database dengan Aturan Firebase Realtime Database khusus untuk tiap instance database.

CARA KERJA REALTIME DATABASE

Firebase Realtime Database memungkinkan Anda untuk membuat aplikasi kolaboratif dan kaya fitur dengan menyediakan akses yang aman ke database, langsung dari kode sisi klien. Data disimpan di drive lokal. Bahkan saat offline sekalipun, peristiwa realtime terus berlangsung, sehingga pengguna akhir akan merasakan pengalaman yang responsif. Ketika koneksi perangkat pulih kembali, Realtime Database akan menyinkronkan perubahan data lokal dengan update jarak jauh yang terjadi selama klien offline, sehingga setiap perbedaan akan otomatis digabungkan.

Realtime Database menyediakan bahasa aturan berbasis ekspresi yang fleksibel, atau disebut juga Aturan Keamanan Firebase Realtime Database, untuk menentukan metode strukturisasi data dan kapan data dapat dibaca atau ditulis. Ketika diintegrasikan dengan Firebase Authentication, developer dapat menentukan siapa yang memiliki akses ke data tertentu dan bagaimana mereka dapat mengaksesnya.

Realtime Database adalah database NoSQL, sehingga memiliki pengoptimalan dan fungsionalitas yang berbeda dengan database terkait. API Realtime Database dirancang agar hanya mengizinkan operasi yang dapat dijalankan dengan cepat. Hal ini memungkinkan Anda untuk membangun pengalaman realtime yang luar biasa dan dapat melayani jutaan pengguna tanpa mengorbankan kemampuan respons. Oleh karena itu, perlu dipikirkan bagaimana pengguna mengakses data, kemudian buat struktur data sesuai dengan kebutuhan tersebut.

5. Firebase Storage

Cloud Storage dirancang untuk developer aplikasi yang perlu menyimpan dan menampilkan konten buatan pengguna, seperti foto atau video.

Cloud Storage for Firebase adalah layanan penyimpanan objek yang andal, sederhana, dan hemat biaya yang dibuat untuk skala Google. Firebase SDK untuk Cloud Storage menambahkan keamanan Google pada upload dan download file untuk aplikasi Firebase Anda, bagaimanapun kualitas jaringannya. Anda dapat menggunakan SDK kami untuk menyimpan gambar, audio, video, atau konten buatan pengguna lainnya. Di server, Anda dapat menggunakan Google Cloud Storage untuk mengakses file yang sama.

Kemampuan Utama

1. Operasi yang stabil : Firebase SDK untuk Cloud Storage melakukan upload dan download, bagaimanapun kualitas jaringannya. Upload dan download bersifat stabil, artinya proses ini akan dilanjutkan dari posisi berhenti, sehingga menghemat waktu dan bandwidth pengguna.
2. Keamanan yang kuat : Firebase SDK untuk Cloud Storage terintegrasi dengan Firebase Authentication untuk menyediakan autentikasi yang mudah dan intuitif bagi developer. Anda dapat menggunakan model keamanan deklaratif kami untuk mengizinkan akses berdasarkan nama file, ukuran, jenis konten, dan metadata lainnya.
3. Skalabilitas tinggi : Cloud Storage for Firebase dibuat untuk skala exabyte ketika aplikasi Anda menjadi viral. Berkembang dari prototipe menjadi produksi dengan mudah menggunakan infrastruktur yang juga memberdayakan Spotify dan Google Foto.

Cara Kerja Firebase Storage

Developer menggunakan Firebase SDK untuk Cloud Storage untuk mengupload dan mendownload file langsung dari klien. Jika koneksi jaringan buruk, klien bisa mencoba operasi ini lagi dari posisi terakhir saat pengoperasian terhenti, sehingga menghemat waktu dan bandwidth pengguna.

Cloud Storage menyimpan file Anda di bucket Google Cloud Storage, sehingga membuatnya mudah diakses melalui Firebase dan Google Cloud. Dengan begitu, Anda memiliki fleksibilitas untuk mengupload dan mendownload file dari klien seluler melalui Firebase SDK, dan melakukan pemrosesan sisi server seperti pemfilteran gambar atau transcoding video menggunakan Google Cloud Platform. Cloud Storage akan diskalakan secara otomatis, yang berarti tidak perlu bermigrasi ke penyedia lain. Pelajari lebih lanjut tentang semua manfaat integrasi dengan Google Cloud Platform kami.

Firebase SDK untuk Cloud Storage terintegrasi sempurna dengan Firebase Authentication untuk mengidentifikasi pengguna, dan kami menyediakan bahasa keamanan deklaratif yang dapat Anda gunakan untuk menyetel kontrol akses pada masing-masing file atau kumpulan file, sehingga Anda bisa menyetelnya sebagai publik atau pribadi sesuai keinginan.

6. Firebase Hosting

Firebase Hosting menyediakan hosting yang cepat dan aman untuk aplikasi web serta konten yang statis dan dinamis. Firebase 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 sederhananya, 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 waktunya 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.

Kemampuan Utama

1. Menayangkan konten melalui koneksi yang aman : Web modern itu aman. SSL tanpa konfigurasi sudah disertakan dalam Firebase Hosting, sehingga konten selalu dikirim dengan aman.
2. Menghosting konten statis dan dinamis plus layanan mikro : Firebase Hosting mendukung semua jenis konten untuk hosting, mulai dari file CSS dan HTML hingga API atau layanan mikro Express.js Anda.
3. Mengirim konten dengan cepat : Setiap file yang Anda upload akan di-cache di SSD pada berbagai server edge CDN seluruh dunia. Di mana pun pengguna berada, konten akan dikirim dengan cepat.
4. Menerapkan versi baru dengan satu perintah : Dengan Firebase CLI, aplikasi Anda dapat aktif dan berjalan dalam hitungan detik. Fitur command-line mempermudah penambahan target penerapan ke dalam proses build Anda.
5. Melakukan rollback dengan satu klik : Penerapan cepat itu baik, tapi akan jadi lebih baik jika disertai kemampuan mengurungkan kesalahan. Firebase Hosting menyediakan pembuatan versi lengkap dan pengelolaan rilis dengan rollback sekali klik.

Cara Kerja Firebase Hosting

Firebase Hosting dibuat untuk developer web modern. Dengan hadirnya framework JavaScript front-end seperti Angular dan fitur generator statis seperti Jekyll, situs dan aplikasi kini jadi lebih tangguh daripada sebelumnya. Jika Anda menerapkan halaman landing aplikasi yang sederhana atau Progressive Web App (PWA) yang kompleks, Hosting menyediakan infrastruktur, fitur, dan rangkaian sarana yang disesuaikan untuk menerapkan serta mengelola situs dan aplikasi.

Dengan menggunakan Firebase CLI, Anda menerapkan file dari direktori lokal di komputer Anda ke server Hosting. Selain menayangkan konten statis, Anda dapat menggunakan Cloud Functions for Firebase atau Cloud Run untuk menayangkan konten dinamis dan menghosting layanan microservice di situs Anda. Semua

konten ditayangkan melalui koneksi SSL dari server edge terdekat di CDN global kami.

Firebase Hosting memiliki opsi konfigurasi hosting yang ringan bagi Anda untuk membuat PWA yang canggih. Anda dapat menulis ulang URL untuk perutean sisi klien atau menyiapkan header kustom dengan mudah.

Untuk menayangkan konten Anda, Firebase menawarkan beberapa opsi domain dan subdomain:

- Secara default, setiap project Firebase memiliki subdomain gratis di domain web.app dan firebaseapp.com. Kedua situs ini melayani konten dan konfigurasi yang sama-sama diterapkan.
- Anda dapat membuat beberapa situs jika memiliki situs dan aplikasi terkait yang menayangkan konten berbeda tetapi tetap memiliki resource project Firebase yang sama (misalnya jika Anda memiliki blog, panel admin, dan aplikasi publik).
- Anda dapat menghubungkan nama domain sendiri ke situs yang di-hosting oleh Firebase.

Firebase secara otomatis memberikan sertifikat SSL untuk semua domain Anda, sehingga semua konten Anda ditayangkan dengan aman.

6. Firebase Cloud Messaging

Firebase Cloud Messaging (FCM) adalah solusi pengiriman pesan lintas platform yang memungkinkan Anda mengirimkan pesan dengan tepercaya tanpa biaya.

Dengan FCM, Anda dapat memberi tahu aplikasi klien bahwa email baru atau data lainnya tersedia untuk disinkronkan. Anda dapat mengirim pesan notifikasi untuk mendorong interaksi kembali dan retensi pengguna. Untuk kasus penggunaan seperti instant messaging, pesan dapat mentransfer payload hingga 4 KB ke aplikasi klien.

Kemampuan Utama

1. Mengirim pesan notifikasi atau pesan data : Mengirim pesan notifikasi yang ditampilkan kepada pengguna. Atau mengirim pesan data dan menentukan sepenuhnya apa yang terjadi dalam kode aplikasi.

2. Penargetan pesan serbaguna : Mendistribusikan pesan ke aplikasi klien dengan salah satu dari 3 cara — ke sebuah perangkat, ke grup perangkat, atau ke perangkat yang berlangganan topik.
3. Mengirim pesan dari aplikasi klien : Mengirim notifikasi, chat, dan pesan lain dari perangkat ke server melalui saluran koneksi FCM yang andal dan hemat baterai.

Cara Kerja Firebase Cloud Messaging

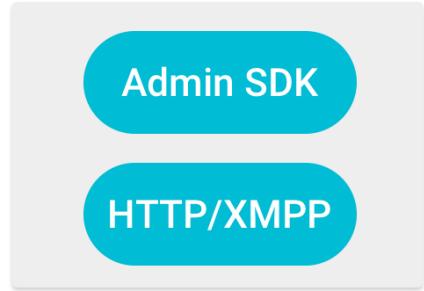
Implementasi FCM mencakup dua komponen utama untuk mengirim dan menerima pesan:

1. Lingkungan tepercaya seperti Cloud Functions for Firebase atau server aplikasi yang akan digunakan untuk membuat, menargetkan, dan mengirim pesan.
2. Aplikasi klien iOS, Android, atau Web (JavaScript) yang menerima pesan.

Anda dapat mengirim pesan melalui Firebase Admin SDK atau protokol server FCM. Untuk menguji atau mengirim pesan pemasaran atau engagement dengan penargetan dan analisis bawaan yang andal, Anda dapat menggunakan Notifications composer.



Notifications Console GUI



Trusted Environment



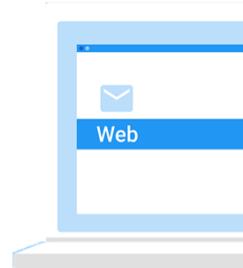
Firebase
Cloud Messaging



Android



iOS



Web

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

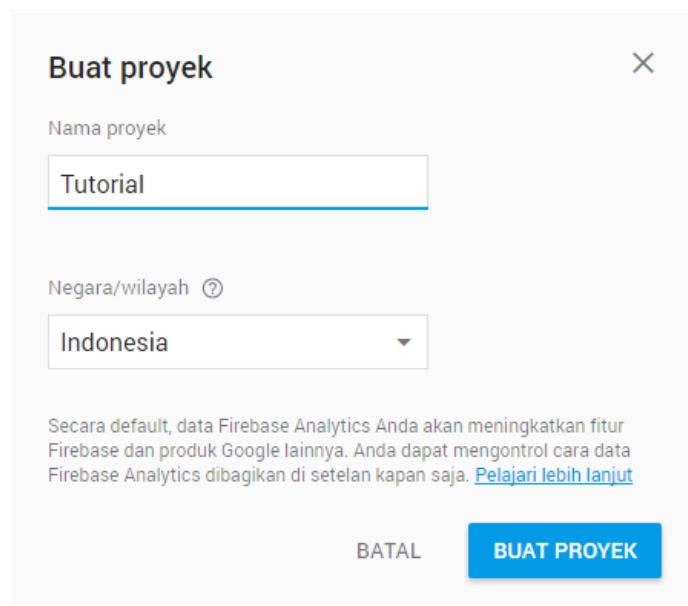
Firebase Dynamic Links adalah URL cerdas yang secara dinamis mengubah perilaku untuk memberikan pengalaman terbaik di berbagai platform (situs web / iOS / Android) serta tautan dalam ke APP. Dynamic Links berfungsi di semua pemasangan aplikasi: jika pengguna membuka Dynamic Links di iOS atau Android 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.

4.5. Cara Menyambungkan Aplikasi Android pada Firebase

1. Masuk ke console.firebaseio.google.com
Disini akan membuat project baru

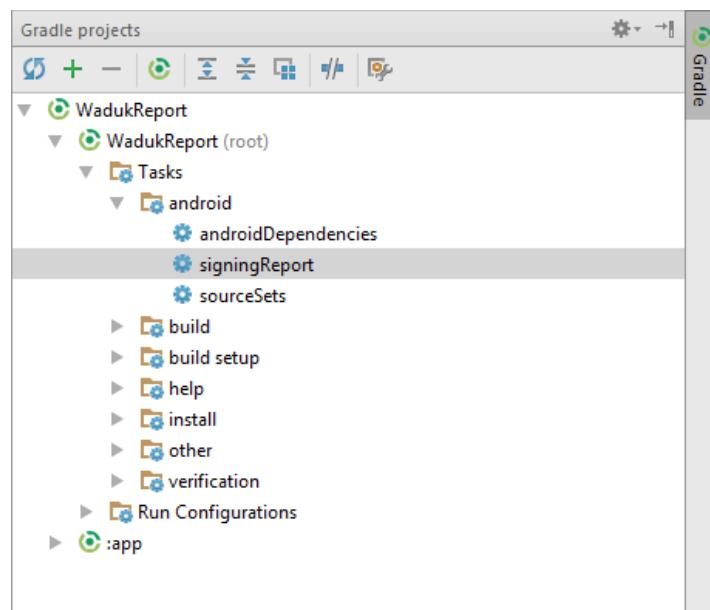


Dsadasda

Saat kita create project pertama kali di project tersebut bisa terdapat dalam beberapa aplikasi lagi, jadi dalam 1 project bisa terdapat lebih dari 1 aplikasi

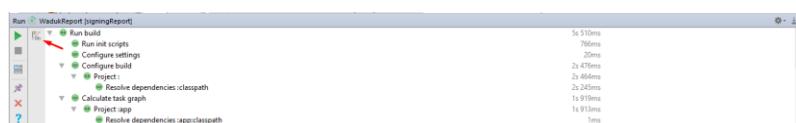
2. Pilih Aplikasi Android dan daftarkan Aplikasi kita
Masukkan SHA --- 1 pada android

- a) Buka Android Studio, dan masuk kesini



Asdasd

- b) Setelah itu akan muncul tampilan seperti ini
Klik tombol yang ditunjuk anak panah



Asdasd

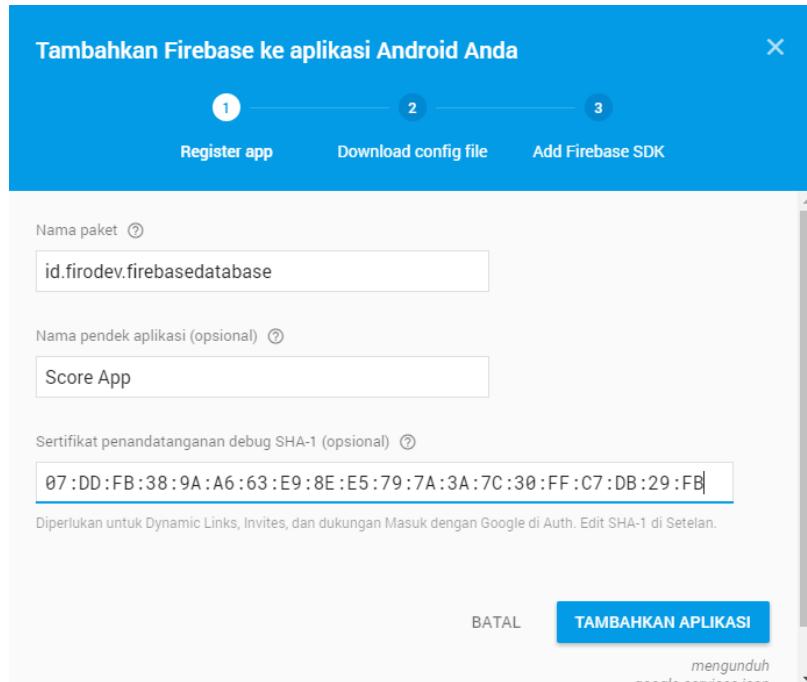
- c) Maka akan muncul SHA – 1 nya

```
Run > Windows Task Manager [signingReport]
MD5: 12:4B:95:00:14:1A:FE:C1:48:05:D4:A9:16:71:FC:96
SHA1: 07:D0:9D:35:9A:AE:63:E9:8E:E5:79:A7:3A:7C:30:FF:C7:DB:29:F8
Valid until: Friday, March 23, 2046
-----
BUILD SUCCESSFUL
-----
Total time: 0.752 secs
? 15.27.21: External task execution finished "signingReport".

```

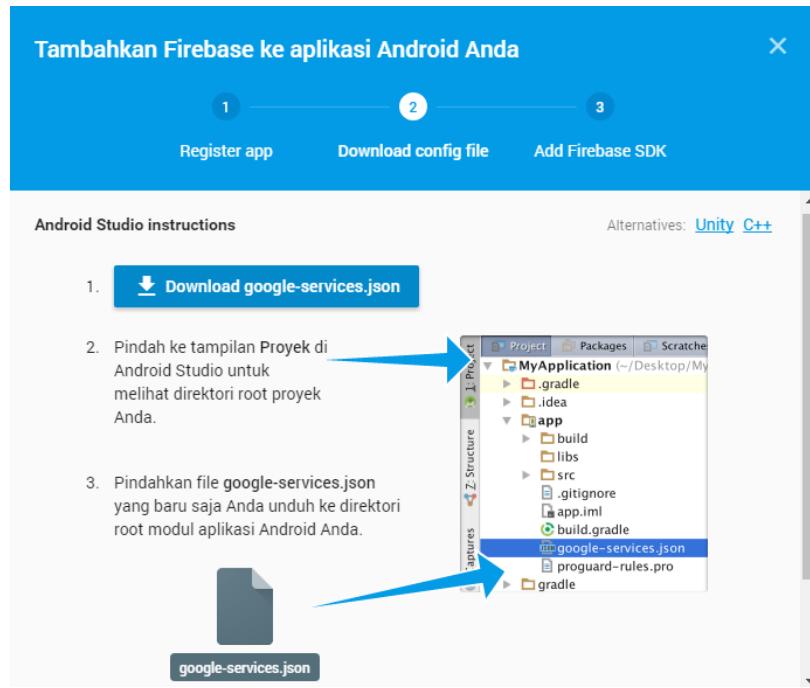
Asdasdasd

3. Masukan SHA nya pada kolom dibawah ini



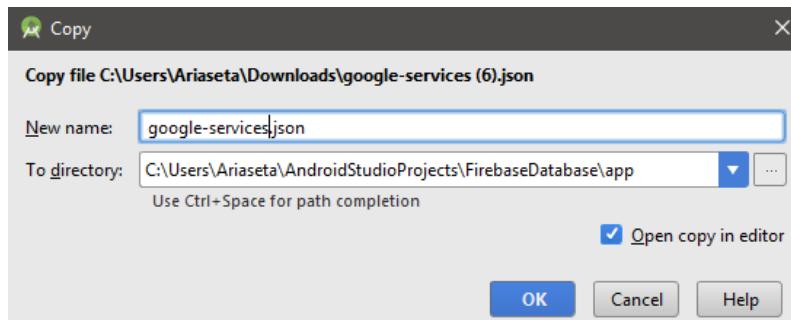
Asdasd

4. Kemudian nanti akan mendownload file .json. Lha, json ini fungsinya untuk menggabungkan aplikasi kita dengan Firebase.

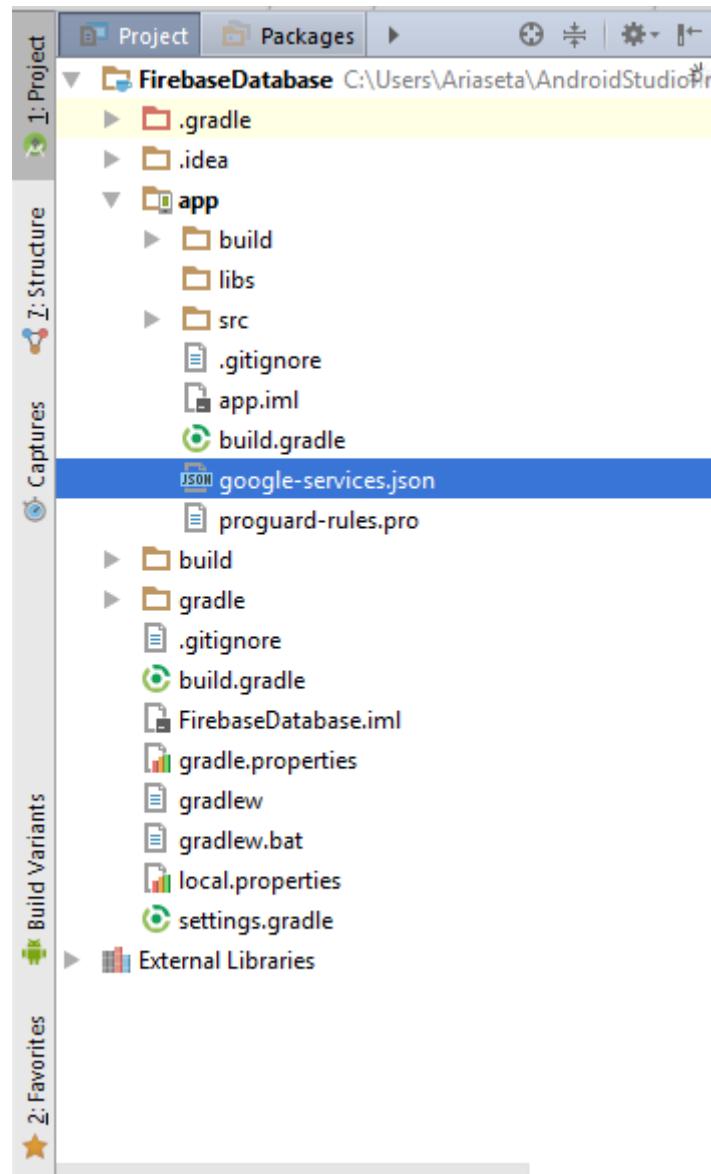


Sadasda

5. Kemudian copy file google-services.json tersebut kedalam directory yang sudah disebutkan pada gambar diatas



Sadasdas



Asdasda

6. Masukan dependencies sesuai platform Firebase yang akan digunakan
Pertama masukkan kode classpath berikut ke folder **build.gradle**

```
buildscript {
    // ...
    dependencies {
        // ...
        classpath 'com.google.gms:google-services:3.0.0'
    }
}
```

Asdasds

7. Setelah itu, kita juga memasukkan apply plugin ke folder **app/build.gradle**

```
apply plugin: 'com.android.application'

android {
    // ...
}

dependencies {
    // ...
}

// Masukkan di bagian paling bawah
apply plugin: 'com.google.gms.google-services'
```

Asdasda

8. Aplikasi Android telah terpasang ke Firebase

4.6. Membaca dan Menulis Data di Android Studio

Data Firebase dituliskan ke referensi FirebaseDatabase dan diambil dengan menambahkan pemroses asinkron ke referensi tersebut. Pemroses dipicu satu kali untuk status awal data dan dipicu lagi setiap kali data berubah.

4.6.1. Mendapatkan DatabaseReference

Untuk membaca atau menulis data dari database, Anda memerlukan instance DatabaseReference:

Untuk java :

```
private DatabaseReference mDatabase;// ...
mDatabase = FirebaseDatabase.getInstance().getReference();
```

Untuk kotlin :

```
private lateinit var database: DatabaseReference// ...
database = FirebaseDatabase.getInstance().reference
```

4.6.2. Membaca dan Menulis Data

1. Operasi Tulis Dasar

Untuk operasi tulis dasar, Anda dapat menggunakan `setValue()` untuk menyimpan data ke referensi yang ditentukan, sehingga menggantikan data yang ada di jalur tersebut. Anda bisa menggunakan metode ini untuk:

- Meneruskan jenis yang cocok dengan jenis JSON yang tersedia berikut ini:
 - String
 - Long
 - Double
 - Boolean
 - Map<String, Object>
 - List<Object>
- Meneruskan objek Java kustom, jika class yang menentukannya memiliki konstruktor default yang tidak membutuhkan argumen, dan memiliki getter publik untuk properti yang akan ditetapkan.

Jika Anda menggunakan objek Java, konten objek Anda akan otomatis dipetakan ke lokasi turunan secara bertingkat. Jika objek Java digunakan, kode Anda biasanya jadi lebih mudah dibaca dan dipertahankan. Misalnya, jika Anda memiliki aplikasi dengan profil pengguna dasar, objek User Anda mungkin akan terlihat seperti berikut:

Untuk java :

```
@IgnoreExtraProperties
public class User {

    public String username;
    public String email;

    public User() {
        // Default constructor required for calls to DataSnapshot.getValue(User.class)
    }

    public User(String username, String email) {
        this.username = username;
        this.email = email;
    }

}
```

Untuk kotlin :

```
@IgnoreExtraProperties
data class User(
    var username: String? = "",
    var email: String? = ""
)
```

Anda dapat menambahkan pengguna dengan `setValue()` seperti berikut:

Untuk java :

```
private void writeNewUser(String userId, String name, String email) {
    User user = new User(name, email);

    mDatabase.child("users").child(userId).setValue(user);
}
```

Untuk kotlin :

```
private fun writeNewUser(userId: String, name: String, email: String?) {
    val user = User(name, email)
    database.child("users").child(userId).setValue(user)
}
```

Penggunaan `setValue()` seperti ini akan menimpa data di lokasi yang ditentukan, termasuk semua node turunan. Namun, Anda masih dapat mengupdate turunan tanpa menulis ulang seluruh objek. Jika ingin mengizinkan pengguna memperbarui profilnya, Anda dapat memperbarui nama pengguna seperti berikut:

Untuk java :

```
mDatabase.child("users").child(userId).child("username").setValue(name);
```

Untuk kotlin :

```
database.child("users").child(userId).child("username").setValue(name)
```

2. Memproses Peristiwa Nilai

Untuk membaca data di suatu lokasi dan memproses perubahan, gunakan metode `addValueEventListener()` atau `addListenerForSingleValueEvent()` untuk menambahkan `ValueEventListener` ke `DatabaseReference`.

Pemroses	Callback peristiwa	Penggunaan standar
<code>ValueEventListener</code>	<code>onDataChange()</code>	Membaca dan memproses perubahan di seluruh konten jalur.

Anda dapat menggunakan metode `onDataChange()` untuk membaca snapshot statis konten di jalur tertentu, saat konten tersebut ada selama peristiwa terjadi. Metode ini terpicu satu kali ketika pemroses terpasang dan terpicu lagi setiap kali terjadi perubahan pada data, termasuk pada setiap turunannya. Callback peristiwa mendapatkan snapshot yang berisi semua data di lokasi tersebut, termasuk data turunan. Jika tidak ada data, snapshot akan menampilkan false ketika Anda memanggil `exists()`, dan null ketika Anda memanggil `getValue()` pada snapshot tersebut.

Contoh berikut menunjukkan aplikasi blogging sosial yang mengambil detail suatu postingan dari database:

Untuk java :

```
ValueEventListener postListener = new ValueEventListener() {
    @Override
    public void onDataChange(DataSnapshot dataSnapshot) {
        // Get Post object and use the values to update the UI
        Post post = dataSnapshot.getValue(Post.class);
        // ...
    }

    @Override
    public void onCancelled(DatabaseError databaseError) {
        // Getting Post failed, log a message
        Log.w(TAG, "loadPost:onCancelled", databaseError.toException());
        // ...
    }
};
mPostReference.addValueEventListener(postListener);
```

Untuk kotlin :

```
val postListener = object : ValueEventListener {
    override fun onDataChange(dataSnapshot: DataSnapshot) {
        // Get Post object and use the values to update the UI
        val post = dataSnapshot.getValue(Post::class.java)
        // ...
    }

    override fun onCancelled(databaseError: DatabaseError) {
        // Getting Post failed, log a message
        Log.w(TAG, "loadPost:onCancelled", databaseError.toException())
        // ...
    }
}
postReference.addValueEventListener(postListener)
```

Pemroses akan menerima DataSnapshot yang berisi data di lokasi yang ditentukan dalam database saat peristiwa terjadi. Pemanggilan getValue() pada snapshot akan menampilkan perwakilan objek Java dari data. Jika tidak ada data di lokasi, null akan ditampilkan saat getValue() dipanggil.

Dalam contoh ini, ValueEventListener juga menentukan metode onCancelled() yang dipanggil jika operasi baca dibatalkan. Misalnya, proses baca dapat dibatalkan jika klien tidak memiliki izin untuk membaca dari lokasi database Firebase. Metode ini meneruskan objek DatabaseError yang menunjukkan alasan terjadinya kegagalan.

3. Membaca Data Sekali

Dalam beberapa kasus, Anda mungkin ingin agar callback dipanggil satu kali kemudian segera dihapus, seperti ketika melakukan inisialisasi elemen UI yang tidak diharapkan berubah. Anda dapat menggunakan metode `addListenerForSingleValueEvent()` untuk menyederhanakan skenario ini: metode ini dipicu satu kali dan tidak dipicu lagi.

Cara ini berguna untuk data yang hanya perlu dimuat sekali, dan tidak diharapkan untuk sering berubah atau harus aktif memproses. Misalnya, aplikasi blogging pada contoh sebelumnya menggunakan metode ini untuk memuat profil pengguna ketika mulai membuat postingan baru:

4.6.3. Mengupdate atau Menghapus data

1. Mengupdate kolom tertentu

Untuk menulis ke turunan tertentu dari sebuah node secara simultan tanpa menimpa node turunan yang lain, gunakan metode `updateChildren()`.

Saat memanggil `updateChildren()`, Anda dapat mengupdate nilai turunan pada level yang lebih rendah dengan menetapkan jalur untuk kunci tersebut. Jika data disimpan dalam beberapa lokasi agar dapat melakukan penskalaan yang lebih baik, Anda dapat mengupdate semua instance data tersebut menggunakan data fan-out. Misalnya, sebuah aplikasi blog sosial mungkin memiliki class Post seperti ini:

Untuk java :

```

@IgnoreExtraProperties
public class Post {

    public String uid;
    public String author;
    public String title;
    public String body;
    public int starCount = 0;
    public Map<String, Boolean> stars = new HashMap<>();

    public Post() {
        // Default constructor required for calls to DataSnapshot.getValue(Post.class)
    }

    public Post(String uid, String author, String title, String body) {
        this.uid = uid;
        this.author = author;
        this.title = title;
        this.body = body;
    }

    @Exclude
    public Map<String, Object> toMap() {
        HashMap<String, Object> result = new HashMap<>();
        result.put("uid", uid);
        result.put("author", author);
        result.put("title", title);
        result.put("body", body);
        result.put("starCount", starCount);
        result.put("stars", stars);

        return result;
    }
}

```

Untuk kotlin :

```

@IgnoreExtraProperties
data class Post(
    var uid: String? = "",
    var author: String? = "",
    var title: String? = "",
    var body: String? = "",
    var starCount: Int = 0,
    var stars: MutableMap<String, Boolean> = HashMap()
) {

    @Exclude
    fun toMap(): Map<String, Any?> {
        return mapOf(
            "uid" to uid,
            "author" to author,
            "title" to title,
            "body" to body,
            "starCount" to starCount,
            "stars" to stars
        )
    }
}

```

Untuk membuat postingan dan sekaligus mengupdatenya ke feed aktivitas terbaru serta memposting feed aktivitas pengguna, aplikasi blog menggunakan kode seperti ini:

Untuk java :

```
private void writeNewPost(String userId, String username, String title, String body) {
    // Create new post at /user-posts/$userid/$postid and at
    // /posts/$postid simultaneously
    String key = mDatabase.child("posts").push().getKey();
    Post post = new Post(userId, username, title, body);
    Map<String, Object> postValues = post.toMap();

    Map<String, Object> childUpdates = new HashMap<>();
    childUpdates.put("/posts/" + key, postValues);
    childUpdates.put("/user-posts/" + userId + "/" + key, postValues);

    mDatabase.updateChildren(childUpdates);
}
```

Untuk kotlin :

```
private fun writeNewPost(userId: String, username: String, title: String, body: String) {
    // Create new post at /user-posts/$userid/$postid and at
    // /posts/$postid simultaneously
    val key = database.child("posts").push().key
    if (key == null) {
        Log.w(TAG, "Couldn't get push key for posts")
        return
    }

    val post = Post(userId, username, title, body)
    val postValues = post.toMap()

    val childUpdates = HashMap<String, Any>()
    childUpdates["/posts/$key"] = postValues
    childUpdates["/user-posts/$userId/$key"] = postValues

    database.updateChildren(childUpdates)
}
```

Contoh ini menggunakan `push()` guna membuat postingan dalam node yang berisi postingan untuk semua pengguna di `/posts/$postid`, dan sekaligus mengambil kunci dengan `getKey()`. Selanjutnya, kunci tersebut dapat digunakan untuk membuat entri kedua di postingan pengguna pada `/user-posts/$userid/$postid`.

Dengan menggunakan jalur tersebut, Anda dapat menjalankan update simultan ke beberapa lokasi di pohon JSON dengan sekali panggilan ke `updateChildren()`, seperti yang digunakan pada contoh ini untuk membuat postingan baru di kedua lokasi. Melalui update simultan, proses ini berjalan menyeluruh: entah semua update berhasil atau semua update gagal.

2. Menambahkan Callback Penyelesaian

Jika Anda ingin tahu kapan data menerapkan commit, Anda bisa menambahkan pemroses penyelesaian. `setValue()` dan `updateChildren()` membawa pemroses penyelesaian opsional yang dipanggil ketika operasi tulis telah berhasil menerapkan commit pada database. Jika panggilan tidak berhasil, pemroses akan diteruskan ke objek error yang menunjukkan penyebab terjadinya kegagalan.

Untuk java :

```
mDatabase.child("users").child(userId).setValue(user)
    .addOnSuccessListener(new OnSuccessListener<Void>() {
        @Override
        public void onSuccess(Void aVoid) {
            // Write was successful!
            // ...
        }
    })
    .addOnFailureListener(new OnFailureListener() {
        @Override
        public void onFailure(@NonNull Exception e) {
            // Write failed
            // ...
        }
});
```

Untuk kotlin :

```
database.child("users").child(userId).setValue(user)
    .addOnSuccessListener {
        // Write was successful!
        // ...
    }
    .addOnFailureListener {
        // Write failed
        // ...
    }
```

3. Menghapus data

Cara termudah untuk menghapus data adalah dengan memanggil `removeValue()` pada referensi ke lokasi data tersebut.

Penghapusan juga dapat dilakukan dengan menentukan `null` sebagai nilai untuk operasi tulis lainnya, seperti `setValue()` atau `updateChildren()`. Teknik ini dapat digunakan dengan `updateChildren()` untuk menghapus beberapa turunan dengan sebuah panggilan API..

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 :

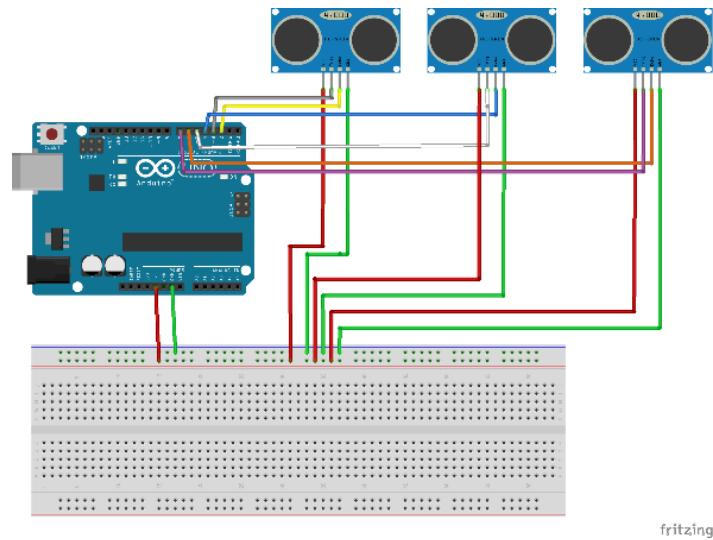
1. Arduino Uno R3
2. Kabel USB Arduino
3. PC/Laptop
4. *Software Arduino IDE*
5. *Jumper Wires*
6. *HC-SR04 Ultrasonic Sensor*

1.1.3. Langkah Pengujian

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

1. Merangkai *ultrasonic sensor* dengan *microcontroller*

Arduino Uno seperti dibawah ini :



fritzing

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);
}
```

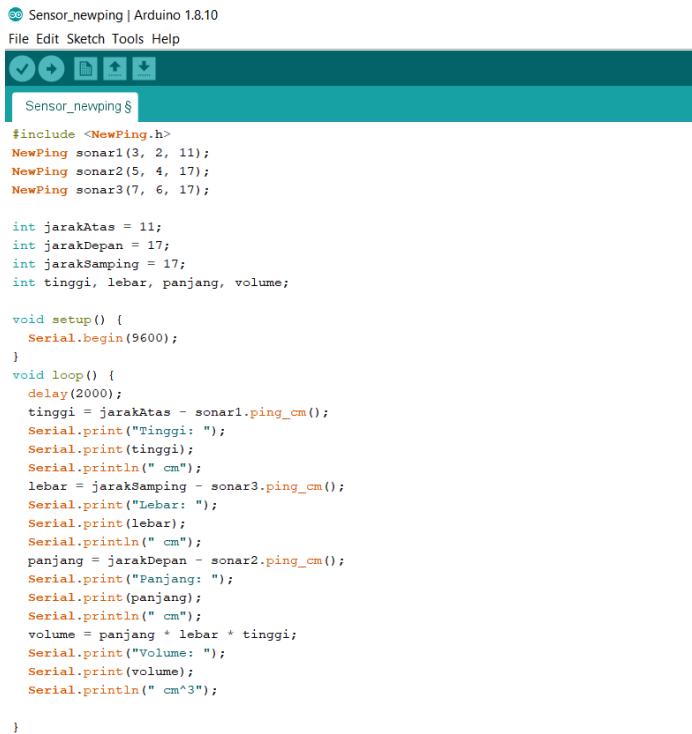
6. Kemudian pada metode Loop() kita tambahkan code yang dimana akan mendapatkan nilai tinggi, lebar panjang, dan perhitungan volume.

```
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

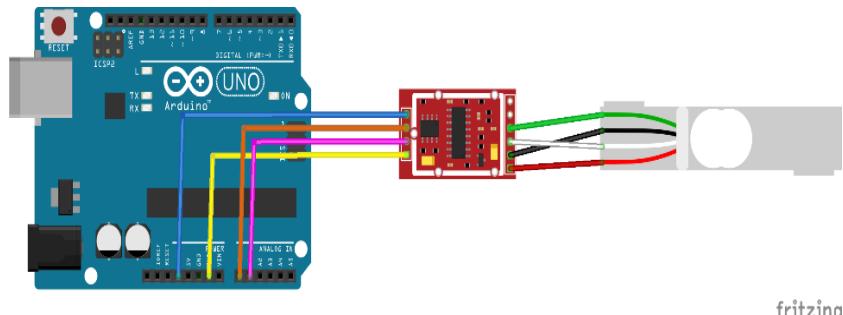
Alat yang digunakan untuk pengujian *ultrasonic sensor* adalah:

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

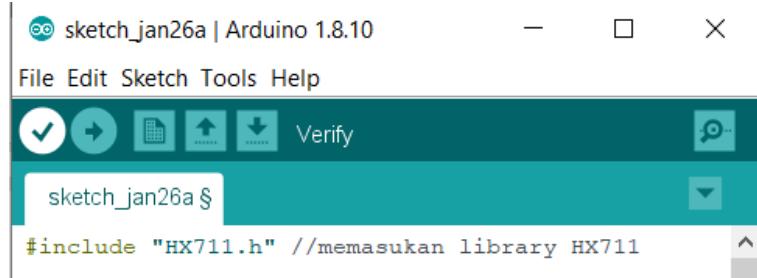
Langkah-langkah pengujian *HX711 dan Load Cell* adalah sebagai berikut :

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



Gambar 4.6 Rangkaian Pengujian HX711 dan LoadCell

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



```
sketch_jan26a | Arduino 1.8.10
File Edit Sketch Tools Help
Sketch_jan26a §
#include "HX711.h" //memasukan library HX711
```

- Kemudian kita mendefinisikan pin Arduino yang digunakan untuk *HX711*

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

- Kemudian kita membuat konstruktor sesuai dengan *library HX711*

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

- Membuat inisiasi variable kalibrasi

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

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

- 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 / mengenolkan 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
#define CLK A0 //mendefinisikan pin arduino yang terhubung dengan
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 / mengenolkan pembacaan berat

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

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();
}
}
```

10. Setelah itu buka serial monitor, dan melakukan kalibrasi sesuai dengan berat yang digunakan untuk kalibrasi, catat nilai calibration_factor.
11. Kemudain masukan niali 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
sketch_jan15b §

#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

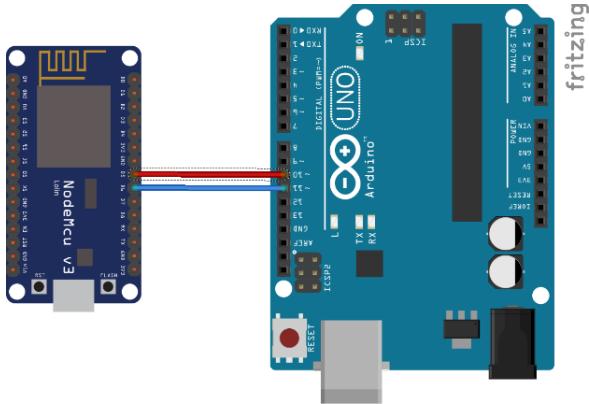
Alat yang digunakan yang digunakan untuk pengujian *Node Mcu* adalah:

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 MCUs*
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 *NodeMCU* 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 "TOuGgQ4K4zviJSKalkJVzR8fk2c1cTyP7RdWm4vw"
#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 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());
                    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
unsigned long previousMillis=0; // millis() returns:
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 "TOuGgQ4K4zviJSKalkJVzR8fk2c1cTyP7RdWm4vw"
#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
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 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());
                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.

1.4. Pengujian Aplikasi Android

1.4.1. Tujuan Pengujian

Pengujian Aplikasi Android bertujuan untuk memastikan Aplikasi Android bisa berjalan dengan baik. Aplikasi Android digunakan untuk menampilkan hasil pengukuran barang.

1.4.2. Alat yang Digunakan

Alat yang digunakan untuk pengujian Aplikasi *Android* adalah:

1. PC/Laptop
2. *Handphone*
3. *Software Android Studio*

1.4.3. Langkah Pengujian

Langkah-langkah pengujian *Android* adalah sebagai berikut :

1. Merangkai kode *source* pembuatan aplikasi *android* pada *activity DataPengukuran.java*:

```
package com.ilhammuhammadariq.pengukurandimensibarang;
import ...;

public class DataPengukuran extends AppCompatActivity {
    private TextView panjang,lebar,tinggi,volume,berat,beratD;
    private DatabaseReference databaseReference;
    private SwipeRefreshLayout swipe;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_data_pengukuran);

        swipe = findViewById(R.id.swipe);
        panjang = findViewById(R.id.tvpanjang);
        lebar = findViewById(R.id.tvlebar);
        tinggi = findViewById(R.id.tvttinggi);
        volume = findViewById(R.id.tvvolumne);
        berat = findViewById(R.id.tvberat);
        beratD = findViewById(R.id.tvberatD);

        swipe.setOnRefreshListener(() -> {
            firebase();
            swipe.setRefreshing(false);
        });
    }
}
```

```

3     public void firebase() {
4         DatabaseReference = FirebaseDatabase.getInstance().getReference();
5         Query lastdata = DatabaseReference.child("/Pengukuran").orderByKey().limitToLast(1);
6         lastdata.addListenerForSingleValueEvent( new ValueEventListener() {
7             @Override
8             public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
9                 for(DataSnapshot data : dataSnapshot.getChildren()) {
10                     String pjg = data.child("panjang").getValue().toString();
11                     String lbr = data.child("lebar").getValue().toString();
12                     String tgi = data.child("tinggi").getValue().toString();
13                     String vlm = data.child("volume").getValue().toString();
14                     String brt = data.child("berat").getValue().toString();
15                     String brtdmns = data.child("beratdimensi").getValue().toString();
16
17                     panjang.setText(pjg + " cm");
18                     lebar.setText(lbr + " cm");
19                     tinggi.setText(tgi + " cm");
20                     volume.setText(vlm + " cm³");
21                     berat.setText(brt + " gram");
22                     beratd.setText(brtdmns + " gram");
23
24                     int dtberat = Integer.parseInt(brt);
25                     int dtberatd = Integer.parseInt(brtdmns);
26                     int selisih = 0;
27                     int selisihl = 0;
28                     selisih = dtberat - dtberatd;
29                     selisihl = dtberatd - dtberat;
30
31                     if(dtberat < dtberatd) {
32                         AlertDialog.Builder dialog = new AlertDialog.Builder( context: DataPengukuran.this );
33                         dialog.setTitle("Pemberitahuan");
34                         dialog.setMessage("BERAT DIMENSI LEBIH BESAR DARIPADA BERAT ASLI . SELISIHNYA SEBESAR " + selisih + " GRAM");
35                         dialog.setCancelable(true);
36                         AlertDialog alert = dialog.create();
37                         alert.show();
38                     } else if(dtberat > dtberatd) {
39                         AlertDialog.Builder dialog = new AlertDialog.Builder( context: DataPengukuran.this );
40                         dialog.setTitle("Pemberitahuan");
41                         dialog.setMessage("BERAT ASLI LEBIH BESAR DARIPADA BERAT DIMENSI . SELISIHNYA SEBESAR " + selisih + " GRAM");
42                         dialog.setCancelable(true);
43                         AlertDialog alert = dialog.create();
44                         alert.show();
45                     } else if(dtberat == dtberatd) {
46                         AlertDialog.Builder dialog = new AlertDialog.Builder( context: DataPengukuran.this );
47                         dialog.setTitle("Pemberitahuan");
48                         dialog.setMessage("BERAT DIMENSI SAMA BESAR DENGAN BERAT ASLI");
49                         dialog.setCancelable(true);
50                         AlertDialog alert = dialog.create();
51                         alert.show();
52                     }
53                 }
54             }
55             @Override
56             public void onCancelled(@NonNull DatabaseError databaseError) {
57             }
58         });
59     }

```

Gambar 4.8 *Source Code* DataPengukuran.java

Codingan ini menjelaskan bahwa *Activity* ini dijalankan untuk menampilkan data pengukuran barang.

2. Pembuatan *source code* pada *activity* SplashScreen.java

```

package com.ilhammuhammadariq.pengukurandimensibarang;

import ...

public class SplashScreen extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_splash_screen);

        new Handler().postDelayed(() -> {
            Intent home = new Intent( packageContext: SplashScreen.this, DataPengukuran.class);
            startActivity(home);
            finish();
        }, delayMillis: 3000);
    }
}

```

Gambar 4.9 *Source code* SplashScreen.java

Codingan ini menjelaskan bahwa *Activity* ini dijalankan pertama kali untuk memunculkan gambar logo aplikasi beberapa detik.

3. Pembuatan *source code* pada layout *Activity* Data Pengukuran

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.swiperefreshlayout.widget.SwipeRefreshLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/swipe"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:background="#2196F3"
    android:orientation="vertical"
    tools:context=".DataPengukuran">

    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_weight="1">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:orientation="vertical">

            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:orientation="vertical">

```

```
<TextView  
    android:id="@+id/textView"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_marginTop="50dp"  
    android:layout_weight="1"  
    android:gravity="center_horizontal"  
    android:text="DATA PENGUKURAN"  
    android:textColor="@android:color/white"  
    android:textSize="30sp"  
    android:textStyle="bold" />  
  
<LinearLayout  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_gravity="center"  
    android:layout_marginTop="50dp"  
    android:layout_marginBottom="50dp"  
    android:layout_weight="1"  
    android:orientation="horizontal">  
  
    <LinearLayout  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:layout_gravity="center"  
        android:layout_marginRight="50dp"  
        android:orientation="vertical">  
  
        <LinearLayout  
            android:layout_width="wrap_content"  
            android:layout_height="wrap_content"  
            android:layout_marginBottom="30dp"  
            android:orientation="vertical">  
  
            <TextView  
                android:id="@+id/textView2"  
                android:layout_width="wrap_content"  
                android:layout_height="wrap_content"  
                android:layout_gravity="center_horizontal"  
                android:layout_marginBottom="10dp"  
                android:layout_weight="1"  
                android:text="Panjang"  
                android:textColor="@android:color/white"  
                android:textSize="15sp" />  
  
            <TextView  
                android:id="@+id/tvpanjang"  
                android:layout_width="wrap_content"  
                android:layout_height="wrap_content"  
                android:layout_weight="1"  
                android:background="@drawable/circle"  
                android:gravity="center"  
                android:text="0"  
                android:textColor="@android:color/white"  
                android:textSize="15sp" />  
        </LinearLayout>
```

```
<LinearLayout  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_weight="1"  
    android:orientation="vertical">  
  
    <TextView  
        android:id="@+id/textView3"  
        android:layout_width="match_parent"  
        android:layout_height="wrap_content"  
        android:layout_marginBottom="10dp"  
        android:layout_weight="1"  
        android:gravity="center_horizontal"  
        android:text="Lebar"  
        android:textColor="@android:color/white"  
        android:textSize="15sp" />  
  
    <TextView  
        android:id="@+id/tvlebar"  
        android:layout_width="match_parent"  
        android:layout_height="wrap_content"  
        android:layout_weight="1"  
        android:background="@drawable/circle"  
        android:gravity="center"  
        android:text="0"  
        android:textColor="@android:color/white"  
        android:textSize="15sp" />  
    </LinearLayout>  
  
</LinearLayout>  
  
<LinearLayout  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_gravity="center"  
    android:layout_weight="1"  
    android:orientation="vertical">  
  
    <LinearLayout  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:layout_marginBottom="30dp"  
        android:orientation="vertical">  
  
        <TextView  
            android:id="@+id/textView4"  
            android:layout_width="match_parent"  
            android:layout_height="wrap_content"  
            android:layout_marginBottom="10dp"  
            android:layout_weight="1"  
            android:gravity="center_horizontal"  
            android:text="Tinggi"  
            android:textColor="@android:color/white"  
            android:textSize="15sp" />
```

```
<TextView  
    android:id="@+id/tvinggi"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_weight="1"  
    android:background="@drawable/circle"  
    android:gravity="center"  
    android:text="0"  
    android:textColor="@android:color/white"  
    android:textSize="15sp" />  
  
</LinearLayout>  
  
<LinearLayout  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_weight="1"  
    android:orientation="vertical">  
  
    <TextView  
        android:id="@+id/textView5"  
        android:layout_width="match_parent"  
        android:layout_height="wrap_content"  
        android:layout_marginBottom="10dp"  
        android:layout_weight="1"  
        android:gravity="center_horizontal"  
        android:text="Volume"  
        android:textColor="@android:color/white"  
        android:textSize="15sp" />  
  
<TextView  
    android:id="@+id/tvvolume"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_weight="1"  
    android:background="@drawable/circle"  
    android:gravity="center"  
    android:text="0"  
    android:textColor="@android:color/white"  
    android:textSize="15sp" />  
    </LinearLayout>  
    </LinearLayout>  
  
<LinearLayout  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_gravity="center"  
    android:layout_weight="10"  
    android:orientation="horizontal">  
  
    <LinearLayout  
        android:layout_width="match_parent"  
        android:layout_height="match_parent"  
        android:layout_marginRight="30dp"  
        android:orientation="vertical">
```

```
        )
```

```
        <TextView  
            android:id="@+id/textView7"  
            android:layout_width="match_parent"  
            android:layout_height="wrap_content"  
            android:layout_gravity="center_horizontal"  
            android:layout_marginBottom="10dp"  
            android:gravity="center"  
            android:text="Berat"  
            android:textColor="@android:color/white"  
            android:textSize="20sp" />
```

```
    )
```

```
    <TextView  
        android:id="@+id/tvberat"  
        android:layout_width="match_parent"  
        android:layout_height="wrap_content"  
        android:background="@drawable/circle2"  
        android:gravity="center"  
        android:text="0"  
        android:textColor="@android:color/white"  
        android:textSize="20sp" />
```

```
    )
```

```
</LinearLayout>
```

```
)
```

```
<LinearLayout  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:orientation="vertical">
```

```
    )
```

```
        <TextView  
            android:id="@+id/textView9"  
            android:layout_width="match_parent"  
            android:layout_height="wrap_content"  
            android:layout_marginBottom="10dp"  
            android:gravity="center_horizontal"  
            android:text="Berat D"  
            android:textColor="@android:color/white"  
            android:textSize="20sp" />
```

```
    )
```

```
        <TextView  
            android:id="@+id/tvberatD"  
            android:layout_width="match_parent"  
            android:layout_height="wrap_content"  
            android:background="@drawable/circle2"  
            android:gravity="center"  
            android:text="0"  
            android:textColor="@android:color/white"  
            android:textSize="20sp" />
```

```
    )
```

```
        </LinearLayout>
```

```
</LinearLayout>
```

```
)
```

```
</LinearLayout>
```

```
)
```

```
<RelativeLayout  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:layout_marginTop="40dp">
```

```
    )
```

```
        <TextView  
            android:id="@+id/tv_copy"  
            android:layout_width="match_parent"  
            android:layout_height="wrap_content"  
            android:layout_alignParentBottom="true"  
            android:gravity="center_horizontal"  
            android:text="@string/copyright"  
            android:textColor="@android:color/white" />
```

```
    )
```

```
    </RelativeLayout>
```

```
)
```

```
</LinearLayout>
```

```
)
```

```
</ScrollView>
```

```
)</androidx.swiperefreshlayout.widget.SwipeRefreshLayout>
```

Asdas

Codingan ini digunakan untuk menampilkan layout *Activity*
Data Pengukuran

4. Pembuatan *source code* pada layout *Activity* Splash Screen

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#FFFFFF"
    android:gravity="center"
    tools:ignore="ContentDescription">

    <ImageView
        android:id="@+id/logo"
        android:layout_width="260dp"
        android:layout_height="160dp"
        android:src="@drawable/logoapp" />
</RelativeLayout>
```

5. Pembuatan *Source code* Drawable Circle.xml

```
<?xml version="1.0" encoding="utf-8"?>
<!-- Background circle for the magnitude value -->
<shape xmlns:android="http://schemas.android.com/apk/res/android"
    android:shape="ring"
    android:useLevel="false"
    android:innerRadius="32dp"
    android:thicknessRatio="15">

    <gradient
        android:startColor="#FFFFFF"
        android:centerColor="#FFFFFF"
        android:endColor="#FF000000"
        android:angle="135"/>

    <size
        android:width="80dp"
        android:height="80dp" />
</shape>
```

6. Pebuatan *Source code* Drawable Circle2.xml

```

<?xml version="1.0" encoding="utf-8"?>
<!-- Background circle for the magnitude value -->
<shape xmlns:android="http://schemas.android.com/apk/res/android"
    android:shape="ring"
    android:useLevel="false"
    android:innerRadius="50dp"
    android:thicknessRatio="15">

    <gradient
        android:startColor="#FFFFFF"
        android:centerColor="#FFFFFF"
        android:endColor="#FF000000"
        android:angle="135"/>

    <size
        android:width="120dp"
        android:height="120dp"/>
</shape>
```

7. Edit File AndroidManifest.xml

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.ilhammuhammadariq.pengukurandimensibarang">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="PengukuranDimensiBarang"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity"></activity>
        <activity android:name=".DataPengukuran" />
        <activity android:name=".SplashScreen">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

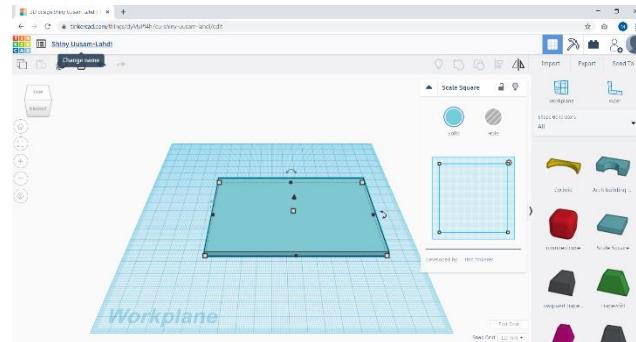
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

1.4.4. Hasil Pengujian

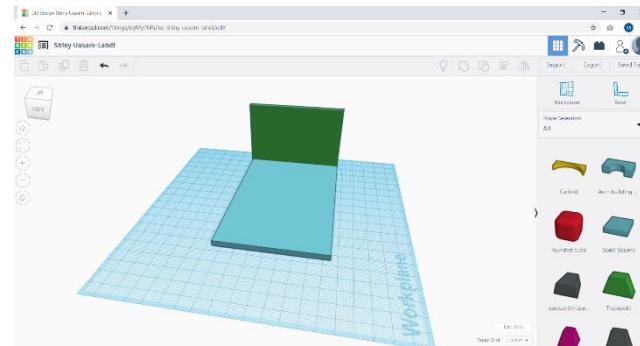
Dari pengujian Aplikasi *Android* yang telah dilakukan, membuktikan bahwa Aplikasi *Android* dapat berfungsi dengan baik. Maka Aplikasi *Android* dapat menampilkan data pengukuran barang.

2. Selanjutnya kita membuat/mendesign Hardware dari prototype yang dimana digunakan sebagai wadah atau tempat yang digunakan. Pada buku ini kita melakukan design hardware prototype menggunakan “Tinkercad”. Kenapa menggunakan tinkercad karena gratis dan berbasis web sehingga kita tidak perlu mendownload dan install lagi.

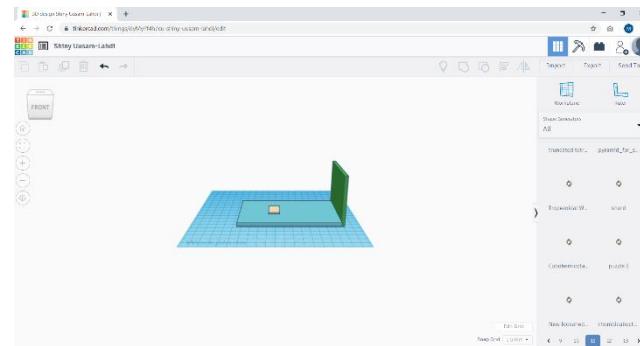
- 2.1. Buka website tinkercad : <https://www.tinkercad.com/>
- 2.2. Lakukan login terlebih dahulu, kita dapat menggunakan google account
- 2.3. Setelah login, pilih Create new design
- 2.4. Kemudian kita akan memilih objek yang akan digunakan untuk membangun sebuah hardware yang kita inginkan
- 2.5. Pada hardware ini kita membutuhkan papan/triplek sebagai alasnya, oleh karena itu pilih shape generators (all), pada halaman 10 pilih “Scale Squere”. Seperti gambar dibawah:



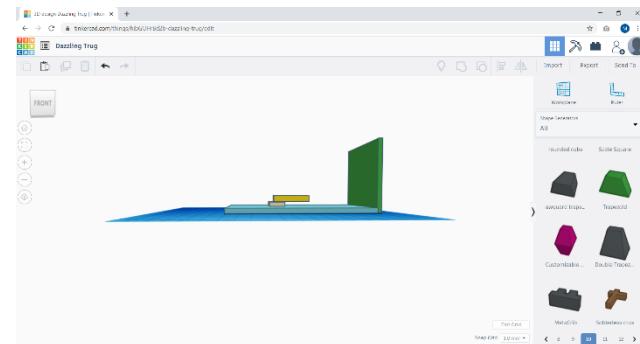
- 2.6. Dan kita juga butuh papan/triplek pada sisi kanan, sebagai batas dan tempat sensor ultrasonic. Seperti gambar dibawah:



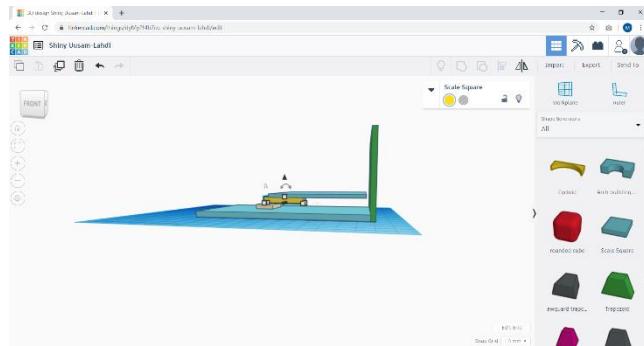
2.7. Kemudian kita butuh papan/triplek lagi sebagai penopang load cell yang digunakan untuk mendapatkan sinyal listrik dari berat benda diatasnya. Seperti gambar dibawah:



2.8. Letakan load cell pada penopang yang telah di buat tadi. Pada design ini kita menggambarkan load cell menggunakan objek persegi Panjang berwarna kuning. Seperti gambar dibawah:



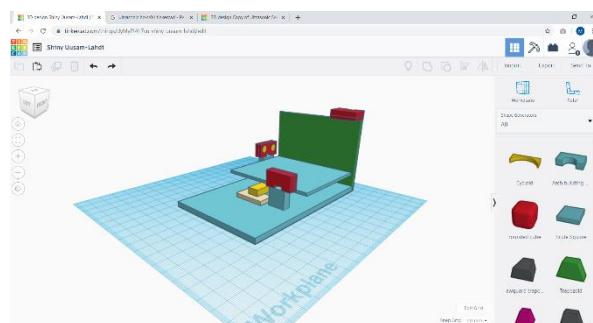
2.9. Lalu kita membutuhkan akrilik sebagai alas untuk benda yang di timbang/ukur pada prototype ini. Akrilik tersebut diletakan di atas load cell. Seperti gambar dibawah:



2.10. Kemudian kita membutuhkan sensor ultrasonic untuk mengetahui dimana titik 0, sehingga kita dapat meletak sensor ultrasonic pada tempatnya.

2.11. Pada tinkercad kami mengambil objek sensor ultrasonic pada <https://www.tinkercad.com/things/dzxjh1kFYYY-ultrasonic-sensor-hc-sr04>

2.12. Kemudian kita copy pada project yang telah kita buat tadi, selanjutnya kita melakukan penyesuaian sensor ulrsasonik tersebut dari tata letak maupun ukuran dan tempatnya. Seperti gambar dibawah:



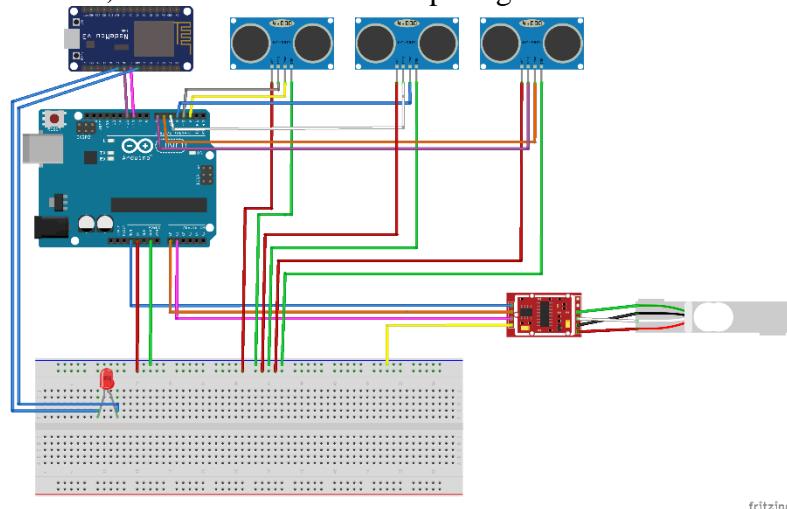
3. Setelah melakukan pengujian pada setiap alat yang kita akan gunakan untuk membangun prototype ini, selanjutnya kita akan menggabungkan alat yang di atas semua menjadi satu dan ditambahkan komponen LED sebagai lampu indicator.

3.1. 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

3.2. 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();
}

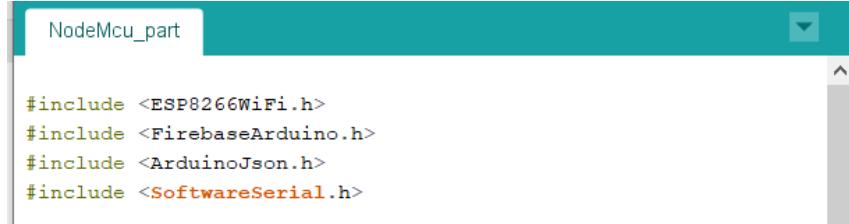
void loop() {
StaticJsonBuffer<1000> jsonBuffer;
JsonObject& root = jsonBuffer.createObject();
delay(1000);

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

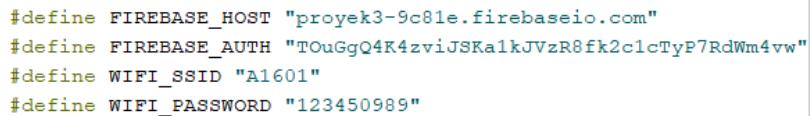
internet dan firebase sehingga bisa meneruskan data json tersebut ke firebase. Pertama seperi biasa kita mengimport library yang kita butuhkan, seperti berikut :



```
NodeMcu_part

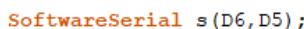
#include <ESP8266WiFi.h>
#include <FirebaseArduino.h>
#include <ArduinoJson.h>
#include <SoftwareSerial.h>
```

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



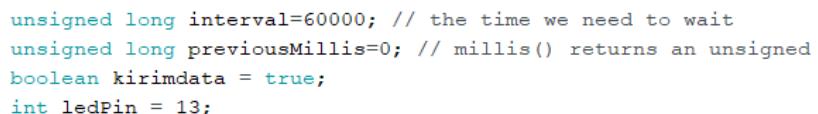
```
#define FIREBASE_HOST "proyek3-9c81e.firebaseio.com"
#define FIREBASE_AUTH "TOuGgQ4R4zviJSKa1kJVzR8fk2c1cTyP7RdWm4vw"
#define WIFI_SSID "A1601"
#define WIFI_PASSWORD "123450989"
```

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 karena 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(700);
    digitalWrite(ledPin,HIGH);
}
```

18. Lalu compile dan upload ke board NodeMCU menggunakan port yang terdeteksi

4. Serlanjutnya kita membuat Hardware dari prototype yang dimana digunakan sebagai wadah atau tempat yang digunakan.