



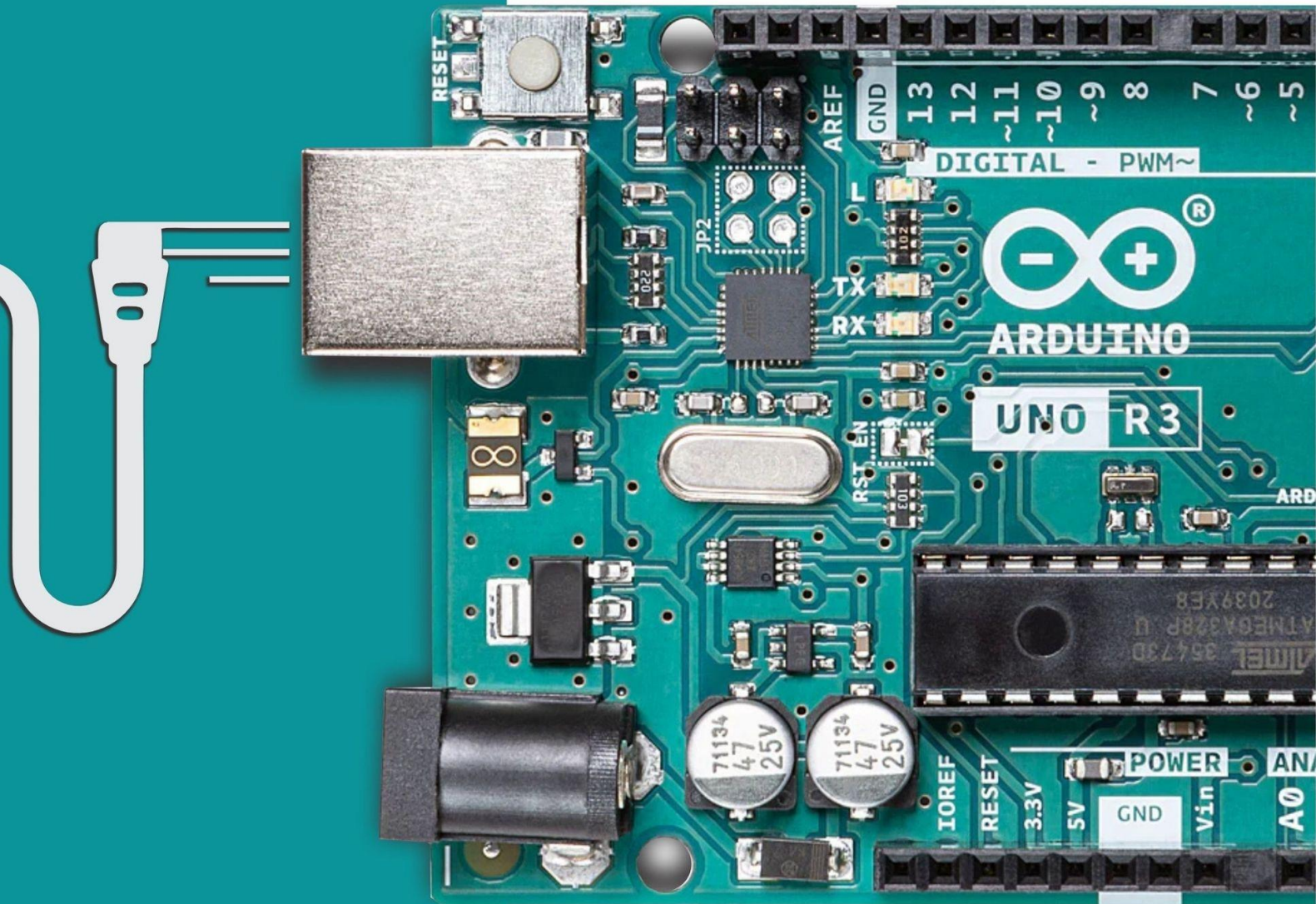
Panduan Praktis Proyek



loop()

setup()

ARDUINO



Safiq Rosad
Adhiya Nur Septio
Muhamad Rifqi Kamali

PANDUAN PRAKTIS PROYEK ARDUINO

Penulis :

Safiq Rosad

Adhiya Nur Septio

Mohamad Rifki

PENERBIT:



HADLA
MEDIA INFORMASI

UU No 28 tahun 2014 tentang Hak Cipta

Pasal 113

- 1)** Setiap Orang yang dengan tanpa hak melakukan pelanggaran hak ekonomi sebagaimana dimaksud dalam Pasal 9 ayat (1) huruf i untuk Penggunaan Secara Komersial dipidana dengan pidana penjara paling lama 1 (satu) tahun dan/atau pidana denda paling banyak Rp 100.000.000 (seratus juta rupiah).
- 2)** Setiap Orang yang dengan tanpa hak dan/atau tanpa izin Pencipta atau pemegang Hak Cipta melakukan pelanggaran hak ekonomi Pencipta sebagaimana dimaksud dalam Pasal 9 ayat (1) huruf c, huruf d, huruf f, dan/atau huruf h untuk Penggunaan Secara Komersial dipidana dengan pidana penjara paling lama 3 (tiga) tahun dan/atau pidana denda paling banyak Rp. 500.000.000,00 (lima ratus juta rupiah).
- 3)** Setiap Orang yang dengan tanpa hak dan/atau tanpa izin Pencipta atau pemegang Hak Cipta melakukan pelanggaran hak ekonomi Pencipta sebagaimana dimaksud dalam Pasal 9 ayat (1) huruf a, huruf b, huruf e, dan/atau huruf g untuk Penggunaan Secara Komersial dipidana dengan pidana penjara paling lama 4 (empat) tahun dan/ atau pidana denda paling banyak Rp1.000.000.000,00 (satu miliar rupiah).
- 4)** Setiap Orang yang memenuhi unsur sebagaimana dimaksud pada ayat (3) yang dilakukan dalam bentuk pembajakan, dipidana dengan pidana penjara paling lama 10 (sepuluh) tahun dan/atau pidana denda paling banyak Rp. 4.000.000.000,00 (empat miliar rupiah).

Panduan Praktis Proyek Arduino

Tim Penulis:

Safiq Rosad

Adhiya Nur Septio

Mohamad Rifki

Editor:

Linda Susi Susanti

Desain Cover:

Safiq Rosad

Tata Letak:

Safiq Rosad

ISBN:

-

Cetakan Pertama:

Desember, 2025

Hak Cipta 2025, Pada Penulis

Hak Cipta Dilindungi Oleh Undang-Undang

Copyright © 2025

by HADLA Media Informasi

All Right Reserved

Dilarang keras menerjemahkan, memfotokopi, atau memperbanyak sebagian atau seluruh isi buku ini tanpa izin tertulis dari Penerbit.

PENERBIT:



Website: www.media.hadlacorp.com

PRAKARTA

Semakin populernya teknologi prosesor saat ini, terutama teknologi *Microcontroller* saat ini sudah menjadi barang konsumsi publik yang telah umum digunakan, tidak hanya bagi kalangan profesional dan kalangan akademisi saja, namun para penghobi juga turut meramaikan kancah persaingan karya berteknologi dengan berbasiskan mikrokontroler. Sehingga berbagai sumber informasi mengenai mikrokontroler semakin mudah ditemukan, baik dari perkumpulan para penghobi, tutorial diinternet, maupun buku aplikasi terapan mikrokontroler. Tidak hanya itu saja, para pengembang juga ikut serta berlomba-lomba membuat terobosan untuk menciptakan kemudahan bagi para pengguna mikrokontroler dari segi perangkat keras maupun perangkat lunak untuk dapat lebih mudah dipelajari dan semakin cepat tingkat produktifitasnya.

Dengan kepopuleranya mikrokontroler, saat ini terciptalah platform perangkat keras yang dikenal dengan Arduino. Dengan platform ini, semakin mempermudah dalam pengembangan proyek berbasis mikrokontroler, tidak diwajibkan seseorang berproduktivitas dengan mikrokontroler harus mengerti dengan rangkaian elektronik. Bahkan bagi para pemula yang tanpa ada dasar perangkat keras pun mampu berkreasi dengan mikrokontroler. Apalagi dilihat dari segi perangkat lunak. Banyak pilihan bahasa pemrograman yang dapat diambil, yang paling populer saat ini seperti bahasa Lua, C, C++, dan MicroPython. Bahasa yang paling populer dan mudah untuk dipelajari terutama banyaknya dukungan informasi adalah bahasa C atau C++ yang terkemas didalam platform yang sangat terkenal yaitu Arduino IDE. Dengan dukungan ratusan library, dan komunitas Arduino IDE, menjadikan Arduino IDE semakin banyak diminati oleh banyak orang.

Sebegitu mudahnya kita mendapatkan informasi mengenai mikrokontroler dan Arduino IDE, apalagi banyaknya tutorial/contoh aplikasi penunjang eksperimen yang disajikan secara instan kadang akan berdampak buruk juga, kita akan merasa malas dan enggan untuk menggali konsep dasar secara teori mengenai metode yang digunakan. Sehingga kreativitas kita pun akan bergantung pada tutorial yang ada (tanpa tutorial/contoh tidak bisa berkarya). Sehingga bila terdapat teknologi terbaru atau dengan metode algoritma yang beda, kita akan mempelajarinya dari awal lagi.

Tidak hanya itu saja, bila mana kita mempelajari mikrokontroler bertujuan untuk menghasilkan suatu produk yang sekiranya akan dipasarkan ke masyarakat luas, dalam perancangan hardware maupun software juga akan dituntut untuk mengikuti sesuai dengan prosedur konsep dasar yang ada. Bisa dibayangkan bilamana kita akan memproduksi sesuatu alat yang didalamnya menggunakan development board Arduino dengan konektivitas menggunakan kabel jumper. Tentu saja kestabilan produk alat tersebut kurang diandalkan dan rentan terhadap kerusakan, karena

kabel jumper dan Arduino board hanyalah diperuntukkan pada saat kita membangun prototipe. Hal inilah yang sering penulis temui di lapangan.

Dari latar belakang inilah, penulis menyuguhkan pembahasan yang tak terputus dengan konsep dasar yang ada akan digali lebih lanjut. Sehingga istilah rangkaian yang ditemui, tidak menjadi kebingungan bagi para pembacanya. Dengan metode pembahasan seperti ini, pembaca akan mendapatkan konsep dasar yang mendalam sebagai peningkatan pengembangan dan kreativitas. Sehingga apabila dihadapkan dengan teknologi yang baru lagi, kita hanya cukup sedikit untuk beradaptasinya.

Buku ini menyajikan secara lengkap untuk memulai mempelajari pemrograman Arduino menggunakan Arduino IDE. Pembaca dituntun untuk benar-benar mengerti konsep dasar dan mengetahui secara detail proses-proses yang terjadi. Hasil yang diharapkan dengan pendekatan pada buku ini, kita akan mudah untuk beradaptasi dengan berbagai permasalahan yang dihadapi dalam pembuatan perangkat lunak maupun perangkat keras.

Akhir kata, sebagai penulis, mengucapkan terimakasih kepada semua pihak yang telah membantu pada saat penyusunan buku ini, spesial untuk SHATOMEDIA yang telah memberikan semua fasilitas yang dibutuhkan dan Universitas Nahdlatul Ulama Al Ghazali Cilacap khususnya Program Studi Informatika sebagai penyemangat penyusunan buku ini. Semoga buku ini memberikan pengetahuan bagi pembacanya dan menjadi amal jariyah bagi penulis dan keluarga. Masih banyak kelemahan dan kekurangan didalam buku ini, penulis menghaturkan mohon maaf sebesar-besarnya.

Yogyakarta, Desember 2025

Penulis

DAFTAR ISI

PANDUAN PRAKTIS PROYEK ARDUINO.....	1
PRAKARTA	5
DAFTAR ISI.....	7
DAFTAR GAMBAR.....	12
DAFTAR TABEL	14
ARDUINO.....	16
Pengenalan Arduino	17
Evolusi Arduino.....	18
Setting Up	22
Instalasi Arduino IDE.....	23
Struktur Bahasa Pemrograman	27
Fungsi void setup().....	28
Fungsi void loop()	28
Sintaks Arduino.....	29
Tipe Data Arduino C.....	32
Arduino Mode Umum.....	33
Komponen Lingkup Pengembangan.....	35
LIBRARY	42
Menambahkan Library pada Arduino IDE.....	43
Mengimpor Library dari File ZIP.....	44
Menambahkan Library Secara Manual	46
Daftar Library Proyek pada Buku.....	47
ALAT UKUR MULTIMETER	48
Bagian-bagian Multimeter	50
Multimeter Analog	52
Multimeter Digital.....	53
LED Berkedip.....	58
Light Emitting Diode	59

Konfigurasi LED Aktif Tinggi	59
Konfigurasi LED Aktif Rendah.....	59
Rangkaian LED Berkedip	60
Sketch Led Berkedip.....	61
Penjelasan Kode Program:.....	62
Catatan rogram:.....	63
LED Berjalan	64
Rangkaian LED Berjalan.....	65
Sketch Led Berjalan	66
Catatan Program:	67
Kendali LED Dengan Tombol.....	68
Rangkaian Kendali LED dengan Tombol	70
Sketch Kendali LED dengan Tombol	71
Catatan Program:	71
Menampilkan Nilai Potensiometer Pada Serial Monitor	72
Potensiometer.....	73
Rangkaian Menampilkan Nilai Potensiometer.....	74
Sketch Menampilkan Nilai Potensiometer pada Serial Monitor	75
Catatan Program:	75
Sensor Analog LM35	76
Pengertian Sensor LM35.....	77
Rangkaian Sensor Analog LM35	78
Sketch Sensor Analog LM35	79
Catatan Program:	79
LED DIMMER.....	80
Mengatur Tegangan DC	81
Rangkaian LED Dimmer.....	82
Sketch LED Dimmer	83
Catatan Program:	83
Sensor ULTRASONIC.....	84
Sensor Ultra Sonic	85
Rangkaian Ultrasonic HC-SR-04	86
Sketch Sensor ULTRASONIC HC-SR04.....	87
Catatan Program:	88

Kendali Motor SERVO.....	89
Prinsip Kerja Motor Servo.....	90
Rangkaian Kendali Motor Servo	91
Sketch Kendali Motor Servo	93
Catatan Program:.....	94
Kendali Motor DC.....	95
Mengendalikan Motor DC.....	96
Rangkaian Kendali Motor DC.....	98
Sketch Kendali Motor DC.....	99
Catatan Program:.....	100
Membaca RFID.....	101
Tentang RFID	102
Rangkaian Kendali LED dengan Tombol.....	103
Sketch Kendali LED dengan Tombol.....	104
Catatan Program:.....	105
Display LCD	107
Menampilkan Teks Pada Display LCD	108
Rangkaian LCD 16x2 dan Potensiometer.....	109
Sketch Menampilkan Teks Pada LCD.....	110
Catatan Program:.....	110
Display LCD I2C	111
Tentang I2C.....	112
Protokol I2C.....	114
Rangkaian LCD dengan Modul I2C.....	117
Sketch LCD Menggunakan Modul I2C.....	118
Catatan Program:.....	118
KTP Reader dengan Suara.....	119
Tentang DF Mini Player.....	120
Rangkaian KTP Reader.....	122
Sketch KTP Reader.....	123
Catatan Program:.....	125
Running Teks.....	127
Tentang LED Matriks P10	128
Rangkaian LED Matrix P10	129

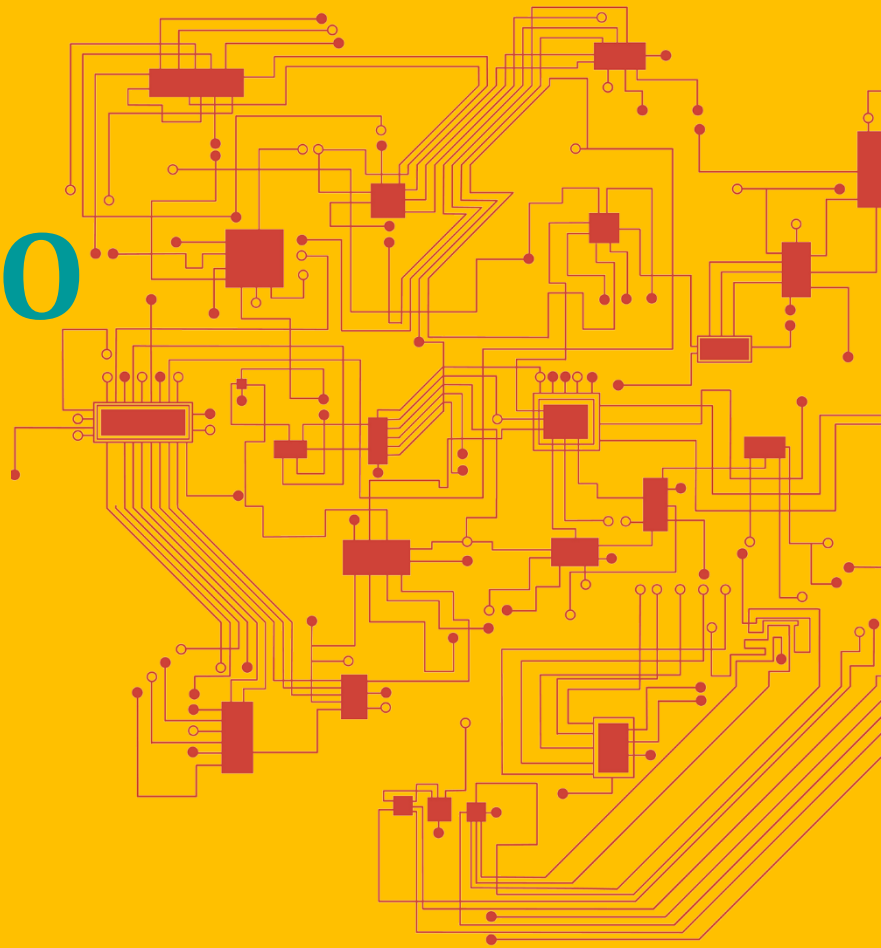
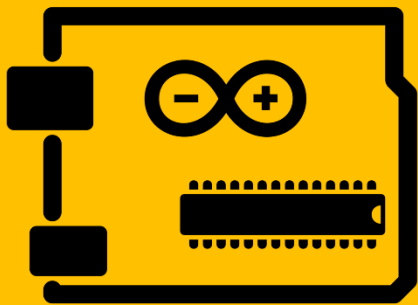
Sketch LED Matrix P10	130
Catatan Program:	131
DAFTAR PUSTAKA.....	133
Tentang Penulis	134

DAFTAR GAMBAR

DAFTAR TABEL

Tabel 1. Tabel Daftar Tipe Data Nilai Dasar	32
Tabel 2. Tabel Fungsi Selektor pada multimeter Digital atau multimeter Analog.....	50

ARDUINO



Saat ini manusia sudah mulai ketergantungan dengan perangkat elektronika, semakin kompleks tugas perangkat elektronika dalam membantu kegiatan manusia, semakin kompleks pula sistem yang dirancang. Sehingga manusia selalu mengembangkan dan berusaha menyederhanakan kekomplekan kebutuhan perangkat elektronika. Setiap hari kita menggunakan teknologi, kebanyakan dari kita enggan untuk mencari tahu dan mengetahui bagaimana teknologi tersebut terwujud sampai teknologi tersebut bisa dinikmati oleh setiap orang didunia ini, karena teknologi seperti dalam lingkup elektronika itu dianggap rumit dan sulit, terlebih lagi dengan kode-kode program untuk mewujudkannya menjadi suatu teknologi. Anggapan tersebut pada saat sekarang mungkin tidak relevan lagi, karena, ternyata teknologi tercipta bukan hanya dari kalangan programmer profesional maupun para peneliti, namun para penghoby, seniman, siswa dari segala usia belajar menciptakan sesuatu teknologi yang dapat diterapkan dilingkungan mereka masing-masing. Hal ini tidak terlepas dari peran ARDUINO.

ARDUINO menampilkan keadaan yang dianggap sulit dan rumit, serta kode program yang bertel-tele mengharuskan keahlian lebih tinggi, semuanya menjadikan mudah, menyenangkan dan megasyikkan. Selama bertahun-tahun Arduino telah digunakan sebagai “otak” bagi ribuan proyek. Di seluruh dunia komunitas pencipta teknologi telah berkumpul di sekitar platform Open Source Arduino ini, beralih dari komputasi personal ke personal fabrikasi, dan berkontribusi pada dunia baru, kerjasama dan berbagi.

Pengenalan Arduino



Arduino didefinisikan sebagai sebuah platform elektronik yang *open source*. Berbasis pada software dan hardware yang fleksibel dan mudah digunakan, ditujukan untuk penghobi, seniman, desainer, dan setiap orang yang tertarik dalam membuat sebuah objek pada lingkungan elektronik. Dengan sifatnya *open source*, para pengguna mendapat kebebasan untuk belajar, mengutak-ngatik, mengubah, menambah bagian-bagian tertentu, ataupun memperbaiki kode sumber (*source code*) dari sebuah perangkat lunak dan kemudian menyebarkan ulang perangkat lunak tersebut. Kebebasan menjadi pertimbangan utama ketika dilepas ke publik. *Open source* pada Arduino mencakup hardware dan software. Pengguna dapat mendownload software dan gambar rangkaian arduino tanpa harus membayar ke pembuat Arduino. Arduino Memiliki begitu banyak pengguna dan komunitas di internet dapat membantu setiap kesulitan yang dihadapi terutama oleh programmer pemula. Dengan dukungan informasi dan komunitas yang luas inilah, salah satu pertimbangan utama untuk menggunakan Development Board Arduino.

Arduino sebagai sebuah platform komputasi fisik (*Physical Computing*) yang *open source* pada board input output sederhana, yang dimaksud dengan platform komputasi fisik disini adalah membuat suatu sistem yang dapat menghubungkan dunia nyata (*physical world*) dengan dunia virtual pada komputer. Sistem yang dibangun merupakan kombinasi perangkat keras (*hardware*) dan perangkat lunak (*software*) yang bersifat interaktif, yaitu dapat menerima rangsangan dari lingkungan luar dan merespon balik. *Physical computing* merupakan sebuah konsep untuk memahami hubungan manusiawi antara lingkungan yang sifat alaminya adalah analog dengan dunia digital.

2005 2006 2007 2008

EVOLUTION OF ARDUINO

In 2005, a group at Italy's Interaction Design Institute Ivrea developed Arduino as a low-cost, easy-to-use electronics platform for students and artists. It borrows its name from nearby watering hole **Bar di Re Arduino**. Since exploding onto the maker scene, Arduino has cultivated a flourishing community of inventors, engineers, and hackers dedicated to sharing code and developing hardware under an open-source banner.



Old-style RS-232 serial port rather than USB.

Serial



Single-Sided Serial

Designed to be built on a home-etched PCB.



First board to bear the Arduino name.



Extreme

Adds female pin headers, data transfer LEDs.



NG (Nuova Generazione)



Severino (aka S3V3)

Most advanced etch-it-yourself PCB design.

Atmel's 8-bit megaAVR microcontroller family is an Arduino signature.

ATmega8

First to ship with ATmega168.



NG+



Bluetooth

The ATmega168 doubles on-board memory to 16KB, but is otherwise nearly identical to the ATmega8.

ATmega168

Large 28-pin Plastic Dual In-line Package (PDIP-28) for through-hole soldering.

Sew-through contact pads for connecting conductive thread.



LilyPad

Underside pins for breadboard connection.

Thin Quad Flat Package (TQFP) designed for surface-mount soldering.

First board to use surface-mount processor.



Mini



Diecimila



Duemilanove

Designed for battery-powered wireless projects.

Very thin Quad Flat No-lead (VQFN) package replaces leads with underside pads.

Auto-selects power supply. First to ship with ATmega328.



Duemilanove



Nano

On-chip memory doubles again to 32KB.

ATmega328

Designed for semipermanent installation.



Pro



LilyPad

Designed for battery-powered wireless projects.

Replaces sew-through contacts with button snaps.

Mini form-factor compatible.



Pro Mini



Uno



Ethernet

Official reference model for Arduino platform.

Replaces sew-through contacts with button snaps.

The Mega took Arduino to a new level, quadrupling on-chip memory to 128KB and more than tripling the total number of I/O and input pins in a significantly larger form factor.

ATmega1280



Mega

Emulates mouse and keyboard over USB.

Replaces sew-through contacts with button snaps.

ATmega32u4

Integrates USB controller into processor chip.



LilyPad Simple



LilyPad Simple Snap



LilyPad USB



Micro



Yun

First dual-core model, combining ATmega32u4 with MIPS embedded Linux machine.

With the Mega2560, memory doubled again to 256KB. Though larger, the new form factor remains pin-compatible with the standard Arduino shield footprint.

ATmega2560



Mega2560

Mega update for use with Android Accessory Development Kit (ADK).



ADK

The Due marks Arduino's first departure from the AVR architecture. The ATSAM3X8E is an ARM Cortex M3 processor with twice the memory and four times the clock speed of the ATmega2560.

ATSAM3X8E



Co-founder Banzai interviewed on TWiT.

Arduino IDE begins rapid development.

300,000 Arduinos "in the wild."



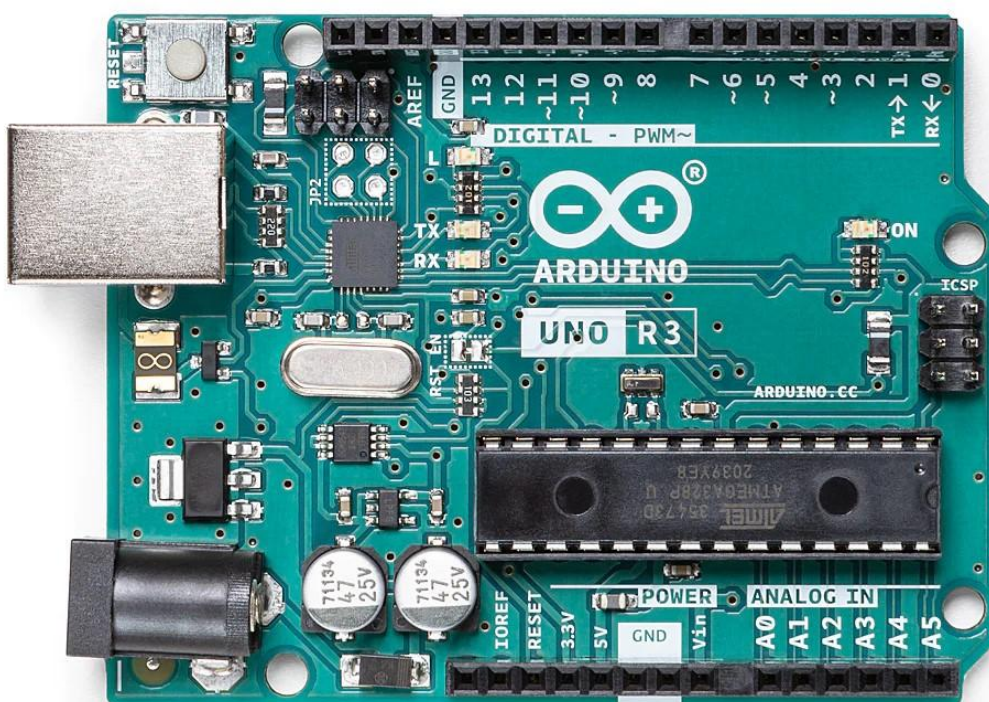
First Arduino to mount 32-bit processor. Runs at 3.3V rather than 5V.

2009 2010 2011 2012 2013

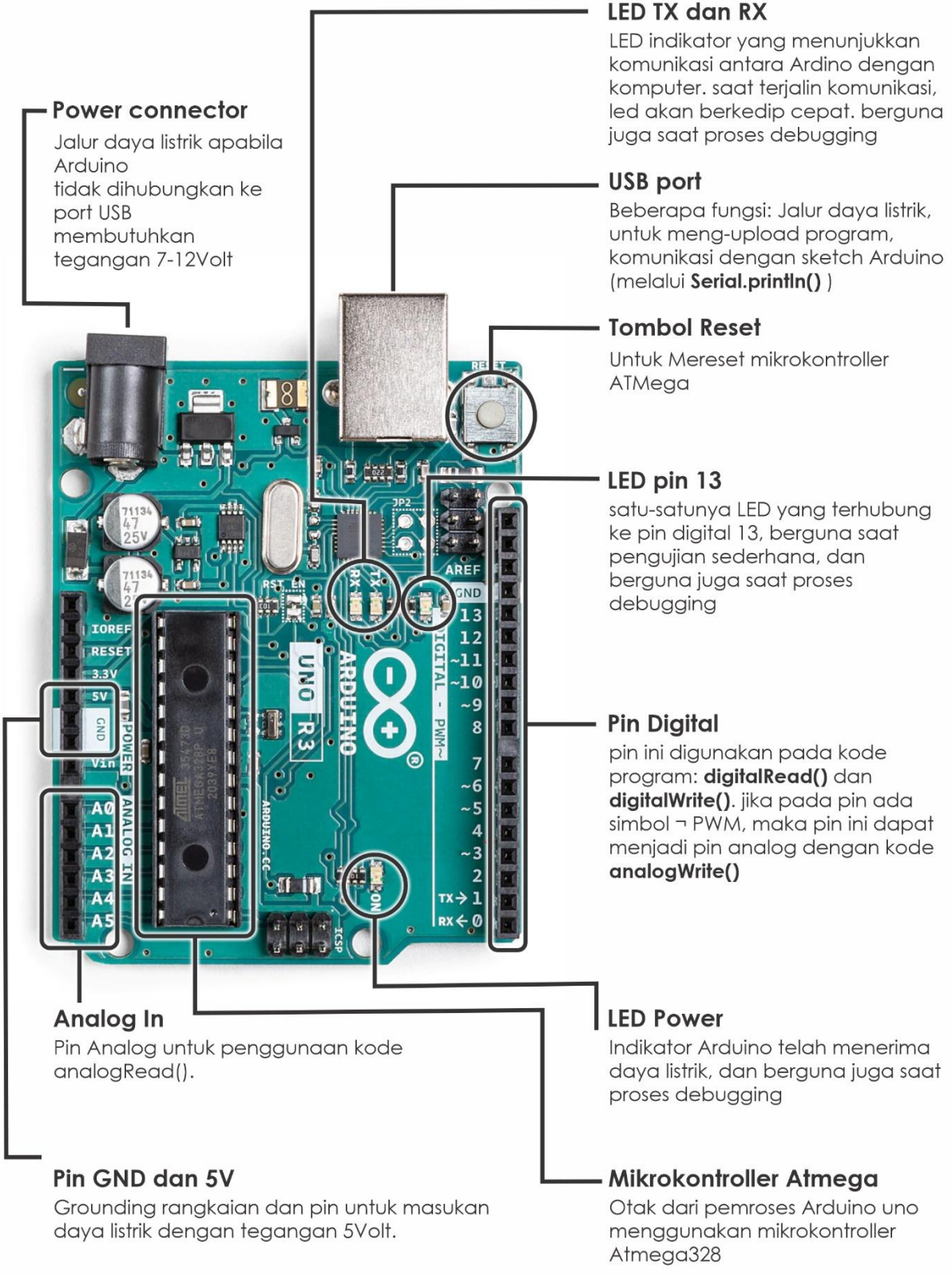
Arduino merupakan platform mikrokontroler, kemampuan kinerja dan peripheral yang terdapat pada arduino bergantung kepada mikrokontroler yang ditanamkan didalam arduino.

Salah satu contoh adalah tipe Arduino Uno.

Arduino tipe ini didalamnya terdapat mikrokontroler dengan tipe ATmega328. Mikrokontroler ATmega328 pada awalnya dibuat oleh Atmel, hingga Atmel diakuisisi oleh Microchip Technology pada tahun 2016. Beberapa dokumen di situs website EE459 mungkin masih menyebutnya sebagai Atmel ATmega328P, tetapi mikrokontrolernya Sama dan hanya Nama perusahaannya yang berubah. Sehingga untuk mendapatkan informasi apa saja yang ada didalam Arduino, kita bisa mengacu terhadap DataSheet mikrokontroler ATmega328.



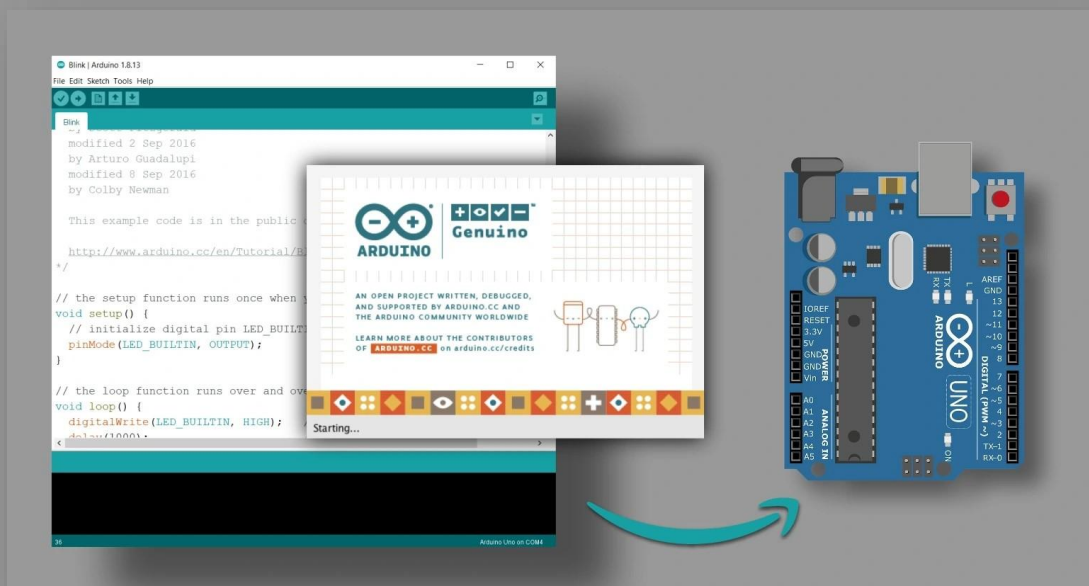
ARDUINO BOARD



Setting Up

Platform Arduino tidak hanya dari segi perangkat keras yang berbasiskan mikrokontrolernya saja, namun perangkat lunak juga. Sehingga Arduino juga menyediakan tempat dalam lingkungan pengembangan perangkat lunak (IDE) untuk dapat membuat suatu rancangan program pada arduino.

Pengembangan perangkat lunak yang dikeluarkan oleh arduino secara resmi saat ini adalah: Arduino IDE, Arduino Web Editor dan Arduino-CLI.



Arduino IDE adalah software Integrated Development Environment (IDE) yang digunakan untuk membuat kode program atau dengan kata lain arduino IDE sebagai media untuk pemrograman pada board yang diprogram. Arduino IDE digunakan untuk mengedit, membuat, mengupload ke board yang ditentukan, dan membuat kode program tertentu secara offline. Arduino IDE merupakan multiplatform, yang dapat dijalankan di berbagai sistem operasi, seperti Windows, Macintosh dan Linux.

Arduino IDE mendukung bahasa pemrograman C dan C++. Program atau kode yang ditulis dalam Arduino IDE sering disebut sebagai sketch. Kita perlu menghubungkan board Arduino dengan Arduino IDE untuk mengunggah sketch yang ditulis dalam perangkat lunak Arduino IDE.

Sketch disimpan dalam file yang ekstensi '.ino'.

Instalasi Arduino IDE

Untuk melakukan instalasi Software Arduino IDE, lakukan beberapa langkah seperti berikut:

Buka halaman resmi unduhan Arduino IDE di link:

<https://www.arduino.cc/en/software>



Arduino IDE 1.8.19

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. This software can be used with any Arduino board.

Refer to the [Getting Started](#) page for Installation instructions.

SOURCE CODE

Active development of the Arduino software is [hosted by GitHub](#). See the instructions for [building the code](#). Latest release source code archives are available [here](#). The archives are PGP-signed so they can be verified using [this](#) gpg key.

DOWNLOAD OPTIONS

- Windows** Win 7 and newer
- Windows** ZIP file
- Windows app** Win 8.1 or 10 
- Linux** 32 bits
- Linux** 64 bits
- Linux** ARM 32 bits
- Linux** ARM 64 bits
- Mac OS X** 10.10 or newer

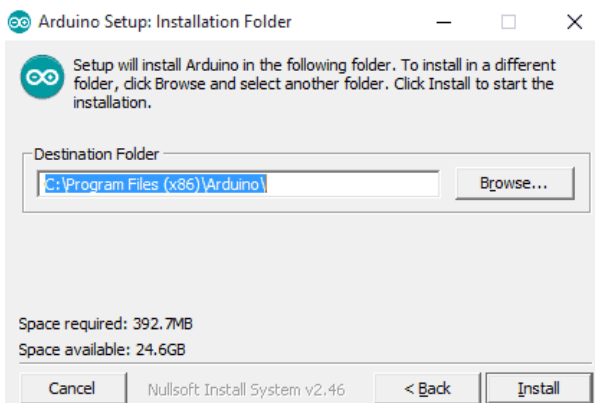
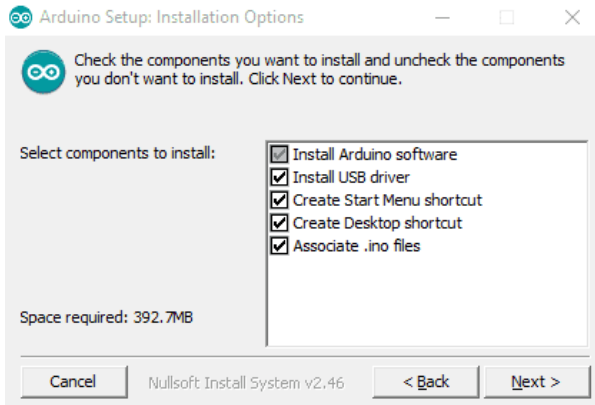
[Release Notes](#)

[Checksums \(sha512\)](#)

Halaman Unduh Arduino IDE

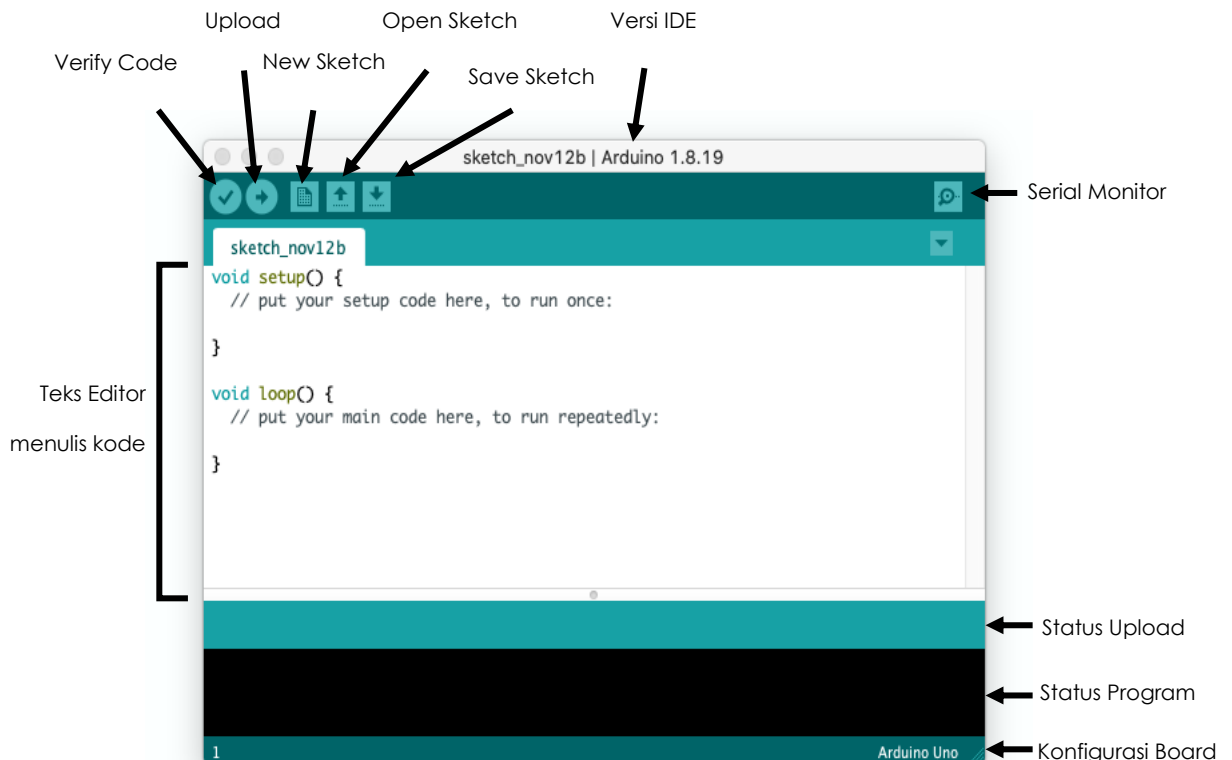
Pilih dan sesuaikan sistem operasi yang digunakan. Arduino IDE mendukung sistem operasi Windows, Linux, dan Mac OS. Kita dapat memilih antara paket Installer (.exe) dan ZIP. Disarankan menggunakan paket Installer (.exe), yang dapat menginstal langsung semua yang di butuhkan untuk menggunakan Software Arduino IDE, termasuk driver Arduino. Dengan memilih paket ZIP kita perlu menginstal driver secara manual. File ZIP berguna jika kita ingin membuat instalasi portabel yang tidak memerlukan proses instalasi, namun tetap memerlukan instalasi driver Arduino. Setelah memilih file unduhan, kemudian masuk pada halaman "**Support the Arduino IDE**", pilih klik "**Just Download**" untuk memulai proses unduhan. Tunggu beberapa saat sampai proses unduhan selesai.

Setelah pengunduhan selesai, lanjutkan dengan penginstalan dan izinkan proses penginstalan driver saat mendapatkan peringatan dari sistem operasi.

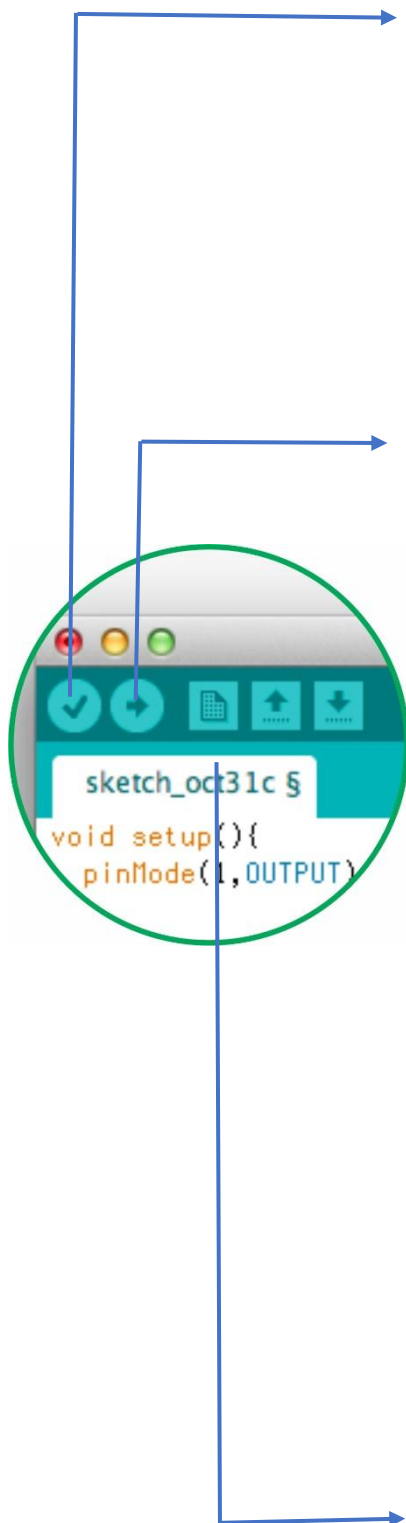


Pastikan semua komponen di ceklist untuk diinstal dan pastikan direktori untuk penginstalan Arduino. Kemudian klik install, maka proses instalasi mulai berjalan. Tunggu beberapa saat sampai proses instalasi selesai, Proses instalasi akan mengekstrak dan menginstal semua file yang diperlukan untuk menjalankan Arduino Software (IDE) dengan benar.

Setelah proses instalasi selesai, kita bisa menjalankan software Arduino IDE dengan klik icon arduino IDE pada desktop maupun menu aplikasi yang bersesuaian dengan sistem operasi masing-masing. Tampilan standar Arduino IDE seperti pada gambar berikut:



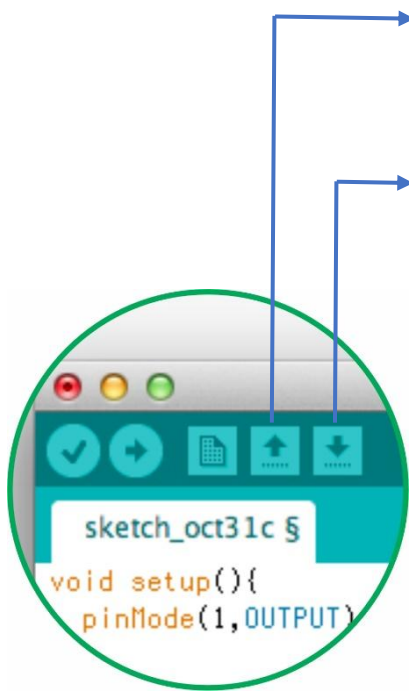
Tampilan Arduino IDE seperti gambar diatas merupakan tampilan standard dengan fungsi setiap bagian sebagai berikut:



Verify Code: Tombol ini digunakan untuk memeriksa kesalahan kompilasi sketch atau kesalahan kode yang ditulis. Apabila terdapat kode program yang salah atau tidak sesuai dengan struktur/sintaks bahasa pemrograman, maka pada bagian kolom 'Status Program' akan muncul pesan kesalahan. Apabila kode program telah sesuai dengan penulisan sintaks, maka proses selanjutnya kode program tersebut dapat di upload ke arduino.

Upload Sketch: Proses pengisian program kedalam memori program mikrokontroler biasanya dinamakan dengan Download Program, namun di Arduino dinamakan dengan Upload. Upload digunakan untuk mengirimkan atau memasukan sketch ke dalam Board Arduino yang ditentukan. Tombol Upload berfungsi untuk mengkompilasi seperti pada tombol 'verify code', namun apabila kode program tidak ada pesan kesalahan, program akan langsung memasukkan kode program ke dalam board arduino. Sebelum sketch di upload, pastikan terlebih dahulu jenis board arduino dan port (COM pada komputer) telah dipilih dengan benar. Dibutuhkan koneksi antara Board Arduino ke komputer menggunakan kabel USB, status konfigurasi COM dan jenis Board Arduino yang digunakan dapat dilihat pada kolom 'Konfigurasi Board' bagian paling bawah pada Arduino IDE. Setelah dipastikan terpasang kabel USB, memilih jenis arduino dan memilih port pada Com komputer, maka klik tombol Upload untuk proses pengisian Sketch. Apabila proses Upload gagal, maka pada kolom 'status program' akan menampilkan pesan kesalahan.

New Sketch: digunakan untuk membuka proyek baru atau membuka halaman sketch baru.

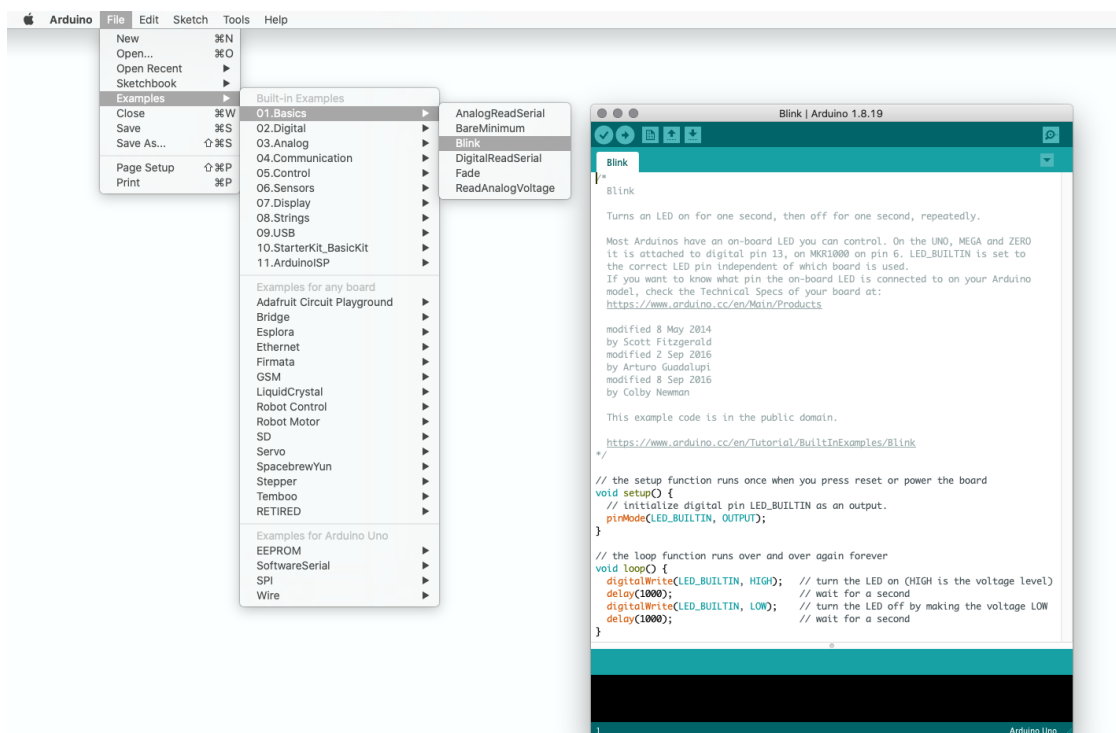


Open Sketch: digunakan untuk membuka proyek yang pernah dibuat, dengan catatan proyek tersebut telah tersimpan.

Save Sketch: digunakan untuk menyimpan sketch atau program yang sudah dibuat.

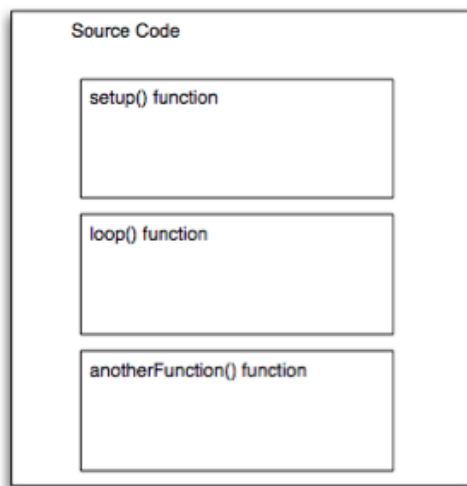
Serial Monitor. Tombol serial monitor berada pada sudut kanan kolom toolbar button, Serial Monitor digunakan untuk menampilkan data yang telah dibuat setelah sketch tersebut di-upload kedalam board yang diperlukan, kemudian nantinya akan dijalankan, dan bisa dilihat pada serial monitor. Ketika kita menghubungkan serial monitor, board Arduino akan mereset pada sistem operasi Windows, Linux dan Mas OS. Jika kita akan memproses kontrol karakter dalam sketch kita, perlu menggunakan program serial terminal eksternal. Program serial terminal harus terhubung ke COM yang sesuai ketika menghubungkan board Arduino ke komputer agar bisa terjadi komunikasi.

Arduino IDE telah dilengkapi contoh-contoh sketch yang cukup lengkap untuk dapat digunakan secara langsung dan di upload ke Board Arduino. Contoh sketch yang ada di board Arduino dapat dilakukan seperti pada gambar berikut:



Struktur Bahasa Pemrograman

Arduino memiliki bahasa pemrograman yang baku untuk menjalankan fungsi sederhana menggunakan bahasa pemrograman C. *Source Code* mengacu pada rangkaian pernyataan bahasa C yang dinamakan program. Program adalah *source code* yang diurai oleh kompiler C (membaca, memeriksa kesalahan sintaks dan semantik) dan akhirnya diterjemahkan ke dalam kode biner (yaitu, 1 dan 0) yang dipahami oleh mikrokontroler atau mikroprosesor. Untuk beberapa aplikasi pemrograman, biasanya menyebut hasil *source code* dengan istilah *script*, *listing*, atau program. Untuk Arduino IDE sebutannya berbeda, yaitu “**sketch**”, yang apabila diterjemahkan ke bahasa Indonesia berarti “sketsa”.



Secara umum, terdapat tiga tahapan yang harus dilakukan agar suatu sketch (program) bisa dijalankan pada Arduino. Tahapan tersebut antara lain:

1. Menulis program
2. Kompilasi program
3. Upload program

Singkatnya, suatu program bisa dijalankan pada *board* Arduino ketika telah selesai ditulis, di-*compile*, dan di-*upload* ke board Arduino menggunakan software Arduino IDE.

Setiap program Arduino mempunyai template dasar atau dua fungsi dasar yang harus ada keberadaannya pada setiap program. Yaitu **void setup()** dan **void loop()**. Perlu diketahui bahwa fungsi void merupakan kata kunci (keyword) yang digunakan untuk mendeklarasikan fungsi (membuat fungsi umum pada setiap program), dalam hal ini Nama **setup** dan **loop** yang merupakan sebuah Nama fungsi. Tujuannya agar fungsi atau program yang sudah di deklarasikan atau sudah dipanggil tidak mengembalikan informasi atau perintah ke fungsi atau program asal pemanggilan. Struktur program atau template sketch utama Arduino yaitu:

-
1. **void setup() {**
 2. **Statement;**
 3. **}**
 - 4.
 5. **void loop() {**
 6. **Statement;**
 7. **}**
-

Kedua fungsi `setup` dan `loop` mempunyai keterkaitan yang tidak bisa terpisahkan, apabila salah satu tidak ada pada sketch/program Arduino, bisa dipastikan program tidak bisa berjalan. Pastikan untuk menulis kedua fungsi tersebut meskipun salah satunya tidak digunakan atau tidak memiliki statement di dalamnya.

Fungsi void setup()

`void setup()` pada Arduino merupakan fungsi yang digunakan sebagai bentuk inisialisasi, pengenalan atau mengkondisikan awal semua komponen yang akan digunakan pada program dan hanya dieksekusi sekali sejak program dijalankan. Setiap program dalam Arduino IDE, memiliki fungsi `void setup()`. Meskipun terdapat statement dan arahan program yang kadang muncul sebelum fungsi `void setup()`, namun fungsi `void setup()` lah yang menandai dimulainya program yang sebenarnya. Tujuan dari fungsi `void setup()` adalah untuk mengatur lingkungan dimana program dijalankan, Salah satu contoh fungsi ini digunakan untuk mendefinisikan mode pin, baudrate komunikasi serial, dan lain sebagainya. Contoh fungsi `void setup()` di bawah ini:

```
1. void setup() {
2.     pinMode (13, OUTPUT);
3. }
```

Dalam kode program diatas terdapat pernyataan atau statement yang berisi penginisialisasian/mengkondisikan pin 13 dijadikan sebagai pin output atau keluaran.

Fungsi void loop()

Setelah `void setup()` dijalankan, yang berfungsi untuk mengkondisikan mode kerja komponen yang akan digunakan, selanjutnya program akan menjalankan fungsi `void loop()`. Statemen yang ada didalam fungsi `void loop()` ini akan dieksekusi terus-menerus secara berurutan hingga program berhenti dijalankan atau sumber daya listrik dimatikan. contoh fungsi `void loop()` di bawah ini:

```
1. void loop() {
2.     digitalWrite (13, HIGH);
3.     delay (1000);
4. }
```

Fungsi ini berisi statement program utama. Apabila terdapat statement didalam fungsi ini, seperti pada contoh diatas. Maka pada program baris 5 akan dieksekusi untuk

mengkondisikan pin 13 menjadi logika HIGH. Setelah selesai, program akan mengeksekusi program pada baris selanjutnya yaitu baris ke 6, untuk memberikan tundaan waktu selama 1000 milidetik. Sampai disini proses pada fungsi **void loop()** telah mencapai satu siklus eksekusi program, karena dibawahnya pada baris 7 terdapat tutup kurawal yang menandakan fungsi telah selesai.

Sampai disini program tidaklah berhenti pada baris 6 saja, akan tetapi akan lompat kembali pada awal statement pada fungsi **void loop()** yaitu pada baris 5, Kemudian mengeksekusi baris ke 6 lagi, begitu seterusnya berulang ulang sampai tidak ada lagi sumber daya listrik pada mikrokontroler/Arduino/NodeMCU. Sekalipun pada fungsi **void loop()** tidak ada statement didalamnya (antara buka kurung kurawal sampai tutup kurung kurawal), program akan tetap berputar terus menerus difungsi **void loop()** ini.



Catatan pada program Arduino IDE:

- *Setiap program Arduino pasti memiliki fungsi void setup()*
- *Fungsi void setup() hanya dijalankan satu kali saat program pertama kali dijalankan.*
- *Setiap program Arduino pasti memiliki fungsi void loop().*
- *Fungsi void loop() berulang kali dijalankan hingga daya dimatikan, program direset, atau komponen gagal.*

Untuk beberapa kasus pemrograman yang kompleks, terkadang programmer juga menambahkan fungsi (void) baru untuk mempermudah pemrograman. Jadi *void* baru ini nantinya bisa diletakkan setelah atau sebelum **void loop()**.

Sintaks Arduino

Sintaks Arduino disebut sebagai aturan karakteristik penulisan kode program/sketch Arduino. Sintaks (syntax) dikenal sebagai seperangkat aturan tata bahasa pada kata atau klausa dalam membentuk kalimat. Dalam bahasa pemrograman sintaks diartikan sebagai suatu aturan dalam penulisan kode program sehingga kode program tersebut bisa dimengerti oleh perangkat keras. Beberapa sintaks pada bahasa pemrograman Arduino antara lain:

1. Titik Koma (*semicolon*)

Pada setiap statement didalam fungsi maupun diluar fungsi seperti deklarasi suatu variabel, harus diikuti sertakan titik koma (;) pada akhir setiap statement. Titik koma menandakan kompilator segera mengambil semua informasi pada statement untuk diproses. Jadi pada dasarnya, program mengeksekusi setiap baris perintah, bukan kolom perintah. Apabila didalam blok fungsi terdapat beberapa statement, dari setiap satu statement di batasi dengan titik koma untuk mengetahui pembeda antara tiap barisnya. Contoh penggunaan titik koma:

```
int x = 13;
```

2. Kurung Kurawal (*curly braces*)

Buka kurung kurawal dan tutup kurung kurawal menandakan awal dan akhir dari sebuah blok fungsi. Didalam fungsi kadang tidak hanya terdapat satu buka dan tutup kurung kurawal, tapi kurung kurawal juga mengikuti kontrol statement. Jadi fungsi kadang berisi beberapa buka tutup kurung kurawal. Yang perlu diperhatikan adalah, setiap buka kurung kurawal harus ada pasangannya yaitu tutup kurung kurawal. Apabila dalam satu fungsi terdapat tiga buka kurung kurawal, berarti dalam fungsi tersebut harus ada tiga tutup kurung kurawal.

```
1. void HidupkanAC(){
2.     If(ACMati){
3.         ACOOn();
4.     }
5.     AmbilDataSuhu()
6. }
```

3. Define (`#define`)

Sebenarnya `#define` merupakan sebuah statement yang biasa digunakan untuk membuat atau mendeklarasikan *fixed variables* (variabel tetap). Dalam Arduino terdapat juga `const` untuk mendeklarasikan variabel, Cara kerja dari statement define sama dengan deklarasi variabel menggunakan `const`. hanya saja sintaks penggunaannya.

```
#define LED1 13

const int LED2 = 13;
```

Pada penggunaan define, harus memberikan simbol tagar '#' pada awal statement dilanjutkan nama variabel kemudian angka setelahnya dan tidak ada titik koma diakhir deklarasi. Seperti pada contoh diatas, mendefinisikan variabel LED1 dengan angka 13. Artinya program/sketch mengganti angka 13 menjadi nama variabel yang didefinisikan pada statement `#define`. Sebagai contoh program sebagai berikut:

```
1. #define LED 13
2. void setup(){
```

```
3.     pinMode(LED, OUTPUT)
4.     }
```

#define ditujukan untuk penggunaan variabel yang berulang-ulang dengan nilai yang tetap dan tidak menggunakan ruang memori, dengan seperti ini, tidak diperlukan untuk mengingat nilai/angka yang di tuju. Variabel yang didefinisikan menggunakan **const** merupakan variabel biasa atau menggunakan ruang memori khusus untuk menyimpan nilainya.

Dalam penerapannya, antara **#define** dengan **const** lebih disenangi menggunakan **const**, karena **const** merupakan variabel umum yang mungkin nantinya dalam nama variabel ini dapat dimanipulasi kembali nilainya seiring berjalanya program.

4. Include (#include)

Didalam pemrograman Arduino, mengenal dengan istilah file header. File header merupakan file tambahan untuk memanggil fungsi tambahan dari sebuah library maupun file eksternal dari file utama arduino berekstensi .ino. fungsi include untuk memasukkan libraries ke dalam sketch, dimana libraries sendiri adalah kumpulan fungsi yang siap digunakan sehingga pembuatan program tidak diharuskan membuat kode program mulai dari nol. Cara penggunaan sintaks include ini dapat dilihat pada program berikut:

```
#include <OneWire.h>
```

5. Komentar

Salah satu kualitas terpenting dari program yang baik adalah *readability* (keterbacaannya). Pentingnya dokumentasi dalam pembuatan perangkat lunak adalah sesuatu yang mutlak, sebisa mungkin perangkat lunak yang dibangun mudah untuk dipahami dalam waktu singkat, apalagi terhadap orang lain. Selain itu menghindari pemeliharaan perangkat lunak yang cukup mahal dan penanganan kesalahan yang cepat. Sebagai seorang programmer, kita dapat menempatkan baris komentar di program, hal ini berfungsi untuk menjelaskan tujuan serta logika dari setiap langkah penting. Bahasa C menyediakan dua jenis baris komentar:

Format baris tunggal (*Single Line*):

Komentar satu baris dimulai dengan `//` dan dapat dimulai di sembarang tempat dalam satu baris. *Compiler* mengabaikan semua yang muncul setelah dua karakter `//` berturut-turut.

Format beberapa baris (end-of-Line):

Komentar beberapa baris dimulai dengan `/*` dan diakhiri dengan `*/`. Segala sesuatu yang muncul di antara `/*` dan `*/` diabaikan oleh *compiler*.

Contoh penggunaan komentar satu baris dan beberapa baris:

-
1. `#include <OneWire.h> // ini adalah komentar satu baris`
 2. `#define LED1 13 /* ini adalah komentar beberapa`
 3. `baris untuk mendokumentasikan kode program agar`
 4. `keterbacaan program tetap terjaga sehingga mudah`
 5. `dimengerti */`
 6. `const int LED2 = 13;`
-

Tipe Data Arduino C

Saat merujuk ke tipe data C, kita mengacu pada atribut yang dimiliki oleh potongan data program. Tipe data tertentu lebih cocok untuk tugas tertentu dari pada tipe data lainnya. **Memilih tipe data yang tepat akan menghasilkan program yang berjalan lebih cepat dan lebih sedikit ruang memori.** Apalagi program yang dibuat dilingkungan mikrokontroler yang pastinya mempunyai keterbatasan ruang memori yang tersedia, sehingga kita sebagai programmer harus lebih jeli dan tepat menggunakan tipe data tersebut.

Dalam tipe data tidak terlepas dari suatu variabel. **Variabel** adalah pemesanan dan penamaan dari ruang memori. Saat pemberian nama variabel atau penamaan ruang memori tersebut, kita harus memberitahukan kompiler jenis data apa yang harus dikaitkan dengan variabel itu. Tipe data variabel ini yang menentukan berapa banyak byte memori yang didedikasikan untuk variabel itu, dan jenis data apa yang dapat disimpan dalam variabel tersebut. Komputer hanya mengetahui dua hal: On (1) atau Off (0). Komputer menyusun digit biner, atau bit, ke dalam pengelompokan 8bit. Setiap kelompok 8bit yang diambil sebagai satu unit disebut byte. Karena satu byte hanya dapat memiliki dua status, 1 atau 0, byte paling tepat menggunakan aritmatika basis 2, dari pada basis 10 yang biasa kita gunakan. Jadi satu byte dapat menyusun bit-nya dalam 256 kombinasi unik. Karena 0 adalah angka yang valid, satu byte memori komputer dapat mewakili nilai 0–255.

Terdapat dua tipe dasar variabel: tipe value dan tipe referensi. Jika variabel didefinisikan sebagai tipe value, ada kemungkinan rentang value numerik yang sangat spesifik dengan tipe data tersebut.

Tabel 1. Tabel Daftar Tipe Data Nilai Dasar

Tipe Data	Lebar Data	Jangkauan value
boolean	1 byte	Terbatas logika true dan false
char	1 byte	-128 s/d 127
unsigned char	1 byte	0 s/d 255
byte	1 byte	0 s/d 255
int	2 byte	-32768 s/d 32767
unsigned int	2 byte	0 s/d 65535
word	2 byte	1 s/d 65535
long	4 byte	-2147438648 s/d 2147438647
unsigned long	4 byte	0 s/d 4294967295
float	4 byte	-3.4028235E+38 s/d 3.4028235E+38
string	?	Tipe data dari karakter array
String	?	Tipe data referensi objek
array	?	Tipe value yang direferensikan oleh nama variabel tunggal
void	0	Tipe pengembalian saat fungsi tidak mengembalikan value

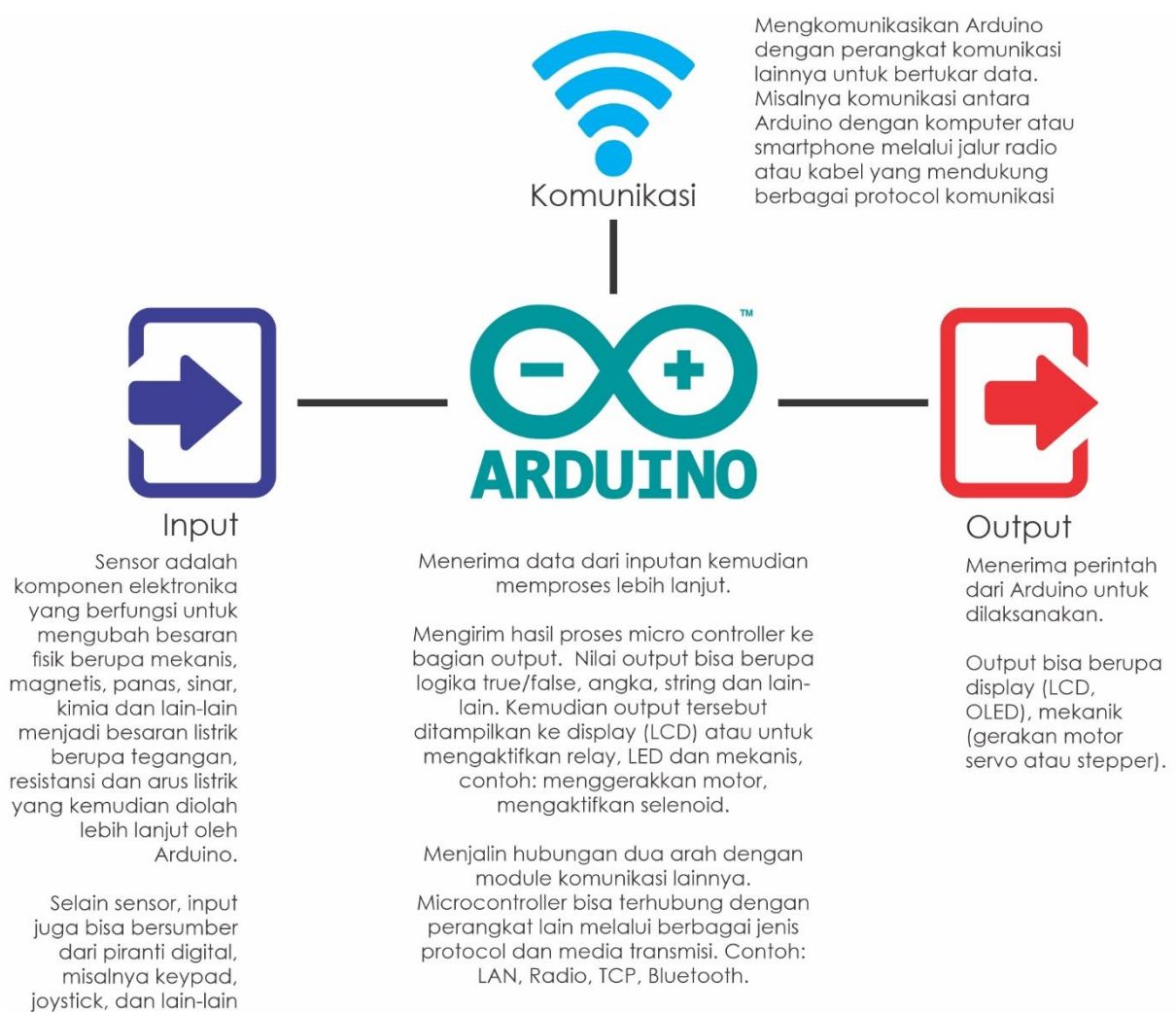
Format penulisan tipe data Arduino dalam sketch dengan menempatkan didepan variabel yang akan dideklarasikan diantarai satu spasi. Misalkan seperti pada contoh berikut:

```
int percobaan = 13;
```

Fungsi **int** pada pada contoh diatas, letaknya diawal variabel untuk mendeklarasikan bahwa variabel tersebut bertipe data integer.

Arduino Mode Umum

Pengendali mikro (bahasa Inggris: microcontroller) adalah sistem mikroprosesor lengkap yang terkandung di dalam sebuah chip. Mikrokontroler berbeda dari mikroprosesor serba guna yang digunakan dalam sebuah komputer, karena di dalam sebuah mikrokontroler umumnya telah berisi komponen pendukung sistem minimal mikroprosesor, yakni memori dan antarmuka I/O. Sedangkan di dalam mikroprosesor umumnya hanya berisi CPU.



Komponen Lingkup Pengembangan



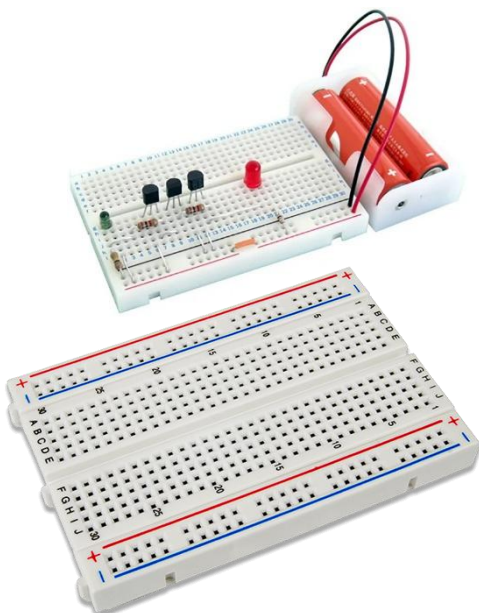
Arduino Uno

Development Board Mikrokontroler yang menjadi jantung dalam project. Arduino Board adalah komputer sederhana, akan tetapi komputer tersebut belum ada cara untuk ber-interaksi dengan Arduino. Kita sebagai perancang / pendesain / programmer membuat rangkaian dan antarmuka untuk berinteraksi, dan memberi tahu mikrokontroler agar dapat berkomunikasi / berinteraksi dengan komponen yang lainnya.



Baterai Snap

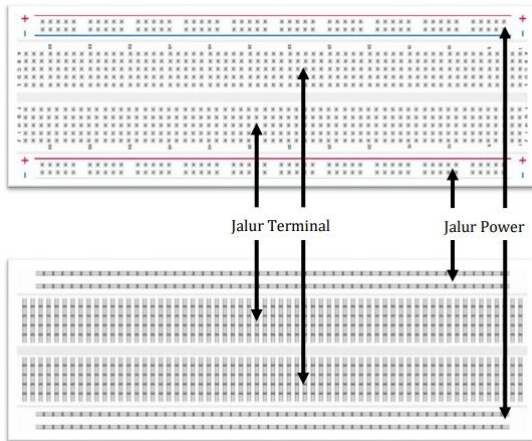
Digunakan untuk menghubungkan Baterai 9Volt ke kabel listrik yang dapat dengan mudah di tancapkan kedalam papan Breadboard atau papan Arduino.



Breadboard

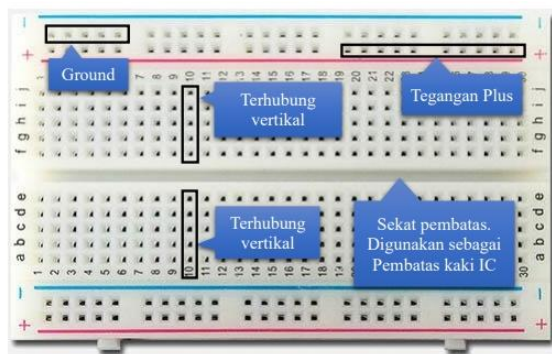
Breadboard terbuat dari bahan plastik yang sudah terdapat lubang didalamnya. Lubang tersebut membentuk pola yang didasarkan pada persambungan konduktor di dalamnya. Breadboard memiliki lubang kurang lebih 830 (meskipun ada yang mempunyai lubang 170 dan 400 titik lubang). Di sini bisa dilihat beberapa tulisan mulai dari A sampai dengan J pada sisi sebelah kanan vertikal. Ini berarti kolom A, B, C, D, dan E merupakan baris yang terhubung kolom vertikal bersama. Kemudian kolom F, G, H, I, dan J juga baris

yang terhubung secara vertikal bersama. Namun dalam baris A sampai E dengan F sampai J tidak terhubung secara vertikal.



Setelah itu masih ada angka 1,5, 10, 15, maupun 20. Perpaduan antara huruf dan juga angka tersebut merupakan sebuah koordinat. Misalnya, A1, B1, sampai dengan E1 saling terkoneksi berdasarkan pola koneksinya. Sementara untuk A2 sampai dengan E2, A3 sampai dengan E3, F1 sampai dengan J1, dan seterusnya. Dengan memahami pola tersebut, kita tidak dibingungkan ketika harus menempatkan komponen-komponen elektronik sesuai dengan tempatnya masing-masing.

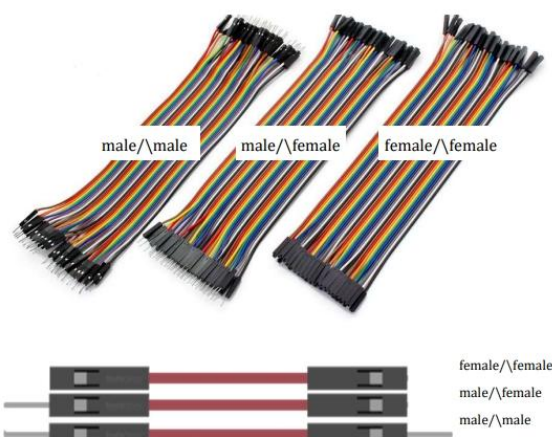
Breadboard memiliki prinsip kerja sebagai berikut:



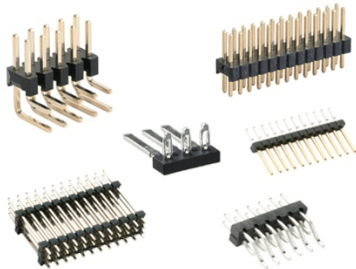
- 2 pasang pada jalur bawah dan atas terkoneksi secara horizontal sampai menuju ke bagian tengah pada breadboard. Biasanya, akan difungsikan sebagai jalur dari tombol power maupun jalur sinyal.
- 5 lubang yang terdapat di komponen bagian tengah digunakan sebagai lokasi untuk melakukan perakitan komponen. Jalur kelima tersebut terkoneksi secara vertikal sampai menuju ke bagian tengah pada breadboard.

Kabel Jumper

Kabel jumper digunakan untuk membuat koneksi diantara komponen yang ada di breadboard maupun antara breadboard dengan modul lainnya. Kabel jumper merupakan kabel inti padat dengan penahan plastik yang dibentuk disetiap ujungnya, memudahkan kita untuk memasukkan dan melepaskan kabel. Warna kabel jumper biasanya berwarna warni untuk



memudahkan kita mengingat jalur yang digunakan. Sedangkan jenis kabel jumper terdapat tiga perbedaan dilihat dari antar ujung pada kabel jumper.

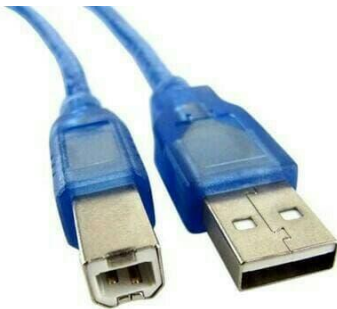


Pin Male Header

Pin male header dapat membantu lebih mudah dalam menghubungkan berbagai koneksi pada setiap perangkat didalam Breadboard

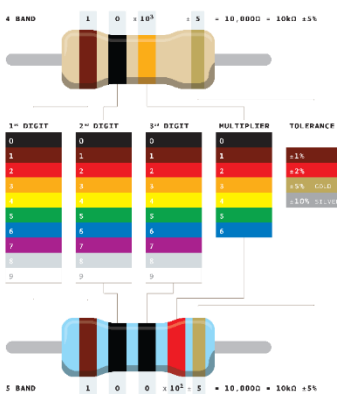
Kabel USB

Berfungsi untuk menghubungkan antara Arduino Uno dengan perangkat komputer atau Android (perlu OTG adapter), sebagai jalur pemograman. Melalui kabel USB ini juga berfungsi untuk memberikan sumber tegangan untuk setiap rangkaian proyek yang terhubung dengan Arduino.



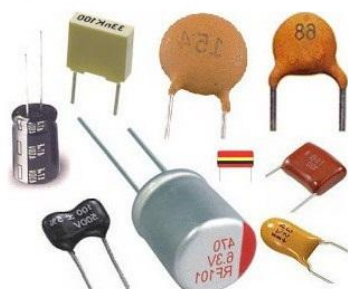
Resistor

Hambatan aliran energi listrik pada suatu rangkaian elektonika. Dapat menghasilkan perubahan pada tegangan dan arus. Nilai resistor diukur dalam Ohm. Garis-garis (gelang) berwarna di tubuh resistor menunjukkan nilainya (lihat tabel daftar warna nilai resistor).



Kapasitor

Komponen ini menyimpan dan melepaskan energi listrik dalam suatu rangkaian. Ketika tegangan rangkaian lebih tinggi dari tegangan yang disimpan kapasitor, hal ini memungkinkan arus mengalir masuk, memberi muatan pada kapasitor. Saat rangkaian tegangan lebih rendah, muatan



yang tersimpan dilepaskan. Sering diterapkan berparalel dengan sensor dan dekat dengan Ground (GND) sensor atau motor, dan untuk membantu menghaluskan lonjakan listrik dalam tegangan Transistor.

Motor DC

Ketika listrik dialirkan akan mengubah energi listrik menjadi energi mekanik. Gulungan kawat di dalam motor menjadi termagnetisasi ketika arus mengalir melaluinya, Medan magnet ini menarik dan menolak inti magnet sehingga menyebabkan poros berputar. Jika arahnya listriknya dibalik maka motor akan berputar dalam arah yang berlawanan



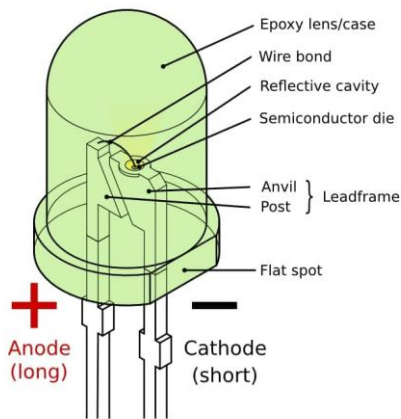
Motor Servo

Sejenis Motor DC yang hanya dapat berputar 180 derajat. Dikendalikan oleh pengiriman pulsa listrik (PWM: Pulse Width Modulation) dari Arduino. Pulsa ini menentukan posisi motor servo yang harus digerakkan.



Dioda

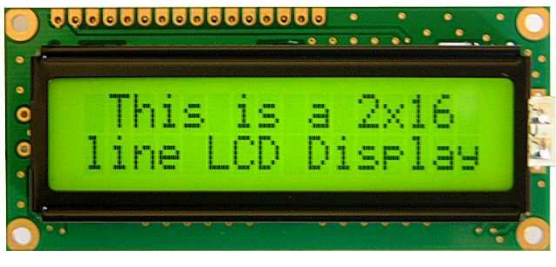
Komponen aktif dua kutub yang pada umumnya bersifat semikonduktor, yang memperbolehkan arus listrik mengalir ke satu arah (kondisi panjar maju) dan menghambat arus dari arah sebaliknya (kondisi panjar mundur). Dioda dapat disamakan sebagai fungsi katup di dalam bidang elektronika. Pada motor DC, dioda yang melintasi terminal, akan mencegah lonjakan EMF balik yang merusak transistor driver yang sensitif. Hal ini juga akan mencegah motor terbalik, karena dioda akan



menyebabkan arus pendek, jika suplai dibalik.

Light Emitting Diodes (LED)

LED (Light Emitting Diode) adalah Sebuah lampu kecil yang digunakan sebagai penanda atau pointer. Light Emitting Diode adalah salah satu komponen elektronika yang terbuat dari bahan semi konduktor jenis dioda yang mampu mengeluarkan cahaya. Strukturnya juga sama dengan dioda, tetapi pada LED elektron menerjang sambungan P-N (Positif-Negatif). Untuk mendapatkan emisi cahaya pada semikonduktor, doping yang pakai adalah galium, arsenic dan phosphorus. Jenis doping yang berbeda menghasilkan warna cahaya yang berbeda pula.



Liquid Crystal Display (LCD)

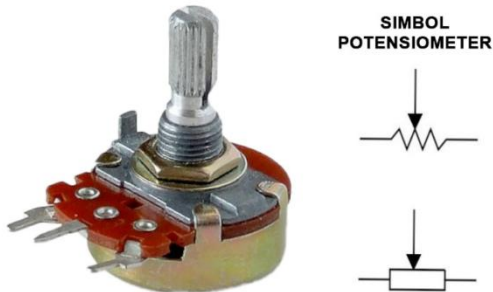
Jenis tampilan alfanumerik atau grafis berdasarkan kristal cair (LCD: Liquid Crystal Display). Terdapat berbagai ukuran dan yang umum digunakan adalah ukuran 2 baris dengan 16 karakter setiap barisnya (LCD 16x2).



Buzzer

merupakan komponen elektronika yang berfungsi merubah getaran listrik menjadi getaran suara. Terdapat dua tipe buzzer yaitu buzzer aktif (memproduksi getaran sendiri hanya dengan diberi tegangan 5V) dan buzzer pasif (membutuhkan simulasi pulsa tegangan untuk menghasilkan getaran suara), pada Kit ini menggunakan active buzzer 5V. Kamu dapat menggunakan

buzzer untuk menghasilkan music yang sederhana.



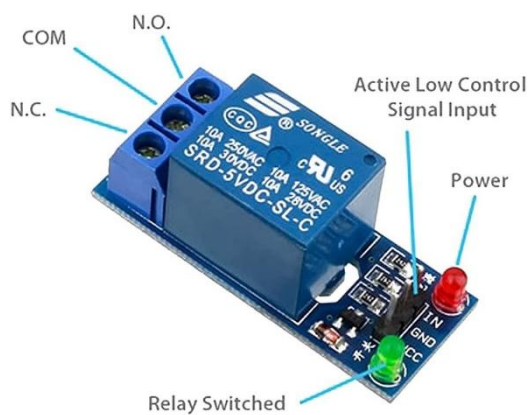
Potensiometer

resistor yang nilai hambatannya dapat diatur sesuai posisi putaran knob. Ketika pada kedua sisi kaki potensiometer diberi tegangan ground dan positif 5V, maka pada kaki tengahnya akan mengeluarkan variabel tegangan dari 0 - 5V sesuai dengan posisi putaran knob.



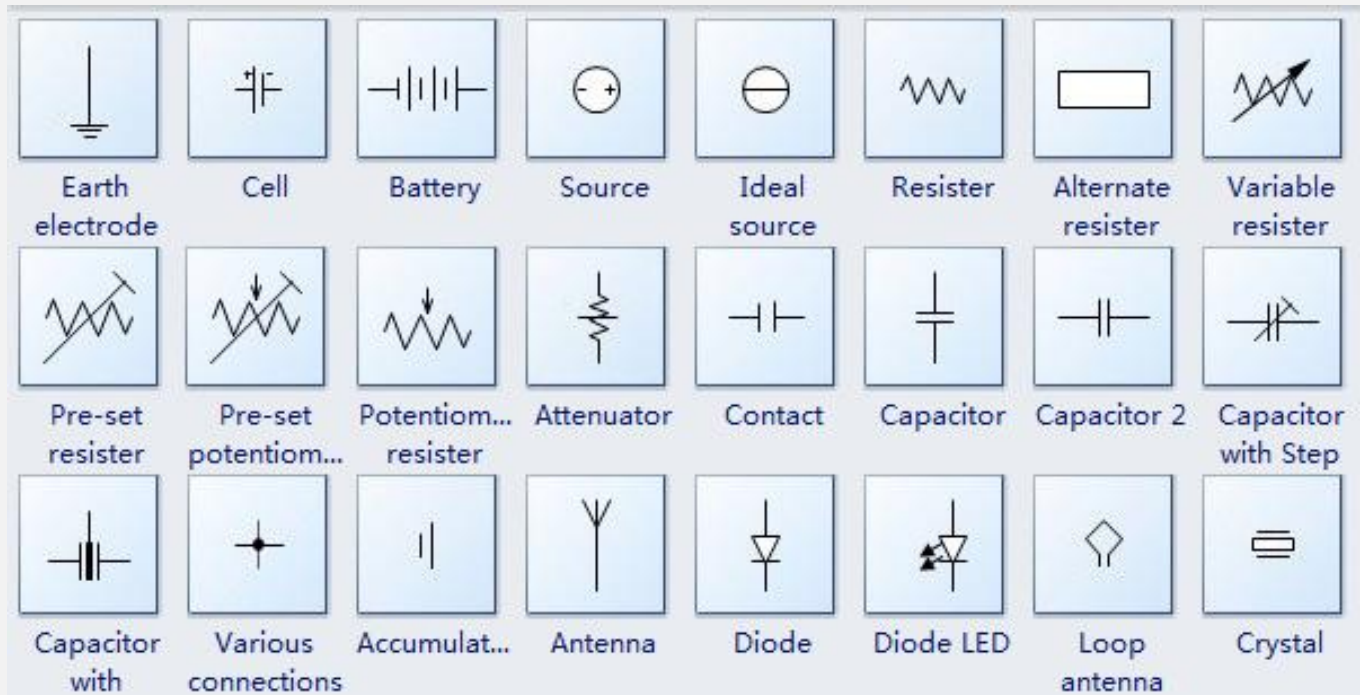
Push Button

Sebuah saklar yang menghubungkan arus pada suatu rangkaian ketika knobnya ditekan. Saklar sangat cocok dipasang pada pin digital input untuk mendeteksi sinyal on/off.



Modul Relay

Pada dasarnya relay adalah sebuah saklar yang dikontrol secara elektronik, untuk mengubah kondisi saklar menjadi on atau off menggunakan arus listrik yang dikenakan pada lilitan magnet di dalamnya. Pada modul relay ini selain terdapat komponen relay sudah dilengkapi juga dengan komponen tabahan seperti led sebagai indikator, transistor sebagai pengendali, dan terminal, sehingga mudah untuk menggunakannya bersama Arduino dengan tegangan kerja 5V.



Listrik adalah sejenis energi, seperti panas, gravitasi, atau cahaya.

Energi listrik mengalir melalui konduktor, seperti kawat. Kita dapat mengubah energi listrik menjadi energi lain bentuk energi untuk melakukan sesuatu yang menarik, seperti menyalakan lampu atau membuat suara yang keluar dari speaker.

Komponen yang mungkin kita gunakan untuk melakukan hal ini, seperti speaker atau bola lampu, adalah transduser listrik. Transduser mengubah jenis energi lain menjadi energi listrik dan sebaliknya.

Hal yang mengubah bentuk energi lain menjadi energi listrik sering disebut sensor, dan benda yang mengubah energi listrik menjadi bentuk lain energi disebut aktuator.

Dalam suatu rangkaian, listrik mengalir dari titik yang energi potensialnya lebih tinggi (biasanya disebut daya atau +) ke titik yang energi potensialnya lebih rendah. Ground (sering dilambangkan dengan - atau GND) umumnya merupakan titik dengan energi potensial paling kecil dalam suatu rangkaian.



LIBRARY

Library adalah kumpulan kode yang memudahkan kita untuk terhubung ke sensor, tampilan, modul, dan lain sebagainya. Misalnya, library LiquidCrystal memudahkan untuk berkomunikasi dengan tampilan karakter LCD. Library merupakan file yang ditulis dalam C atau C++ (.c, .cpp) yang dibuat oleh pihak ketiga untuk mempermudah dan mempercepat dalam pembuatan kode program dengan fungsionalitas tambahan tanpa perlu menulis setiap baris kode, dan tanpa harus mengetahui karakteristik secara detail dari setiap komponen/modul tambahan. Bayangkan apabila dalam implementasi pembuatan program kita harus membuat struktur komunikasi serial secara manual (tanpa library) untuk mengendalikan tampilan LCD, maka diperlukan puluhan baris kode program yang akan ditulis, namun dengan library kita hanya mengetikkan kode program dengan beberapa baris saja. Kita tidak lagi menuliskan sub-program kecil untuk menangani masalah antarmuka hardware, terutama dalam hal level pin dan elektrikalnya. Terdapat tiga jenis library Arduino:

Library Standar: Library ini sebagai pelengkap dari Arduino IDE untuk membantu pemula dengan contoh proyek yang paling banyak digunakan. Library standar ditujukan untuk komunikasi dasar antar perangkat atau sensor eksternal yang paling umum digunakan seperti penampil LCD, motor servo, dan lain sebagainya. Library Standar terletak di folder "Libraries", dalam sistem Operasi Windows terletak di `Dokuments -> Arduino -> libraries`.

Library Manager: Arduino IDE mempunyai Library Manager sebagai tempat pencarian dan menginstall library, dibuat oleh pihak ketiga yang terdaftar pada Arduino untuk dapat digunakan oleh pengguna.

Library Installed: Ada library yang dapat diinstall oleh pengguna sendiri, dimana library ini tersedia dari sumber eksternal seperti Arduino Playground dan Github.

Library pada dasarnya adalah perintah singkat perangkat lunak, disimpan untuk mempermudah pengguna menggunakan kode program. Biasanya library akan menentukan pin GPIO, protokol komunikasi, dan metode kerja (algoritma). Dengan demikian pengguna akan menggunakan perintah program dengan mudah dan mendapatkan hasil kode yang disederhanakan. Namun kadang kebebasan penggunaan PIN dan metode komunikasi menjadikan ketidak leluasaan kita dalam penerapan program yang kompleks.

Menambahkan Library pada Arduino IDE

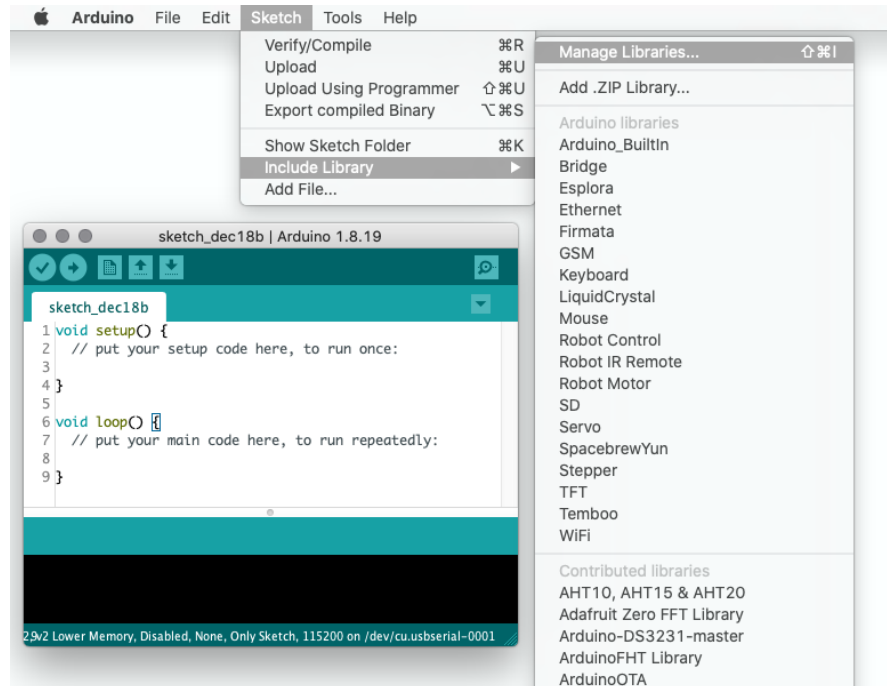
Cara menambahkan library pada Arduino IDE setidaknya dapat dilakukan dengan tiga cara, yaitu:

1. Menggunakan Library Manager

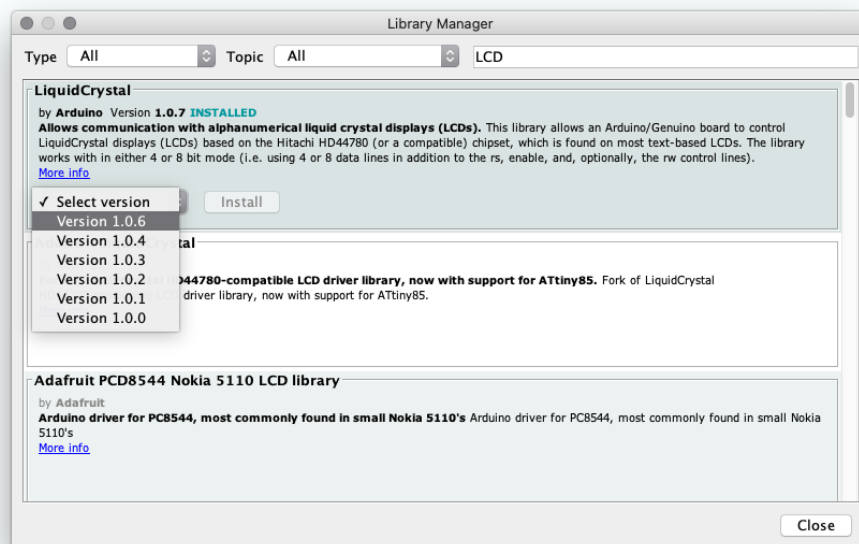
Arduino IDE memberikan fasilitas Untuk menambahkan library dengan mencari dan menginstallnya dalam satu tempat, yaitu menggunakan Library Manager. Untuk menambahkan library menggunakan library Manager dibutuhkan koneksi internet. Sehingga pastikan perangkat komputer kita sudah terkoneksi dengan internet untuk proses download dan instalasi.

- a. Buka Arduino IDE dan pilih menu **Sketch -> Include Library -> Manage Libraries**.

Tunggu loading tampilan Library Manager terbuka, biasanya akan membutuhkan waktu yang lama, karena akan menampilkan daftar library yang tersedia melalui internet.



- b. Mencari file library yang diinginkan, pilih versi dan klik install. Setelah tampilan library terbuka, kita bisa mengetikkan nama library yang dituju pada kolom pencarian "Filter your search..". kemudian klik pilihan versi dan klik install.

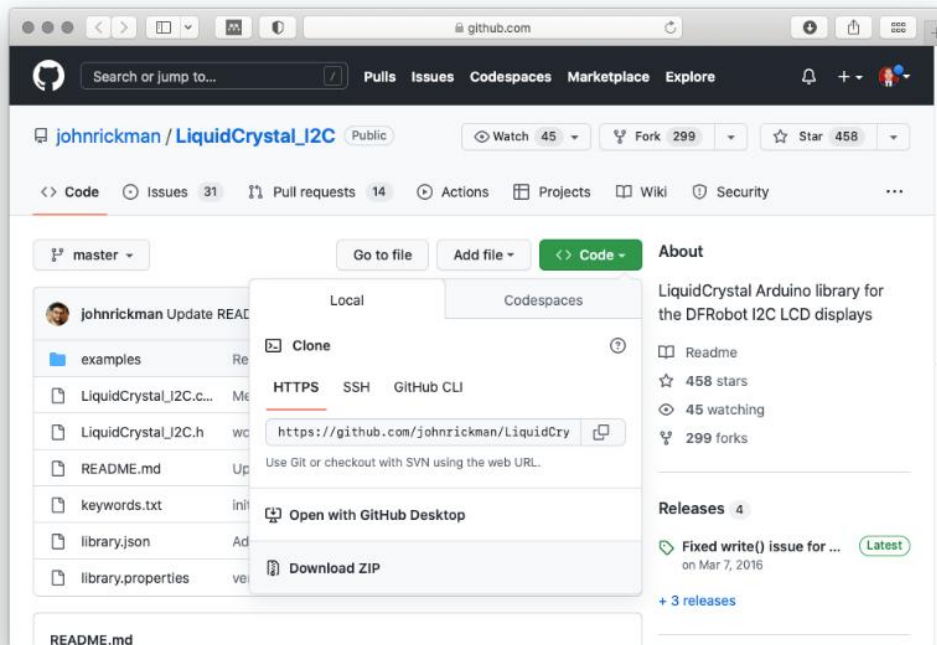


Mengimpor Library dari File ZIP

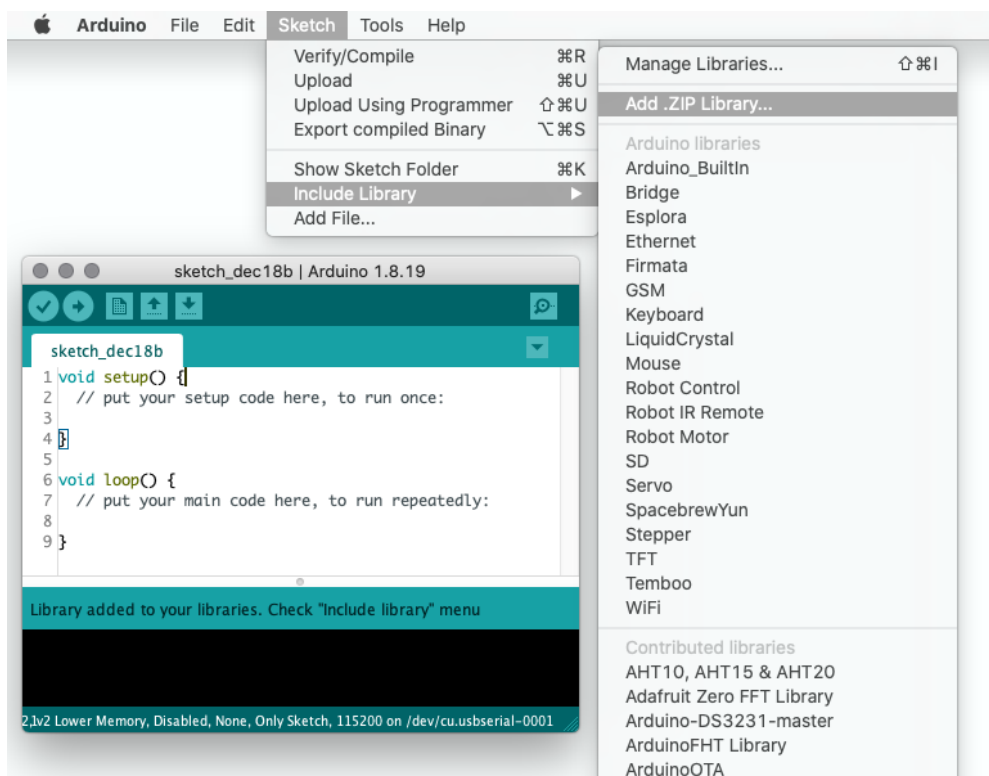
Kita bisa mendapatkan banyak library dalam bentuk file ZIP di internet, salah satunya adalah dari Github. Pada contoh kali ini, kita akan memasukkan library untuk LCD 16x2 menggunakan komunikasi serial I2C. buka software browser pada komputer, kemudian pergi ke link:

https://github.com/johnrickman/LiquidCrystal_I2C

Download file library tersebut dengan cara klik tombol “<> **Code**” kemudian tarik kebawah klik menu “**Download ZIP**”.

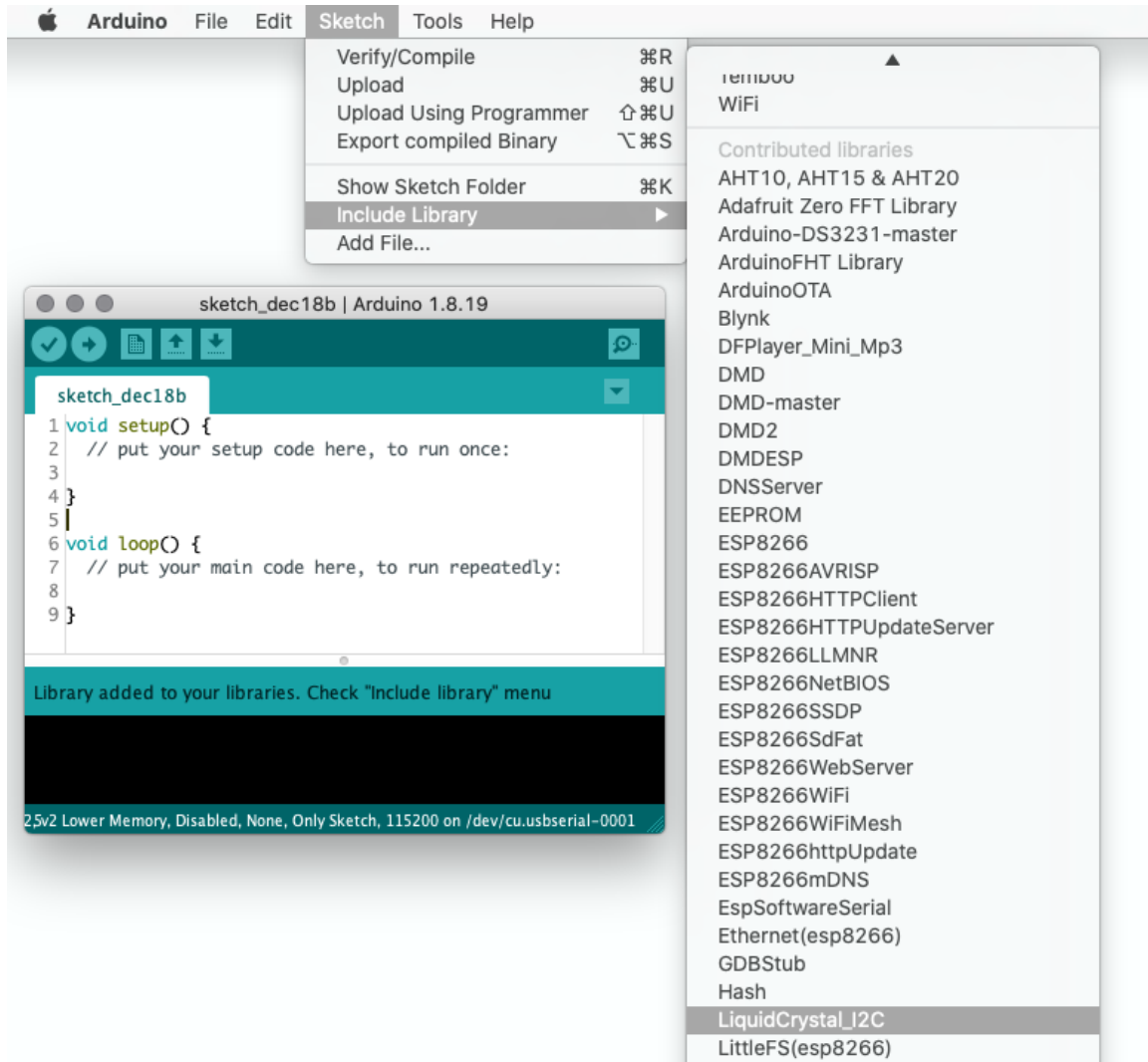


Setelah proses download selesai, pergi ke Arduino IDE dengan mengklik menu **Sketch -> Include Library -> Add .ZIP Library...**



pilih file ZIP yang telah di download tadi, kemudian klik **OK / Choose**. Tunggu sebentar sampai proses instalasi selesai dengan tanda pada bagian keterangan aplikasi Arduino IDE akan muncul pesan “Library added to your libraries. Check include library menu”.

Untuk memastikan proses instalasi library sudah berhasil atau tidak, kita bisa mengecek pada Arduino IDE di menu **Sketch -> Include Library -> ...**



Menambahkan Library Secara Manual

Selain kedua cara penginstalan library, masih ada cara lain untuk menambahkan library ke Arduino IDE, yaitu dengan cara instalasi manual. Caranya yaitu: ekstrak file library yang berekstensi ZIP (hasil download dari Github) disatukan kedalam folder baru/folder yang tersedia, kemudian folder yang berisi file Library / hasil ekstrak tersebut dipindahkan ke direktori:

- Pada windows : *C:\user\NamaPC\Documents\Arduino\libraries*
- Pada Mac OS: */Users/NamaPC/Documents/Arduino/libraries*

ALAT UKUR MULTIMETER



Multimeter sering disebut dengan AVOMeter atau multitester (singkatan dari Ampere Volt Ohm Meter), Multimeter merupakan alat yang berfungsi untuk mengukur Voltage (Tegangan), Ampere (Arus Listrik), dan Ohm (Hambatan/resistansi) dalam satu unit. Terdapat 2 jenis Multimeter dalam menampilkan hasil pengukurannya yaitu Analog Multimeter (AMM) dan Digital Multimeter (DMM). Sehubungan dengan tuntutan akan keakurasian nilai pengukuran dan kemudahan pemakaiannya serta didukung dengan harga yang semakin terjangkau, Digital Multimeter (DMM) menjadi lebih populer dan lebih banyak dipergunakan oleh para Teknisi Elektronika ataupun dalam penelitian untuk mengetahui pengukuran secara nyata.

Dengan perkembangan teknologi, kini sebuah Multimeter atau Multitester tidak hanya dapat mengukur Ampere, Voltage dan Ohm atau disingkat dengan AVO, tetapi dapat juga mengukur Kapasitansi, Frekuensi dan Induksi dalam satu unit (terutama pada Multimeter Digital). Beberapa kemampuan pengukuran Multimeter yang banyak terdapat di pasaran antara lain:

- Voltage (Tegangan), dengan jenis Arus AC dan arus DC dalam satuan Volt.
- Current (Arus Listrik), baik arus AC maupun arus DC dalam satuan Ampere. Rentang frekuensi pengukuran arus AC yang akurat itu merupakan hal penting, tergantung pada desain dan konstruksi rangkaian, dan harus spesifik, sehingga pengguna dapat mengevaluasi pembacaan yang mereka lakukan. Beberapa multimeter mampu mengukur arus rendah hingga jangkauan milliampere atau bahkan microampere. Semua multimeter memiliki pembebanan tegangan karena didalam multimeter menggunakan kombinasi shunt dan desain internal rangkaian. Beberapa multimeter memiliki pembebanan tegangan yang cukup tinggi sehingga pembacaan arus rendah sangat terganggu.
- Resistance (Hambatan), dalam satuan Ohm.
- Kapasitansi, dalam satuan farad, tetapi biasanya multimeter dibatasi jangkauannya antara beberapa ratus atau ribu mikro farad atau justru beberapa pico farad. Sangat sedikit multimeter serbaguna yang mampu mengukur aspek penting lainnya dari status kapasitor, misal; ESR, faktor disipasi, atau kebocoran kapasitor.
- Konduktansi, dalam satuan Siemens, yang merupakan kebalikan dari resistansi yang diukur.
- Frequency (Frekuensi), dalam satuan Hertz.
- Inductance (Induktansi), dalam satuan Henry. Seperti pengukuran kapasitansi, ini biasanya lebih baik ditangani oleh pengukur induktansi / kapasitansi yang dirancang khusus.
- Suhu, dalam satuan derajat Celcius atau Fahrenheit, dengan probe penguji suhu yang sesuai, biasanya menggunakan thermocouple.

Selain fungsi diatas, multimeter digital dapat juga digunakan untuk memeriksa rangkaian dan atau komponen elektronika diantaranya:

- Uji Kontinuitas, *buzzer* berbunyi ketika resistansi rangkaian cukup rendah (seberapa rendah resistansinya itu sangat bervariasi dari setiap merk dan seri multimeter).
- Dioda, (mengukur *forward drop* pada persimpangan dioda).
- Transistor, (mengukur penguatan arus dan parameter lain pada beberapa jenis transistor)
- Pemeriksaan baterai untuk baterai 1,5 V dan 9 V. Ini adalah pengukuran pemuatan arus, dengan mensimulasikan beban baterai yang sedang digunakan; rentang tegangan yang normal sangat sedikit menguras arus dari baterai.

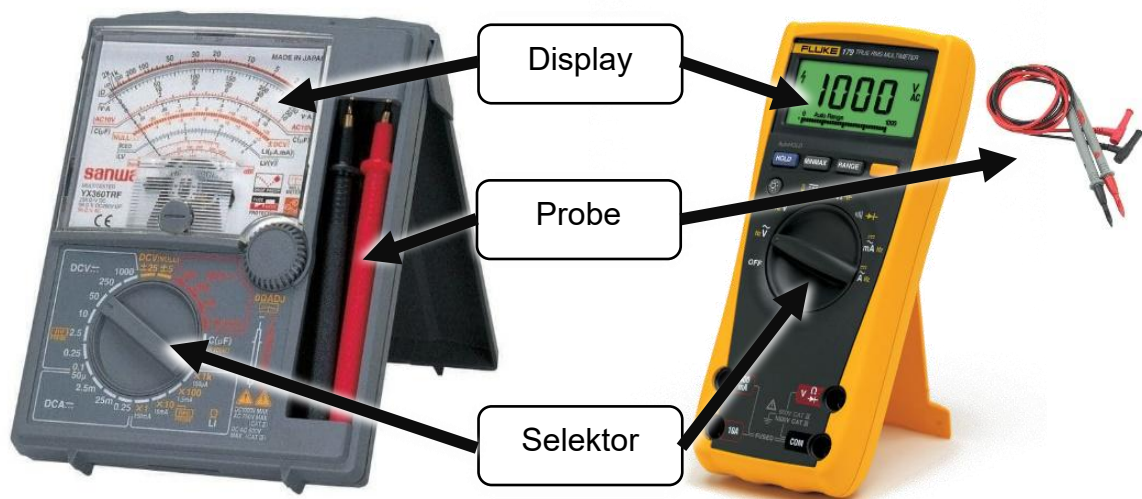
Berbagai sensor juga dapat digunakan bersama dengan multimeter untuk melakukan beberapa jenis pengukuran, hal ini tergantung dari tiap-tiap merk dan jenis multimeter. Semakin mahal harga multimeter semakin lengkap fasilitas yang ada didalamnya dan juga semakin tinggi tingkat proteksi untuk pengamanan alat multimeter tersebut.

Bagian-bagian Multimeter

Multimeter atau multimeter pada umumnya terdiri dari 3 bagian penting, diantaranya adalah:

1. Display
2. Saklar Selektor
3. Probe

Gambar dibawah ini adalah bentuk Multimeter Analog dan Multimeter Digital beserta bagian-bagian pentingnya.



Main Display

Main Display atau *Layar Utama* merupakan area untuk menampilkan nilai hasil pengukuran. Nilai akan muncul dalam bentuk angka berukuran besar.

Probe

Test Probe adalah perangkat fisik yang digunakan untuk menghubungkan multimeter ke perangkat yang diuji atau *Device Under Test* (disingkat: *DUT*). *Test Probe* tersedia dari perangkat yang sangat sederhana dan kuat hingga probe kompleks yang canggih, mahal, dan *fragile* (mudah pecah). Ada beberapa jenis *Test Probe* yang tersedia, misal; probe osiloskop, probe arus dan lain sebagainya. *Test Probe* merupakan perangkat yang selalu disertakan didalam paket pembelian multimeter, dan biasanya disertakan pula probe suhu. Namun demikian, *Test Probe* juga dijual terpisah dengan berbagai bentuk, fungsi dan ujung yang sederhana hingga ke yang sangat unik.

Selektor

Function Selector atau *Selektor Fungsi* merupakan saklar utama yang menggunakan saklar jenis putar atau Rotary Switch untuk memilih salah satu fungsi multimeter. Pada fungsi-fungsi tersebut, ada simbol yang berwarna kuning diatas simbol utama, itu menunjukkan ada sub fungsi yang dibisa dipilih melalui *Sub Selection Button*.

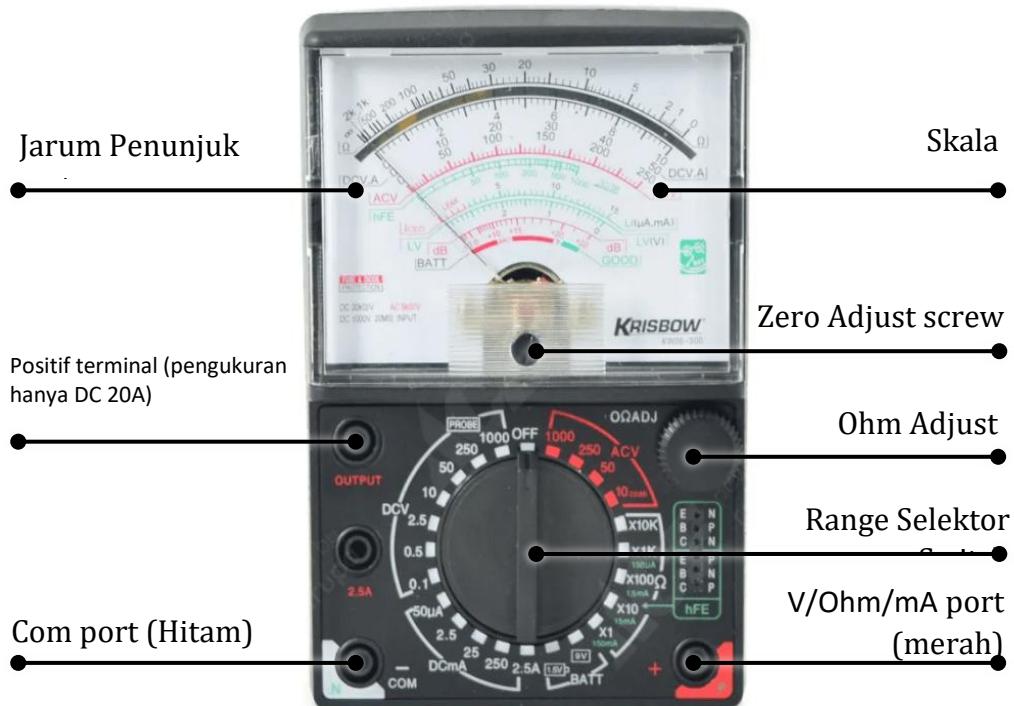
Selector menentukan fungsi multimeter. Fungsi-fungsi menu multimeter dijelaskan pada tabel dibawah ini:

Tabel 2. Tabel Fungsi Selektor pada multimeter Digital atau multimeter Analog.

NO	SIMBOL	FUNGSI	SUB FUNGSI	DESKRIPSI
1.	OFF	Multimeter mati	-	-
2.	V	Pengukuran tegangan	- Arus DC - Arus AC	- $\leq 1000V$ - $\leq 750V$
3.	mV	Pengukuran tegangan	- Arus DC - Arus AC	- $\leq 600mV$ - $\leq 600mV$
4.	- Ohm (Ω) - Buzzer - Dioda - Kapasitor	- Pengukuran Resistansi - Pengukuran Kontinuitas - Pemeriksaan Dioda - Pengukuran Kapasitor	- Resistansi - Kontinuitas - Tes Dioda - Kapasitansi	- $\leq 60M\Omega$ - - - $\leq 9.999mF$
5.	- Hz - %	- Pengukuran Frekuensi - Pengukuran Duty Cycle	- Frekuensi - Siklus Kerja	- Tegangan rendah - 1% - 99%
6.	A	Pengukuran arus	- Arus DC - Arus AC	- $\leq 10A$ - $\leq 10A$
7.	mA	Pengukuran arus	- Arus DC - Arus AC	- $\leq 600mA$ - $\leq 600mA$
8.	$^{\circ}C$ $^{\circ}F$	Pengukuran suhu	- Celcius - Fahrenheit	- (-10) ~ 1000 - (-4) ~ 1832

Multimeter Analog

Multimeter Analog atau Multimeter Jarum adalah alat pengukur besaran listrik yang menggunakan tampilan dengan jarum yang bergerak ke range-range yang di ukur dengan probe. Analog tidak digunakan untuk mengukur secara detail suatu besaran nilai komponen tetapi kebanyakan hanya di gunakan untuk baik atau buruknya komponen pada waktu pengukuran atau juga di gunakan untuk memeriksa suatu rangkaian apakah sudah tersambung dengan baik sesuai dengan rangkaian blok yang ada.



Dari gambar multimeter dapat dijelaskan bagian-bagian dan fungsinya:

1. **Display:** Jarum penunjuk meter (Knife -edge Pointer) pada multimeter analog, Berfungsi sebagai penunjuk besaran yang diukur.
2. **Skala:** Berfungsi sebagai skala pembacaan meter, yaitu: skala tegangan skala arus dan skala resistor.
3. **Zero adjust screw:** Berfungsi untuk mengatur kedudukan jarum penunjuk dengan cara memutar sekrupnya ke kanan atau ke kiri dengan menggunakan obeng pipih kecil (-).
4. **Zero Ohm Adjust Knob:** Berfungsi untuk mengatur jarum penunjuk pada posisi nol. Caranya, saklar pemilih diputar pada posisi (ohm), test lead + (merah) dihuungkan ke test lead - (hitam), kemudian tombol pengatur diputar ke kiri atau ke kanan sehingga jarum menunjuk pada 0 Ohm.
5. **Lubang kutub + :** Berfungsi sebagai tempat masuknya test lead kutub (+) yang berwarna merah.

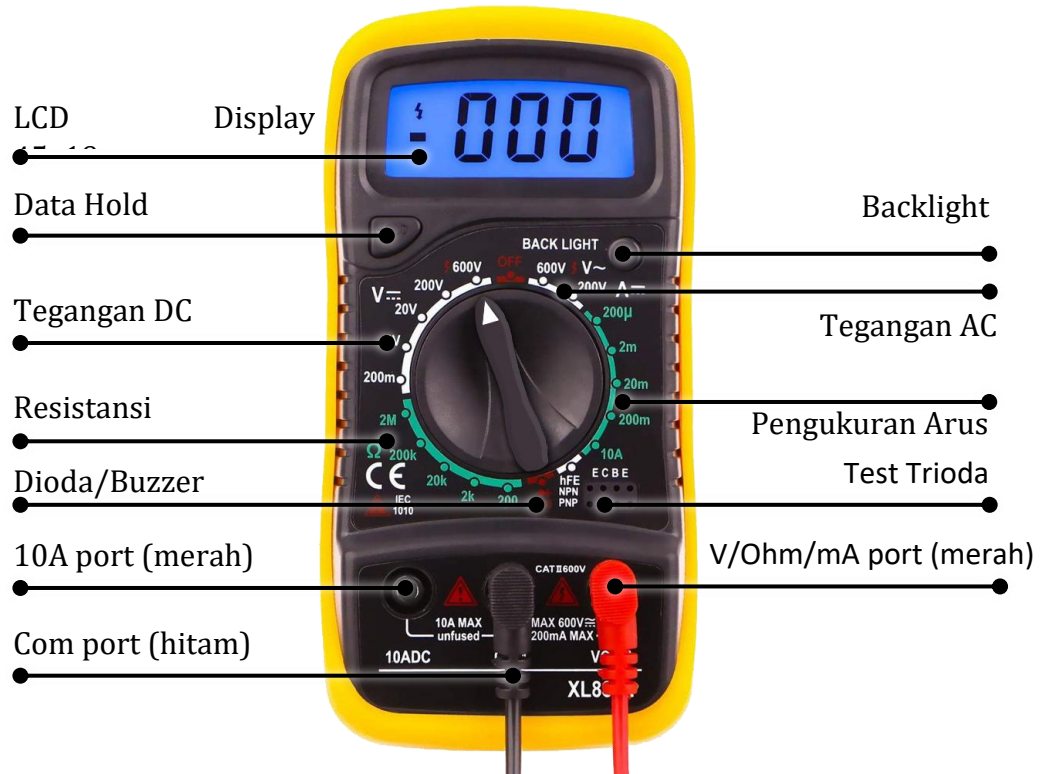
6. **Positive Terminal (20 A DC Only):** Berfungsi sebagai tempat masuknya test lead kutub (+) yang berwarna merah ketika mengukur Arus DC dari 0-20 Ampere
7. **Range Selector Switch:** Berfungsi untuk memilih posisi pengukuran dan batas ukurannya. Multimeter biasanya terdiri dari empat posisi pengukuran, yaitu:
 - a. **Posisi W (Ohm)** berarti multimeter berfungsi sebagai ohmmeter, yang terdiri dari tiga batas ukur: x 1; x 10; dan K W
 - b. **Posisi ACV (Volt AC)** berarti multimeter berfungsi sebagai voltmeter AC yang terdiri dari lima batas ukur: 10; 50; 250; 500; dan 1000.
 - c. **Posisi DCV (Volt DC)** berarti multimeter berfungsi sebagai voltmeter DC yang terdiri dari lima batas ukur: 10; 50; 250; 500; dan 1000.
 - d. **Posisi DCmA (miliampere DC)** berarti multimeter berfungsi sebagai mili amperemeter DC yang terdiri dari tiga batas ukur: 0,25; 25; dan 500. Tetapi ke empat batas ukur di atas untuk tipe multimeter yang satu dengan yang lain batas ukurannya belum tentu sama.
8. **Lubang kutub - :** Berfungsi sebagai tempat masuknya test lead kutub (-) yang berwarna hitam.

Multimeter Digital

Penggunaan multimeter digital tidaklah sesulit multimeter analog, hal ini karena fitur-fitur yang ada pada multimeter digital dikembangkan mendekati keinginan para penggunanya. Ada fitur auto-polarity, auto-range, auto-power off merupakan fitur yang tidak akan ditemukan didalam multimeter analog. Namun demikian, penggunaan multimeter digital tetaplah harus hati-hati, karena multimeter digital juga masih memiliki batasan dalam kinerjanya. Misal; pada saat mengukur resistansi maka jangan menghubungkan multimeter ke sumber listrik, hal ini secara logika saja akan merusak multimeter.

1. **Main Display:** *Main Display* atau *Layar Utama* merupakan area untuk menampilkan nilai hasil pengukuran. Nilai akan muncul dalam bentuk angka berukuran besar.
2. **Data Hold:** Sedangkan fungsi HOLD digunakan untuk mempertahankan nilai pengukuran jika pengguna tidak ingin terus menerus menempelkan probe pada objek yang sedang diperiksa. Hal ini biasanya dilakukan ketika pengguna ingin mencatat hasil pemeriksaan. Maksud dari *mempertahankan nilai hasil pengukuran* yaitu ketika pengguna ingin membaca hasil pengukuran dengan keadaan ingin melepas probe dari objek setelah dilakukan pengukuran. Prosedur penggunaannya sebagai berikut:
 - Hubungkan probe multimeter pada objek yang akan diperiksa
 - Nilai hasil pengukuran akan tampil pada layar utama
 - Tekan pendek tombol HOLD tanpa melepas probe dari objek yang sedang diperiksa
 - Pastikan simbol HOLD telah muncul pada layar sub menu

- Lepas probe dari objek yang sedang diperiksa
- Baca nilai hasil pengukuran pada layar utama
- Nilai hasil pengukuran tidak akan berubah selama simbol HOLD masih muncul pada layar sub menu walaupun probe sudah dilepas dari objek yang diperiksa
- Tekan pendek tombol HOLD untuk melepaskan nilai yang tertahan
- Pastikan simbol HOLD telah hilang dari layar jika ingin melakukan pemeriksaan ulang atau akan memeriksa objek lainnya



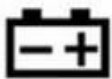
3. **Range Selector Switch:** Yang termasuk dalam fungsi range selector switch pada multimeter digital adalah:



- a. **Posisi Tegangan DCV (Volt DC)** berarti multimeter berfungsi sebagai voltmeter DC yang terdiri dari Lima batas ukur: 200mV; 2V; 20V; 200V; dan 600V.
- 200mV artinya akan mengukur tegangan yang maximal 0,2 Volt
 - 2V artinya akan mengukur tegangan yang maximal 2 Volt
 - 20V artinya akan mengukur tegangan yang maximal 20 Volt
 - 200V artinya akan mengukur tegangan yang maximal 200V
 - 600V artinya akan mengukur tegangan yang maximal 750V
- b. **Posisi Resistansi (Ohm)** berarti multimeter berfungsi sebagai ohmmeter, yang terdiri dari lima batas ukur: 200; 2k; 20k; 200k; dan 2M. Perhatikan skala Pengukuran pada Ohm Meter.
- 200 artinya mengukur hambatan yang nilainya max. 200 Ohm.
 - 2k artinya mengukur hambatan yang nilainya max. 2000 Ohm (2kOhm).
 - 20k artinya mengukur hambatan yang nilainya max. 20.000 Ohm (20kOhm).
 - 200K artinya mengukur hambatan yang nilainya max. 200.000 Ohm (200K Ohm).
 - 2M artinya mengukur hambatan yang nilainya 2.000.000 Ohm (2000kOhm atau 2 Mega Ohm).
- Bila tidak diketahui besaran nilai yang mau diukur, dianjurkan pilih skala tengah misalnya skala 20k, lalu lakukan pengukuran. Jika hasilnya 1 (Overload) maka naikkan skala. - Jika hasilnya digit dibelakang koma kurang akurat, maka turunkan skala. Contoh pembacaan hasil: - Pada skala 2k hasilnya 1,76 itu artinya hambatan yang terukur adalah 1,76 k Ohm. - Pada skala 2k hasilnya 0,378 itu artinya hambatan yang terukur adalah 0,378 k Ohm alias 378 Ohm. (KOhm ke Ohm dikali 1000). - Pada skala 20K hasilnya 1, artinya obyek yang mau diukur melebihi skala 20k, maka naikan skala menjadi 200k, hasilnya menjadi 38,78 itu artinya hambatan yang terukur adalah sebesar 38,78 kOhm.
- c. **Buzzer:** digunakan untuk pemeriksaan kontinuitas. Pada fungsi ini, pengguna tidak perlu memperhatikan nilai di layar, cukup dengarkan bunyi *beep* yang keluar dari *buzzer* di dalam multimeter.
- d. **hFE NPN/PNP:** multimeter digunakan untuk menguji Dioda dan Transistor
- e. **Posisi DCmA (miliampere DC)** berarti multimeter berfungsi sebagai mili amperemeter DC yang terdiri dari tiga batas ukur: 200u; 2mA; 20mA; 200mA dan 10A. Khusus untuk pengukuran arus DC yang melebihi batas antara 200mA sampai 10A, maka probe port warna merah harus diletakkan pada lubang port tertulis 10A port (lihat gambar multimeter digital).

- f. **Posisi ACV (Volt AC)** berarti multimeter berfungsi sebagai voltmeter AC atau pengukur tegangan AC yang terdiri dari dua batas ukur: 200Volt; dan 600Volt.
4. **Backlight:** Ini adalah tombol yang memiliki fungsi *backlight* atau pencahayaan layar aktif. *Backlight* biasanya digunakan pada saat melakukan pengukuran objek di area yang kurang pencahayaan hal ini untuk memperjelas tampilan layar.





Low power

Switch

Transistor / diode socket

Continuity/diode

Resistance

Tripolar Zero FireWire

DC voltage

Black test lead jack

Red test lead socket

Indicator light

Capacitance

DC current

Alternating current

AC voltage

Current / 1.5V capacitance

20A current

AUTO POWER OFF

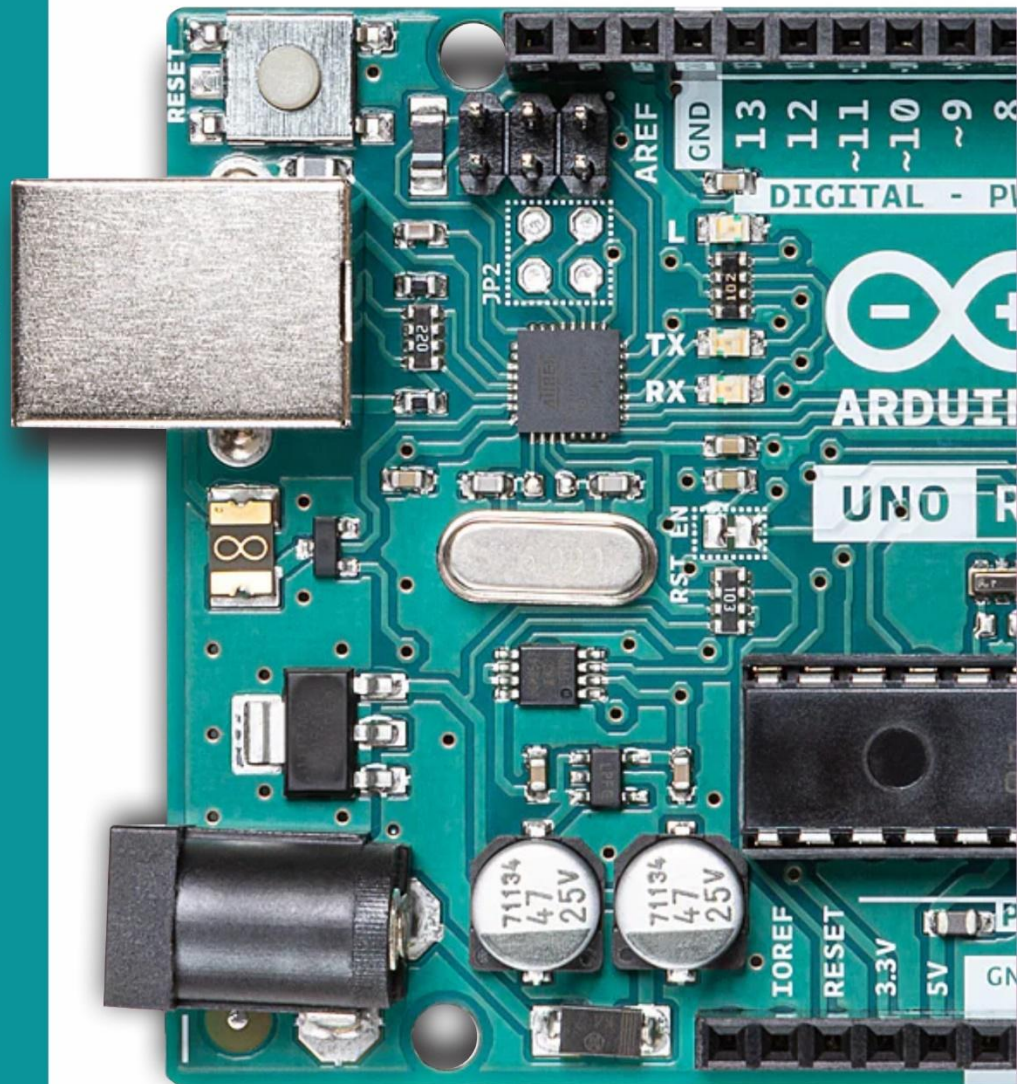
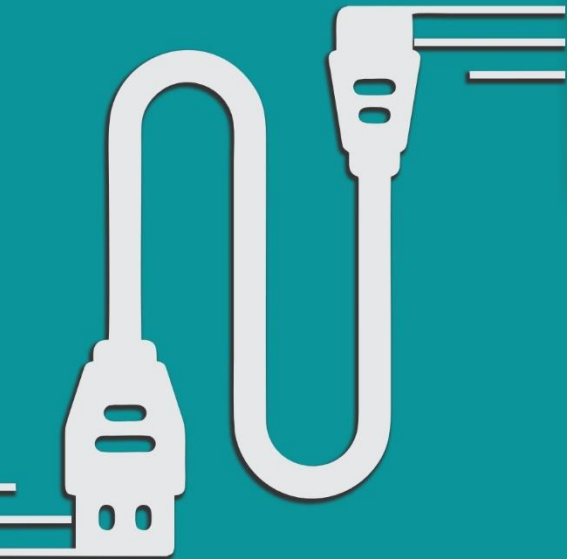
01

Panduan Praktis Proyek

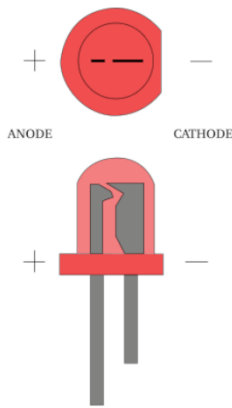
ARDUINO

LED Berkedip

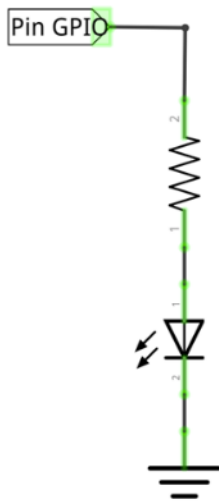
Semua bahasa pemrograman biasanya memiliki contoh program dasar seperti “Hello World”. Contoh program ini menunjukkan bahwa kode program bisa aktif dan berjalan dengan benar/normal. Di Arduino IDE, ada contoh program dasar yang disebut “Blink” digunakan untuk board Arduino.



Light Emitting Diode



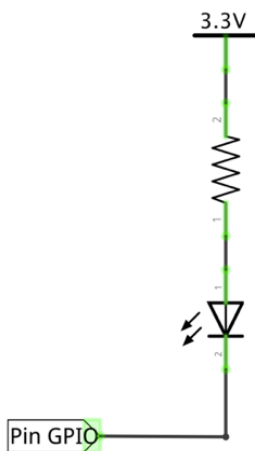
Berbagai kemasan LED dipasaran tersedia bermacam-macam bentuk yang berbeda, perbedaanya pada bentuk lensanya. Semua jenis lensa yang tersedia bertujuan untuk mengontrol cahaya yang dihasilkannya. Hal terpenting pada LED adalah kecerahan dan sudut pandang. Kecerahan biasanya diukur dalam microcandelas (mcd), dapat berkisar antara beberapa ribu sampalebih pendek adalah katoda yang terhubung ke catudaya negatif (-) atau ground. Selain panjang pendek kaki LED, kita bisa memperhatikan kutub positif dan kutub negatif dengan melihat selungkup epoksi yang memiliki sisi datar merupakan kutub katoda/negatif/ground.



Konfigurasi LED Aktif Tinggi

Disaat pin digital diatur ke logika 1, pin digital akan bertegangan 3,3Volt. Hal ini mengakibatkan terjadi bias tegangan maju melewati dioda LED yang menyebabkan arus mengalir melalui LED dan memancarkan cahaya. Saat pin digital dirubah dari 1 ke logika ke 0, pin digital akan terjadi penurunan tegangan pada LED, sehingga tidak ada arus yang mengalir melalui LED dan padam (tidak menyala).

PIN	LED
HIGH	ON
LOW	OFF

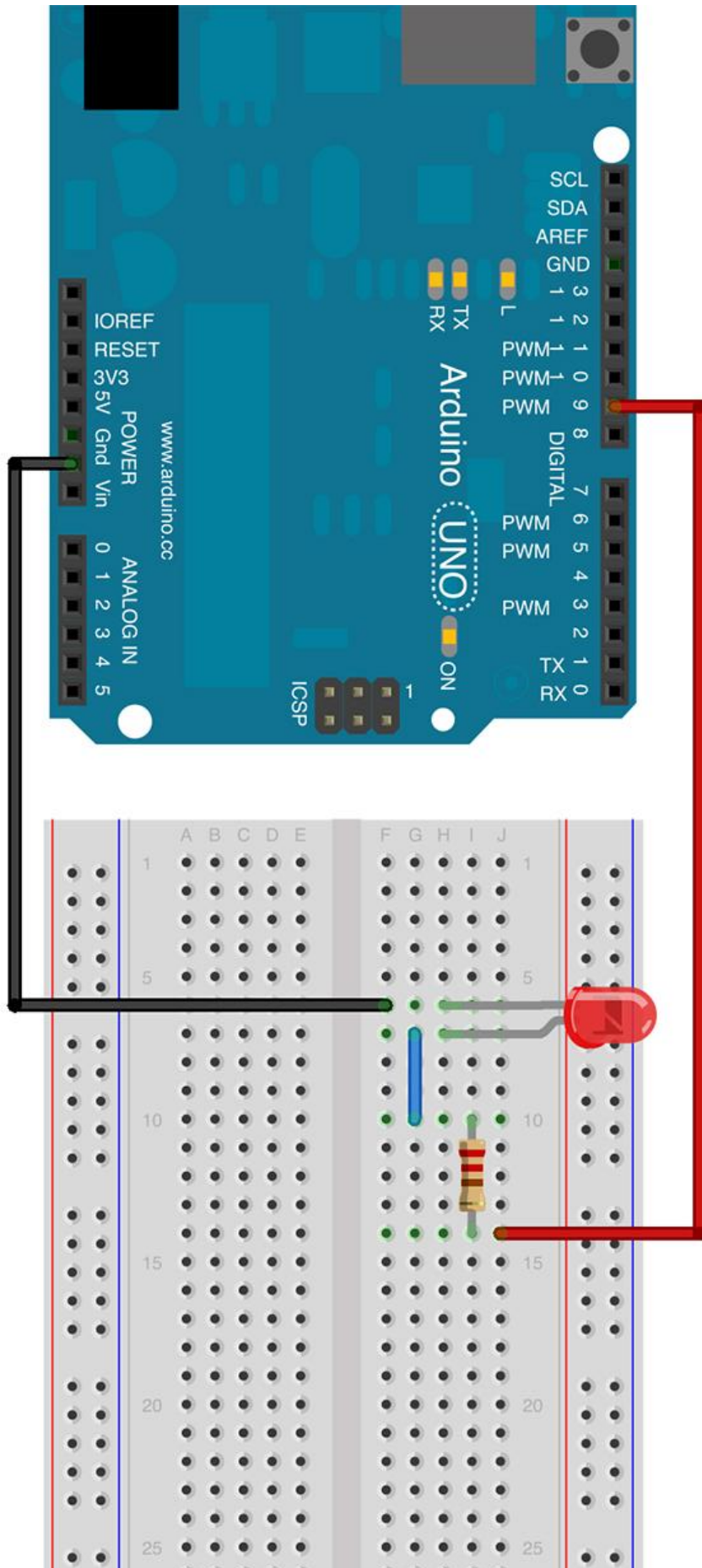


Konfigurasi LED Aktif Rendah

Disaat pin digital diatur ke logika 0, pin digital menjadi bertegangan 0Volt. Hal ini mengakibatkan terjadi bias tegangan maju melewati dioda LED yang menyebabkan arus mengalir melalui LED dan memancarkan cahaya. Saat GPIO dirubah ke logika dari 0 ke 1, pin digital akan menjadi 3,3Volt. akan terjadi penurunan tegangan pada LED, sehingga tidak ada arus yang mengalir melalui LED dan padam.

PIN	LED
LOW	ON
HIGH	OFF

Rangkaian LED Berkedip



Proyek pertama ini, kita memulai dengan membuat sebuah LED (*Light Emittin Diode*) berkedip menggunakan Arduino, secara singkatnya LED akan menyala dan mati sesuai dengan delay yang diatur pada program (sketch). Pada proyek ini kita mengatur agar LED menyala selama 250 milidetik. Setelah itu LED akan mati selama 1 detik atau 1000 milidetik, kemudian mati kembali. Kondisi ini berulang terus menerus sampai Arduino di matikan atau catudaya listrik tidak diberikan.

Sketch Led Berkedip

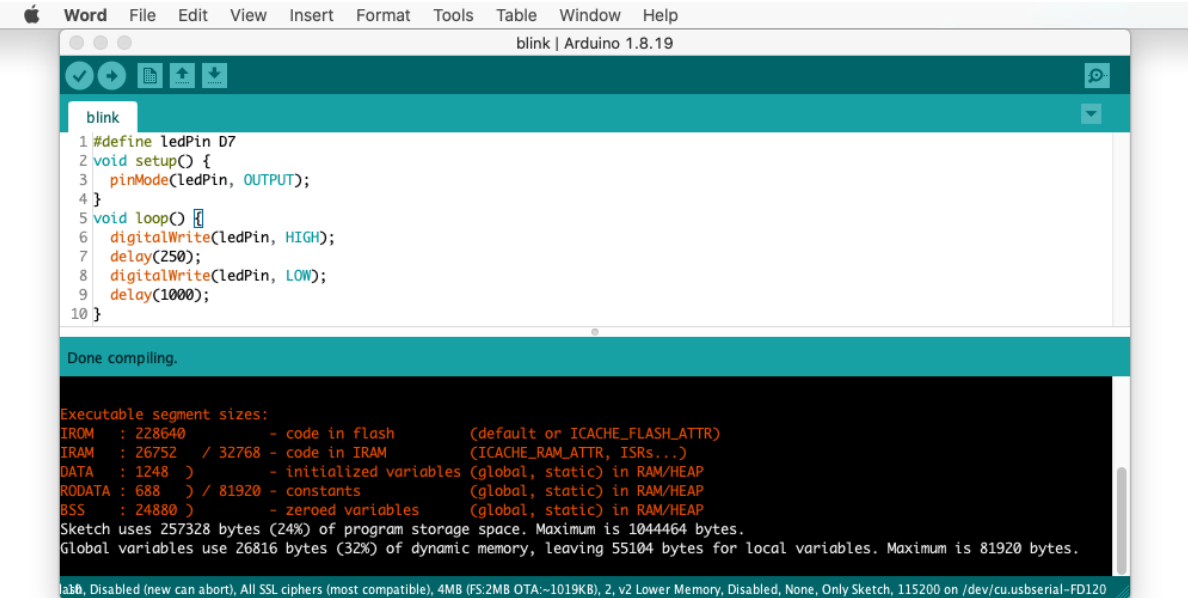
Program mengatur pin yang terhubung ke LED yaitu pin D13 dijadikan sebagai OUTPUT. Pin D13 ini didalam board Arduino merupakan pin yang telah disediakan LED onboard didalamnya. Konfigurasi rangkaiannya LED onboard Arduino menggunakan konfigurasi aktif rendah. Sehingga, apabila kita memprogram LED di pin D13 kita bisa saja tanpa dihubungkan seperti pada rangkaian LED diatas. Dalam proyek perdana ini, kita mencoba untuk mengendalikan LED yang terhubung dengan pin D13. Pin D13 diberi logika HIGH (konfigurasi rangkaian LED Aktif Rendah) untuk mematikan LED selama 250milidetik (1 detik). Kemudian dinyalakan dengan memberikan logika LOW dan di sertai tundaan waktu (delay) selama 1000milidetik atau satu detik. Proram akan berulang terus menerus tanpa batas sampai terjadi reset atau catudaya dimatikan.

Ketik kode program berikut dengan membuat proyek baru di Arduino IDE dengan meng klik **File -> New Sketch**.

```
1. #define ledPin D13
2. void setup() {
3.     pinMode(ledPin, OUTPUT);
4. }
5. void loop() {
6.     digitalWrite(ledPin, HIGH);
7.     delay(250);
8.     digitalWrite(ledPin, LOW);
9.     delay(1000);
10. }
```

Setelah kode program diketik dalam proyek baru, lakukan penyimpanan file dengan klik **File -> Save As** kemudian berinama file proyek tersebut dengan nama "Blink". Nanti akan menghasilkan nama file dengan ekstensi `.ino` pada proyek Arduino, yaitu menjadi "Blink.ino".

Untuk menguji kode program telah benar dalam sintak penulisan dan tidak ada kesalahan ketik. Kita bisa mengujinya dengan menggunakan tool tombol "**verify**" yang ada di bagian kolom menu Arduino IDE. Apabila penulisan sintak dan pengetikan sudah benar, maka pada bagian bawah "status monitor" akan tertampil "**Done Compiling**". Apabila masih ada kesalahan, maka tertampil error yang kemudian di tampilan windows bawah status, akan dijelaskan pada bagian baris dan kolom kesalahan yang terjadi. Lihat pada gambar berikut, menunjukkan **verify** / kompilasi telah dilakukan dan tidak terdapat **error** atau kesalahan pada kode program.



```

Word  File  Edit  View  Insert  Format  Tools  Table  Window  Help
blink | Arduino 1.8.19
blink
1 #define ledPin D7
2 void setup() {
3   pinMode(ledPin, OUTPUT);
4 }
5 void loop() {
6   digitalWrite(ledPin, HIGH);
7   delay(250);
8   digitalWrite(ledPin, LOW);
9   delay(1000);
10 }

Done compiling.

Executable segment sizes:
IRAM : 228640 - code in flash (default or ICACHE_FLASH_ATTR)
IRAM : 26752 / 32768 - code in IRAM (ICACHE_RAM_ATTR, ISRs...)
DATA : 1248 - initialized variables (global, static) in RAM/HEAP
RODATA : 688 / 81920 - constants (global, static) in RAM/HEAP
BSS : 24880 - zeroed variables (global, static) in RAM/HEAP
Sketch uses 257328 bytes (24%) of program storage space. Maximum is 1044464 bytes.
Global variables use 26816 bytes (32%) of dynamic memory, leaving 55104 bytes for local variables. Maximum is 81920 bytes.

```

Gambar Verify Kode Program

Penjelasan Kode Program:

Program dimulai dengan mendeklarasikan variabel tetap (*fixed variable*) pada pin D0 dengan diberi nama `ledPin`, dibaris 1. Bila variabel tetap `ledPin` (D13) sudah dideklarasikan, maka pada kode program selanjutnya kita tidak diperlukan lagi mengingat pin yang digunakan untuk antarmuka ke LED. Kita hanya mengingat D13 dengan nama `ledPin`. Namun, variabel `ledPin` ini harus di konfigurasi menjadi OUTPUT, karena pin D13 terhubung dengan LED. Untuk perintah konfigurasi ini berada pada fungsi `void setup()` yaitu pada baris 3.

Fungsi void `setup()` pada program dibaris 2 sampai 4, memiliki perintah yang akan di baca sekali. Pada program ini berfungsi untuk menginisialisasi variable dan mendeklarasikan pin yang digunakan berupa INPUT atau OUTPUT.

Fungsi void `loop()` pada baris 5 sampai 10, memiliki perintah yang akan di baca berulang-ulang. Fungsi `digitalWrite` dibaris 6, memberikan perintah `ledPin` berlogika HIGH (1), sehingga membuat LED akan padam. Kondisi LED yang padam di berikan jeda waktu sampai 1000milidetik (1 detik), perintah untuk penundaan waktu ini pada baris 7.

Setelah kondisi LED padam selama 1000 milidetik, LED dinyalakan atau ON dengan memberikan logika LOW pada pin `ledPin` (D0) di baris 8. LED akan menyala selama 1detik (1000milidetik) karena terdapat perintah tundaan waktu pada baris 9. Setelah proses tundaan waktu dibaris 9, selanjutnya program akan menemui tutup kurawal di baris 10, yang menandakan akhir dari fungsi diatasnya, yaitu fungsi void `loop()`. Karena tutup kurawal ini masuk dalam body fungsi void `loop()`, maka program akan mengulang dari atas dalam body void `loop()`, yaitu memulai di baris 6 lagi.

Sampai disini telah terjadi satu siklus perulangan pada void `loop()`. Siklus perulangan ini akan berjalan terus menerus didalam body fungsi void `loop()`sampai catudaya dimatikan.

Catatan rogram:



DAFTAR PUSTAKA

- Filip Rak, J. W. (2021). *Comparison of ESP programming platforms*. Computer Science and Information Technologies, 2, 77-86.
- Geddes, M. (2017). *Arduino Project Handbook, Volume 2*. San Francisco: William Pollock.
- Geddes, M. (2016). *Arduino Project Handbook*. San Francisco: William Pollock.
- Craft, B. (2013). *Arduino Projects For Dummies*. England: John Wiley & Sons, Ltd.
- Wiley. (2015). *Arduino Sketches: Tools and Techniques for Programming Wizardry*. Canada: Indiana Published simultaneously .
- Purdum, J. (2015). *Beginning C for Arduino, Second Edition: Learn C Programming for the Arduino*. Ohio: Springer Science.
- Javed, A. (2016). *Building Arduino Projects for the Internet of Things: Experiments with Real-World Applications*. Illinois: Springer Science.
- Bayle, J. (2013). *C Programming for Arduino*. Birmingham B3 2PB: Packt Publishing.
- Expressife. (2020). *ESP8266 Technical Reference*. -: Expressif System.
- Thakur, M. R. (-). *NodeMCU ESP8266 Communication Methods and Protocols Programming with Arduino IDE*. -: -.
- Monk, S. (2016). *Programming Arduino: Getting Started with Sketches, Second Edition*. San Francisco: McGraw-Hill Education.
- Kolban, N. (2015). *Book On ESP8266*. Texas: -.
- Wisnu Jatmiko, P. M. (2015). *Real Time Operating System (RTOS) Teori dan Aplikasi*. Jakarta: Fakultas Ilmu Komputer Universitas Indonesia.
- Rosad, S. (2022). *Bahasa Pemrograman Java Tingkat Dasar*. Yogyakarta: Nasmedia. ISBN: 978-623-351-615-0
- Rosad, S. (2023). *Pemrograman NodeMCU ESP8266 Menggunakan Arduino IDE*. Cilacap: UNUGHAPRESS. ISBN: 978-623-09-1963-3

Tentang Penulis



Safiq Rosad, seorang akademisi dan praktisi teknologi yang menggabungkan latar belakang Teknik Elektro dan Teknik Informatika. Beliau merupakan Dosen di Jurusan Informatika Universitas Nahdlatul Ulama Al Ghazali Cilacap, sekaligus pendiri SHATOMEDIA (sejak 2008) yang bergerak di bidang pengembangan hardware, software, dan fabrikaasi elektronika. Dengan keahlian utama pada Embedded System, Internet of Things (IoT), Cyber Physical System, dan Sound Engineering, aktif berkontribusi dalam riset dan saat ini sedang menempuh studi doktoral di Universitas Ahmad Dahlan Yogyakarta



Adhiya Nur Septio, kelahiran Cilacap 2001, memperoleh gelar sarjana dari Universitas Nahdlatul Ulama Al Ghazali Cilacap jurusan Informatika pada tahun 2022. Bidang yang digeluti adalah Pengembangan Perangkat Lunak.



Muhamad Rifqi Kamali, kelahiran Cilacap 1999, memperoleh gelar sarjana dari Universitas Nahdlatul Ulama Al Ghazali Cilacap jurusan Informatika pada tahun 2022. Bidang yang digeluti adalah Internet Of Things.

