

Mengajak Anak SD Berpetualang dengan Coding Arduino

Muhamad Irsadul Ngibad



Mengajak Anak SD Berpetualang dengan Coding Arduino

Penulis :

Muhamad Irsadul Ngibad

Desain Sampul :

Sambas R

Penata Letak :

W. Slam

Cetakan Pertama :

November 2025

ISBN :

Penerbit :

PT Kirei Nara Press (Anggota IKAPI)

Jl. Dahlia No. 90 Perum. Pesona Bumi Mandiri 2

RT 06 RW 03 Beran Tambaharjo Pati

Kab. Pati Provinsi Jawa Tengah

Website : www.kireinarapress.com

Email : redaksi.kireinarapress.com

☎ 082135593898 / 08232 888 9482

KATA PENGANTAR

Alhamdulillah robbil álamiiin, pada akhirnya berkat limpahan rahmat dari Allah SWT, penulis mampu kembali menghadirkan buku berjudul, "Mengajak Anak SD Berpetualang dengan Arduino."

Penulisan buku ini didasari oleh kebutuhan akan buku coding untuk anak-anak usia sekolah dasar. Kalau buku sebelumnya, "Mengajak Anak Berpetualang Coding dengan Arduino dan IoT", lebih banyak digunakan untuk siswa tingkat SMP dan SMA, maka muncul permintaan buku serupa yang bisa digunakan untuk tingkat SD.

Berdasarkan permintaan dan kebutuhan tersebut, dengan segenap kemampuan, penulis berupaya untuk menulis buku coding arduino dengan sentuhan rasa anak-anak sekolah dasar. Sehingga nantinya anak-anak SD, guru-guru SD atau pun orang tua siswa lebih mudah memahami dan memparktekkkan apa yang ada dalam buku ini.

Dengan begitu dengan kehadiran buku ini, mampu membantu tujuan utama adalah memperkenalkan dunia coding sejak dini. Apabila di usia dini sudah diperkenalkan coding, maka diharapkan mapu tercetak generasi yang mampu menganalisa masalah dan membuat penyelesaian masalah/solusi. Sehingga di kemudian hari akan tercipta generasi yang solutif inovatif

Demikian, penulis berharap semoga buku ini dapat bermanfaat bagi perkembangan kreatifitas dan semngat berinovasi bagi anak-anak sejak dini.. Saran, masukan dan kritik membangun penulis nantikan untuk perbaikan selanjutnya.

Selamat berpetualang bersama coding Arduino!

Pati, 2025

Muhamad Irsadul Ngibad

DAFTAR ISI

Halaman Judul

Kata Pengantar

Daftar Isi

Petualangan 1 : Ayo, Mengenal Arduino!

Petualangan 2 : Bermain Komponen!

Petualangan 3 : Yuk, Siapkan Codingmu!

Petualangan 4 : Rangkai Kreasimu!

Petualangan 5 : Asyiknya Simulasi!

Petualangan 6 : Nikmati Petualangamu!

Daftar Pustaka

Tentang Penulis

π

"Anak adalah peniru terbaik, jadi berikanlah mereka sesuatu yang hebat untuk ditiru."

- Anonim -

Petualangan 1

AYO, MENGENAL ARDUINO!

Sekarang saatnya berkenalan dengan Arduino. Pasti Ayahbunda sudah pernah mendengarnya, bukan? Tapi sebelumnya, apakah Ayahbunda sudah pernah mendengar mikrokontroler? Baiklah, kita akan mengingat kembali sedikit tentang mikrokontroler.

Mikrokontroler sering dikenal dengan sebut μC , uC , atau MCU. Mikrokontroler adalah komputer yang berukuran mikro dalam satu chip IC (integrated circuit) yang terdiri dari processor, memory, dan antarmuka yang bisa diprogram. Jadi disebut komputer mikro karena dalam IC atau chip mikrokontroler terdiri dari CPU, memory, dan I/O yang bisa kita kontrol dengan memprogramnya. I/O juga sering disebut dengan GPIO (General Purpose Input Output Pins) yang berarti, pin yang bisa kita program sebagai input atau output sesuai kebutuhan.

Nah, Arduino sebenarnya juga merupakan sebuah mikrokontroler, yang berawal dari sebuah thesis oleh Hernando Barragan, di Institute Ivrea, Italia pada tahun 2005, yang kemudian dikembangkan oleh Massimo Banzi dan David Cuartielles dan diberi nama *Arduino of Ivrea*. Lalu diganti nama menjadi Arduino yang dalam bahasa Italia berarti teman yang berani.

Arduino kemudian dikenal banyak orang karena memiliki beberapa kelebihan dibanding mikrokontroler non Arduino, diantaranya adalah :

1. *Open Source Hardware Software* ataupun hardware Arduino bersifat *open source* sehingga kita akan dapat membuat tiruan.
2. Tidak butuh *chip programming, downloader Chip* yang ada pada Arduino dilengkapi bootloader yang nantinya akan dapat menangani proses upload dari computer.
3. Koneksi USB Sambungan dari board Arduino ke komputer dapat menggunakan USB dan tidak memerlukan parallel port ataupun serial port. Ini akan membuat sambungannya semakin mudah

terutama bagi laptop yang tidak dilengkapi serial ataupun parallel port.

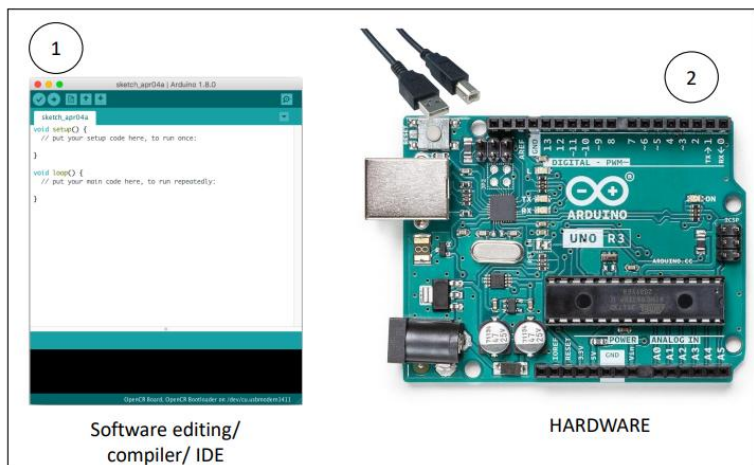
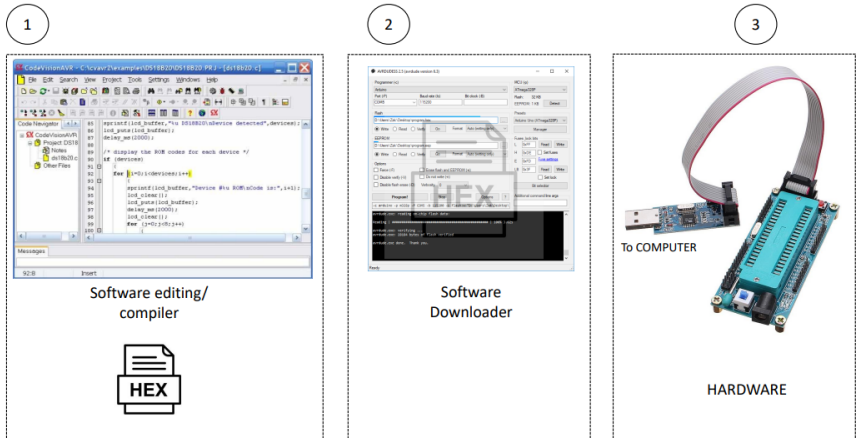
4. Fasilitas Chip Yang Lengkap Arduino telah menggunakan chip AVR AT mega 168/328 dengan kapasitas PWM, ADC, timer, komunikasi serial, interrupt, I2C serta SPI. Ini berarti, Arduino dapat digabung bersama modul maupun alat lain yang protokolnya berbeda-beda.
5. Ukuran Kecil Arduino memiliki ukuran yang relatif kecil sehingga dapat dengan mudah dibawa kemanapun kita pergi. Arduino bisa dibawa bersama laptop ataupun dimasukkan dalam saku.
6. Bahasa Pemrograman Relatif Mudah Walau bahasa pemrograman yang digunakan ialah bahasa C/C++ namun dengan ditambahkannya library serta beragam fungsi standar membuat Arduino bisa dipelajari dengan lebih mudah.
7. Library Gratis Tersedia library gratis yang akan menghubungkan Arduino dengan beragam sensor, modul komunikasi serta aktuator. Sebagai contoh adalah library khusus keyboard, mouse, GPS, servo dan lain-lain. Mengingat Arduino bersifat open

source, maka library yang ada juga bisa didownload secara gratis di website resmi Arduino.

8. Pengembangan Aplikasinya Mudah Mengembangkan aplikasi elektronik satu ini terbilang mudah, apalagi dengan adanya bahasa pemrograman yang mudah serta library dasar yang terbilang lengkap. Contohnya adalah apabila kita hendak membuat sebuah sensor suhu. Cukup beli IC sensor suhu dan sambungkan ke Arduino. Apabila suhu hendak ditampilkan di LCD, belilah LCD dan tambahkan ke library.
9. Komunitas Open Source Saling Mendukung Ada komunitas saling mendukung di dalamnya demi pengembangan proyek. pengembangan software maupun hardwarenya didukung para pecinta elektronika di seluruh dunia.

Kelebihan-kelebihan inilah yang membuat orang-orang kemudian banyak menggunakan mikrokontroler Arduino di banding mikrokontroler non-Arduino. Dengan kata lain, dengan Arduino kita bisa melakukan pemrograman dan bisa langsung dieksekusi oleh hardware tanpa harus

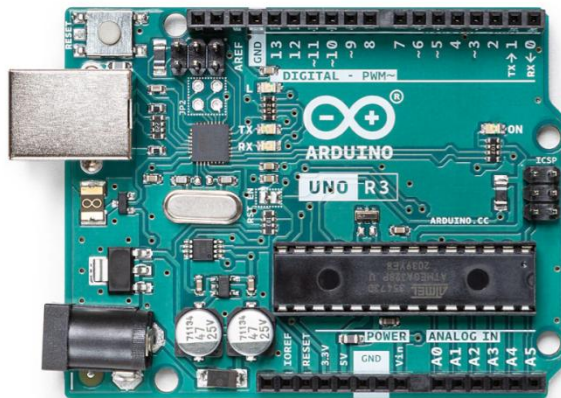
melalui tambahan software downloader. Hal ini tentu saja memangkas satu langkah dari mikrokontroller non-Arduino. Gambarannya adalah sebagai berikut :



Varian atau jenis dari Arduino begitu beragam, berkembang sesuai dengan kebutuhan. Berikut beberapa varian dari Arduino :

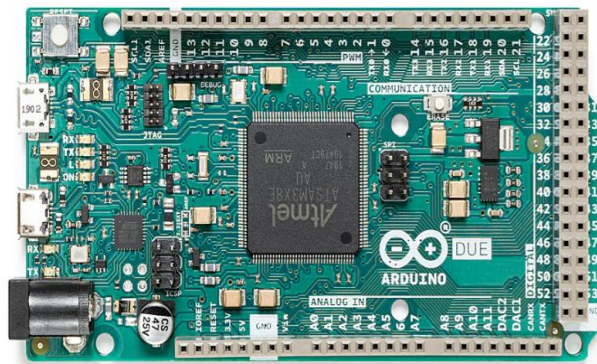
Arduino Uno

Jenis yang ini adalah yang paling banyak digunakan. Terutama untuk pemula sangat disarankan untuk menggunakan Arduino Uno. Banyak sekali referensi yang membahas Arduino Uno. Versi yang terakhir adalah Arduino Uno R3 (Revisi 3), menggunakan ATMEGA328 sebagai pengendali mikronya, memiliki 14 pin I/O digital dan 6 pin input analog. Menggunakan kabel USB type A to B. Sama seperti yang digunakan pada USB printer.



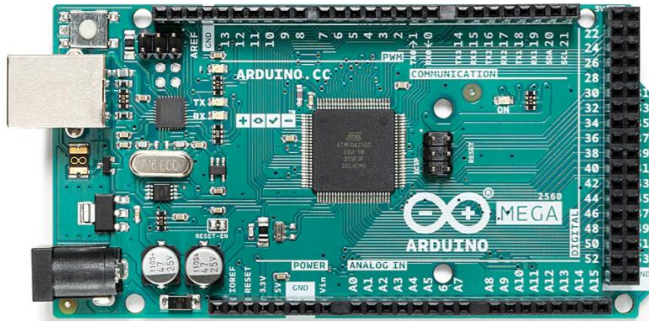
Arduino Due

Berbeda dengan saudaranya, Arduino Due tidak menggunakan ATMEGA, melainkan dengan chip yang lebih tinggi ARM Cortex CPU. Memiliki 54 I/O pin digital dan 12 pin input analog. Untuk pemrogramannya menggunakan Micro USB, terdapat pada beberapa handphone.



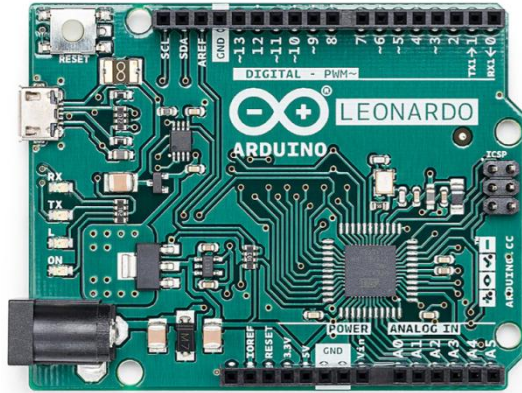
Arduino Mega

Mirip dengan Arduino Uno, sama-sama menggunakan USB type A to B untuk pemrogramannya. Tetapi Arduino Mega, menggunakan Chip yang lebih tinggi ATMEGA2560. Dan tentu saja untuk Pin I/O Digital dan pin input Analognya lebih banyak dari Uno.



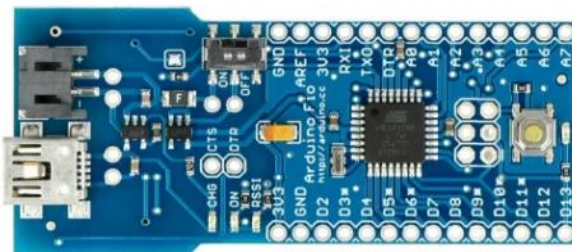
Arduino Leonardo

Saudara kembar dari Uno. Namun banyaknya pin I/O digital, pin input Analog, dan input output PWM berbeda dari Arduino Uno. Hal ini dikarenakan Arduino Leonardo menggunakan pengendali mikro ATMEGA32U4. ATMEGA32U4 ini telah memiliki fitur komunikasi USB yang terintegrasi, sehingga pada Arduino Leonardo tidak memerlukan tambahan pengendali mikro pengatur komunikasi USB ke TTL Serial seperti yang terdapat pada board Arduino Uno yang menggunakan tambahan ATMEGA16U2 disamping ATMEGA328P. Dengan ATMEGA32U4, memungkinkan Arduino Leonardo untuk berkomunikasi sebagai keyboard/mouse dengan protokol USB-HID menggunakan pustaka keyboard.



Arduino Fio

Bentuknya lebih unik, terutama untuk socketnya. Walau jumlah pin I/O digital dan input analognya sama dengan uno dan leonardo, tapi Fio memiliki Socket XBee. XBee membuat Fio dapat dipakai untuk keperluan proyek yang berhubungan dengan wireless.



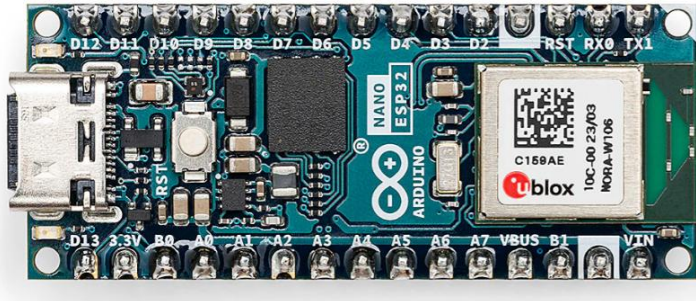
Arduino LilyPad

Bentuknya yang melingkar membuat LilyPad dapat dipakai untuk membuat projek unik. Seperti membuat amor iron man misalkan. Hanya versi lamanya menggunakan ATMEGA168, tapi masih cukup untuk membuat satu projek keren. Dengan 14 pin I/O digital, dan 6 pin input analognya.



Arduino Nano

Sepertinya namanya, Nano yang berukuran kecil dan sangat sederhana ini, menyimpan banyak fasilitas. Sudah dilengkapi dengan FTDI untuk pemograman lewat Micro USB. 14 Pin I/O Digital, dan 8 Pin input Analog (lebih banyak dari Uno). Dan ada yang menggunakan ATMEGA168 atau ATMEGA328.



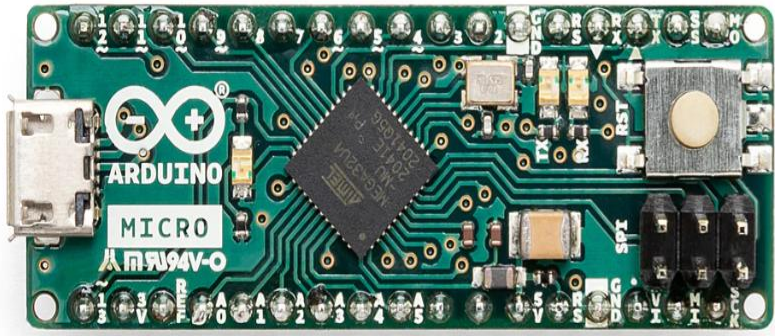
Arduino Mini

Fasilitasnya sama dengan yang dimiliki Nano. Hanya tidak dilengkapi dengan Micro USB untuk pemrograman. Dan ukurannya hanya 30 mm x 18 mm saja.



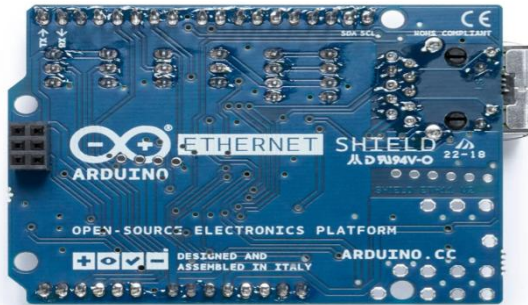
Arduino Micro

Ukurannya lebih panjang dari Nano dan Mini. Karena memang fasilitasnya lebih banyak yaitu; memiliki 20 pin I/O digital dan 12 pin input analog.



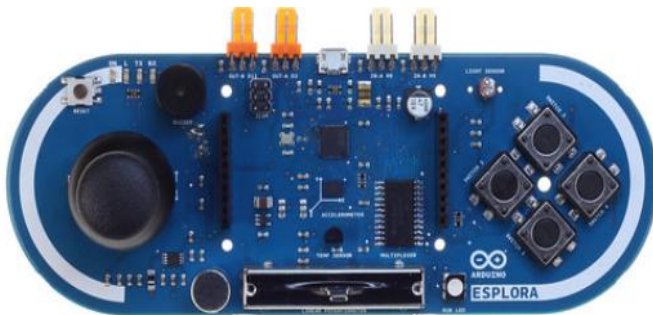
Arduino Ethernet

Ini arduino yang sudah dilengkapi dengan fasilitas ethernet. Membuat Arduino Ayahbunda dapat berhubungan melalui jaringan LAN pada komputer. Untuk fasilitas pada Pin I/O Digital dan Input Analognya sama dengan Uno.



Arduino Esplora

Rekomendasi bagi kamu yang mau membuat gadget seperti Smartphone, karena sudah dilengkapi dengan Joystick, button, dan sebagainya. Ayahbunda hanya perlu tambahkan LCD, untuk lebih mempercantik Esplora.



Arduino Robot

Ini adalah paket lengkap dari Arduino yang sudah berbentuk robot. Sudah dilengkapi dengan LCD, Speaker, Roda, Sensor Infrared, dan semua yang Ayahbunda butuhkan untuk robot sudah ada pada Arduino ini.



Petualangan 2

BERMAIN KOMPONEN!

Setelah anak-anak merasa asyik dengan permainan harta karun yang melibatkan kode dengan sistem bilangan, kini saatnya mereka dibawa ke pemandangan yang lebih asyik. Apa itu? Yaitu, Arduino Kit untuk pemula, yang berisi komponen-komponen yang akan digunakan untuk membuat rangkaian. Perhatikan dengan antusias bahwa komponen-komponen tersebut memiliki daya sulap.

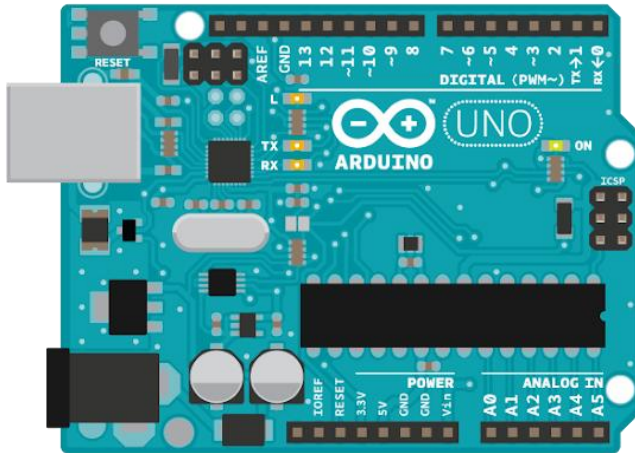
Mendengar kata sulap, anak-anak pasti akan mudah tertarik. Bagaimana pun imajinasi anak-anak lebih menerima kata-kata seperti sulap, magic, bim salabim atau sejenisnya. Ceritakan, bagaimana LED yang berwarna-warni itu akan bisa menyala sesuai keinginan kita? Bagaimana jari kita mampu membuat sebuah LED menyala atau padam? Bagaimana kita mampu memunculkan suhu atau kelembaban di sebuah LCD? Bagaimana kita mampu mengeluarkan angka-angka

hanya dengan menggunakan potensiometer? Bagaimana kita mampu membuat sebuah tulisan berjalan? Bagaimana kita mampu membuat angka menyala dari 0-9 atau sebaliknya dengan seven segmen? Atau berbagai macam kreasi lainnya yang tentu saja akan menumbuhkan rasa penasaran di pikiran anak-anak. Bagaimana caranya dan kapan dimulai sulapnya?

Biarkan mereka melihat-lihat dulu komponen-komponen yang ada. Biarkan mereka bertanya-tanya. Setelah mereka benar-benar tertarik dan ingin tahu, terangkan satu per satu komponen beserta fungsi-fungsinya. Supaya Ayahbunda yakin dengan jawaban yang ditanyakan anak-anak, berikut daftar komponen-komponen yang perlu Ayahbunda sediakan. Ayahbunda tidak perlu khawatir, dalam buku ini akan dipaparkan beberapa komponen yang dibutuhkan beserta fungsinya. Dengan membaca buku ini, diharapkan Ayahbunda tahu terlebih dahulu dibanding anak-anak. Sehingga Ayahbunda akan terlihat sebagai sosok yang *smart*. Pahami kan, maksudnya? Hehe...

Arduino

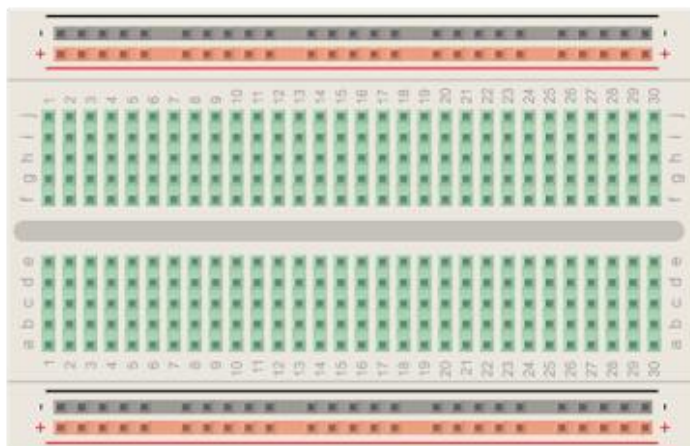
Sebuah sistem minimum yang akan menjadi otak dari semua proyek yang akan Ayahbunda buat.



Breadboard

Sebuah papan percobaan dimana Ayahbunda dapat membangun berbagai rangkaian elektronik tanpa harus menyolder dari satu kaki komponen ke kaki komponen lainnya, untuk menghubungkannya hanya cukup menancapkannya pada lubang dari setiap barisnya. Kontruksi bagian dalam pada sebuah breadboard terlihat seperti pada gambar, dari setiap baris lubang 1 - 30, 5 lubang yang berlabel a - e terhubung semua,

begitu juga 5 lubang f - j terhubung semua. Pada bagian kedua sisi lubang yang terdapat label - (negatif) dan + (positif) terhubung semua dari ujung kiri sampai ujung kanan.



Kabel jumper

Digunakan untuk menghubungkan tiap kaki komponen yang berjauhan pada rangkaian breadboard, dan untuk menghubungkan dari kaki komponen ke papan Arduino. Kabel jumper ada tiga jenis, yaitu : male-male (M-M), male-female (M-F) dan female-female (F-F).

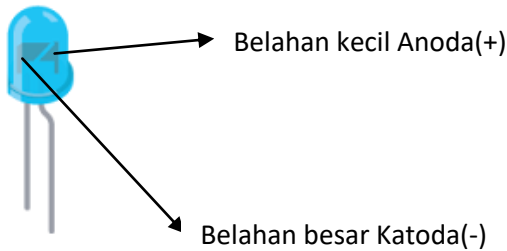


Light Emitting Diode (LED)

Merupakan tipe dioda yang dapat berpendar/bercahaya ketika dilalui arus listrik. Seperti semua tipe dioda, arus listrik hanya mengalir satu arah melalui komponen ini. Ayahbunda sering menjumpai LED sebagai indikator pada peralatan elektronik. Kaki anoda, yang biasanya dihubungkan dengan tegangan positif memiliki kaki yang lebih panjang, sedang kaki katoda ukurannya lebih pendek.



Bagaimana jika karena suatu hal, seperti kakinya terpotong atau sengaja dipotong sehingga panjang kaki jadi sama satu sama lain? Kita bisa memperhatikan pada fisik LED, didalamnya ada bagian yang seperti terbelah dua. Ada belahan kecil dan ada belahan besar. Belahan kecil adalah bagian yang terhubung dengan kaki anoda (+).

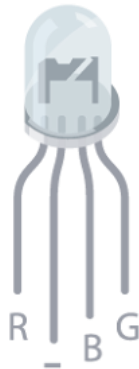


Yang perlu Ayahbunda perhatikan adalah bahwa pemasangan LED tidak boleh terbalik. Karena jika terbalik, LED tidak akan menyala.

LED RGB

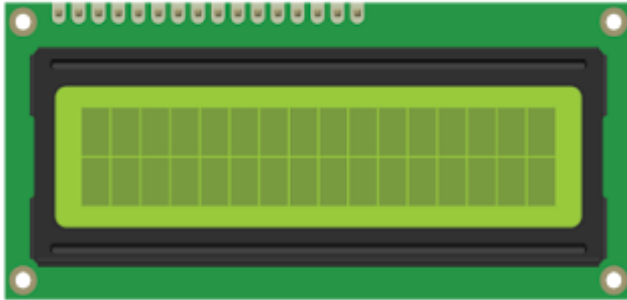
Adalah sebuah LED yang di dalamnya terdapat tiga nyala LED berwarna dasar red-green-blue (merah-hijau-biru), perpaduan dari tiga warna tersebut dengan

tingkat kecerahan yang berbeda dapat menghasilkan semua kombinasi warna.



Liquid Crystal Display (LCD)

Merupakan media penampil alfanumerik yang menggunakan kristal cair sebagai medianya. LCD tipe ini terdapat dalam beberapa ukuran dan bentuk. Yang digunakan pada Kit ini mempunyai 2 baris dengan 16 karakter tiap barisnya. Jika Ayahbunda ingin menampilkan karakter lebih banyak bisa memilih dengan ukuran yang lebih besar.



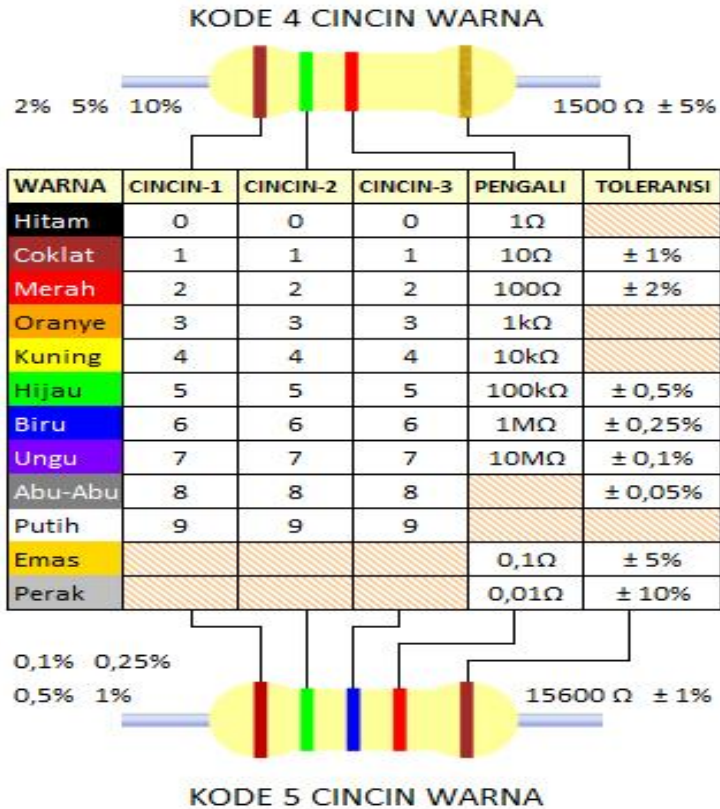
Buzzer

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. Ayahbunda dapat menggunakan buzzer untuk menghasilkan musik yang sederhana.



Resistor

Berfungsi untuk menghambat aliran listrik pada rangkaian, sehingga dapat menghasilkan perubahan arus dan tegangan. Nilainya tergantung pada warna gelang yang ada padanya. Berikut cara membaca nilai resistor :



Photoresistor

Disebut juga sebagai LDR (*Light Dependent Resistor*) sebuah variabel resistor yang nilai hambatannya bergantung dari intensitas cahaya yang diterima pada permukaan sensor tersebut. Jadi, misalkan Ayahbunda ingin membuat LED mati sendiri saat ada cahaya dan menyala sendiri saat tidak ada cahaya, LDR ini sangat merupakan pilihan tepat.



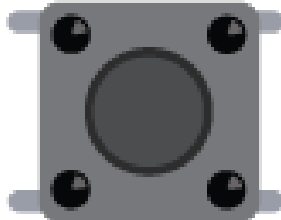
Potensiometer

Adalah sebuah 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

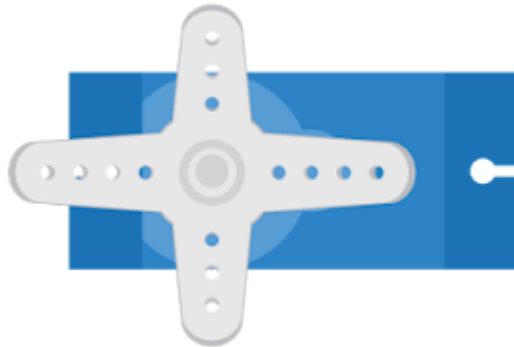
Adalah sebuah saklar yang menghubungkan arus pada suatu rangkaian ketika knobnya ditekan. Saklar sangat cocok dipasang pada pin digital input untuk mendeteksi sinyal on/off.



Servo

Adalah motor yang mempunyai gearbox dan dapat berputar dengan posisi yang presisi sesuai yang diperintahkan, untuk jenis ini hanya bisa berputar 180 derajat. Putarannya dapat dikontrol dengan memberikan pulsa tegangan dari analog output Arduino, pulsa

tegangan tersebut yang memerintahkan servo untuk berputar pada posisi tertentu.



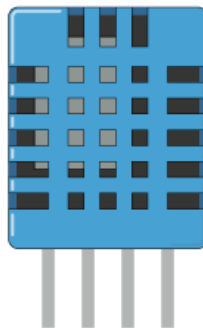
Modul Relay

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.



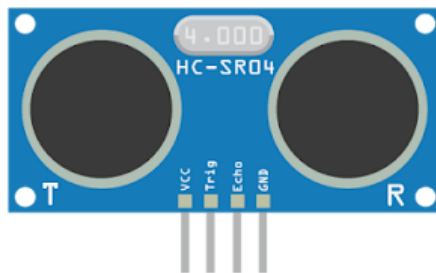
DHT11

Merupakan sensor untuk mengukur suhu dan kelembapan udara sekitar. Output yang keluar dari sensor ini sudah berupa data, sehingga memerlukan penggunaan library pada sketch programnya. Untuk membaca data dari sensor ini hanya memerlukan satu sinyal pin digital dengan pembacaan data tiap 2 detik sekali, serta pin VCC (Power 5V) dan GND (Ground). Selain DHT11 ada juga jenis sensor serupa dengan tipe DHT 22.



Modul Ultrasonik

Dapat berfungsi sebagai sensor jarak untuk mengukur jarak dari 2cm - 400cm dan memiliki tingkat akurasi sekitar 3mm. Pada modul ini terdapat sensor ultrasonik penerima, pemancar dan rangkaian kontrol. Hanya terdapat empat pin untuk menghubungkannya dengan Arduino: VCC (Power 5V), Trig (Trigger), Echo (Receive), dan GND (Ground).



Kabel USB

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



Jenis kabel USB yang digunakan tergantung port yang tersedia di Board Arduino. Jika portnya berbentuk kabel data mini, maka jenis kabel USB adalah seperti yang ada pada gambar di atas. Namun, jika port yang sedia berbentuk seperti colokan pada printer, maka yang diperlukan adalah kabel USB Arduino Printer. Namun, Ayahbunda tidak perlu risau, karena saat pembelian board Arduino, sudah dilengkapi kabel USB.



Bagaimana Ayahbunda? Semakin asyik, kan? Tentu setelah mengenal komponen-komponen yang akan digunakan, maka Ayahbunda akan semakin tertantang untuk melangkah ke langkah selanjutnya. Begitu pun dengan putera-puteri Ayahbunda. Semangat Ayahbunda pasti akan menular kepada mereka semua. Mereka pasti juga akan semakin penasaran dan bergairah.

Petualangan 3

YUK, SIAPKAN CODINGMU!

Sebelumnya Ayahbunda sudah berkenalan dengan Si Biner yang menarik, para komponen yang asyik dan Arduino yang menggelitik. Pertanyaan berikutnya adalah bagaimana mereka bekerja sama mewujudkan sebuah project? Atau bagaimana Ayahbunda bisa membuat kreasi-kreasi yang menakjubkan menggunakan mereka semua? Tentu saja sebelum bisa melakukan semua itu, Ayahbunda harus menyiapkan sebuah komputer atau laptop dan komponen-komponen yang sudah Ayahbunda kenal sebelumnya. Kenapa Ayahbunda memerlukan komputer atau laptop? Komputer inilah yang akan Ayahbunda gunakan untuk membuat coding. Apa itu coding? Mari kita melangkah lebih lanjut ke permainan yang lebih seru.

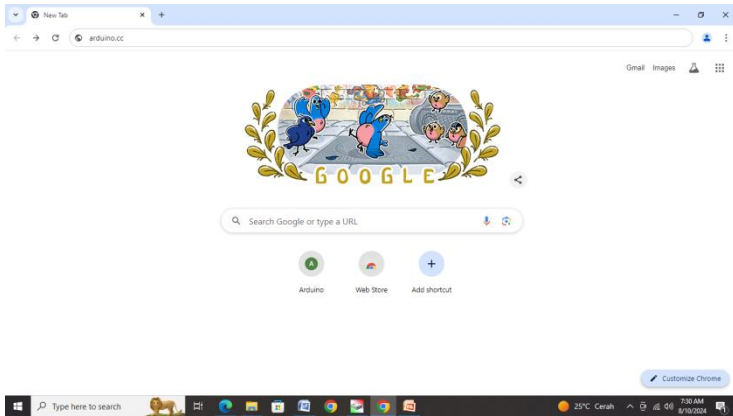
Coding adalah salah satu tindakan dari langkah-langkah pemrograman dengan menuliskan kode atau skrip dalam bahasa pemrograman. Supaya skrip tersebut dapat dipahami oleh komputer, maka saat

proses *coding* Ayahbunda harus mengikuti aturan sintaks yang berlaku. Aturan sintaks sangat tergantung dari bahasa pemrograman apa yang Ayahbunda gunakan saat menuliskan skrip. Dengan kata lain *coding* merupakan kegiatan dimana Ayahbunda memberitahukan komputer apa yang harus dia kerjaka. Sebuah kode yang ada pada skrip bisa diibaratkan layaknya bahasa sehari-hari. Setiap kode yang Ayahbunda tulis akan membantu komputer untuk mengetahui dan memahami apa yang ingin Ayahbunda lakukan pada komputer. Komputer akan menerima perintah ini dan komputer akan melakukan operasi berdasarkan perintah yang Ayahbunda tuliskan.

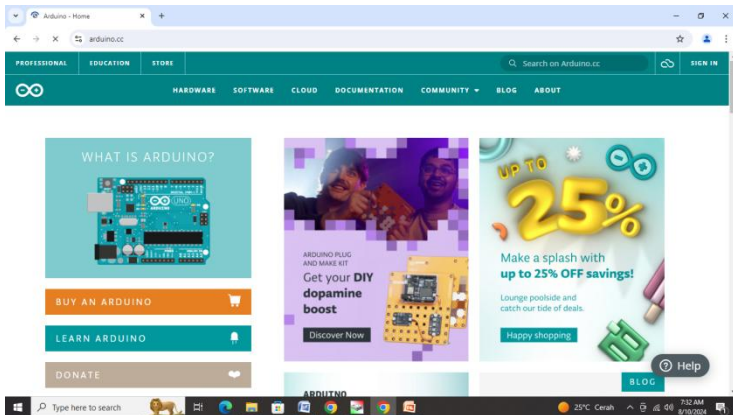
Dalam membuat kreasi-kreasi, nantinya Ayahbunda akan menggunakan perangkat Arduino. Karena itu langkah pertama yang harus Ayahbunda lakukan adalah menginstal software terkait Arduino, yaitu Arduino IDE (*Integrated Development Enviromnet*). Software inilah yang akan Ayahbunda gunakan dalam menuliskan skrip/kode-kode pemrograman. Pemrograman Ardiuno menggunakan bahasa pemrograman C. Mari kita mulai dari langkah paling awal!

Mendownload Arduino IDE

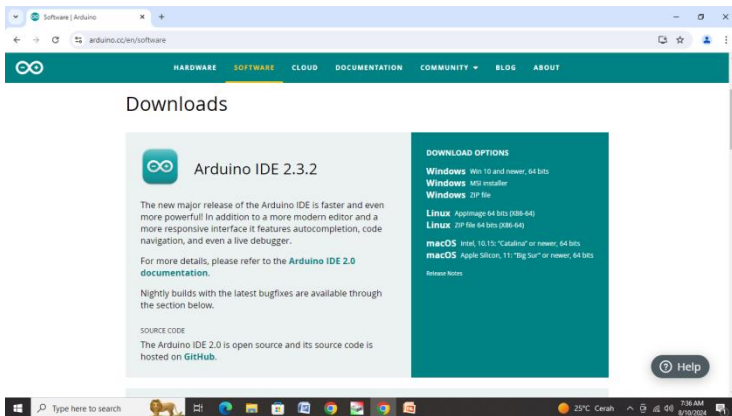
Buka website Arduino dengan mengetikkan *arduino.cc*



Tekan Enter dan akan muncul website arduino.

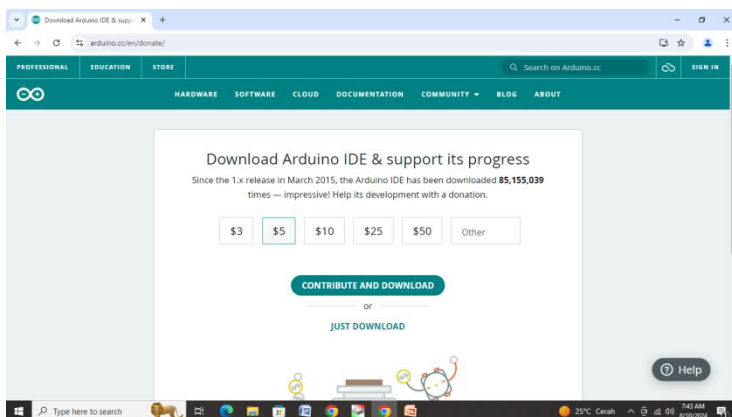


Klik software maka dilayar komputer Ayahbunda akan muncul tipe Arduino IDE tipe terbaru.

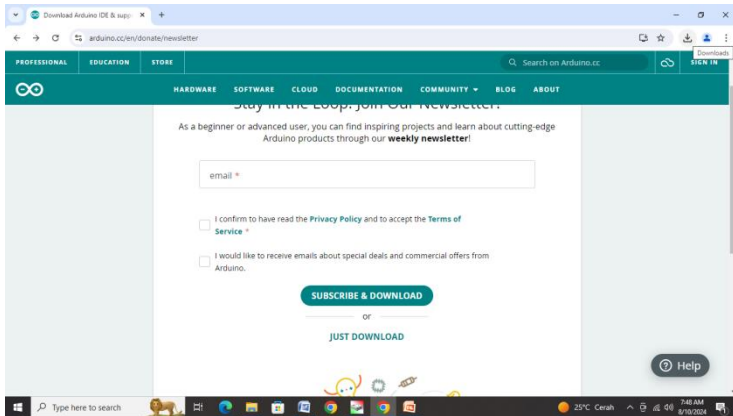


Download software tersebut, yang perlu Ayahbunda ingat adalah Windows yang ada di komputer Ayahbunda minimal Windows 10. Di bawah Windows 10 kurang kompatibel untuk software ini. Artinya, jika menggunakan komputer dibawah Windows 10, dimungkinkan akan terjadi kendala saat Ayahbunda membuat coding. Tentu saja hal ini akan membuat Ayahbunda tidak bisa berkreasi secara leluasa. Jadi, pastikan komputer yang Ayahbunda gunakan sudah memakai Windows 10.

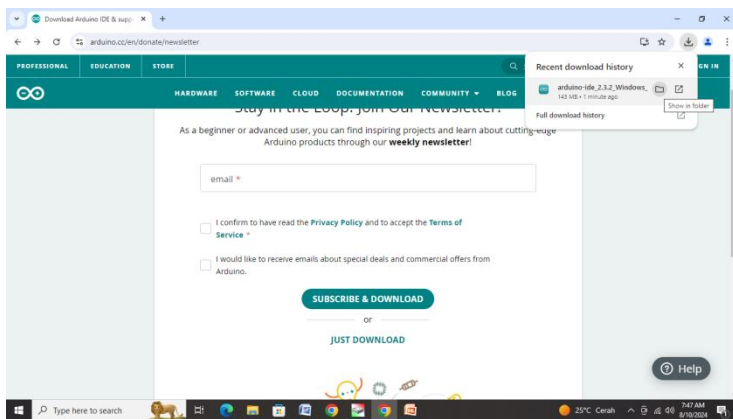
Saat mendownload, Ayahbunda akan diberikan pilihan Contribute & Download atau Just Download.



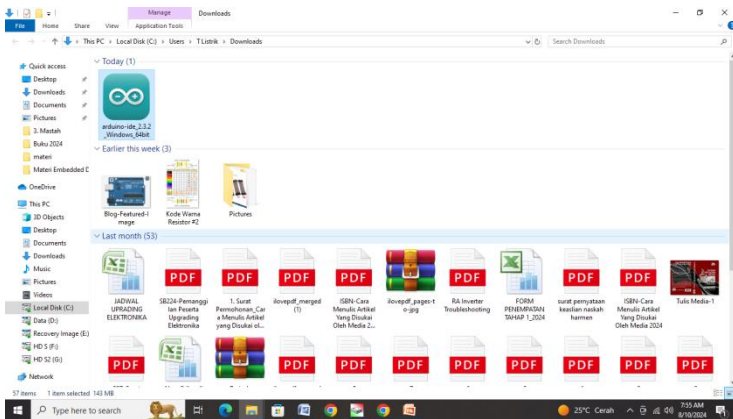
Maka Ayahbunda, pilih saja Just Download. Selanjutnya, Ayahbunda sekali lagi akan diberi pilihan Subscribes & Download atau Just Download. Pilih saja Just Download. Setelah itu tunggu sampai proses download selesai. Untuk mengetahui dimana file Arduino IDE hasil download berada, Ayahbunda bisa melihatnya di bagian pojok kanan atas browser. Jika Ayahbunda menggunakan Google Chrome maka berkas file yang didownload akan terlihat seperti pada gambar berikut :



Klik pada tanda download, akan tampil nama file yang terdownload, yaitu arduino-ide_2.3.2_Windows_64bit.

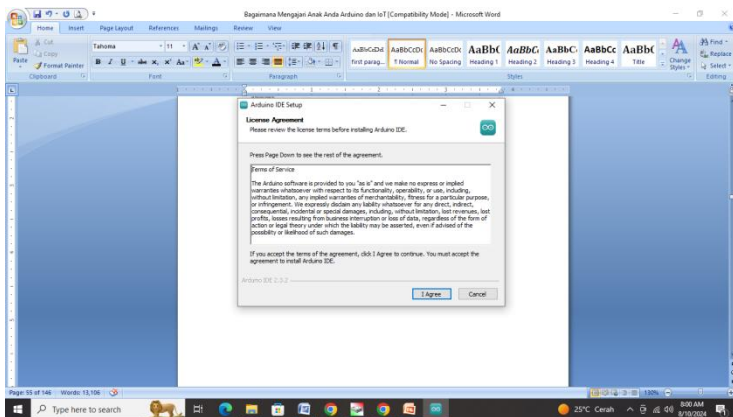


Klik Show of Folder, Ayahbunda akan dibawa ke tempat file tersebut tersimpan.

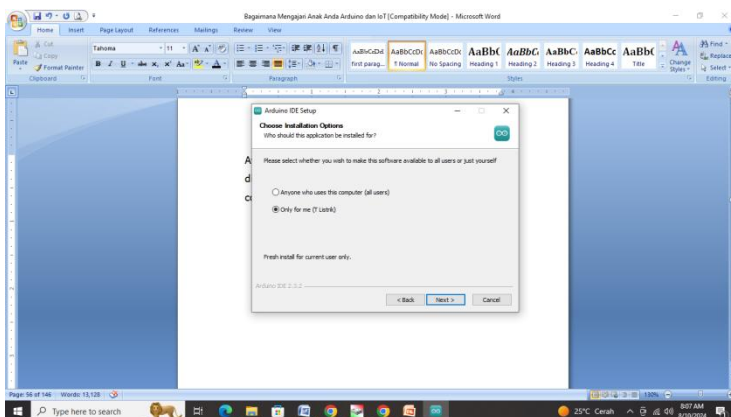


Menginstal Arduino IDE

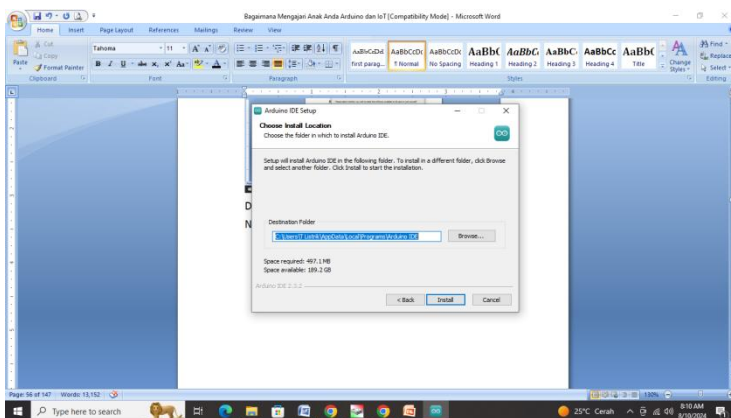
Klik dua kali file arduino-ide_2.3.2_Windows_64bit. Ayahbunda akan diminta persetujuan untuk melakukan terlebih dahulu sebelum melakukan proses instal lebih lanjut.



Ayahbunda klik saja, I Agree. Ayahbunda akan dihadapkan lagi pada pilihan, Anyone who use this computer (all users) atau Only for me.



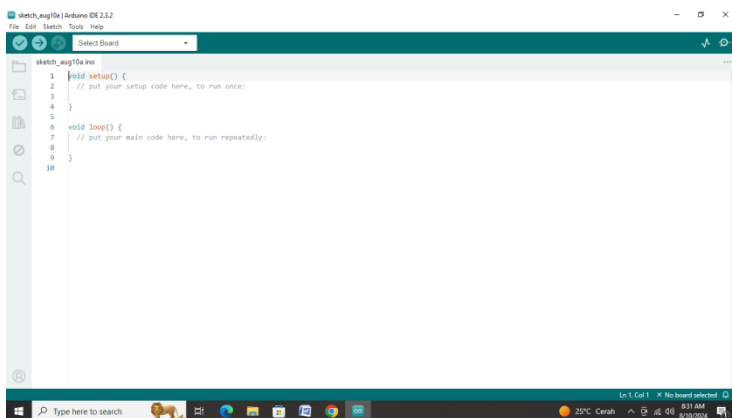
Dalam hal ini saya memilih Anyone who use this computer(all users), kemudian klik Next. Kemudian akan muncul pilihan Install.



Klik Install dan tunggu sampai proses penginstalan selesai dan muncul layar komputer Ayahbunda pilihan untuk mengakhiri proses. Klik Finish. Untuk memastikan Arduino IDE sudah terinstal atau belum bisa di lihat di list program di komputer Ayahbunda.

Membuka Arduino IDE

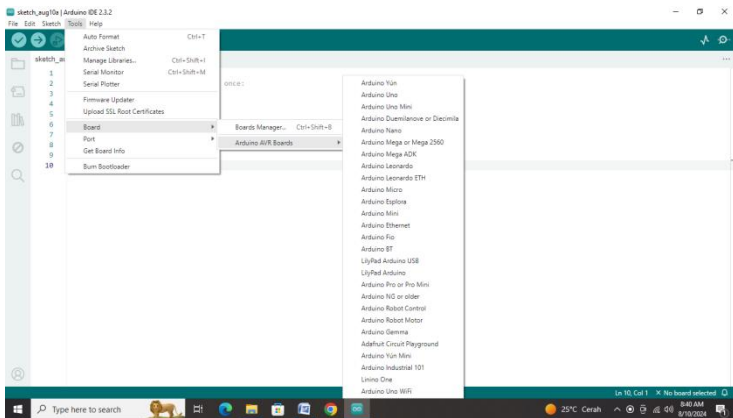
Klik Windows Start pada pojok kiri bawah. Klik Arduino IDE. Maka akan muncul lembar kerja Arduino IDE seperti di bawah ini.



Jika Ayahbunda sudah bisa masuk ke halaman tersebut maka Arduino IDE benar-benar sudah terinstal. Namun, untuk menggunakannya, Ayahbunda harus melewati

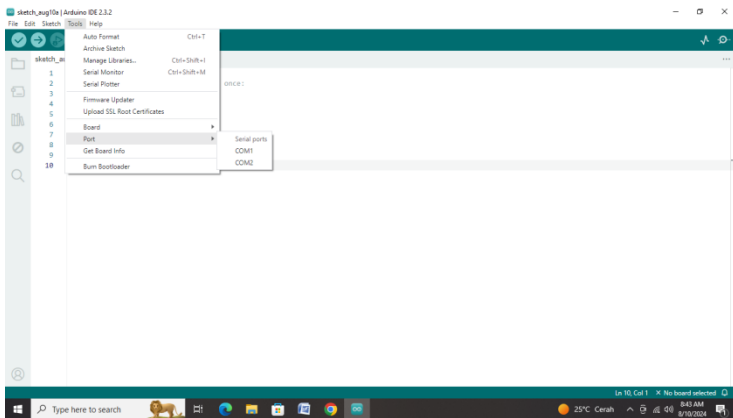
satu langkah lagi, yaitu memastikan board dan port yang bisa digunakan. Board disesuaikan dengan jenis Arduino yang digunakan, sedangkan port disesuaikan dengan port USB yang terdata di komputer. Untuk menentukan Board, ikuti langkah berikut :

Klik Tool → Board → Arduino AVR Board → Pilih Jenis Arduino yang dipakai (dalam hal ini Ayahbunda akan memakai Arduino Uno, maka pilih Arduino Uno).



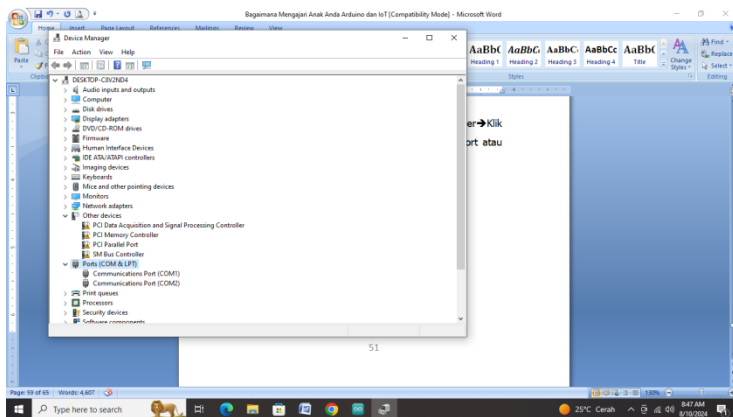
Sedangkan untuk menentukan Portnya, langkahnya adalah :

Klik Tool → Port → Pilih Port yang terdaftar (dalam hal ini di komputer saya ada di COM1 atau COM 2)



Bagaimana untuk melihat keberadaan port di komputer Ayahbunda? Caranya sangat mudah.

Klik kanan tombol Start → Klik kiri Device Manager → Klik Kiri Port (COM & PLT) → maka akan terlihat Port atau COM di komputer Ayahbunda.



Memulai Membuat Coding

Sampai di sini perjalanan Ayahbunda akan semakin seru dan asyik. Semua ide-ide Ayahbunda segera bisa diwujudkan. Ayahbunda akan segera menjelajah alam imajinasi yang sangat luas dan tidak terbatas. Imajinasi Ayahbunda tidak hanya tersimpan dalam otak saja. Kini saatnya, Ayahbunda menuliskan imajinasi-imajinasi tersebut ke dalam bentuk coding atau pemrograman. Di mana, coding ini nanti yang akan memerintah Arduino untuk mengeksekusi kreasi Ayahbunda. Bagaimana mengawalinya? Mari, kita segera berkenalan dengan bahasa pemrograman lebih lanjut.

Bahasa pemrograman Arduino terbagi menjadi tiga bagian utama, yaitu :

- Function/fungsi, untuk mengendalikan papan Arduino dan melakukan perhitungan.
- Value/nilai, ada dua jenis value yaitu variable/variabel (nilainya bisa berubah-ubah) dan constants (nilai tetap atau konstan).
- Structure/struktur, biasanya terdiri dari header, deklarasi variabel global, fungsi set up dan loop.

Berikut adalah contoh struktur sebuah pemrograman dengan Arduino IDE. Ayahbunda, bisa mencoba membuka Arduino IDE yang sudah terinstal di komputer. Ikuti langkah berikut :

- Klik Tombol Start di pojok kiri bawah.
- Klik Arduino IDE
- Klik File→Example→01.Basics→Blink

Akan muncul sebuah contoh struktur pemrograman Blink, sebagai berikut :

```
/*
  Blink

  Turns an LED on for one second, then off for one second,
  repeatedly.

  Most Arduinos have an on-board LED you can control. On
  the UNO, MEGA and ZERO
  it is attached to digital pin 13, on MKR1000 on pin 6.
  LED_BUILTIN is set to
  the correct LED pin independent of which board is used.
  If you want to know what pin the on-board LED is
  connected to on your Arduino
  model, check the Technical Specs of your board at:
  https://www.arduino.cc/en/Main/Products

  modified 8 May 2014
  by Scott Fitzgerald
  modified 2 Sep 2016
  by Arturo Guadalupi
  modified 8 Sep 2016
  by Colby Newman
```

This example code is in the public domain.

```

https://www.arduino.cc/en/Tutorial/BuiltInExamples/Blink
*/

// the setup function runs once when you press reset or
power the board

void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

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

```

Kita akan membahas satu per satu struktur tersebut mulai dari bagian atas hingga ke bagian bawah.

`/*...*/` adalah block comment, komentar dalam beberapa baris. Komentar ini tidak akan terbaca oleh Arduino sebagai coding. Ini hanya semacam catatan dari kita untuk bisa dibaca oleh orang lain. Dalam contoh di atas Arduino memebrikan catatan bahwa program Blink dibuat untuk menghidupkan LED selama

satu detik dan mematikan LED selama satu detik secara berulang terus- menerus.

// adalah single comment, sebaris komentar. Sama seperti block comment, ini juga tidak akan terbaca sebagai program. Dalam contoh di atas, single lcomment pertama, Tim Arduino ingin memberitahu bahwa fungsi set up digunakan untuk menjalankan satu kali saja program yang dibuat.

Fungsi void setup() digunakan untuk mengkonfigurasi pin Arduino digunakan sebagai INPUT atau OUTPUT. Setiap perintah yang ada dalam fungsi void setup hanya akan dijalankan sekali oleh Arduino. Penginisasian harus berada dalam kurung kurawal {}. Dalam contoh di awal tertulis :

```
void setup() {  
  // initialize digital pin LED_BUILTIN as an output.  
  pinMode(LED_BUILTIN, OUTPUT);  
}
```

Dalam hal itu, LED-BUILTIN akan dijadikan sebagai OUPUT. LED-BUILTIN adalah LED yang sudah menempel pada Arduino Uno. Jika kita ingin menambahkan LED tersendiri pada breadboard maka

peng-codingan di atas bisa ditambahkan satu lagi pinMode sebagai berikut :

```
void setup() {  
  pinMode(LED_BUILTIN, OUTPUT);  
  pinMode(13, OUPUT);  
}
```

Dengan tambahan tersebut maka bisa diartikan bahwa, nantinya akan dua output, yaitu LED Builtin dan LED baru yang salah satu kakinya akan terhubung dengan pin 13 Arduino Uno. Tetapi, sebelumnya (di atas voidsetup) harus dideklarasikan dulu pin 13 sebagai terhubung dengan output tambahan, misalnya LED Merah. Programnya jadi seperti ini :

```
int lampu_merah = 13;  
void setup() {  
  pinMode(LED_BUILTIN, OUTPUT);  
  pinMode(13, OUPUT);  
}
```

Jadi, setiap ada penambahan input atau output harus menambah "deklarasi" dan menambah "pinMode". Karena kalau tidak ditambahkan pada deklarasi dan pinMode maka Arduino tidak akan bisa membaca penambahan tersebut. Pada akhirnya, juga akan menambah perintah pada bagian "void loop".

Fungsi void loop() digunakan untuk pemrograman yang sifatnya berulang.

```
void loop() {  
  digitalWrite(LED_BUILTIN, HIGH);  
  delay(1000);  
  digitalWrite(LED_BUILTIN, LOW);  
  delay(1000);  
}
```

Karena didepan ada penambahan Led Merah sebagai output maka di void loop juga harus ditambahkan pada "digitalWrite", menjadi seperti :

```
void loop() {  
  digitalWrite(LED_BUILTIN, HIGH);  
  delay(1000);  
  digitalWrite(LED_BUILTIN, LOW);  
  delay(1000);  
  digitalWrite(13, HIGH);  
  delay(1000);  
  digitalWrite(13, HIGH);  
  delay(1000);  
}
```

Hasil pemrograman secara utuh akan menjadi seperti struktur di bawah ini :

```
int lampu_merah = 13;  
  
void setup() {  
  pinMode(LED_BUILTIN, OUTPUT);  
  pinMode(13, OUPUT);  
}  
  
void loop() {  
  digitalWrite(LED_BUILTIN, HIGH);  
  delay(1000);  
  digitalWrite(LED_BUILTIN, LOW);  
  delay(1000);  
}
```

```
digitalWrite(13, HIGH);  
delay(1000);  
digitalWrite(13, HIGH);  
delay(1000);
```

Jika diartikan dalam bahasa manusia, maka coding di atas bisa diartikan sebagai :

Perintah kepada Arduino Uno untuk menghidupkan dan mematikan LED Builtin (yang sudah menempel di Board Arduino Uno) dan Lampu LED Warna Merah yang akan dipasang di Breadboard. Dimana salah satu kaki LED Merah tersambung ke Board Arduino Uno. Dua LED tersebut diprogram untuk hidup selama satu detik dan mati selama satu detik secara terus menerus.

Sebagai sebuah perintah, maka coding ini harus dipastikan benar. Baik secara logika atau secara syntax (atau tata cara penulisan sebuah perintah program). Untuk mengetahui, coding kita sudah benar atau tidak bisa dicek melalui tombol "Verify" dengan simbol tanda centang (✓). Jika ada kesalahan maka akan ada catatan error pada bagian konsol di Arduiono IDE. Jika sudah benar, maka hasil coding kita sudah bisa diupload ke Arduino dengan menekan tombol Upload dengan simbol tanda panah (➔).

Penulisan syntax harus benar, karena seringkali kesalahan/eror terjadi hanya karena masalah penulisan (syntax). Lalu, bagaimana menuliskan syntax-nya?

Deklarasi

Langkah pertama dalam coding adalah mendeklarasikan pin-pin yang akan digunakan. Ada beberapa cara menuliskan syntax deklarasi, antara lain :

Type data variable = value;

Contoh : `int lampu_merah = 13;`

Artinya, pin 13 Arduino Uno dideklarasikan untuk Lampu Merah. Pemberian nama variable bebas, yang penting mudah untuk mengingatnya.

Macam-macam type data :

Type Data	Ukuran Memori byte	Range
boolean	1	True or false
char	1	-128 to +127, tipe data digunakan untuk penyimpanan nilai character merujuk pada tabel ASCII
unsigned Char	1	0 to 255
byte	1	0 to 255
int	2	-32768 to 32767
unsigned int	2	0 to 65535
word	2	0 to 65535
long	4	-2.147.483.648 to 2.147.483.647
unsigned long	4	0 to 4.294.967.295
float	4	-3.4028235E+38 to 3.4028235E+38
double	4	-3.4028235E+38 to 3.4028235E+38

#define variable value

Contoh : `#define lampu_merah 13.` Artinya sama dengan syntax di sebelumnya.

#include<LibraryFile.h>

#include"LocalFile.h"

Ini digunakan untuk menyertakan library luar.

Penentuan Input/Output

Langkah kedua dalam coding adalah menentukan pin-pin arduino sebagai input atau output.

`pinMode(pin,mode);`

Contoh : `pinMode(13,OUTPUT);`

Artinya, pin 13 Arduino Uno ditentukan sebagai output. Ada tiga mode yang ada, INPUT, OUTPUT dan INPUT_PULLUP. Jika mode INPUT yang digunakan maka perangkat yang akan dipasang harus dilengkapi resistor. Jika menggunakan mode INPUT_PULLUP, berarti rangkaian kita menggunakan internal resistor, jadi tidak memerlukan resistor dalam rangkaian.

Pembacaan dan Penulisan Logika Proses

Langkah ketiga dalam coding adalah menentukan nilai-nilai pin Arduino sebagai ON (High/1) atau OFF (Low/0).

`digitalWrite(pin,value);`

Contoh : `digitalWrite(13,HIGH);` atau `digitalWrite(13,LOW)`

Artinya, perintah untuk output dari pin menghidupkan atau mematikan. Jika output ingin hidup value diisi HIGH atau 1. Jika output ingin mati maka value diisi LOW atau 0.

`digitalRead(pin);`

Contoh : `digitalRead(13);` atau `digitalRead(10)`

Artinya, perintah untuk membaca kondisi pin Arduino HIGH atau LOW).

`analogWrite(pin,value);`

Contoh : `analogWrite(A0,HIGH);` atau `analogWrite(A1,LOW)`

Untuk menulis nilai analog (gelombang PWM pulse width modulation) ke pin. Dapat digunakan untuk menyalakan LED pada berbagai kecerahan atau menggerakkan motor pada berbagai kecepatan. Dengan siklus kerja 0 ~ 255.

analogRead(pin);

Contoh : `analogRead(A0);` atau `analogRead(A1)`

Membaca nilai dari pin analog, ADC (Analog to Digital Converter); contoh 10-bit ADC, berarti merubah tegangan 0 ~ (5V atau 3.3V) menjadi integer 0 ~1023. Ini menghasilkan resolusi $5V/1024=0,0049V$ (4,9mV).

BOARD	OPERATING VOLTAGE	USABLE PINS	MAX RESOLUTION
Uno	5 Volts	A0 to A5	10 bits
Mini, Nano	5 Volts	A0 to A7	10 bits
Mega, Mega2560, MegaADK	5 Volts	A0 to A14	10 bits
Micro	5 Volts	A0 to A11*	10 bits
Leonardo	5 Volts	A0 to A11*	10 bits
Zero	3.3 Volts	A0 to A5	12 bits**
Due	3.3 Volts	A0 to A11	12 bits**
MKR Family boards	3.3 Volts	A0 to A6	12 bits**

*A0 through A5 are labelled on the board, A6 through A11 are respectively available on pins 4, 6, 8, 9, 10, and 12

**The default `analogRead()` resolution for these boards is 10 bits, for compatibility. You need to use `analogReadResolution()` to change it to 12 bits.

BOARD	PWM PINS	PWM FREQUENCY
Uno, Nano, Mini	3, 5, 6, 9, 10, 11	490 Hz (pins 5 and 6: 980 Hz)
Mega	2 - 13, 44 - 46	490 Hz (pins 4 and 13: 980 Hz)
Leonardo, Micro, Yún	3, 5, 6, 9, 10, 11, 13	490 Hz (pins 3 and 11: 980 Hz)
Uno WiFi Rev2, Nano Every	3, 5, 6, 9, 10	976 Hz
MKR boards *	0 - 8, 10, A3, A4	732 Hz
MKR1000 WiFi *	0 - 8, 10, 11, A3, A4	732 Hz
Zero *	3 - 13, A0, A1	732 Hz
Nano 33 IoT *	2, 3, 5, 6, 9 - 12, A2, A3, A5	732 Hz
Nano 33 BLE/BLE Sense	1 - 13, A0 - A7	500 Hz
Due **	2-13	1000 Hz
101	3, 5, 6, 9	pins 3 and 9: 490 Hz, pins 5 and 6: 980 Hz

delay(ms);

Contoh : `delay(1000);` artinya delay selama 1000 milli detik atau 1 detik = 1 detik. Untuk menjeda program selama 1000 milli detik atau 1 detik.

**TypeData Variable [jumlah data] =
{data1,data2, data3,...data-n};**

Contoh : `int led [5]={13,12,11,10,9};` Artinya, program diperintah untuk mengelompokkan sebanyak lima data (pin 13,12,11,10 dan 9) dengan tipe yang sama yaitu integer. Ini disebut dengan Array dengan tujuan untuk mengelompokkan banyak data dengan tipe yang sama dengan satu perintah.

**for(typedata
variable=nilai;variable=batas nilai;i++);**

Contoh : `for(int i=0;i<5;i++);` Artinya, program diperintah untuk memanggil data ke 0 sampai dengan data ke 4, dan naik satu-satu secara berurutan. `i++` dinamakan increment.

**if(nilai logika) {pernyataan benar();
} else { pernyataan salah();}**

Contoh:

```
if(i>10){digitalWrite(led_merah,1);  
}  
else{digitalWrite(Led_merah,0);  
}
```

Artinya, jika nilai variable `i` lebih besar dari 10 maka LED Merah akan menyala. Sebaliknya, jika nilai variable `i` lebih kecil dari 10 maka LED Merah akan padam.

Melihat syntax ini, mungkin Ayahbunda jadi semakin penasaran. Itu hanya contoh syntax yang akan sering kita gunakan. Untuk melihat syntax yang lebih lengkap bisa dilihat di www.arduino.cc/references. Mari kita mulai berkreasi dengan coding dan rangkaian-nya!

Petualangan 4

RANGKAI KREASIMU!

Dari sini Ayahbunda akan lebih merasakan keseruan bermain coding. Karena kalau dari petualangan 1-4 adalah sekadar dibahas hal-hal yang mendasari coding, maka mulai petualangan 5 ini, Ayahbunda akan bisa merasakan bagaimana nikmatnya membuat coding dan merangkai komponennya. Saya yakin, putera-puterinya juga sudah tidak sabar lagi untuk segera memulai permainan asyik bersama Arduino.

Langkah pertama, adalah Ayahbunda beserta putera-puterinya harus membuat coding atau program. Bagaimana caranya? Pada bab sebelumnya Ayahbunda sudah mengenal software Arduion IDE dan syntax-syntax yang diperlukan untuk memulai coding. Bagaimana jika Ayahbunda masih bingung? Ayahbunda tidak perlu cemas, untuk tahap awal akan kita berikan beberapa hasil coding. Ayahbunda tinggal menyalinnya ke dalam sketch/lembar kerja Arduino IDE yang ada di

komputer. Langkah menyalin ini tentu saja hanya untuk melatih logika Ayahbunda. Nanti, semakin banyak berlatih maka logika Ayahbunda akan semakin terlatih dan mampu untuk membuat coding sendiri. Bagaimana jika Ayahbunda ingin meningkatkan kreasi tapi programnya tidak ada di buku ini? Atau Ayahbunda sudah mencoba membuat program tetapi belum bisa jalan? Sekali lagi, Ayahbunda tidak perlu panik. Ada banyak cara untuk mendapatkan hasil coding untuk berlatih. Ayahbunda bisa mencarinya di internet. Karena banyak teman kita yang mempublikasikan hasil codingnya di internet secara gratis.

Lets, play it!

Berpetualang dengan Satu LED Kedap-Kedip

(LED hidup 1 detik dan mati 1 detik terus menerus)

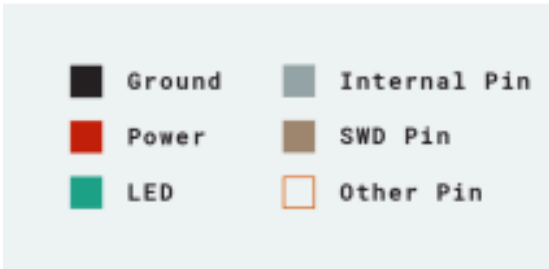
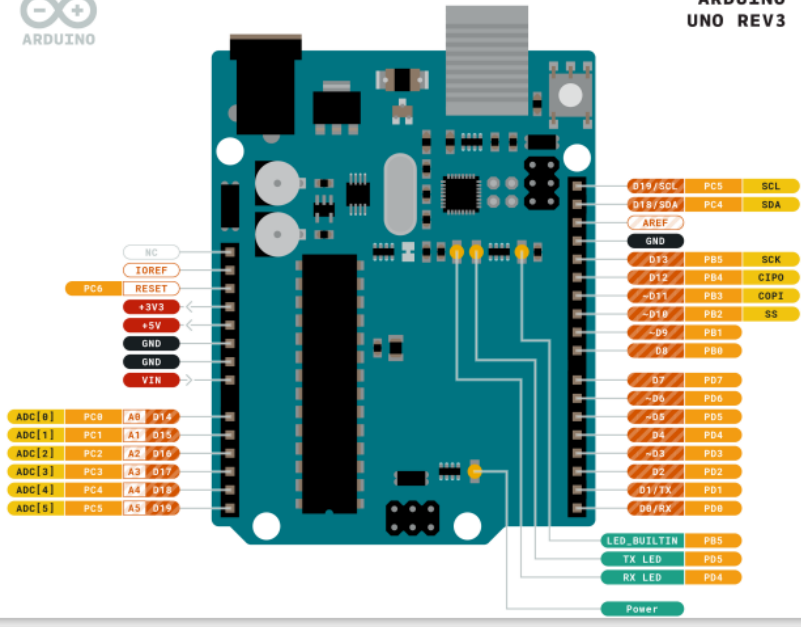
Pertama, Ayahbunda perlu menyiapkan laptop/PC dan satu set Arduino Kit. Dalam praktik kali ini kita akan menggunakan Arduino Uno R3. Bagaimana bentuk fisik dari Ardduino Uno ini? Mari kita berkenalan dengannya.



MICROCONTROLLER	ATmega328P
OPERATING VOLTAGE	5V
INPUT VOLTAGE (RECOMMENDED)	7-12V
INPUT VOLTAGE (LIMIT)	6-20V
DIGITAL I/O PINS	14 (of which 6 provide PWM output)
PWM DIGITAL I/O PINS	6
ANALOG INPUT PINS	6
DC CURRENT PER I/O PIN	20 mA
DC CURRENT FOR 3.3V PIN	50 mA
FLASH MEMORY	32 KB (ATmega328P) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
CLOCK SPEED	16 MHz
LED_BUILTIN	13
LENGTH	68.6 mm
WIDTH	53.4 mm
WEIGHT	25 g



ARDUINO UNO REV3



Kedua, hidupkan laptop. Buka Arduino IDE. Nah, saatnya Ayahbunda mulai membuat coding. Untuk kali ini, mari kita lakukan bersama-sama dengan memakai buku ini sebagai panduan.

```
int led=13;

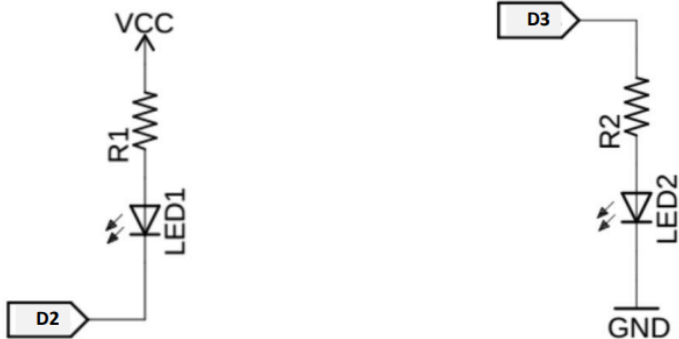
void setup() {
  pinMode(led,OUTPUT);
}

void loop() {
  digitalWrite(led,1);
  delay(1000);
  digitalWrite(led,0);
  delay(1000);
}
```

Untuk memastikan codingan kita sudah benar atau belum lakukan Verify. Jika sudah benar maka upload.

Ketiga, merangkai komponen. Siapkan breadboard. Ambil satu LED, misal warna merah atau warna kesukaan Ayahbunda. Jangan lupa, Ayahbunda juga harus menyiapkan resistor, pilih yang bernilai 220 Ohm atau 330 Ohm. Kabel jumper yang akan Ayahbunda gunakan adalah kabel jumper jenis male-male. Bagaimana cara menghubungkannya? Ingat LED tidak

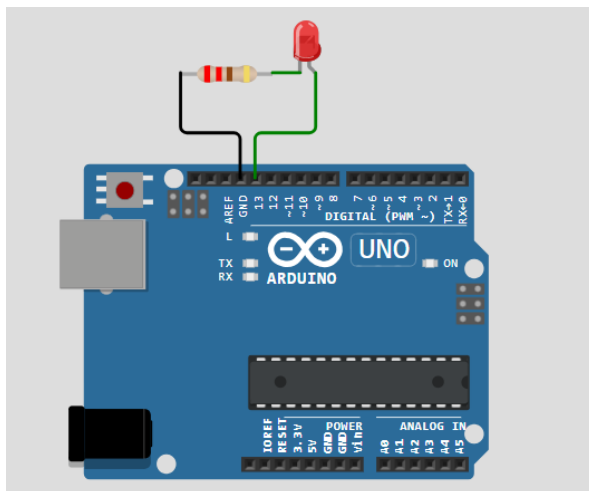
boleh terpasang secara terbalik. Jadi, harus ingat kembali kaki Anoda (+) dan Katoda (-) dari LED. Kaki anoda lebih panjang dari pada kaki katoda. Atau, secara fisik lihat LED-nya. Kaki yang mendapatkan belahan kecil adalah kaki Anoda. Setelah siap semua, Ayahbunda bisa mencoba rangkaiannya. Untuk tahap awal, ikuti saja gambar rangkaian berikut :



Gambar kiri menunjukkan jika Anoda mendapatkan Pin Arduino, maka kaki lainnya mendapatkan VCC atau 3.3 Volt. Tetapi jika coding D2 diberi nilai 1, LED justru malah akan mati. Namun jika, pada D2 diberi nilai 0, LED akan hidup. Rangkaian ini disebut dengan AKTIF_LOW, karena LED akan aktif atau menyala jika coding D2 di beri nilai 0 atau LOW.

Sedangkan gambar sebelah kanan, sebaliknya. Jika Anoda mendapatkan Ground (GND) maka kaki lainnya mendapatkan pin Arduino. Jika pada coding D3 diberi nilai 1, LED akan hidup. Namun jika D3 diberi nilai 0, LED akan mati. Rangkaian ini disebut dengan AKTIF_HIGH, karena LED akan aktif/menyalakan saat diberi nilai 1 atau HIGH.

Pakai saja, mana yang Ayahbunda suka. Dalam contoh berikut, kita memakai rangkaian AKTIF_HIGH. Jadi, rangkaian sesungguhnya adalah seperti berikut ini :



Apa hasilnya? LED Merah akan berkedap-kedip terus menerus. Sampai kapan? Sampai, power Arduino dimatikan. Atau kabel USB dicopot dari laptop atau PC Ayahbunda.

Sekarang dengan cara yang sama, coba Ayabunda membuat coding supaya dua LED bisa kedap-kedip secara bersamaan. Pasti Ayahbunda langsung bisa mengeksekusi latihan ini. Oke! Kita tunggu, ya? Yang perlu diperhatikan oleh Ayahbunda adalah Jika ada penambahan input atau output maka,

- Dideklarasikan juga harus ditambahkan.
- Pada pinMode juga harus ditambahkan
- Pada digitalWrite juga harus ditambahkan

Berpetualang dengan Tiga LED Hidup Bergantian

(LED1=merah hidup, kemudian mati. Berganti LED2 =hijau hidup, kemudian mati. Berganti LED3=kuning hidup, kemudian mati. Kemudian kembali ke LED1 lagi. Demikian secara terus menerus).

Kembali ke laptop, dan Ayahbunda bisa mencoba membuat coding terlebih dahulu. Mari kita simak saja, codingan berikut ini.

Karena ada tiga LED, maka kita akan siapkan dulu tiga deklarasi sebagai berikut :

```
int led_merah=13; //led merah pada pin 13
int led_hijau=12; //led hijau pada pin 12
int led_kuning=11; //led kuning pada pin 11
```

pada pinMode juga harus ditambahkan :

```
pinMode(13,OUTPUT);
pinMode(12,OUTPUT);
pinMode(11,OUTPUT);
```

pada digitalWrite juga harus ditambahkan :

```
digitalWrite(13,1);
delay(1000);
digitalWrite(13,0);
```

```
digitalWrite(12,1);
delay(1000);
digitalWrite(12,0);
```

```
digitalWrite(11,1);
delay(1000);
digitalWrite(11,0);
```

Kemudian masukkan ketiganya ke dalam sketch/lembar kerja Arduino IDE. Masukkan sesuai tempatnya masing-masing. Jadinya adalah seperti di bawah ini :

```
int led_merah=13; //led merah pada pin 13
int led_hijau=12; //led hijau pada pin 12
int led_kuning=11;//led kuning pada pin 11

void setup() {
  pinMode(13,OUTPUT);
  pinMode(12,OUTPUT);
  pinMode(11,OUTPUT);
}

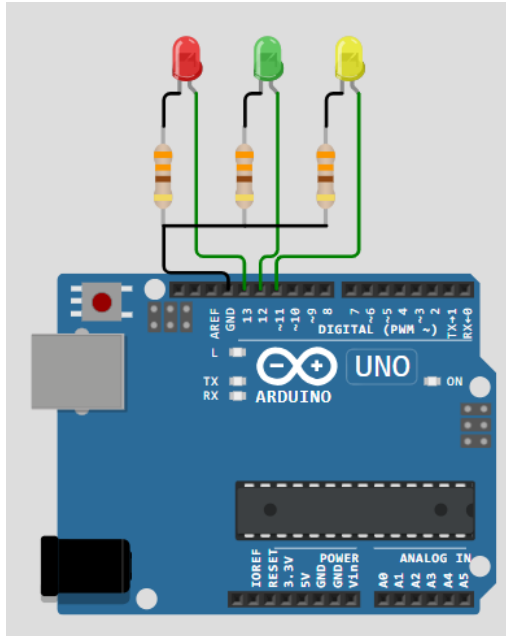
void loop() {
  digitalWrite(13,1);
  delay(1000);
  digitalWrite(13,0);

  digitalWrite(12,1);
  delay(1000);
  digitalWrite(12,0);

  digitalWrite(11,1);
  delay(1000);
  digitalWrite(11,0);
}
```

Setelah di Verify dan tidak terjadi kesalahan maka coding bisa diupload. Setelah itu rangkai komponen

seperti yang diminta pada soal. Hasilnya adalah sebagai berikut :



Nah, tantangan Ayahbunda berikutnya adalah silahkan membuat coding untuk Lima LED yang bekerja seperti soal di atas. Lima LED hidup secara bergantian dan berurutan secara terus menerus. Pasti Ayahbunda sudah sangat lihai untuk menyelesaikan tantangan tersebut. Sebagai *clue*, cara kerjanya sama, hanya LED nya yang ditambah, jika semula tiga LED sekarang menjadi lima LED. Bagaimana? Oke, kan Ayahbunda?

Berpetualang dengan 5 LED Hidup Maju-Mundur

(Lima LED, silahkan pilih warna sesukanya. LED akan hidup secara bergantian dari LED1, LED2, LED3, LED4 dan LED5. Kemudian LED hidup berbalik, dari LED4, LED3, LED2, LED1. Begitu seterusnya).

Kita siapkan deklarasinya dulu. Jika dari tadi kita menggunakan syntax `Typedata_variabel=pin`, sekarang kita coba pakai syntax yang lain, yaitu `#define variabel pin`. LED ada lima, maka deklarasi juga harus ada lima.

```
#define led_merah 13
#define led_hijau 12
#define led_kuning 11
#define led_biru 10
#define led_putih 9
```

pada `pinMode` juga harus ditambahkan :

```
pinMode(13,OUTPUT);
pinMode(12,OUTPUT);
pinMode(11,OUTPUT);
pinMode(10,OUTPUT);
pinMode(9,OUTPUT);
```

pada `digitalWrite` juga harus ditambahkan :

- LED hidup berurutan maju

```
digitalWrite(13,1);
delay(1000);
digitalWrite(13,0);
```

```
digitalWrite(12,1);  
delay(1000);  
digitalWrite(12,0);
```

```
digitalWrite(11,1);  
delay(1000);  
digitalWrite(11,0);
```

```
digitalWrite(10,1);  
delay(1000);  
digitalWrite(10,0);
```

```
digitalWrite(9,1);  
delay(1000);  
digitalWrite(9,0);
```

- LED hidup berurutan mundur

```
digitalWrite(10,1);  
delay(1000);  
digitalWrite(10,0);
```

```
digitalWrite(11,1);  
delay(1000);  
digitalWrite(11,0);
```

```
digitalWrite(12,1);  
delay(1000);  
digitalWrite(12,0);
```

```
digitalWrite(13,1);  
digitalWrite(13,0);
```

Seperti pada contoh sebelumnya, kita satukan struktur tersebut pada sketch/lembar kerja Arduino IDE :

```
#define led_merah 13
#define led_hijau 12
#define led_kuning 11
#define led_biru 10
#define led_putih 9

void setup() {
  pinMode(13,OUTPUT);
  pinMode(12,OUTPUT);
  pinMode(11,OUTPUT);
  pinMode(10,OUTPUT);
  pinMode(9,OUTPUT);
}

void loop() {
  //Lampu berurutan MAJU
  digitalWrite(13,1);
  delay(1000);
  digitalWrite(13,0);

  digitalWrite(12,1);
  delay(1000);
  digitalWrite(12,0);

  digitalWrite(11,1);
  delay(1000);
  digitalWrite(11,0);

  digitalWrite(10,1);
  delay(1000);
  digitalWrite(10,0);
```

```

digitalWrite(9,1);
delay(1000);
digitalWrite(9,0);

//Lampu berurutan MUNDUR
digitalWrite(10,1);
delay(1000);
digitalWrite(10,0);

digitalWrite(11,1);
delay(1000);
digitalWrite(11,0);

digitalWrite(12,1);
delay(1000);
digitalWrite(12,0);

digitalWrite(13,1);
digitalWrite(13,0);
}

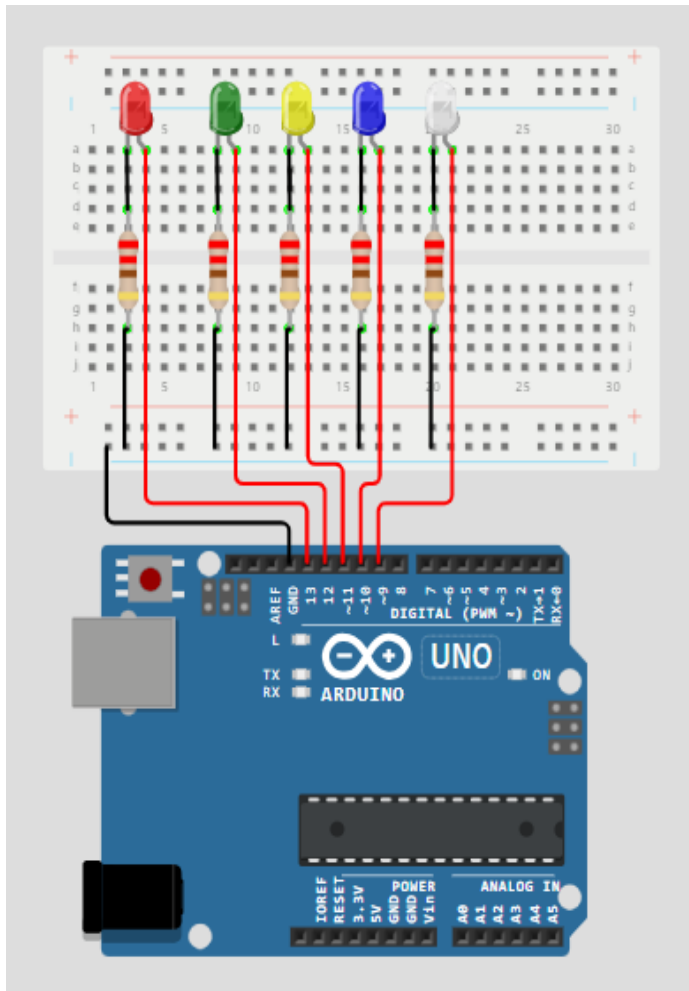
```

Catatan Penting :

Dalam membuat coding dengan Arduino IDE tanda baca sangat penting tidak boleh ada yang hilang atau terhapus. Karena sistem akan membacanya sebagai eror. Tanda baca yang sering ada dalam syntax Arduino IDE antara lain :

- () kurung buka kurung tutup. Setiap membuat syntax yang diawali kurung buka, harus diakhiri oleh kurung tutup.
- { } kurung kurawal buka dan tutup. Setiap membuat syntax yang diawali kurung kurawal buka, harus diakhiri oleh kurung kurawal tutup.
- ; semikolon atau titik koma, ditempatkan di setiap tanda kurung tutup biasa (bukan kurung kurawal).
- , tanda koma (tanpa titik atau hanya koma saja).

Untuk selanjutnya, Ayahbunda akan kita ajak untuk merangkai komponen pada breadboard yang sudah disiapkan. Bagaimana membuat rangkaian tersebut pada breadboard? Perhatikan gambar berikut.



Penjelasan lengkap tentang rangkaian pada breadboard di atas adalah sebagai berikut :

- Pin GND (Ground) sudah terhubung langsung dan masuk ke salah satu kaki resistor (anggap saja sebagai R1, R2, R3, R4 dan R5).
- Keluaran dari :
 - R1 masuk ke Katoda (-) LED Merah
 - R2 masuk ke Katoda (-) LED Hijau
 - R3 masuk ke Katoda (-) LED Kuning
 - R4 masuk ke Katoda (-) LED Biru
 - R5 masuk ke Katoda (-) LED Putih
- Keluaran dari LED
 - Anoda(+) LED Merah langsung ke pin 13
 - Anoda(+) LED Merah langsung ke pin 12
 - Anoda(+) LED Merah langsung ke pin 11
 - Anoda(+) LED Merah langsung ke pin 10
 - Anoda(+) LED Merah langsung ke pin 9

Catatan Penting :

- Lubang breadboard yang sejajar dengan tanda positif (+) saling terhubung secara horizontal.
- Begitu juga lubang breadboard yang sejajar dengan tanda negatif (-) saling terhubung secara horizontal.
- Sedangkan lubang breadboard jalur a, b, c, d dan e terhubung secara vertikal.
- Begitu juga lubang breadboard jalur f, g, h, i dan j terhubung secara vertikal.

Menyingkat Struktur dengan Array

Mari kita perhatikan kembali coding sudah kita buat sebelumnya. Yaitu, coding tentang perintah untuk menhidupkan lima LED secara berurutan Maju-Mundur.

```
#define led_merah 13
#define led_hijau 12
#define led_kuning 11
#define led_biru 10
#define led_putih 9

void setup() {
  pinMode(13,OUTPUT);
  pinMode(12,OUTPUT);
  pinMode(11,OUTPUT);
  pinMode(10,OUTPUT);
  pinMode(9,OUTPUT);
}

void loop() {
  //Lampu berurutan MAJU
  digitalWrite(13,1);
  delay(1000);
  digitalWrite(13,0);

  digitalWrite(12,1);
  delay(1000);
  digitalWrite(12,0);

  digitalWrite(11,1);
  delay(1000);
  digitalWrite(11,0);
```

```

digitalWrite(10,1);
delay(1000);
digitalWrite(10,0);

digitalWrite(9,1);
delay(1000);
digitalWrite(9,0);

//Lampu berurutan MUNDUR
digitalWrite(10,1);
delay(1000);
digitalWrite(10,0);

digitalWrite(11,1);
delay(1000);
digitalWrite(11,0);

digitalWrite(12,1);
delay(1000);
digitalWrite(12,0);

digitalWrite(13,1);
digitalWrite(13,0);
}

```

Secara logika, tidak ada yang salah dalam coding tersebut. Hasilnya juga bisa memerintahkan output sesuai dengan tugas yang ada. Yang perlu dipikirkan kemudian adalah bagaimana jika item atau jumlah lampu yang akan Ayahbunda kendalikan jumlahnya

banyak? Seratus, misalnya. Ayahbunda harus membuat digitalWrite sebanyak 199 pasang. Tentu itu bukan suatu pekerjaan yang menarik. Karena itu, Ayahbunda perlu menyingkat atau memperpendek coding-nya. Bagaimana caranya? Ayo, kita mulai bersenang-senang. Kita ambil contoh dulu, kreasi dengan sembilan lampu LED dengan cara kerja sama seperti soal di atas yaitu Hidup Maju-Mundur.

Deklarasikan sembilan LED dan tambah dengan Array :

```
//SEMBILAN LAMPU BERURUTAN (PAKAI ARRAY DENGAN TIPE DATA
SAMA)
int merah=13;
int hijautua=12;
int biru=11;
int kuning=10;
int orange=9;
int putih=8;
int ungotua=7;
int ungomuda=6;
int hijau muda=5;

int led[9]={13,12,11,10,9,8,7,6,5}; //Array untuk menyingkat
```

pinMode juga akan berubah karena adanya Array :

```
for (int i=0;i<9;i++) //Pernyataan increment
{pinMode(led[i],OUTPUT);
}
```

Jadi, bisa dibaca yang menjadi OUTPUT adalah variable led dan i (nama variable bisa diganti sesuai selera. Tapi lebih baik yang singkat dan jelas, supaya tidak membingungkan).

digitalWrite juga harus disesuaikan :

```
for(int j=0;j<9;j++){ //Untuk Maju
digitalWrite(led[j],HIGH);
delay(500);
digitalWrite(led[j],LOW);
//delay(500);
}
for(int j=7;j>0;j--){ //Untuk Mundur
digitalWrite(led[j],HIGH);
//delay(500);
digitalWrite(led[j],LOW);
delay(500);
}
```

Jika dibaca, untuk j dari 0 (nol) sampai 8 maka hidup secara berurutan (Maju). Dan kemudian, untuk j dari 7 ke 0 (nol) hidup secara berurutan (Mundur).

Jadi, coding secara keseluruhan akan menjadi :

```
//SEMBILAN LAMPU BERURUTAN (PAKAI ARRAY DENGAN TIPE DATA
SAMA)
int merah=13;
int hijautua=12;
int biru=11;
int kuning=10;
int orange=9;
int putih=8;
int ungutua=7;
int ungumuda=6;
int hijaumuda=5;

int led[9]={13,12,11,10,9,8,7,6,5};

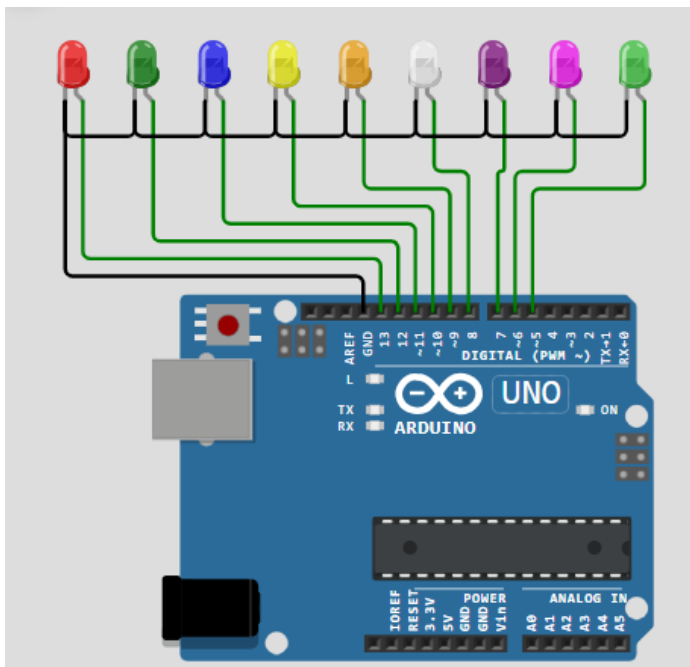
void setup() {
  // put your setup code here, to run once:
  for (int i=0;i<9;i++)
  {pinMode(led[i],OUTPUT);
  }}
void loop () {
  for(int j=0;j<9;j++){
  digitalWrite(led[j],HIGH);
  delay(500);
```

```

digitalWrite(led[j],LOW);
//delay(500);
}
for(int j=7;j>0;j--){
digitalWrite(led[j],HIGH);
//delay(500);
digitalWrite(led[j],LOW);
delay(500);
}}

```

Sedangkan rangkaiannya akan menjadi seperti :








Coba sekarang, Ayahbunda membuat coding pada tugas sebelumnya, yaitu lima LED hidup Maju-Mundur menggunakan Array. Lebih singkat dan asyik, kan?

Petualangan 5

ASYIKNYA SIMULASI!

Jika Ayahbunda sudah membaca sampai halaman ini, berarti Ayahbunda sudah bisa dikatakan sebagai Ayah dan Bunda yang Ahli Coding di mata anak-anak. Sudah saatnya pula Ayahbunda mulai mengajak putera-puterinya untuk ikut berpetualang. Atau, jangan-jangan Ayahbunda sudah berpetualang bersama keluarga selama mempelajari buku ini? Jika benar begitu, maka hal itu merupakan sebuah langkah maju yang hebat. Pasti Ayahbunda sudah membeli perangkat Arduino Kit? Berapa harganya? Harganya pasti bervariasi, tergantung kelengkapan komponen atau perangkatnya. Jika kita searching di salah satu market place ternama di Indonesia, kita akan dihadapkan dengan banyak pilihan paket Arduino Kit. Mulai dari yang sangat sederhana sampai ke yang lengkap. Untuk menentukan mana yang harus Ayahbunda pilih, tergantung dari kreasi yang akan Ayahbunda buat. Jika kreasinya semakin rumit, maka

memerlukan komponen atau perangkat yang lebih lengkap. Jika kreasi yang akan dibuat sederhana, cukup membeli paket Arduino Kit yang sederhana. Hal ini juga bisa menyesuaikan budget yang Ayahbunda miliki. Mari, kita coba intip harga Arduino Kit dengan variasi paketannya.

	Arduino Uno R3 kit paket dasar lengkap untuk pemula smd compatible Bebas Pengembalian	Rp150.000 Rp148.000
	paket combo arduino Uno Mega starter kit paket RFID compatible Bebas Pengembalian	Rp750.000 Rp620.000
	arduino Uno r3 dip kit advance paket pemula belajar compatible Bebas Pengembalian	Rp330.000 Rp299.000
	arduino Uno r3 kit paket dasar compatible Bebas Pengembalian	Rp199.000 Rp179.500
	Arduino Uno R3 starter kit paket belajar compatible 2 Bebas Pengembalian	Rp440.000 Rp379.000

Pertanyaannya adalah, bisakah Ayahbunda tetap berpetualang tanpa Arduino Kit? Jawabannya, tentu saja bisa, yaitu melalui simulasi. Simulasi ini juga dapat digunakan untuk mengecek coding dan rangkaian kita, sudah benar atau belum. Mari, kita mulai simulasinya!

Berpetualang dengan WOKWI!

Wokwi berfungsi sebagai simulator arduino yang dapat digunakan secara bebas baik itu pengajar maupun pelajar. Ayahbunda dapat menggunakannya untuk mensimulasikan Arduino, ESP32, MicroPython, board dan sensor populer lainnya tanpa perlu mengeluarkan biaya untuk membeli perangkat yang sebenarnya. Wokwi Simulator memiliki dua ruang kerja, yaitu ruang pemrograman dan ruang merangkai mikrokontroler dengan interface input output yang tersedia sebagai media pembuatan project. Perancangan menggunakan wokwi simulator sangat mendukung dalam menerapkan pembelajaran mikrokontroler dengan berbagai fitur yang sangat membantu pada setiap rancangan project arduino disertai berbagai sensor yang ada.

Wokwi Simulator menawarkan banyak fitur penunjang pembuatan rancangan mikrokontroler, yang dapat diakses gratis secara online dimana saja. Dalam penggunaan Wokwi simulator ini, Ayahbunda dapat menyimpan dan membagikan project yang telah dibuat ke pengguna lain sebagai media contoh penerapan

konsep anduino. Tersedia juga ruang komunitas yang memungkinkan pengguna untuk mempublish hasil kerja project yang dibuat. Ayahbunda juga bisa melakukan pengamatan terhadap hasil kerja dari pengguna lainnya. sehingga perbaikan dan perkembangan dalam

Di dalam Wokwi simulator ini Ayahbunda dapat memilih jenis board yang kita butuhkan seperti Arduino Uno, Arduino Mega, ESP32, Arduino Nano dan lainnya. Disamping itu, Ayahbunda juga dapat mengatur dan mengubah projek menggunakan berbagai macam sensor seperti sensor PIR, sensor ultra sonic dan sensor-sensor lainnya yang tersedia. Wokwi juga dilengkapi berbagai macam jenis LCD, seperti LCD 16 x 2, LCD 16 x 2 (12C), LCD 20 x 4, LCD 20 x 4 (12C), ILI9341 2 8" TFT-LCD display dan berbagai perangkat lainnya yang dapat mendukung pembuatan rancangan project.

Wokwi Simulator ini juga dapat dijadikan sebagai media ujicoba yang dapat dilakukan pada mikrokontroler serta dapat diakses di mana saja menggunakan web browser.

Wokwi meski sangat menarik, juga memiliki kelebihan dan kekurangan.

Kelebihan Wokwi diantaranya adalah :

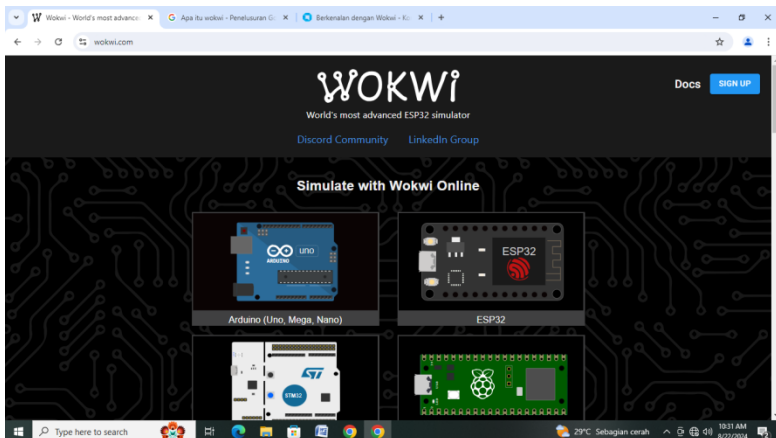
- Tidak perlu menunggu adanya komponen, atau mengunduh software. Tinggal memprogram kode untuk mulai memulai proyek IoT dalam hitungan detik.
- Tidak perlu takut salah dengan terjadinya kerusakan pada perangkat hardware, karena sistemnya virtual hardware.
- Unlimited Hardware tidak perlu membeli perangkat karena adanya berbagai media virtual hardware.
- Komunitas Discord Wokwi Discord Community yang membantu anda dalam penggunaan.
- Support Wifi Simulation yang dapat menggunakan protokol IoT seperti MQTT, HTTP, NTP.
- Virtual Logic Analyzer dapat menggunakan tampilan layar seperti UART, I2C, SPI yang dapat menganalisa data.
- GDB Debugging Support untuk penggunaan Arduino dan Raspberry Pi.

- SD Card dapat menyimpan dan mengambil file atau direktori dari kode yang telah dibuat. serta dapat di sharing.

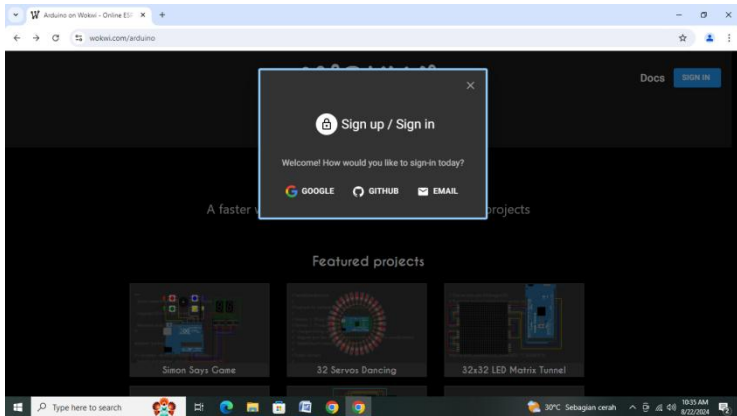
Sedangkan kekurangannya adalah :

- Belum lengkapnya perangkat atau sensor yang diberikan.
- Terjadinya kegagalan program yang mengakibatkan file tidak dapat dibuka.
- Kurang ketersediaan contoh konsep project tutorial.

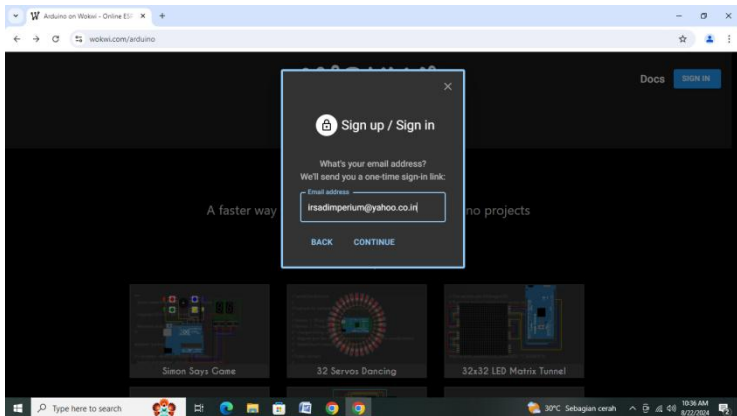
Bagaimana Ayahbunda? Kita mulai saja, ya? Baiklah, pertama-tama Ayahbunda perlu membuka website Wokwi di <http://www.wokwi.com>



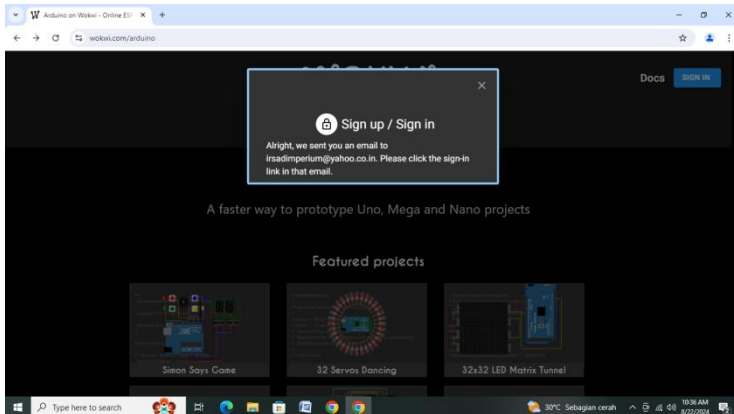
Ayahbunda perlu meng-klik Sign-Up/Sign In untuk memulainya. Karena jika tidak melakukan langkah ini, kreasi Ayahbunda tidak akan tersimpan.



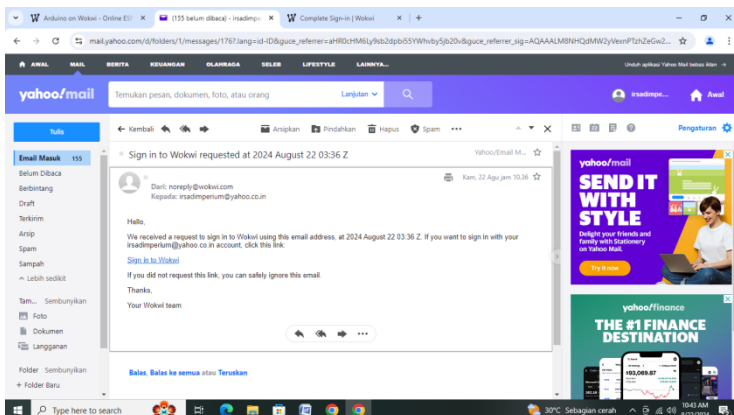
Ayahbunda bisa Sign Up dengan menggunakan akun google atau email.



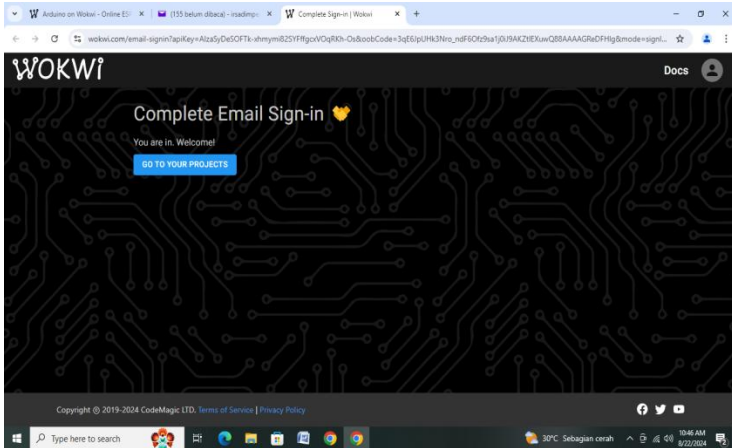
Ayahbunda akan mendapatkan akses link, yang dikirim melalui email yang digunakan untuk Sign Up.



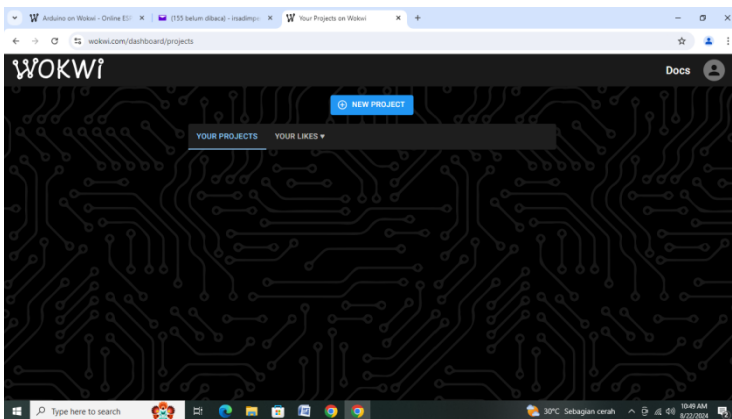
Cek email Ayahbunda dan klik Sign In to Wokwi.



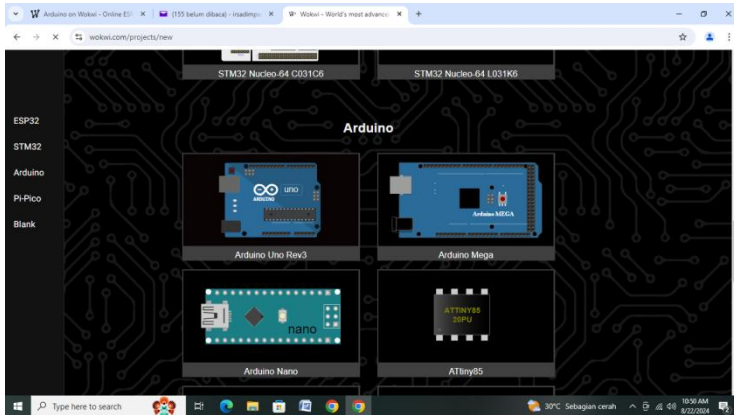
Ayahbunda sudah berhasil masuk ke dalam Wokwi dan ikuti langkah selanjutnya untuk menuju lembar kerja Wokwi.



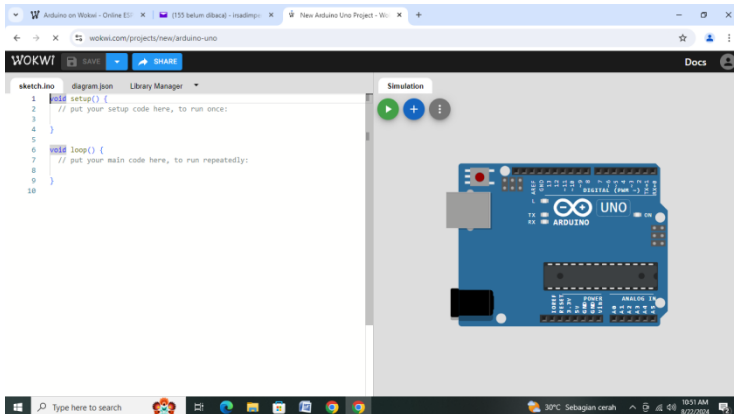
Klik Go to Your Project



Klik New Project, scroll ke bawah dan pilih Arduino Uno.



Maka akan tampil lembar kerja Wokwi



Ayahbunda sudah bisa memulai petualangan bersama Wokwi. Bagian kiri untuk mengetik coding dan bagian kanan untuk merangkai komponen.

Petualangan Tiga LED Hidup Berurutan

(Led1, Led2, Led3 akan hidup secara berurutan terus menerus)

Ayahbunda bisa menyalin coding berikut ke bagian kanan Wokwi.

```
int led_merah=13; //led merah pada pin 13
int led_hijau=12; //led hijau pada pin 12
int led_kuning=11;//led kuning pada pin 11

void setup() {
  pinMode(13,OUTPUT);
  pinMode(12,OUTPUT);
  pinMode(11,OUTPUT);
}

void loop() {
  digitalWrite(13,1);
  delay(1000);
  digitalWrite(13,0);

  digitalWrite(12,1);
  delay(1000);
  digitalWrite(12,0);

  digitalWrite(11,1);
  delay(1000);
  digitalWrite(11,0);
}
```

Selelah itu, Ayahbunda bisa mencoba membuat rangkaian komponennya. Tombol yang akan digunakan dalam membuat rangkaian adalah sebagai berikut :



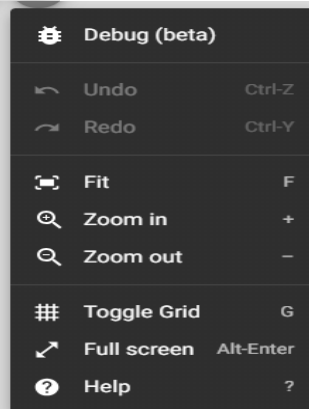
Untuk memainkan kreasi kita (play)



Untuk menambah komponen yang mau dipakai



Untuk melihat fungsi lainnya

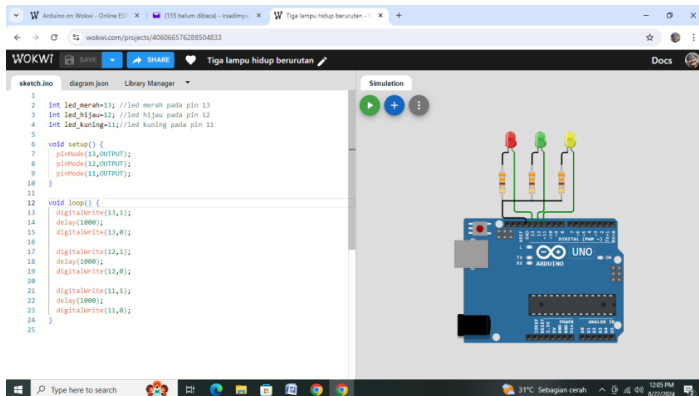


Klik ikon + dan pilih LED sebanyak tiga buah, resistor 220Ω atau 330 Ω sebanyak tiga buah. Ayahbunda bisa menata letak komponen-komponennya dengan cara diklik gambar komponennya dan digeser ke tempat yang diinginkan.

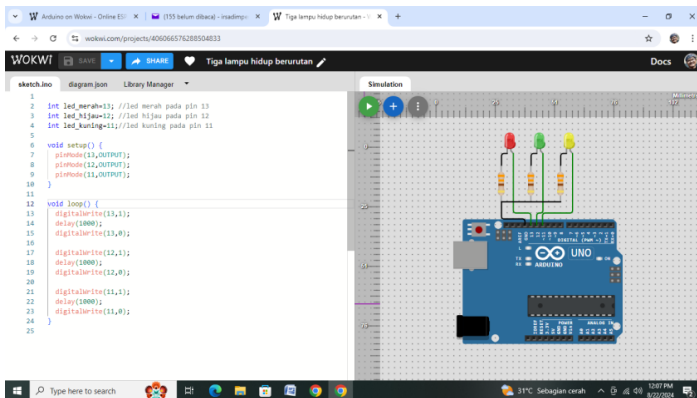
Untuk membuat wiring/pengkabelannya, Ayahbunda tinggal klik kaki-kaki yang mau dihubungkan. Setiap kali mau pindah ke komponen lain tekan Escape terlebih dahulu. Untuk membelokkan garis/wiring cukup diklik dimana akan membelok, maka garis/wiring akan secara otomatis membelok.

Untuk mengubah warna komponen seperti LED atau wiringnya, maka Ayahbunda hanya perlu meng-klik komponen atau wiringnya kemudian pilih warna yang diinginkan.

Ayahbunda tidak perlu merasa takut akan melakukan kesalahan. Jika salah tinggal memperbaikinya. Jika salah pada bagian coding maka tinggal memperbaiki codingnya. Jika salah pada rangkaian, Ayahbunda bisa menghapus dan mengganti dengan yang benar. Semua bisa Ayahbunda lakukan dengan santai dan asyik. Jadi, jika Ayahbunda sudah mencoba petualangan pertama tadi, maka hasilnya adalah sebagai berikut :



Dalam membuat simulasi rangkaian, supaya Ayahbunda lebih mudah membuat wiring, maka tampilannya bisa diubah ke Toggle Grid. Caranya, Klik Titik Tiga → Pilih Toggle Grid. Sehingga akan nampak seperti :



Setelah coding dan pembuatan rangkaian selesai maka Ayahbunda bisa mainkan dengan menekan tombol "play". Rangkaian tersebut akan bekerja sesuai rencana kerja di atas yaitu tiga led akan hidup berurutan secara terus menerus.

Petualangan 6

Nikmati Petualanganmu !

Untuk selanjutnya, Ayahbunda akan kita berikan beberapa coding dan gambar skematiknya. Ayahbunda mempunyai tantangan berpetualang untuk membuat rangkaianannya dengan Wokwi. Apakah Ayahbunda siap? Kalau Ayahbunda berhasil dengan Wokwi maka Ayahbunda pasti sudah lebih siap lagi untuk merangkainya di papan latihan dengan komponen-komponen yang sesungguhnya.

Petualangan LED Aktif HIGH

```
int led = 12;

void setup() {
  // put your setup code here, to run once:
  pinMode(led,OUTPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
  digitalWrite(led,1); //kalau hanya sampai sini tanpa
  delay, lampu hidup terus)
```

```

delay(1000); //ditambahi delay hanya menyala selama
waktu tertentu misal 100 millissecond
digitalWrite(led,0);
delay(1000);
}

```



Petualangan LED Aktif LOW

```

int led = 12;

void setup() {
  // put your setup code here, to run once:
  pinMode(led,OUTPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
  digitalWrite(led,0); //kalau hanya sampai sini tanpa
  delay, lampu hidup terus)
  delay(1000); //ditambahi delay hanya menyala selama
  waktu tertentu misal 100 millissecond
  digitalWrite(led,1);
  delay(1000);
}

```

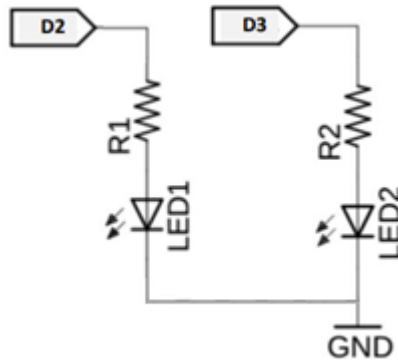


Petualangan LED Flip-Flop Sederhana

```
int led1=13;
int led2=12;

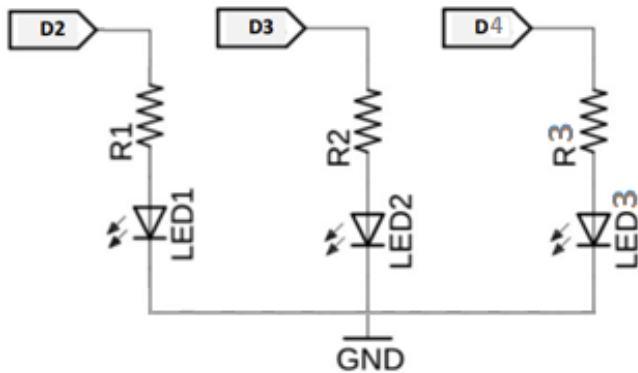
void setup() {
  pinMode(led1,OUTPUT);
  pinMode(led2,OUTPUT);
}

void loop() {
  digitalWrite(led1,1);
  digitalWrite(led2,0);
  delay(1000);
  digitalWrite(led1,0);
  digitalWrite(led2,1);
  delay(1000);
}
```



Petualangan Running Tiga LED

```
int led[3] = { 13, 12, 11 };
void setup() {
  // put your setup code here, to run once:
  for (int i = 0; i < 3; i++) {
    pinMode(led[i], OUTPUT);
  }
}
void loop() {
  // put your main code here, to run repeatedly:
  for (int j = 0; j < 3; j++) {
    digitalWrite(led[j], 1);
    delay(1000);
    digitalWrite(led[j], 0);
  }
}
```



Petualangan Lima LED Maju Mundur

```

int led[5] = { 9, 10, 13, 12, 11 };
void setup() {
  for (int i = 0; i < 5; i++) {
    pinMode(led[i], OUTPUT);
  }
}
void loop() {
  for (int j = 0; j < 5; j++) {
    digitalWrite(led[j], 1);
    delay(500);
    digitalWrite(led[j], 0);
  }
  for (int j = 3; j > 0; j--) {
    digitalWrite(led[j], 1);
    delay(500);
    digitalWrite(led[j], 0);
  }
}

```

Catatan :

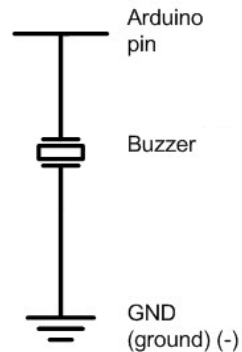
Untuk diagram skematik, sama dengan tiga led pada petualangan sebelumnya. Ayahnya tinggal menambahkan dua led dan dua resistor 220Ω atau 330Ω.

Petualangan Buzzer Hidup 3 Detik Terus Mati

```
int buzzer=13;
void setup() {
  pinMode(buzzer,OUTPUT); // ditaruh di set up karena
  hanya sekali bunyi dan mati (kalo ditaru di loop
  akan berulang terus tidka mau mati)
  digitalWrite(buzzer,1);
  delay(3000);
  digitalWrite(buzzer,0);
}
void loop() {
}
```

Petualangan Buzzer Hidup 0,75 Mati 0,5 Detik

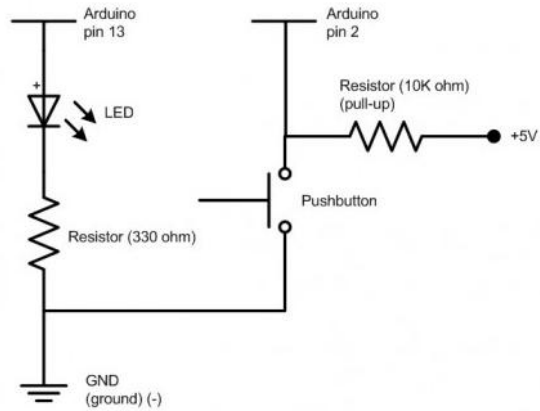
```
int buzzer=8;
void setup() {
  pinMode(buzzer,OUTPUT);
  digitalWrite(buzzer,HIGH);
  delay(750);
  digitalWrite(buzzer,LOW);
  delay(500);
  digitalWrite(buzzer,HIGH);
  delay(750);
  digitalWrite(buzzer,LOW);
  delay(500);
  digitalWrite(buzzer,HIGH);
  delay(750);
  digitalWrite(buzzer,LOW);
  delay(500);
  digitalWrite(buzzer,HIGH);
}
```



```
delay(1200);  
digitalWrite(buzzer,LOW);  
}  
void loop() {  
}
```

Petualangan Satu PB Satu LED, PULLUP

```
int pb=2;  
int led=13;  
  
void setup() {  
  pinMode(pb,INPUT);  pinMode(led,OUTPUT);  
  Serial.begin(9600);  
}  
  
void loop() {  
  // put your main code here, to run repeatedly:  
  int tekan = digitalRead(pb);  
  Serial.println(tekan);  
  if (tekan==1) {  
    digitalWrite(led,0);  
  }  
  if (tekan==0) {  
    digitalWrite(led,1);  
    Serial.println(tekan);  
  }  
}
```



Petualangan Satu PB Satu LED, PULDOWN

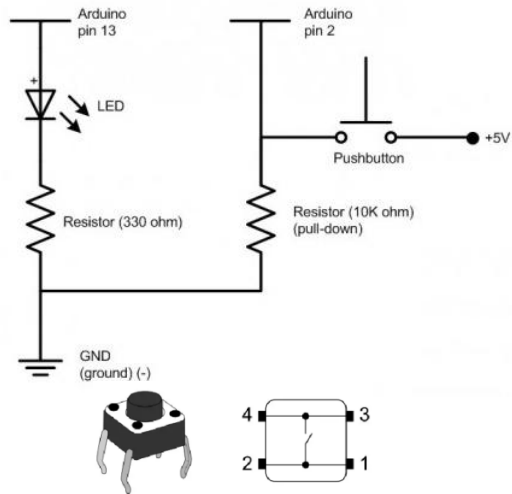
```

int pb=8;
int led=9;

void setup() {
  pinMode(pb,INPUT);
  pinMode(led,OUTPUT);
  Serial.begin(9600);
}

void loop() {
  // put your main code here, to run repeatedly:
  int tekan = digitalRead(pb);
  Serial.println(tekan);
  if (tekan==1) {
    digitalWrite(led,1);
  }
  if (tekan==0) {
    digitalWrite(led,0);
    Serial.println(tekan);
  }
}

```



Jadi, yang bisa dipakai adalah kaki 1-4 atau 2-3 atau menyilang. Misal, inputnya dari 1 maka outputnya 4 atau jika inputnya 2 maka outputnya 3.

Petualangan PB Tekan, LED Hidup Terus, Tekan lagi LED Baru Mati

```
#define LED 13
#define PB 12

int HIDUP = false;

void setup() {
  pinMode(LED, OUTPUT);
  pinMode(PB, INPUT_PULLUP);
}

void loop() {
```

```

if (digitalRead(PB) == 1) {
  delay(50);
  if (digitalRead(PB) == 1) {
    if (HIDUP == false) {
      HIDUP = true;
      digitalWrite(LED, 1);
    } else if (HIDUP == true) {
      HIDUP = false;
      digitalWrite(LED, 0);
    }
    while (digitalRead(PB) == 1)
      ;
  }
}
}
}
}

```

Petualangan PB1 On, PB2 Off, Led dan Buzzer

```

int pb1 = 12;
int led = 13;
int pb2 = 11;
int buzz=10;
bool statusLampu = true;

void setup() {
  // put your setup code here, to run once:
  pinMode(led, OUTPUT);
  pinMode(pb1, INPUT);
  pinMode(pb2, INPUT);
  pinMode(buzz,OUTPUT);
}

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

  if (digitalRead(pb1) == 0 && statusLampu == true)
  { //PB pulldown
    statusLampu = false;
    digitalWrite(led, 1);
  }
}

```

```

        digitalWrite(buzz,1);
        delay(1000);
    } else if (digitalRead(pb2) == 0 && statusLampu ==
false) {
        statusLampu = true;
        digitalWrite(led, 0);
        digitalWrite(buzz,0);
        delay(1000);
    }
}

```

Catatan :

Rangkaannya merupakan gabungan antara PB, LED dan Buzzer.

Petualangan LDR (Light Dependent Resistor) dan LCD I2C

```

#define led 2
#define ldr A0
int tegangan;
#include <LCD_I2C.h>
LCD_I2C lcd(0x27, 16, 2);

void setup() {
    pinMode(led, OUTPUT);
    pinMode(ldr, INPUT);
    Serial.begin(9600);
    lcd.begin();
    from calling Wire.begin()
    lcd.backlight();
}

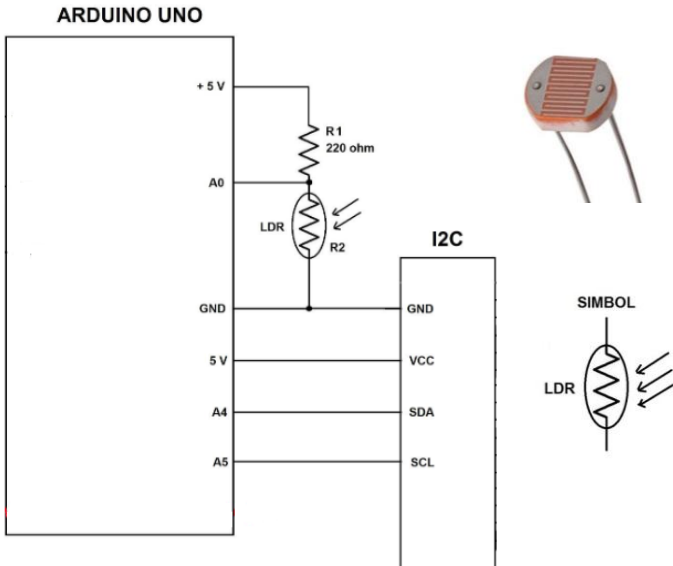
void loop() {
    tegangan = analogRead(ldr);
    Serial.println(tegangan);
}

```

```

Serial.print("Nilai LDR :");
delay(500);
lcd.setCursor(0,0);
lcd.print("Nilai LDR=");
lcd.print(tegangan);
delay(500);
if (tegangan < 350) {
    digitalWrite(led, 1);
    lcd.setCursor(0,1);
    lcd.print("Nyala");
}
else {
    digitalWrite(led, 0);
    lcd.setCursor(0, 1);
    lcd.print("Mati");}
}

```



Petualangan Seven Segmen Mengaktifkan 0-9

```
//7 Segments ---> Common Anode
#define SEGMENT_A 2
#define SEGMENT_B 3
#define SEGMENT_C 4
#define SEGMENT_D 5
#define SEGMENT_E 6
#define SEGMENT_F 7
#define SEGMENT_G 8

unsigned long time = 500;

void setup() {
  Serial.begin(9600);
  pinMode(SEGMENT_A, OUTPUT);
  pinMode(SEGMENT_B, OUTPUT);
  pinMode(SEGMENT_C, OUTPUT);
  pinMode(SEGMENT_D, OUTPUT);
  pinMode(SEGMENT_E, OUTPUT);
  pinMode(SEGMENT_F, OUTPUT);
  pinMode(SEGMENT_G, OUTPUT);
}

void loop() {
  //====>> Angka 0
  digitalWrite(SEGMENT_A, 0);
  digitalWrite(SEGMENT_B, 0);
  digitalWrite(SEGMENT_C, 0);
  digitalWrite(SEGMENT_D, 0);
  digitalWrite(SEGMENT_E, 0);
  digitalWrite(SEGMENT_F, 0);
  digitalWrite(SEGMENT_G, 1);
  delay(time);
  //====>> Angka 1
  digitalWrite(SEGMENT_A, 1);
  digitalWrite(SEGMENT_B, 0);
  digitalWrite(SEGMENT_C, 0);
```

```
digitalWrite(SEGMENT_D, 1);
digitalWrite(SEGMENT_E, 1);
digitalWrite(SEGMENT_F, 1);
digitalWrite(SEGMENT_G, 1);
delay(time);
//====>> Angka 2
digitalWrite(SEGMENT_A, 0);
digitalWrite(SEGMENT_B, 0);
digitalWrite(SEGMENT_C, 1);
digitalWrite(SEGMENT_D, 0);
digitalWrite(SEGMENT_E, 0);
digitalWrite(SEGMENT_F, 1);
digitalWrite(SEGMENT_G, 0);
delay(time);
//====>> Angka 3
digitalWrite(SEGMENT_A, 0);
digitalWrite(SEGMENT_B, 0);
digitalWrite(SEGMENT_C, 0);
digitalWrite(SEGMENT_D, 0);
digitalWrite(SEGMENT_E, 1);
digitalWrite(SEGMENT_F, 1);
digitalWrite(SEGMENT_G, 0);
delay(time);
//====>> Angka 4
digitalWrite(SEGMENT_A, 1);
digitalWrite(SEGMENT_B, 0);
digitalWrite(SEGMENT_C, 0);
digitalWrite(SEGMENT_D, 1);
digitalWrite(SEGMENT_E, 1);
digitalWrite(SEGMENT_F, 0);
digitalWrite(SEGMENT_G, 0);
delay(time);
//====>> Angka 5
digitalWrite(SEGMENT_A, 0);
digitalWrite(SEGMENT_B, 1);
digitalWrite(SEGMENT_C, 0);
digitalWrite(SEGMENT_D, 0);
```

```
digitalWrite(SEGMENT_E, 1);
digitalWrite(SEGMENT_F, 0);
digitalWrite(SEGMENT_G, 0);
delay(time);
//====>> Angka 6
digitalWrite(SEGMENT_A, 0);
digitalWrite(SEGMENT_B, 1);
digitalWrite(SEGMENT_C, 0);
digitalWrite(SEGMENT_D, 0);
digitalWrite(SEGMENT_E, 0);
digitalWrite(SEGMENT_F, 0);
digitalWrite(SEGMENT_G, 0);
delay(time);
//====>> Angka 7
digitalWrite(SEGMENT_A, 0);
digitalWrite(SEGMENT_B, 0);
digitalWrite(SEGMENT_C, 0);
digitalWrite(SEGMENT_D, 1);
digitalWrite(SEGMENT_E, 1);
digitalWrite(SEGMENT_F, 1);
digitalWrite(SEGMENT_G, 1);
delay(time);
//====>> Angka 8
digitalWrite(SEGMENT_A, 0);
digitalWrite(SEGMENT_B, 0);
digitalWrite(SEGMENT_C, 0);
digitalWrite(SEGMENT_D, 0);
digitalWrite(SEGMENT_E, 0);
digitalWrite(SEGMENT_F, 0);
digitalWrite(SEGMENT_G, 0);
delay(time);
//====>> Angka 9
digitalWrite(SEGMENT_A, 0);
digitalWrite(SEGMENT_B, 0);
digitalWrite(SEGMENT_C, 0);
digitalWrite(SEGMENT_D, 0);
digitalWrite(SEGMENT_E, 1);
```

```

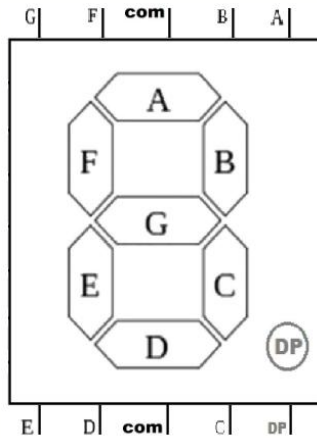
digitalWrite(SEGMENT_F, 0);
digitalWrite(SEGMENT_G, 0);
delay(time);
}

```

Catatan :

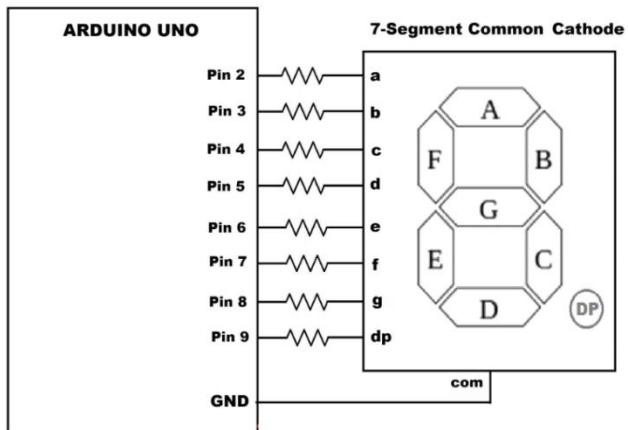
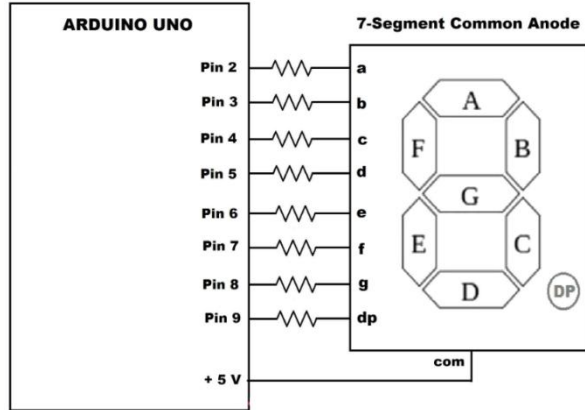
Seven Segmen ada dua jenis : Common Anoda dan Common Katoda. Perbedaannya hanya pada masalah coding. Pada Common Anoda perintah 1 berarti Hidup dan perintah 0 berarti mati. Sebaliknya pada Common Katoda, perintah 0 berarti menghidupkan sedangkan perintah 1 malah mematikan. Bagaimana cara memastikan bahwa seven segmen tersebut Common Anaoda atau Katoda? Berikut langkah-langkahnya :

- Perhatikan gambar Seven Segmen berikut :



- Hubungkan kaki COM pada +5V dan salah satu kaki (pilih salah satu dari A,B,C,D,E,F,G) ke GND. Lihat hasilnya, jika LED menyala maka itu adalah Common Anoda.
- Sekarang Ayahbunda balik posisinya. Hubungkan kaki COM pada GND dan salah satu kaki (pilih

salah satu dari A,B,C,D,E,F,G)ke +5V. Lihat hasilnya, jika LED menyala maka itu adalah Common Katoda. Jika tidak menyala, maka Seven Segmen dalam keadaan rusak.



Untuk menghidupkan angka-angka pada Seven Segmen, bisa berpedoman pada table berikut :

	A	B	C	D	E	F	G
0	1	1	1	1	1	1	0
1	0	1	1	0	0	0	0
2	1	1	0	1	1	0	1
3	1	1	1	1	0	0	1
4	0	1	1	0	0	1	1
5	1	0	1	1	0	1	1
6	1	0	1	1	1	1	1
7	1	1	1	0	0	0	0
8	1	1	1	1	1	1	1
9	1	1	1	1	1	0	1

Petualangan 3 PB, Buzzer dan Seven Segmen PB1 Maju (0-9) PB1 Mundur (9-0) PB3 Reset Setiap PB ditekan Buzzer Bunyi

```

/7 Segments ---> Common Anode
#define SW_UP 9
#define SW_DOWN 10
#define SW_RESET 11
#define BUZZER 12
const int SEGMENT_PIN[7] = {2, 3, 4, 5, 6, 7, 8};

bool digit[10][7] = {
    {1, 1, 1, 1, 1, 1, 0}, //Angka 0
    {0, 1, 1, 0, 0, 0, 0}, //Angka 1
    {1, 1, 0, 1, 1, 0, 1}, //Angka 2
    {1, 1, 1, 1, 0, 0, 1}, //Angka 3

```

```

    {0, 1, 1, 0, 0, 1, 1}, //Angka 4
    {1, 0, 1, 1, 0, 1, 1}, //Angka 5
    {1, 0, 1, 1, 1, 1, 1}, //Angka 6
    {1, 1, 1, 0, 0, 0, 0}, //Angka 7
    {1, 1, 1, 1, 1, 1, 1}, //Angka 8
    {1, 1, 1, 1, 0, 1, 1} //Angka 9
};

int val = 0;

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

    pinMode(SW_UP, INPUT_PULLUP);
    pinMode(SW_DOWN, INPUT_PULLUP);
    pinMode(SW_RESET, INPUT_PULLUP);
    pinMode(BUZZER, OUTPUT);
    for (int i = 0; i < 7; i++)
pinMode(SEGMENT_PIN[i], OUTPUT);
    for (int i = 0; i < 7; i++)
digitalWrite(SEGMENT_PIN[i], HIGH);
}

void loop() {
    displayDIGIT(val);

    if (digitalRead(SW_UP) == 0 && val < 9) {
        tone(BUZZER, 2500);
        while (digitalRead(SW_UP) == 0) {}
        noTone(BUZZER);
        val++;
    }
    if (digitalRead(SW_DOWN) == 0 && val > 0) {
        tone(BUZZER, 2400);
        while (digitalRead(SW_DOWN) == 0) {}
        noTone(BUZZER);
        val--;
    }
}

```

```

    }
    if (digitalRead(SW_RESET) == 0) {
        tone(BUZZER, 2300);
        while (digitalRead(SW_RESET) == 0) {}
        noTone(BUZZER);
        val = 0;
    }
}

void displayDIGIT(int num) {
    for (int i = 0; i < 7; i++) {
        digitalWrite(SEGMENT_PIN[i], 1 - digit[num][i]);
    }
}

```

Catatan :

Rangkaian skematik merupakan gabungan dari PB, Buzzer dan Seven Segmen.

Petualangan Sensor Ultrasonik dan Buzzer

```

//Indikator Jarak Ultrasonik
Const int pingPin = 7;
Const int echoPin = 6;

bool kondisi1 = false;
bool kondisi1 = false;
bool kondisi1 = false;

int buzzer = 8;

void setup() {
    Serial.begin(9600);
    pinMode(pingPin, OUTPUT);
    pinMode(buzzer, OUTPUT);
    pinMode(echoPin, INPUT);
}

```

```

void loop() {
    longduration, cm;
    digitalWrite(pingPin, LOW);
    delayMicroseconds(2);
    digitalWrite(pingPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(pingPin, LOW);
    duration = pulseIn(echoPin,HIGH);
    cm = microsecondsToCentimeters(duration);

    Serial.print("Jarak : ");
    Serial.print("\t");
    Serial.print(cm);
    Serial.print("cm");
    Serial.print();

    if(cm<=25){
        kondisi1 = true;
        kondisi2 = false;
        kondisi3 = false;
    }
    if(cm<=15){
        kondisi1 = false;
        kondisi2 = true;
        kondisi3 = false;
    }

    if(cm<=7){
        kondisi1 = false;
        kondisi2 = false;
        kondisi3 = true;
    }
}
else {
    kondisi1 = false;
    kondisi2 = false;
    kondisi3 = false;
}
}

```

```
}

if(kondisi1==true)
{
    digitalWrite(buzzer,LOW);
    delay(300);
    digitalWrite(buzzer,HIGH);
    delay(300);
    digitalWrite(buzzer,LOW);
    delay(300);
    digitalWrite(buzzer,HIGH);
    delay(300);
    digitalWrite(buzzer,LOW);
}
else
{
    digitalWrite(buzzer,LOW);
}

if(kondisi2==true)
{
    digitalWrite(buzzer,LOW);
    delay(100);
    digitalWrite(buzzer,HIGH);
    delay(100);
    digitalWrite(buzzer,LOW);
    delay(100);
    digitalWrite(buzzer,HIGH);
    delay(100);
    digitalWrite(buzzer,LOW);
}
else
{
    digitalWrite(buzzer,LOW);
}

if(kondisi3==true)
```

```

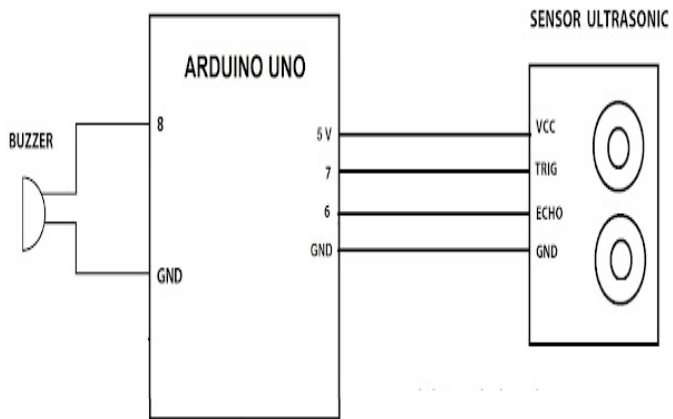
{
  digitalWrite(buzzer,HIGH);
}
else
{
  digitalWrite(buzzer,LOW);
}
}

```

```

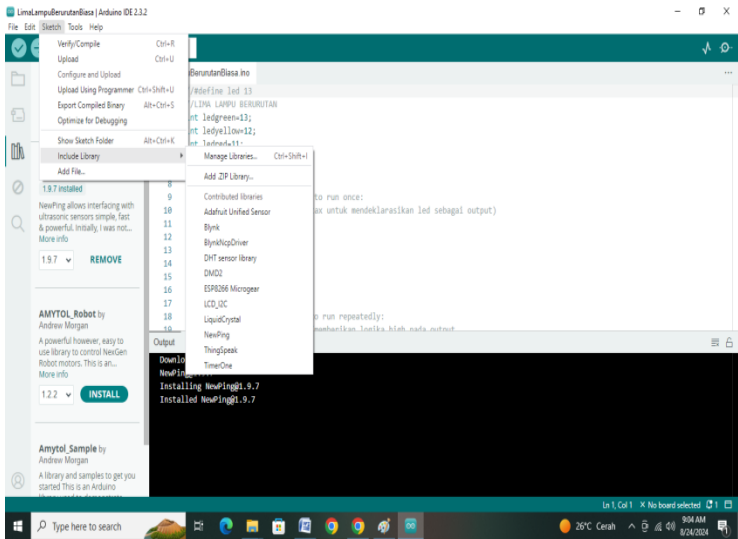
longmicrosecondsToCentimeter(long
micropseconds) {
return microseconds/29/2;
}

```



Catatan :

Jangan lupa untuk menginstal library dari sensor ultrasonik. Sketch → Include Library → Manage Libraries → Library Manager → NewPing By Tim Eckel → Instal.



DAFTAR PUSTAKA

- Banzi, Massimo; Shiloh, Michael (2022). Make: Memulai dengan Arduino: Platform Prototipe Elektronik Sumber Terbuka (edisi ke-4). Make Community.
- Blum, Jeremy (2019). Menjelajahi Arduino: Alat dan Teknik untuk Teknik Sihir (edisi ke-2). Wiley.
- Boxall, John (2021). Arduino Workshop: Pengenalan Praktis dengan 65 Proyek (edisi ke-2). No Starch Press.
- Karvinen, Tero; Karvinen, Kimmo; Valtokari, Ville (2014). Merek: Sensor (Edisi ke-1st). Buat Komunitas.
- Monk, Simon (2022). Pemrograman Arduino: Memulai dengan Sketches (edisi ke-3). McGraw-Hill Education.
- Nussey, John (2018). Arduino For Dummies (edisi ke-2). John Wiley & Sons.
- Schmidt, Maik (2015). Arduino: Panduan Memulai Cepat (edisi ke-2). Pragmatic Bookshelf .
- Handout Pelatihan *Embedded System* Berbasis *IoT*, Balai Besar Pelatihan Vokasi dan Produktivitas Bekasi. 2024.

Sumber Internet :

- <https://www.edtechs.com.au/products/lilypad>
- <https://store.arduino.cc/products/>
- <https://botland.store/withdrawn-products/1591>

- <https://blog.unnes.ac.id/antosupri/arduino-pro-mini/>
- <https://habibi028.wordpress.com/arduino-esplora/>
- <https://docs.arduino.cc/retired/other/arduino-robot/>
- <https://kodingindonesia.com/belajar-iot-bagian-2-monitoring-suhu-dan-kelembaban-dengan-thingspeak/#2>
- <https://newbiely.com/tutorials/esp8266/esp8266-dht11-lcd>
- <https://medium.com/@shfamrsya/mengintegrasikan-sensor-dht-11-dan-bme280-dengan-lcd-i2c-8a23857c365>
- <https://www.arduino.biz.id/2020/10/apa-yang-dimaksud-dengan-nodemcu-esp8266.html>
- <https://blog.famosastudio.com/tutorial/tutorial-arduino-buzzer/171/>
- <https://blog.famosastudio.com/2011/07/tutorial/tutorial-arduino-push-buttons/311/>
- <https://www.scribd.com/document/485329020/Pertemuan-2-Input-Pada-Arduino>
- <https://www.elektronikahendry.com/2020/12/arduino-part-16-tiga-macam-praktik.html>
- <https://www.elektronikahendry.com/arduino-part-9-cara-menghidupkan-7.html>
- <https://www.elektronikahendry.com/arduino-part-20-alarm-jarak-sensor.html>
- <http://www.labelektronika.com/2017/03/tutorial-mendapatkan-hex-file-pada-arduino.html>

TENTANG PENULIS



Muhamad Irsadul Ngibad, S.T., M.M., lahir di Magelang, 23 Juli 1980. Menyelesaikan pendidikan dasar hingga menengah atas di kota kelahirannya. Selepas SMA harus hijrah ke Kota Pahlawan, karena mendapatkan beasiswa dari Yayasan Nala TNI-AL guna menyelesaikan

Program S1 Teknik Elektro di Universitas Hang Tuah Surabaya.

Setelah berhasil menyabet gelar Sarjana Teknik (S.T.), langsung melompat ke Ibu Kota, Jakarta untuk menjalani tugas pertamanya sebagai *Supervisor Maintenance* di PPMI Menara Imperium Jakarta. Keberuntungan membuat penulis menerima bantuan belajar dari perusahaan tempatnya bekerja, untuk menyelesaikan Program S2 Magister Manajemen Produksi/Operasional di Universitas Mercu Buana Jakarta.

Berhasil lulus dengan predikat *cumlaude*, membawanya berkarir di beberapa perusahaan swasta diantaranya, sebagai *Engineer* Mekanikal/Elektrikal di PT. Intiswiss Consulindo dan sebagai *Project Manager* di PT Graha Vitech Elektrindo. Di bidang pendidikan pernah menjabat sebagai *Academic Manager* di LP3I Pati. Saat ini merupakan Instruktur Listrik di UPTD BLK-Dinas Tenaga Kerja Kab. Pati, juga merupakan Asesor Kompetensi di bidang Listrik Domestik, Listrik Industri, *Programable Logic Control (PLC)* dan Teknisi Telepon Seluler. Selain itu, penulis juga menyukai bidang pengkodean seperti Arduino dan IoT.

Beberapa karya yang telah dipublikasikan : Dari Bidadari Merapi Hingga Negeri Paman Sam(2012), Buku Saku Anti Galau (2013), Merancang Listrik Intern(2014), Pedoman Praktis Keselamatan Kerja Kelistrikan(2015), Aktif menulis di Jurnal Engineering Edu (2015-2021), 22 Kaki (2021), Ngleles (2022), Pedoman Praktis Teknisi Handphone Profesional (2023), Jurus Asyik Menjadi Teknisi AC Tersertifikasi (2023).

Aktif di beberapa organisasi keteknikan diantaranya, anggota Persatuan Insinyur Indonesia (PII) dan Dewan Pengawas Perkumpulan Teknisi Refrigerasi dan Tata Udara (PTRTU) DPD Jawa Tengah. Penulis bisa dihubungi melalui :

Email : ng.irsyadmochammad@gmail.com

Ponsel di nomer : 0821-35593898.