

**Belajar Mikrokontroler Arduino
dengan Simulasi Tinkercad**

deepublish / publisher

UU No 28 tahun 2014 tentang Hak Cipta

Fungsi dan sifat hak cipta Pasal 4

Hak Cipta sebagaimana dimaksud dalam Pasal 3 huruf a merupakan hak eksklusif yang terdiri atas hak moral dan hak ekonomi.

Pembatasan Pelindungan Pasal 26

Ketentuan sebagaimana dimaksud dalam Pasal 23, Pasal 24, dan Pasal 25 tidak berlaku terhadap:

- i. Penggunaan kutipan singkat Ciptaan dan/atau produk Hak Terkait untuk pelaporan peristiwa aktual yang ditujukan hanya untuk keperluan penyediaan informasi aktual;
- ii. Penggandaan Ciptaan dan/atau produk Hak Terkait hanya untuk kepentingan penelitian ilmu pengetahuan;
- iii. Penggandaan Ciptaan dan/atau produk Hak Terkait hanya untuk keperluan pengajaran, kecuali pertunjukan dan Fonogram yang telah dilakukan Pengumuman sebagai bahan ajar; dan
- iv. Penggunaan untuk kepentingan pendidikan dan pengembangan ilmu pengetahuan yang memungkinkan suatu Ciptaan dan/atau produk Hak Terkait dapat digunakan tanpa izin Pelaku Pertunjukan, Produser Fonogram, atau Lembaga Penyiaran.

Sanksi Pelanggaran 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 Rp100.000.000 (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 Rp500.000.000,00 (lima ratus juta rupiah).

Belajar Mikrokontroler Arduino dengan Simulasi Tinkercad

**Erwan Eko Prasetyo, S.Pd., M.Eng.
Assist. Prof., Dr., Ir., Iswanto, S.T., M.Eng., IPM.**



Cerdas, Bahagia, Mulia, Lintas Generasi.

BELAJAR MIKROKONTROLER ARDUINO DENGAN SIMULASI TINKERCAD

Erwan Eko Prasetyo & Iswanto

Desain Cover :
Rulie Gunadi

Sumber :
www.shutterstock.com

Tata Letak :
T. Yuliyanti

Proofreader :
Mira Muarifah

Ukuran :
viii, 83 hlm, Uk: 17.5x25 cm

ISBN :
No ISBN

Cetakan Pertama :
Bulan 2023

Hak Cipta 2023, Pada Penulis

Isi diluar tanggung jawab percetakan

Copyright © 2023 by Deepublish Publisher
All Right Reserved

Hak cipta dilindungi undang-undang.
Dilarang keras menerjemahkan, memfotokopi, atau
memperbanyak sebagian atau seluruh isi buku ini
tanpa izin tertulis dari Penerbit.

PENERBIT DEEPUBLISH
(Grup Penerbitan CV BUDI UTAMA)

Anggota IKAPI (076/DIY/2012)

Jl.Rajawali, G. Elang 6, No 3, Drono, Sardonoharjo, Ngaglik, Sleman

Jl.Kaliurang Km.9,3 – Yogyakarta 55581

Telp/Faks: (0274) 4533427

Website: www.deepublish.co.id

www.penerbitdeepublish.com

E-mail: cs@deepublish.co.id

KATA PENGANTAR PENERBIT

Assalamualaikum, w.r. w.b.

Segala puji kami haturkan ke hadirat Allah Swt., Tuhan Yang Maha Esa, atas limpahan rahmat dan karunia-Nya. Tak lupa, lantunan selawat serta salam senantiasa tercurahkan kepada Nabi Besar Muhammad saw.

Dalam rangka mencerdaskan dan memuliakan umat manusia dengan penyediaan serta pemanfaatan ilmu pengetahuan dan teknologi untuk menciptakan industri *processing* berbasis sumber daya alam (SDA) Indonesia, Penerbit Deepublish dengan bangga menerbitkan buku dengan judul ***Belajar Mikrokontroler Arduino dengan Simulasi Tinkercad***.

Terima kasih dan penghargaan terbesar kami sampaikan kepada penulis, Erwan Eko Prasetyo, S.Pd., M.Eng. dan Assist. Prof., Dr., Ir., Iswanto, S.T., M.Eng., IPM yang telah memberikan kepercayaan, perhatian, dan kontribusi penuh demi kesempurnaan buku ini. Semoga buku ini bermanfaat bagi semua pembaca, mampu berkontribusi dalam mencerdaskan dan memuliakan umat manusia, serta mengoptimalkan pemanfaatan ilmu pengetahuan dan teknologi di tanah air.

Wassalamualaikum, w.r. w.b.

Hormat Kami,
Penerbit Deepublish

DAFTAR ISI

KATA PENGANTAR PENERBIT	v
DAFTAR ISI	vi
Bab 1 Pengenalan Mikrokontroler Arduino	1
A. Tujuan	1
B. Alat dan Bahan	1
C. Teori Singkat.....	1
D. <i>Software</i> Arduino IDE.....	10
E. Antarmuka pada Arduino.....	18
F. Sistem Komunikasi Data.....	20
G. Mengenal Kode-kode dalam Pemrograman Dasar Arduino.....	25
H. Menyusun Set Instruksi Pemrograman Arduino.....	31
I. Soal Latihan	35
Bab 2 Simulasi Pemrograman Arduino Menggunakan Tinkercad	37
A. Tujuan	37
B. Alat dan Bahan	37
C. Teori Singkat.....	37
D. Langkah Kerja	38
E. Latihan.....	47
F. Hasil Praktik.....	48
G. Analisis	48
H. Kesimpulan.....	48
Bab 3 Simulasi Arduino Running LED	49
A. Tujuan	49
B. Alat dan Bahan	49
C. Aplikasi Dasar	49
D. Langkah Kerja	50
E. Tugas Praktik	52

F.	Analisis	52
G.	Kesimpulan.....	52
Bab 4	Simulasi Arduino Digital Input.....	53
A.	Tujuan	53
B.	Alat dan Bahan	53
C.	Aplikasi Dasar	53
D.	Langkah Kerja	54
E.	Tugas Praktik.....	56
F.	Analisis	56
G.	Kesimpulan.....	56
Bab 5	Simulasi Arduino Analog Input.....	57
A.	Tujuan	57
B.	Alat dan Bahan	57
C.	Aplikasi Dasar	57
D.	Langkah Kerja	58
E.	Tugas Praktik.....	59
F.	Analisis	60
G.	Kesimpulan.....	60
Bab 6	Simulasi Arduino 7 Segment	61
A.	Tujuan	61
B.	Alat dan Bahan.....	61
C.	Aplikasi Dasar	61
D.	Langkah Kerja.....	62
E.	Tugas Praktik.....	66
F.	Analisis	66
G.	Kesimpulan.....	67
Bab 7	Simulasi Arduino LCD 16x2	68
A.	Tujuan	68
B.	Alat dan Bahan	68
C.	Aplikasi Dasar	68
D.	Langkah Kerja	69
E.	Tugas Praktik.....	70

F. Analisis	71
G. Kesimpulan.....	71
Bab 8 Simulasi Arduino Motor Servo	72
A. Tujuan	72
B. Alat dan Bahan	72
C. Teori dan Aplikasi Dasar.....	72
D. Langkah Kerja	73
E. Tugas Praktik.....	75
F. Analisis	75
G. Kesimpulan.....	75
Bab 9 Simulasi Arduino Pengukur Jarak	77
A. Tujuan	77
B. Alat dan Bahan	77
C. Teori dan Aplikasi Dasar.....	77
D. Langkah Kerja	78
E. Tugas Praktik.....	80
F. Analisis	80
G. Kesimpulan.....	80
Bab 10 Rancang Bangun <i>Embedded System</i> Berbasis Arduino	81
A. Tujuan	81
B. Alat dan Bahan	81
C. Teori dan Aplikasi Dasar.....	81
D. Langkah Kerja	82
E. Tugas Praktik.....	82
F. Analisis	82
G. Kesimpulan.....	82
DAFTAR PUSTAKA.....	83

Bab 1

Pengenalan Mikrokontroler Arduino

A. Tujuan

Praktik ini bertujuan agar praktikan dapat:

1. Mengetahui Arduino dan jenis-jenisnya
2. Mengetahui set instruksi pada pemrograman Arduino
3. Menyusun set instruksi pemrograman Arduino

B. Alat dan Bahan

1. PC atau Laptop
2. *Software* Arduino IDE

C. Teori Singkat

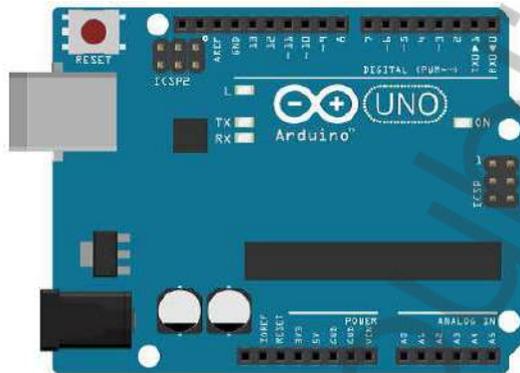
a. Apa itu Mikrokontroler?

Menurut Wikipedia *a microcontroller (sometimes abbreviated μ C, uC or MCU) is a small computer on a single integrated circuit containing a processor core, memory, and programmable input/output peripherals.*

Dalam diskusi sehari-hari, mikrokontroler sering dikenal dengan sebutan μ C, uC, atau MCU. Terjemahan bebas dari pengertian tersebut, bisa dikatakan bahwa mikrokontroler adalah komputer yang berukuran mikro dalam satu *chip* IC (*integrated circuit*) yang terdiri atas *processor*, *memory*, dan *interface* yang bisa diprogram. Jadi disebut komputer mikro karena dalam IC atau *chip* mikrokontroler terdiri atas 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.

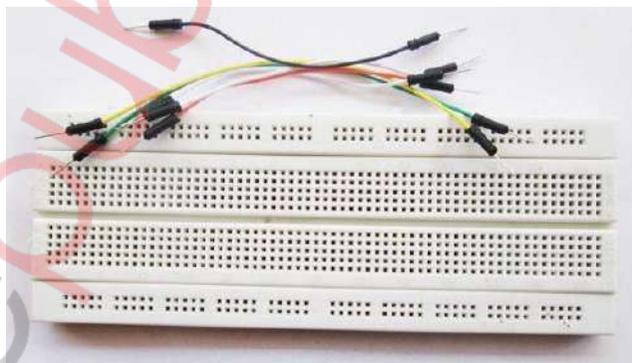
Pada praktik ini kita akan menggunakan *board* Arduino Uno (Gambar 1.1). Arduino *board* terdiri atas *hardware* /modul mikrokontroler yang siap pakai dan *software* IDE yang digunakan untuk memprogram sehingga kita bisa belajar dengan lebih mudah. Kelebihan Arduino yaitu kita tidak perlu lagi

membuat rangkaian minimum sistem dan *programmer* karena sudah *built in* dalam satu *board*. Oleh sebab itu kita bisa fokus ke pengembangan sistem.



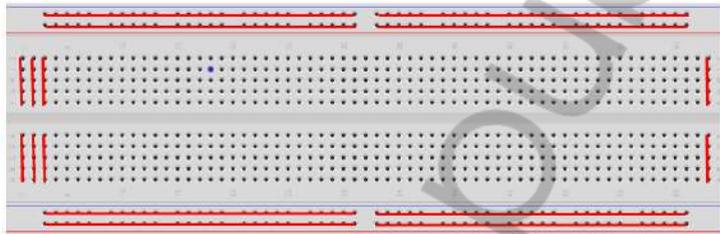
Gambar 1.1. Arduino Uno Board

Pada saat praktik, kita akan menggunakan *project board* (atau disebut dengan istilah *bread board*) dan beberapa kabel *jumper* untuk menghubungkan antara komponen dan Arduino board (Gambar 1.2). Dengan *project board* kita tidak perlu menyolder rangkaian sehingga relatif mudah dan cepat dalam merangkai rangkaian. *Project board* memungkinkan kita untuk membangun dan membongkar rangkaian dengan cepat sehingga sangat cocok untuk eksperimen. Tapi jika kita ingin membuat rangkaian yang permanen, maka kita harus menggunakan PCB. Yang terpenting adalah, kita harus memahami jalur-jalur pada *project board*. *Project board* terdiri atas jalur vertikal dan jalur horizontal.



Gambar 1.2. Project Board dan Kabel Jumper

Jalur vertikal ada di bagian tengah yang terdiri atas 2 x 64 jalur. Masing-masing jalur mempunyai 5 titik vertikal, misal jalur 1A-1B-1C-1D-1E dan jalur 1F-1G-1H-1I-1J yang kedua tidak saling tersambung. Jalur horizontal sebanyak 8 jalur, 4 jalur ada di bagian atas dan 4 jalur lagi di bagian bawah. Jalur ini bisa digunakan untuk jalur *power supply* (VCC dan GND). Untuk lebih jelasnya, silakan perhatikan Gambar 1.3. Garis merah menunjukkan bahwa lubang tersebut terhubung secara fisik.



Gambar 1.3. Peta Jalur pada Project Board

b. Macam-Macam Arduino Board

1. Arduino Uno

Arduino Uno merupakan jenis 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 mikrokontrolernya, memiliki 14 pin I/O digital dan 6 pin input analog. Pemrogramannya cukup menggunakan koneksi USB type A to type B. Sama seperti yang digunakan pada USB printer.



Gambar 1.4 Arduino Uno

2. Arduino Due

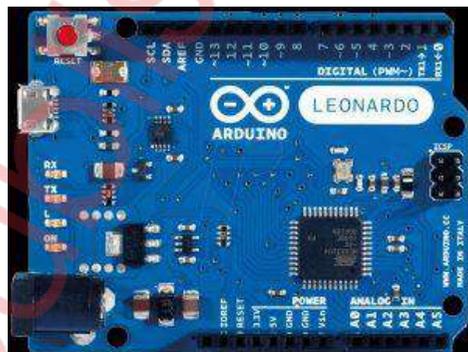
Arduino Due tidak menggunakan mikrokontroler ATMEGA, melainkan dengan *chip* yang lebih tinggi yaitu ARM Cortex CPU. Arduino Due memiliki 54 I/O pin digital dan 12 pin input analog. Pemrogramannya menggunakan Micro USB.



Gambar 1.5 Arduino Due

3. Arduino Leonardo

Arduino Leonardo hampir sama dengan Arduino Uno. Jumlah pin I/O digital dan pin input analognya sama. Perbedaannya hanya Arduino Leonardo menggunakan Micro USB untuk pemrogramannya.



Gambar 1.6 Arduino Leonardo

4. Arduino Mega 2560

Arduino Mega 2560 adalah *board* Arduino yang merupakan perbaikan dari *board* Arduino Mega sebelumnya. Arduino Mega awalnya

memakai *chip* ATmega1280 dan kemudian diganti dengan *chip* ATmega2560, oleh karena itu namanya diganti menjadi Arduino Mega 2560. Selain perbedaan *chip* ATmega yang digunakan, perbedaan lain antara Arduino Mega dengan Arduino Mega 2560 adalah tidak lagi menggunakan *chip* FTDI untuk fungsi *USB to serial converter*, melainkan menggunakan *chip* ATmega16u2. Secara fisik, ukuran Arduino Mega 2560 kurang lebih 2 kali lebih besar dari Arduino Uno, ini untuk mengakomodasi lebih banyaknya pin Digital dan Analog pada *board* Arduino Mega 2560 tersebut. Tampilan Arduino Mega 2560 dapat dilihat pada gambar di bawah ini.



Gambar 1.7 Arduino Mega 2560

5. Arduino Intel Galileo

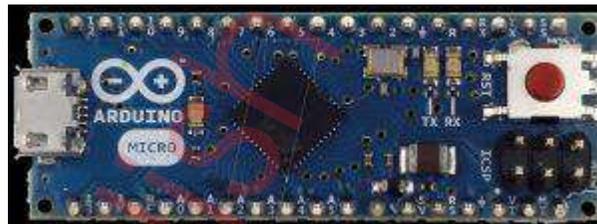
Arduino Intel Galileo adalah *board* mikrokontroler yang menggunakan Intel Quark SoC X1000 Application Processor, 32-bit sistem Pentium kelas Intel. Arduino jenis ini merupakan *board* pertama menggunakan arsitektur Intel. *Board* ini dirancang untuk menjadi *hardware* dan *software* yang kompatibel dengan Arduino Uno R3. Arduino Intel Galileo dirancang untuk mendukung *shield* yang beroperasi di kedua tegangan 3.3V atau 5V. Tegangan operasi Arduino Intel Galileo adalah 3.3V. Namun, *jumper* di *board* memungkinkan konversi tegangan 5V di pin I/O. Hal ini mendukung untuk tegangan 5V yang digunakan pada *shield* Arduino Uno. Dengan memindah posisi *jumper*, konversi tegangan dapat dinonaktifkan untuk menyediakan operasi 3.3V di pin I/O.



Gambar 1.8 Arduino Intel Galileo

6. Arduino Pro Mikro

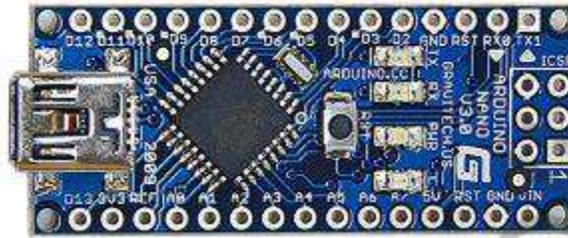
Arduino Mikro adalah *board* mikrokontroler dengan mikrokontroler ATmega32u4 (lihat *datasheet*) yang dikembangkan bersama dengan Adafruit. Arduino Pro Mikro memiliki 20 digital pin input/ output (yang dapat digunakan sebagai output PWM dan 12 input analog), osilator 16 MHz kristal, koneksi USB mikro, *header* ICSP, dan tombol reset.



Gambar 1.9 Arduino Pro Mikro

7. Arduino Nano R3

Arduino Nano adalah salah satu varian produk *board* mikrokontroler keluaran Arduino. Arduino Nano adalah *board* Arduino terkecil, menggunakan mikrokontroler Atmega 328 untuk Arduino Nano 3.x dan Atmega168 untuk Arduino Nano 2.x. Varian ini mempunyai rangkaian yang sama dengan jenis Arduino Duemilanove, tetapi dengan ukuran dan desain PCB yang berbeda. Arduino Nano tidak dilengkapi dengan soket catu daya, tetapi terdapat pin untuk catu daya luar atau dapat menggunakan catu daya dari mini USB port. Arduino Nano didesain dan diproduksi oleh Gravitech.



Gambar 1.10 Arduino Nano R3

8. Arduino Pro Mini Atmega

Arduino Pro Mini merupakan salah satu produk dari Arduino yang menjadi favorit, dengan ukuran dimensi yang kecil membuat modul mikrokontroler ini terlihat sangat praktis. Meskipun dengan ukuran yang kecil namun fitur-fitur yang ada di Arduino Pro Mini ini tak kalah dengan jenis-jenis mikrokontroler Arduino yang lainnya. Tentu dengan harga yang murah membuat modul ini sering digunakan di berbagai peranti-peranti cerdas.



Gambar 1.11 Arduino Pro Mini

9. Arduino Mega ADK

Board Arduino biasa (Uno, Duemillanove, Mega, dll.) pada dasarnya sama dengan *board* Arduino ADK (Android Development Kit) yakni di dalamnya terdapat *bootloader* Arduino. Namun kelebihan dari *board* ADK adalah *board* ini telah memiliki port USB tersendiri sehingga membuat *board* menjadi lebih simpel dan kecil. Sebuah *board* Arduino Mega dapat diubah menjadi Arduino ADK dengan menambahkan *shield* USB pada *board*-nya.

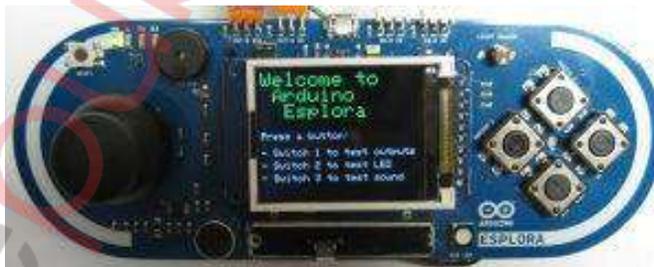
Arduino ADK merupakan *board* mikrokontroler yang dikhususkan untuk berkomunikasi dengan *smartphone* Android via komunikasi USB. *Board* mikrokontroler berfungsi sebagai induk dari *smartphone* Android dengan berfungsi seolah-olah adalah komputer. *Smartphone* Android nantinya akan memanfaatkan mode USB *debugging* saat berkomunikasi dengan *board* mikrokontroler.



Gambar 1.12 Arduino Mega ADK

10. Arduino Esplora

Arduino Esplora adalah Mikrokontroler Arduino yang sudah dilengkapi dengan sensor dan *joystick*. Berbeda dengan *board* Arduino sebelumnya yang hanya berisi mikrokontroler. Arduino Esplora melangkah lebih jauh lagi. Arduino Esplora terdiri atas mikrokontroler dan banyak sensor, antara lain: sensor suhu, sensor *accelerometer* dan sensor cahaya. Selain itu Arduino Esplora juga dilengkapi dengan tombol *joystick*, linier *potentiometer*, 4 *push button*, *microphone* dan LED RGB.



Gambar 1.13 Arduino Esplora

11. Arduino Serial

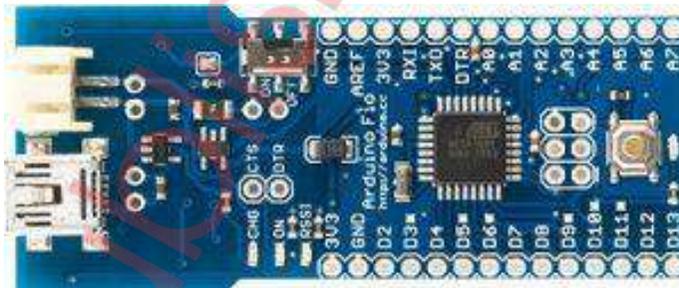
Arduino Serial, yaitu jenis mikrokontroler Arduino yang menggunakan RS232 sebagai antarmuka pemrograman atau komunikasi komputer.



Gambar 1.14 Arduino Serial

12. Arduino Fio

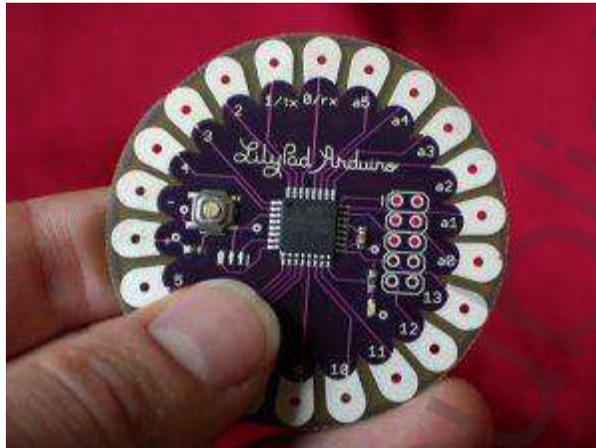
Arduino Fio merupakan mikrokontroler Arduino yang ditujukan untuk penggunaan nirkabel. Arduino Fio ini menggunakan ATmega328P sebagai kontrolernya.



Gambar 1.15 Arduino Fio

13. Arduino Lilypad

Arduino Lilypad adalah mikrokontroler dengan bentuk yang melingkar. Contoh: LilyPad Arduino 00, LilyPad Arduino 01, LilyPad Arduino 02, LilyPad Arduino 03, LilyPad Arduino 04.



Gambar 1.16 Arduino Lilypad

14. Arduino BT

Arduino BT adalah mikrokontroler Arduino yang sudah memiliki modul Bluetooth untuk komunikasi nirkabel.



Gambar 1.17 Arduino BT

D. *Software* Arduino IDE

Software Arduino IDE dapat diunduh secara gratis di laman resmi Arduino yaitu di <https://www.arduino.cc/en/Main/Software>. *Software* Arduino versinya semakin berkembang. Sampai saat modul ini disusun, Arduino IDE sudah sampai versi 2.0.1. Berdasarkan informasi dari laman

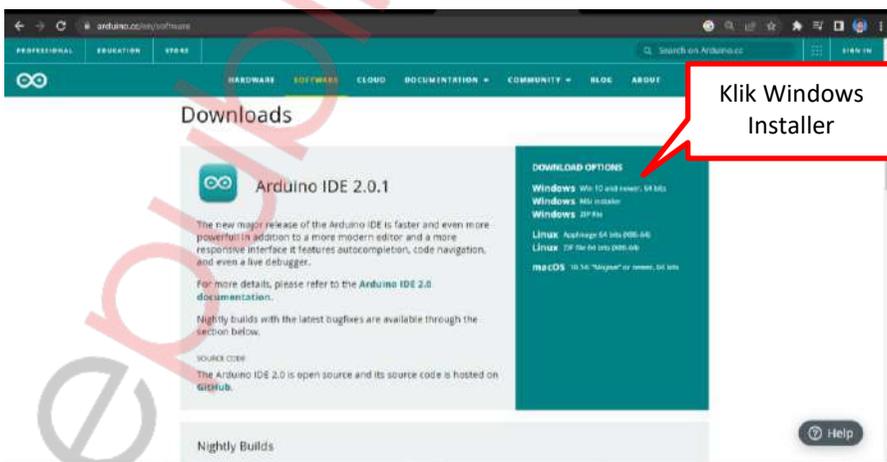
resmi Arduino, Arduino menyediakan *software* Arduino IDE 2.0.1 yang dapat diinstal pada perangkat dengan sistem operasi Windows, Mac dan Linux.



Gambar 1.18. Tampilan Laman Download *Software* Arduino IDE

Berikut ini adalah langkah-langkah mengunduh *software* Arduino IDE dari laman resminya.

1. Buka *browser* di komputer atau laptop Anda. Pastikan sudah terhubung dengan koneksi internet.
2. Buka laman https://www.arduino.cc/en/Main/Software_
3. Klik pilihan Windows Installer seperti pada gambar berikut.



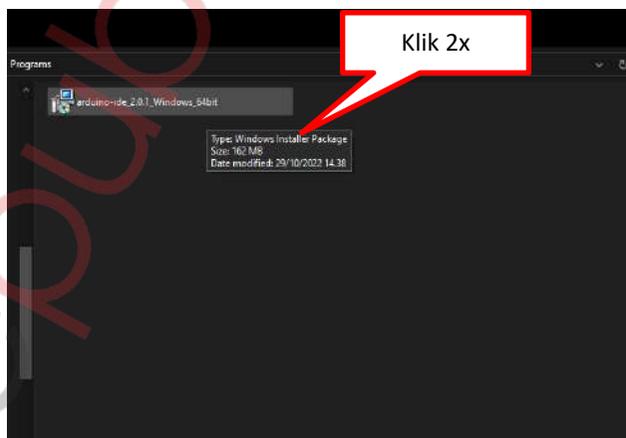
4. Klik pilihan “Just Download” untuk memilih pilihan mengunduh secara gratis atau klik “Contribute & Download” jika Anda akan memberi donasi.



5. Tunggu proses unduh sampai selesai.
6. Simpan *file* hasil unduhan pada folder Anda.

Pada modul ini akan dijelaskan bagaimana cara melakukan instalasi pada sistem operasi Windows. Langkah-langkah instalasi *software* Arduino IDE 2.0.1 dijelaskan sebagai berikut:

1. Buka folder tempat Anda menyimpan *software* Arduino IDE 2.0.1 hasil unduhan Anda.
2. Klik 2x *file* arduino-2.0.1-windows.exe untuk mulai menginstal.



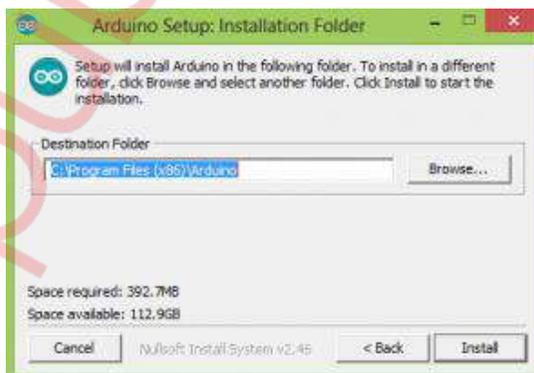
3. Klik "Yes" pada menu opsi.



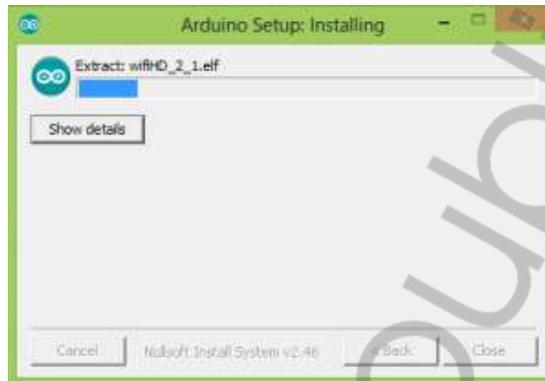
4. Klik "I Agree" untuk menyetujui lisensi dan melanjutkan instalasi.



5. Instal semua komponen yang ada.



- Pilih lokasi folder instalasi *Arduino Software* atau gunakan *default destination folder*, dan klik *Install*.



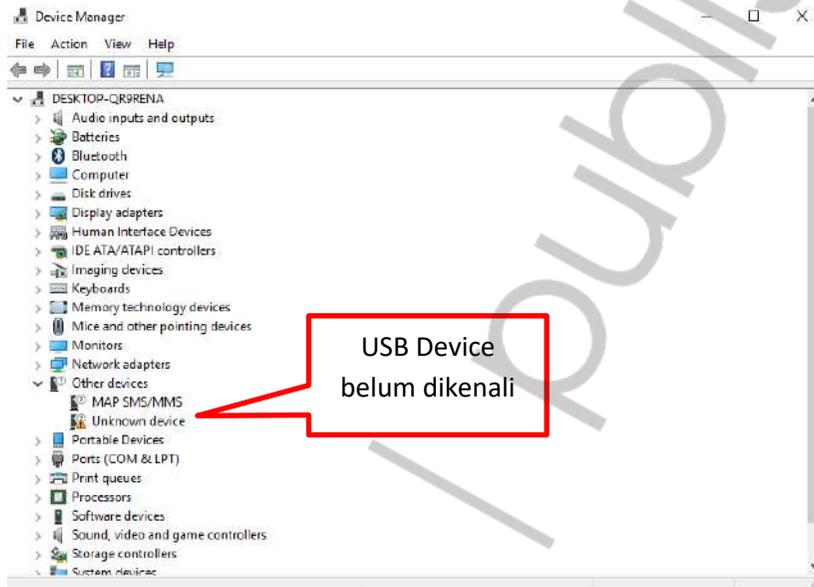
- Tunggu proses instalasi sampai selesai.
- Klik "Close" untuk menutup *dialog windows*.

Setelah semua langkah di atas selesai dilakukan, maka *software* Arduino IDE sudah terinstal pada komputer/laptop Anda. Langkah selanjutnya adalah melakukan proses instalasi *driver Arduino Board* sesuai yang akan digunakan (pada modul ini akan menggunakan Arduino Uno). Langkah-langkah melakukan instalasi *driver Arduino board* sebagai berikut:

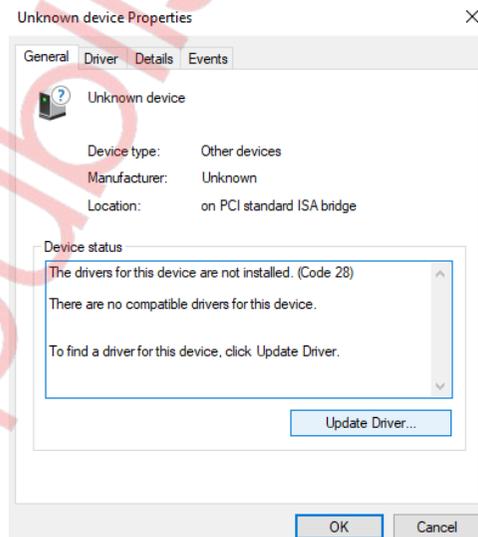
- Hubungkan Arduino Uno *board* pada perangkat komputer/laptop Anda dengan kabel USB.



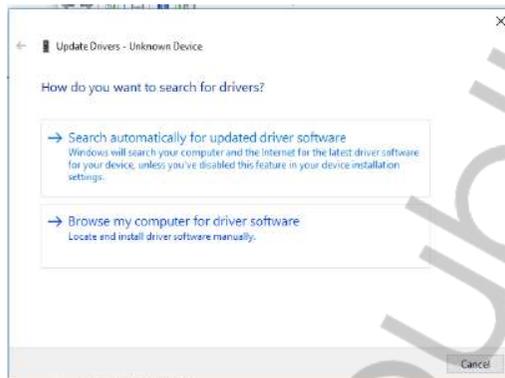
2. Cek pada bagian “Device Manager”. Apakah Arduino Uno *board* sudah dikenali atau belum. Jika belum lakukanlah instalasi *driver* sesuai jenis *chip* modul *USB to serial* yang dipakai.



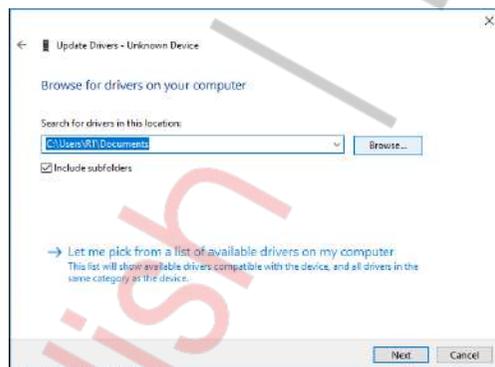
3. Instalasi *driver* Arduino *board* dapat dilakukan dengan cara Klik 2x pada daftar “Unknown device”.



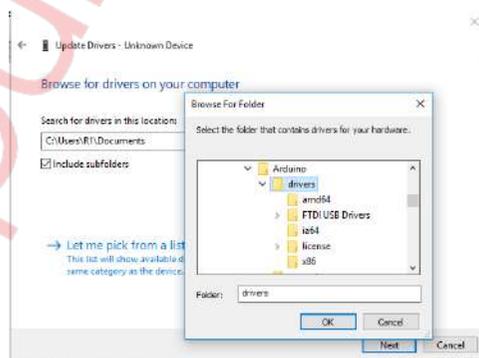
4. Klik "Update Driver".



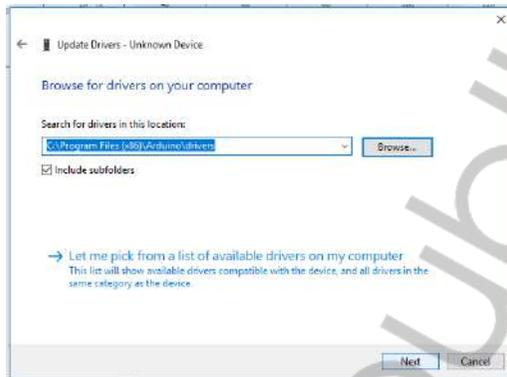
5. Klik "Browse my computer for driver software".



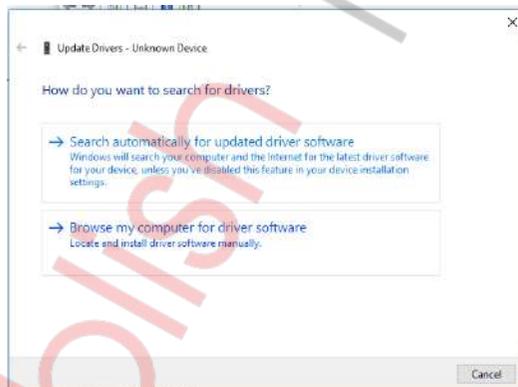
6. Klik "Browse" kemudian arahkan pada folder tempat Anda menginstal Arduino IDE.



7. Pilih folder “Arduino > Drivers” lalu klik “OK”.



8. Klik “Next” dan tunggu sampai proses instalasi *driver* selesai.
9. Jika belum berhasil, maka proses instalasi *driver* dapat dilakukan secara otomatis.



10. Instalasi otomatis dapat dilakukan secara *online* dan harus ada dukungan koneksi internet. Instalasi otomatis dapat dilakukan dengan klik pilihan “Search automatically for updated driver software”.
11. Tunggu proses instalasi sampai selesai.

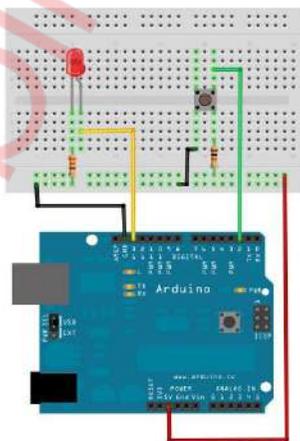
Setelah proses instalasi *driver* selesai, maka dapat dilakukan pengecekan pada Device Manager. Jika berhasil maka Arduino Uno *board* dapat terdeteksi dan dikenali pada Port COM seperti ditunjukkan pada Gambar berikut.



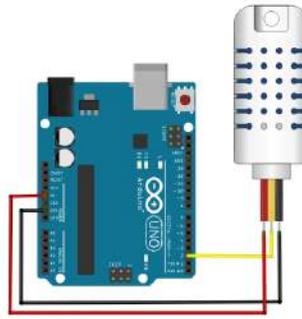
E. Antarmuka pada Arduino

Pada sistem Arduino, komunikasi antarmuka lebih banyak diterapkan antara perangkat satu dengan perangkat yang lainnya. Contohnya adalah antara Arduino dengan sensor, Arduino dengan tombol, Arduino dengan papan *display*/layar monitor, Arduino dengan *actuator*/motor, Arduino dengan unit *memory* atau penyimpanan data, maupun antara Arduino dengan Arduino yang lain.

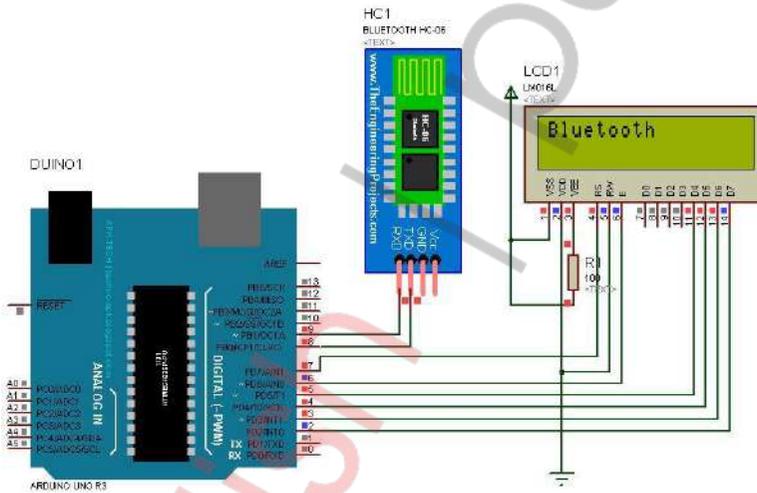
Tujuan komunikasi ini adalah untuk saling berkirim maupun menerima data antarperangkat yang satu dengan perangkat yang lainnya. Berikut ini beberapa contoh antarmuka pada Arduino.



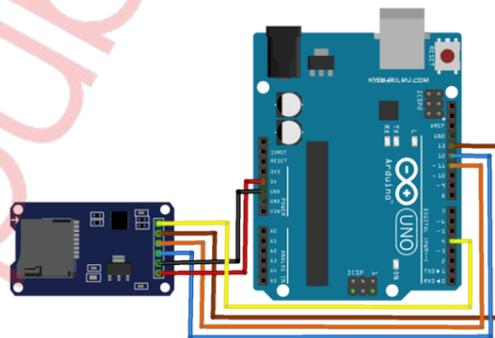
Gambar Contoh Antarmuka Arduino dengan Tombol dan LED



Gambar Contoh Antarmuka Arduino dengan Sensor Suhu DHT-11



Gambar Contoh Antarmuka antara Arduino dengan *Display* LCD 16x2 dan Bluetooth



Gambar Contoh Antarmuka antara Arduino dengan Kartu Memori (MicroSD)

F. Sistem Komunikasi Data

Antarmuka Arduino dengan perangkat lain membutuhkan sistem komunikasi data, agar perangkat dapat terhubung dan saling berkomunikasi dengan Arduino. Sistem komunikasi data ini harus di-*setting* pada saat kita melakukan pemrograman. Agar perangkat-perangkat di luar Arduino dapat saling berkomunikasi dengan baik, maka sistem komunikasi data harus diatur dengan tepat. Sebelum melakukan pengaturan, kita harus mengetahui jenis komunikasi data yang digunakan. Arduino menggunakan sistem komunikasi data serial. Ciri-ciri komunikasi Serial antara lain:

- Beberapa bit dikirim melalui jalur data yang sama secara bergantian
- Selain jalur data dapat menggunakan juga sinyal sinkronisasi (*clock*)

Berikut ini dijelaskan beberapa jenis komunikasi data serial yang sering digunakan pada Arduino, antara lain:

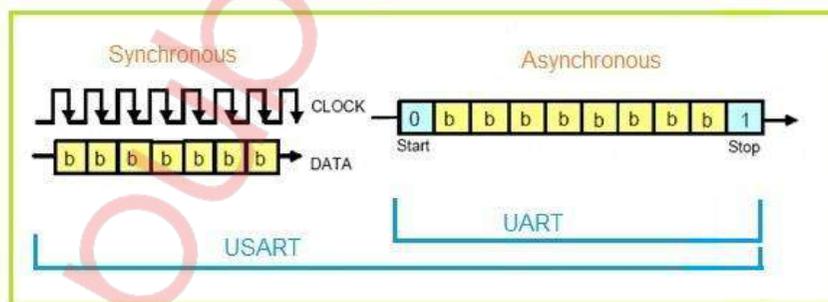
1. UART (*Universal Asynchronous Receiver-Transmitter*)

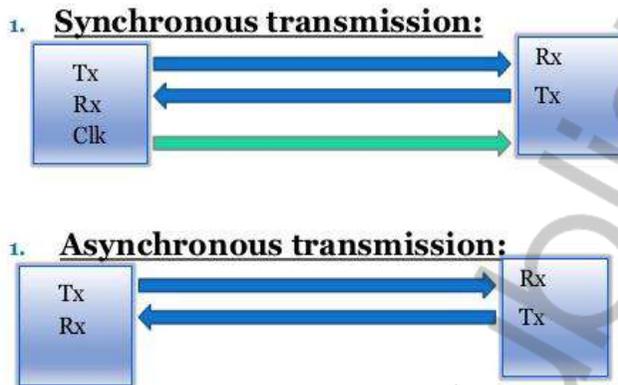
UART biasanya berupa sirkuit terintegrasi yang digunakan untuk komunikasi serial pada komputer atau port serial perangkat *peripheral*.

2. USART (*Universal Synchronous & Asynchronous Receiver Transmitter*)

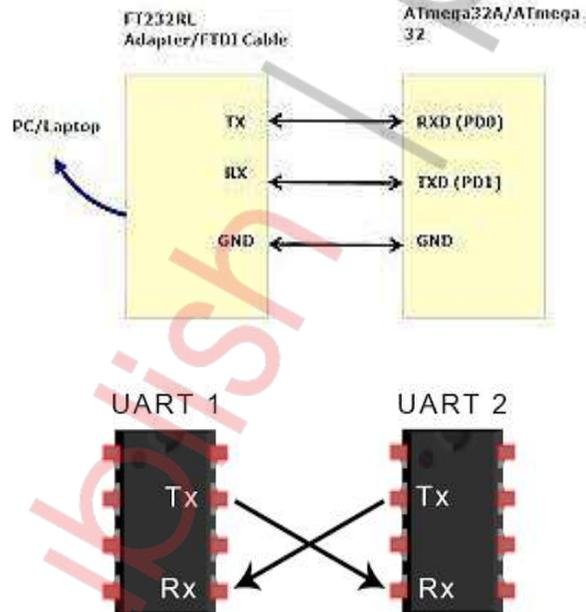
Komunikasi yang memiliki fleksibilitas tinggi, yang dapat digunakan untuk melakukan transfer data antarmikrokontroler maupun dengan modul-modul eksternal termasuk PC yang memiliki fitur UART.

USART memungkinkan transmisi data baik secara *synchronous* maupun *asynchronous*. Berikut ini perbedaan antara UART dengan USART.





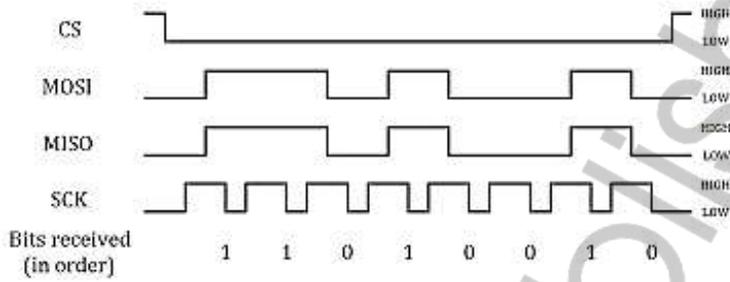
Gambar Perbedaan antara UART dengan USART



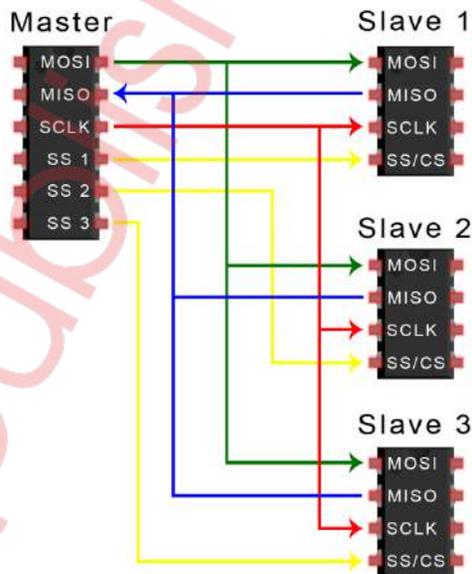
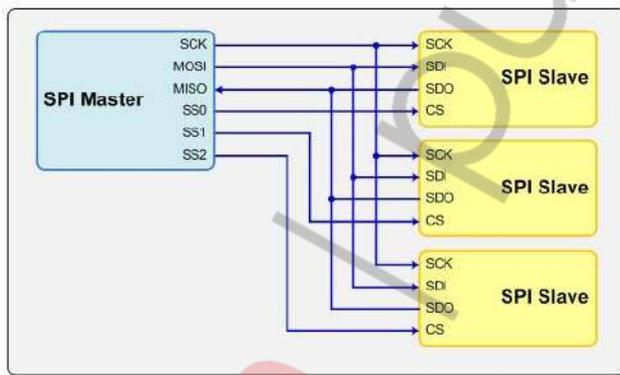
Gambar Contoh Diagram Komunikasi Data

3. SPI (*Serial Peripheral Interface*)

Komunikasi SPI membutuhkan 4 jalur yaitu MOSI, MISO, SCK dan CS. Data dapat saling dikirimkan antarmikrokontroler maupun antara mikrokontroler dengan *peripheral* lain di luar mikrokontroler. Transmisi Data pada Sistem SPI ditunjukkan seperti pada gambar berikut.



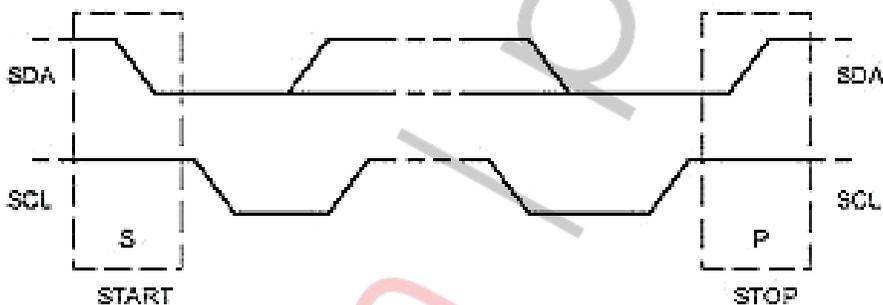
Gambar Transmisi Data pada Sistem SPI



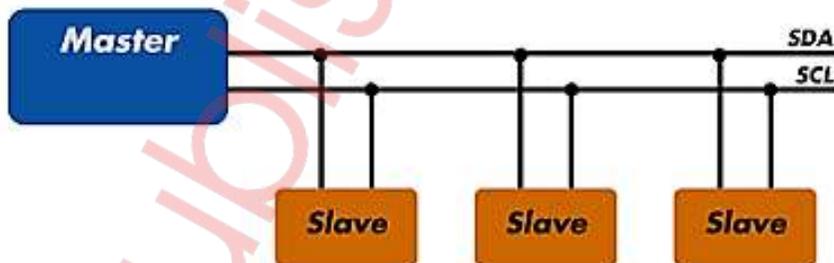
Gambar Contoh Sistem Komunikasi dengan SPI

4. I2C (*Inter Integrated Circuit*)

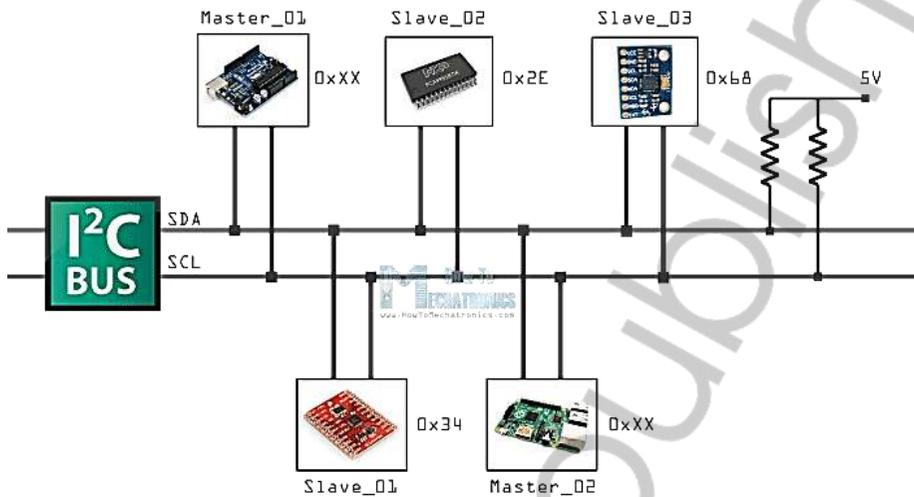
I2C merupakan sistem komunikasi serial dua arah menggunakan dua saluran yang didesain khusus untuk mengirim maupun menerima data. Sistem komunikasi ini terdiri atas saluran SCL (*Serial Clock*) dan SDA (*Serial Data*) yang membawa informasi data antara I2C dengan pengontrolnya. Peranti yang dihubungkan dengan sistem I2C Bus dapat dioperasikan sebagai Master dan Slave. Master adalah peranti yang memulai transfer data pada I2C Bus dengan membentuk sinyal Start, mengakhiri transfer data dengan membentuk sinyal Stop, dan membangkitkan sinyal *clock*. Slave adalah peranti yang dialamati Master.



Gambar Contoh Transmisi Data pada I2C



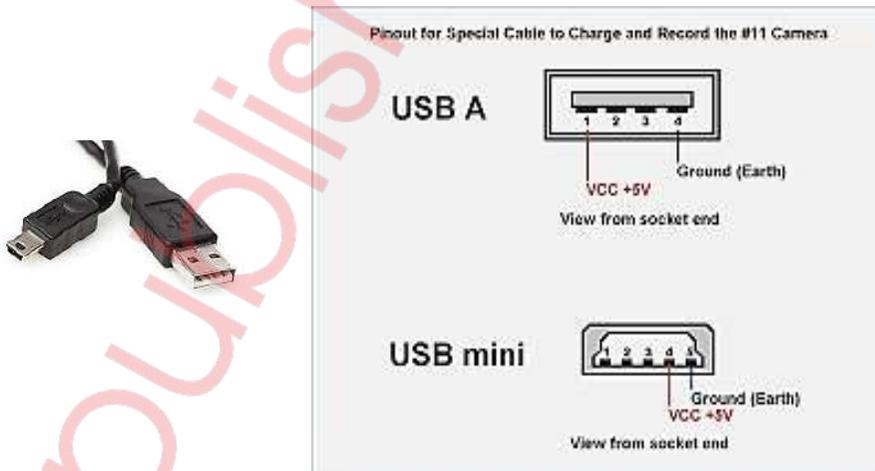
Gambar Contoh Diagram I2C



Gambar Contoh Penggunaan Komunikasi I2C

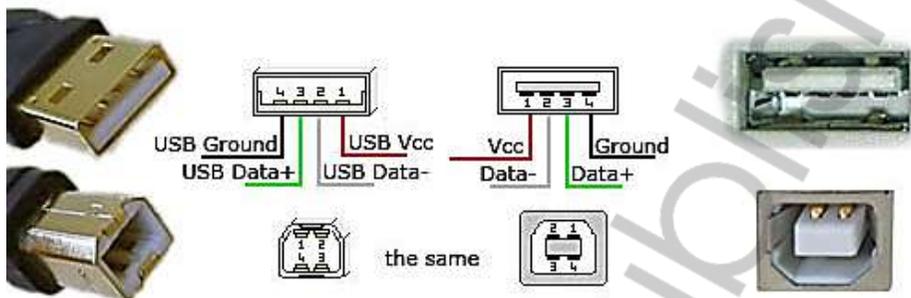
5. USB (Universal Serial Bus)

Standar bus serial untuk perangkat penghubung, biasanya untuk komputer namun juga digunakan di peralatan lainnya seperti konsol permainan, ponsel dan perangkat lainnya.



Gambar Contoh USB dan Konfigurasi Pin-nya

USB pinout



USB is a serial bus. It uses 4 shielded wires: two for power (+5v & GND) and two for differential data signals (labelled as D+ and D- in pinout)

http://pinouts.ru/Slots/USB_pinout.shtml

Gambar USB Pin Out

G. Mengetahui Kode-kode dalam Pemrograman Dasar Arduino

1. Bare Minimum (Bagian Minimal Sebuah Program pada Arduino)

Bagian ini merupakan bagian minimal yang harus ada pada sebuah program. Arduino. Bagian program minimal harus memiliki **void setup()** dan **void loop()**.

Contoh:

```
void setup() {  
  // semua kode yang di sini akan dibaca sekali oleh Arduino  
}  
void loop() {  
  // semua kode yang ada di sini akan dibaca  
  // berulang kali (terus menerus) oleh Arduino  
}
```

2. Void Setup

Semua kode program yang ada dalam **void setup** akan dibaca sekali oleh Arduino. Void setup berisi kode perintah untuk menentukan fungsi pada sebuah pin.

Contoh:

```
pinMode(13, OUTPUT); // menentukan pin 13 sebagai OUTPUT  
pinMode(3, INPUT); // menentukan pin 3 sebagai INPUT
```

Komunikasi antara Arduino dengan komputer dapat dibuat dengan menggunakan kode:

```
Serial.begin(9600); // untuk komunikasi Arduino dengan komputer
```

3. Void Loop

Semua kode program yang berada pada **void loop** akan dibaca setelah void setup. Kode ini akan dibaca terus menerus oleh Arduino. Isinya berupa kode-kode perintah kepada pin INPUT dan OUTPUT pada Arduino.

Contoh:

```
digitalWrite(13, HIGH); //untuk memberikan 5V (nyala) kepada pin 13.  
digitalWrite(13, LOW); //untuk memberikan 0V (mati) kepada pin 13.  
analogWrite(3, 225); //untuk memberikan nilai 225 (setara dengan 5V) kepada pin 3.
```

sedangkan untuk menampilkan nilai pada sebuah sensor pada Serial Monitor, bisa menggunakan:

```
Serial.print(namasensor);  
//menampilkan nilai sensor yang disimpan di variabel nama sensor
```

untuk menampilkan teks, bisa menggunakan:

```
Serial.print("Selamat Datang");  
//menampilkan teks Selamat Datang pada Serial Monitor
```

untuk membuka Serial Monitor pada Arduino, bisa dengan memilih menu **Tools** kemudian pilih **Serial Monitor**, atau dengan menekan kombinasi **CTRL+SHIFT+M** di *keyboard*. Cara lain bisa juga dengan mengklik ikon “Kaca Pembesar” di Arduino IDE, seperti gambar berikut.



Gambar Serial Monitor pada Arduino IDE

4. Catatan pada Kode Program

Anda dapat membuat catatan pada program dan tidak akan dibaca oleh Arduino, dengan cara mengetikkan tanda // kemudian mengetikkan catatannya.

Contoh:

```
void loop() {  
  // catatan pada baris ini tidak akan dibaca oleh program  
}
```

Pemakaian tanda // hanya berfungsi untuk catatan satu baris saja, jika Anda ingin membuat catatan yang panjang yaitu berupa paragraf. Maka pertama kamu ketikkan /* lalu ketikkan catatan kamu, dan jika sudah selesai tutup dengan kode */.

Contoh:

```
void loop() {  
  /* apapun yang kamu mau ketikkan di sini tidak  
  akan dibaca oleh program  
  sepanjang apapun kamu mengetiknya  
  */  
}
```

5. Kurung Kurawal {}

Kurung kurawal digunakan untuk menentukan awal dan akhir dari program. Seperti bahasa pemrograman pada umumnya, Arduino membaca mulai dari atas hingga ke bawah.

Contoh:

```
void loop()  
{  
  ....program  
  ....program  
  ....program  
}
```

6. Titik Koma ;

Tanda titik koma digunakan untuk mengakhiri suatu baris kode program. Setiap baris kode program pada Arduino harus diakhiri dengan tanda “;”

Contoh:

```
void setup(){  
  pinMode(13, OUTPUT);  
}  
void loop(){  
  digitalWrite(13, HIGH);  
}
```

7. Variables

Variabel adalah kode program yang digunakan untuk menyimpan data atau suatu nilai pada sebuah nama. Tipe data yang biasa digunakan di antaranya adalah Integer, Long, Boolean, Float, Character.

int (integer)

Variabel yang dapat menyimpan data sebesar 2 *bytes* (16 bits).

long (long)

Variabel ini nilai datanya lebih besar dari integer yaitu menggunakan 4 *bytes* (32 bits).

boolean (boolean)

Variabel yang hanya menyimpan nilai TRUE dan FALSE saja. Hanya menggunakan 1 bit saja.

float(float)

Variabel yang digunakan untuk *floating point* pada nilai desimal. *Memory* yang digunakan sebesar 4 *bytes* (32 bits).

char(character)

Variabel untuk menyimpan *character* berdasarkan ASCII kode (contoh: 'A'=65) menggunakan 1 *byte* (8 bits).

8. Operator Matematika

Operator matematika digunakan untuk memanipulasi nilai dengan perhitungan matematika sederhana seperti: penjumlahan, pengurangan, perkalian, pembagian sama dengan dan persentase.

Contoh:

= (sama dengan)
% (persentase)
+ (penambahan)
- (pengurangan)
* (perkalian)
/ (pembagian)

9. Operator Perbandingan

Operator ini digunakan untuk melakukan perbandingan secara logika.

Contoh:

== (sama dengan) contoh: **15 == 10 FALSE** atau **15 == 15 TRUE**
!= (tidak sama dengan) contoh: **15 != 10 TRUE** atau **15 != 15 FALSE**
< (lebih kecil dari) contoh: **15 < 10 FALSE** atau **12 < 14 TRUE**
> (lebih besar dari) contoh: **15 > 19 TRUE** atau **15 > 10 FALSE**

10. Struktur Pengendali

Program yang digunakan untuk menentukan sebuah kondisi, dan jika kondisinya terpenuhi maka akan melaksanakan perintah yang sudah ditentukan. Sebaliknya, saat tidak memenuhi kondisinya juga ada perintah yang dilaksanakan oleh Arduino.

Contoh:

```
if(kondisi A)
{
Kode Perintah A
}
else if(kondisi B)
{
Kode Perintah B
}
else
{
Kode Perintah C
}
```

Pertama, Arduino akan melihat **Kondisi A**. Jika terpenuhi, maka akan melaksanakan **Kode Perintah A**. Apabila kondisi TIDAK terpenuhi, maka Arduino akan melihat **Kondisi B**. Jika terpenuhi, maka akan melaksanakan

Kode Perintah B. Namun jika kondisi TIDAK juga terpenuhi, maka Arduino akan melaksanakan **Kode Perintah C.**

11. Kode Perintah untuk Mengatur Pin pada Arduino

Kode ini digunakan untuk pemrograman yang menggunakan Pin Digital pada Arduino. Bentuk penulisannya seperti berikut:

```
pinMode( pin, mode);
```

Kode ini digunakan untuk seting mode pin. **Pin** adalah nomor pin yang akan digunakan, jika Anda menggunakan Arduino Uno, pin digitalnya dari 0-13, sedangkan **mode** berupa INPUT atau OUTPUT. Penulisan kode **pinMode** harus berada di dalam **void**.

Contoh:

```
pinMode(13, OUTPUT); // artinya pin 13 digunakan sebagai
OUTPUT
pinMode(7, INPUT); // artinya pin 7 digunakan sebagai
INPUT
digitalRead(pin);
```

Kode ini digunakan pada pin **INPUT**, untuk membaca nilai sensor yang ada pada pin. Nilai pembacaannya hanya terbatas pada **1 (TRUE)**, atau **0 (FALSE)**. Penulisan kode **digitalRead** harus dimasukkan dalam **void**.

Contoh:

```
digitalRead(13); // artinya kode akan membaca nilai sensor
pada pin 13.
digitalWrite(pin, nilai);
```

Kode ini digunakan untuk pin **OUTPUT** yang sudah Anda seting apakah akan diberikan kondisi **HIGH (+5V)**, atau **LOW (Ground)**. Penulisan kode harus dimasukkan dalam **void**.

Contoh:

```
digitalWrite(13, HIGH); // artinya pin 13 kita diberi
tegangan +5V
digitalWrite(13, LOW); // artinya pin 13 kita diberi
tegangan 0 / Ground
analogWrite(pin, nilai);
```

Meskipun Arduino adalah perangkat digital, namun masih bisa menggunakan fungsi Analognya pada pin Digital Arduino meskipun hanya beberapa pin saja. Fungsi ini biasa disebut PWM (*Pulse With Modulation*). Pada Arduino Uno memiliki 6 pin PWM, yaitu: pin 3,5,6,9, 10, dan 11. Nilai yang dihasilkan menjadi bervariasi dari 0-225, yang setara dengan tegangan antara 0-5V. Penulisan kode analogWrite juga harus dimasukkan dalam void. Contoh:

```
        analogWrite(3, 150); // artinya pin 3 diberikan nilai
        sebesar 150
    analogRead(pin);
```

Kode ini digunakan untuk membaca nilai pada sensor Analog antara 0-1024.

Contoh:

```
    analogRead(A0); // artinya kode akan membaca nilai sensor pada
    pin AO.
```

Kode dalam Arduino adalah *Case Sensitive* sehingga penggunaan huruf kecil atau huruf besar sangat berpengaruh.

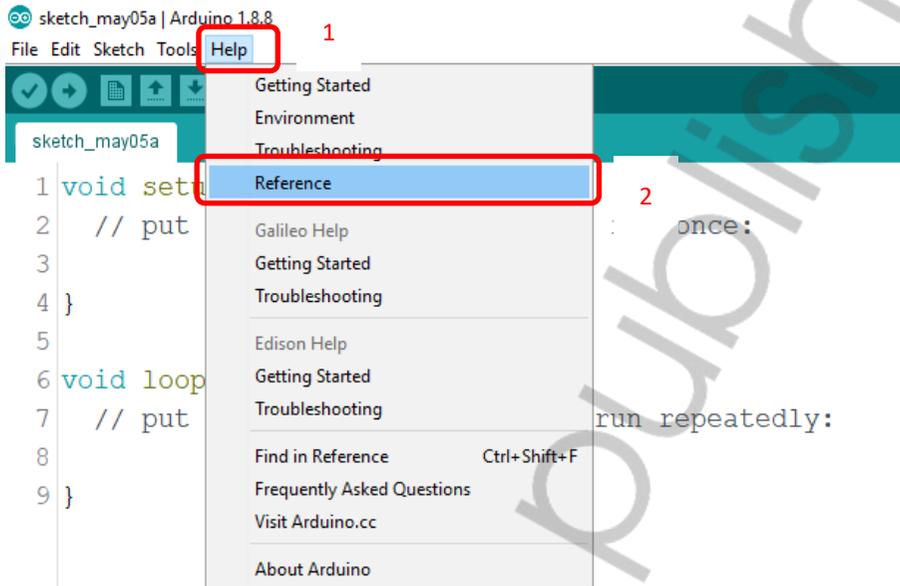
Sumber: <https://kelasrobot.com/belajar-pemrograman-dasar-arduino/>

H. Menyusun Set Instruksi Pemrograman Arduino

Pada dasarnya memrogram Arduino adalah menyusun kode-kode program/set instruksi untuk memerintahkan Arduino melakukan suatu perintah. Perintah ini dibuat atas dasar pemikiran manusia/*programmer*. Berikut ini akan dijelaskan beberapa set instruksi Arduino yang telah ditentukan dalam aplikasi Arduino IDE.

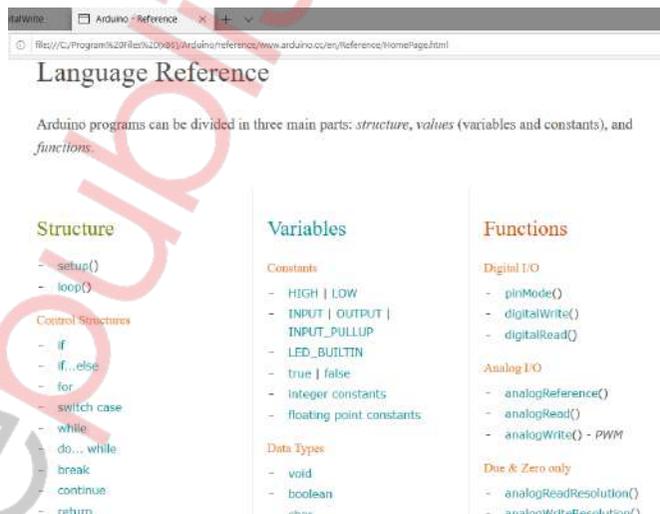
Anda juga dapat mengakses keterangan tentang set-set instruksi yang telah disediakan oleh Arduino IDE melalui cara berikut:

1. Buka Aplikasi Arduino IDE.
2. Klik Help >> Reference



Gambar Tampilan Menu Help pada Arduino IDE

3. Kemudian Anda akan diarahkan masuk ke menu Language Reference. Di sini Anda dapat memilih set instruksi yang ingin Anda pelajari mengenai fungsi dan *syntax* (bentuk) penulisannya beserta contoh penulisannya.



Gambar Tampilan Language Reference

Pada pemrograman dasar sebagai contoh pada program LED Blink (untuk membuat LED berkedip-kedip) Anda akan mengaktifkan fungsi Digital Output. Sehingga dalam membuat program Anda akan memberikan perintah menggunakan set instruksi Digital I/O. Berikut ini Contoh Program Sederhana LED Blink.

```
//Contoh Program LED Blink
void setup() {
  pinMode(13, OUTPUT);
}

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

Pada program LED Blink kita dapat menemukan set instruksi sebagai berikut:

1. void setup();
2. void loop();
3. pinMode(13, OUTPUT);
4. digitalWrite(13, HIGH);
5. delay(1000);

Set instruksi tersebut dapat kita ketahui fungsinya berdasarkan Language Reference yang disediakan Arduino IDE pada menu **Help>>Reference**. Anda dapat membuat penjelasan singkat dari contoh program sederhana LED Blink dengan cara melihat keterangan pada Language Reference. Sebagai contoh Anda akan melihat penjelasan set instruksi **pinMode()** maka Anda dapat melakukannya dengan cara sebagai berikut:

1. Anda klik pinMode (), kemudian Anda dapat melihat penjelasannya.

Functions

Digital I/O

- pinMode()
- digitalWrite()
- digitalRead()

2. Penjelasan set instruksi **pinMode()**

pinMode() berfungsi untuk mengonfigurasi/mengatur pin digital Arduino sebagai Input atau Output. Perlu diketahui bahwa pin digital yang ada pada Arduino dapat digunakan sebagai Input ataupun Output.

pinMode()

Description

Configures the specified pin to behave either as an input or an output. See the description of [digital pins](#) for details on the functionality of the pins.

As of Arduino 1.0.1, it is possible to enable the internal pullup resistors with the mode `INPUT_PULLUP`. Additionally, the `INPUT` mode explicitly disables the internal pullups.

Syntax

```
pinMode(pin, mode)
```

Parameters

pin: the number of the pin whose mode you wish to set

mode: `INPUT`, `OUTPUT`, or `INPUT_PULLUP`. (see the [digital pins](#) page for a more complete description of the functionality.)

Gambar Penjelasan Set Instruksi pinMode()

Dengan cara yang sama Anda juga dapat melihat keterangan set instruksi **digitalWrite()** sebagai berikut:

1. Anda klik digitalWrite (), kemudian Anda dapat melihat penjelasannya.

Functions

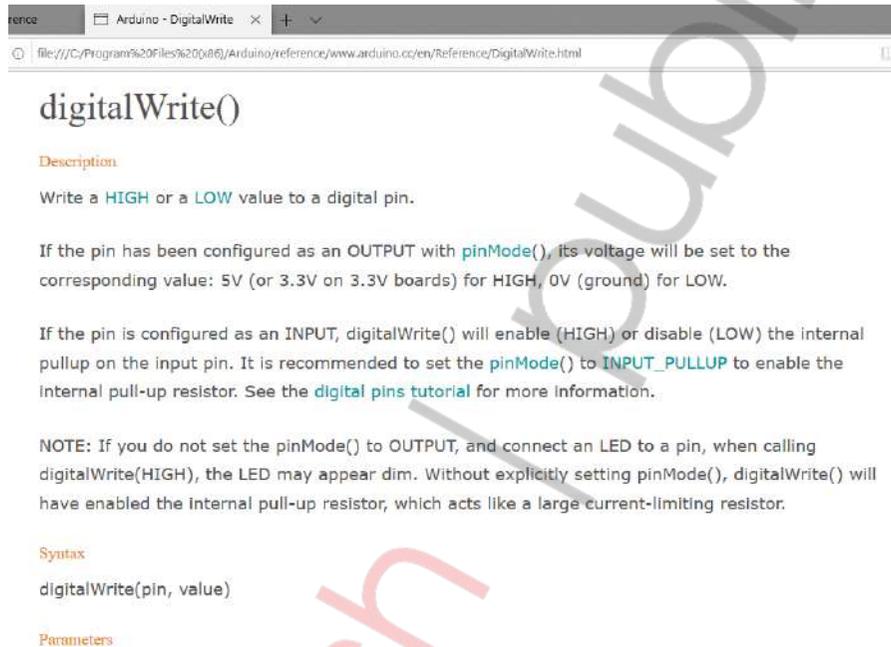
Digital I/O

- pinMode()
- digitalWrite()
- digitalRead()

2. Penjelasan set instruksi **digitalWrite()**

digitalWrite() berfungsi untuk memberikan nilai HIGH atau LOW pada pin Arduino. Pin Arduino yang dimaksud di sini adalah pin Arduino sesuai dengan nomor pin yang kita pilih. Misal dalam program kita

menulis set instruksi `digitalWrite(13, HIGH)`; artinya kita memberikan nilai HIGH pada pin Arduino nomor 13. Perlu diketahui bahwa nilai digital berupa nilai biner yang hanya ada dua keadaan yaitu HIGH atau LOW.



Gambar Penjelasan Set Instruksi `digitalWrite()`

Anda juga dapat melihat beberapa set instruksi yang lain dengan cara yang sama, beserta contoh penulisan programnya pada saat Anda menganalisis atau membuat program, jika Anda belum mengetahui fungsi dan bentuk penulisannya (*syntax*).

Sumber: <https://www.arduino.cc/reference/en/>

I. Soal Latihan

1. Sebutkan komponen mikrokontroler dan jelaskan masing-masing fungsinya!
2. Jelaskan perbedaan antara sistem komunikasi sinkron dengan asinkron!
3. Jelaskan bagaimana cara memprogram sebuah mikrokontroler!

4. Sebutkan bagian-bagian sistem minimum Arduino Uno dan jelaskan fungsinya!
5. Sebutkan macam-macam antar muka pada Arduino!

Bab 2

Simulasi Pemrograman Arduino Menggunakan Tinkercad

A. Tujuan

Praktik ini bertujuan agar praktikan dapat:

1. Mengenal dan memahami cara pemrograman Arduino secara simulasi melalui tinkercad.com
2. Melakukan simulasi pemrograman Arduino secara simulasi melalui tinkercad.com

B. Alat dan Bahan

1. PC atau Laptop
2. *Web browser* (misal: Google Chrome, Mozilla Firefox atau sejenisnya)
3. Koneksi internet

C. Teori Singkat

a. Mengenal Tinkercad

Tinkercad merupakan sebuah platform yang merupakan web penyedia sarana untuk belajar secara *online* terkait desain 3D, rangkaian elektronika dan *codeblock*. Tinkercad merupakan web dari Autodesk yang sudah cukup terkenal. Jika sebelum-sebelumnya Autodesk banyak memperkenalkan *software-software* komputer seperti *software* desain, animasi, kini mereka hadir dengan salah satu platform yang bernama Tinkercad. Sebenarnya, web ini sudah lama didirikan yakni pada tahun 2011. Web ini merupakan web yang cocok digunakan sebagai media pembelajaran secara *online*.

Web ini hampir sama dengan web belajar *online* lainnya, seperti Google Classroom, Edmodo, Schoology, dan web penyedia layanan pembelajaran *online* lainnya. Pada web ini, terdapat fitur Educator untuk guru, dan *student* untuk siswa. Hanya saja Tinkercad sendiri lebih dikhususkan untuk pembelajaran di bidang desain 3D, elektronika, dan

codeblock. Pada Tinkercad juga terdapat fitur Class di mana kita bisa membuat sebuah kelas untuk melakukan pembelajaran antara guru dan siswanya. Namun pada penjelasan kali ini kita fokuskan pada penggunaan layanan elektronika untuk simulasi pemrograman Arduino.



Gambar 1. Halaman Tampilan tinkercad.com

D. Langkah Kerja

Berikut ini adalah langkah-langkah menggunakan layanan simulasi menggunakan Tinkercad.

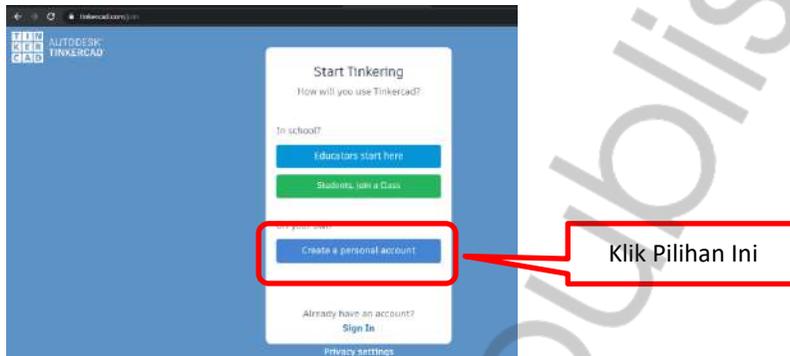
a. Mendaftar akun di tinkercad.com

1. Buka *browser* di komputer atau laptop Anda. Pastikan sudah terhubung dengan koneksi internet.
2. Buka laman <https://www.tinkercad.com/>
3. Klik pilihan Join Now pada pojok kanan atas.



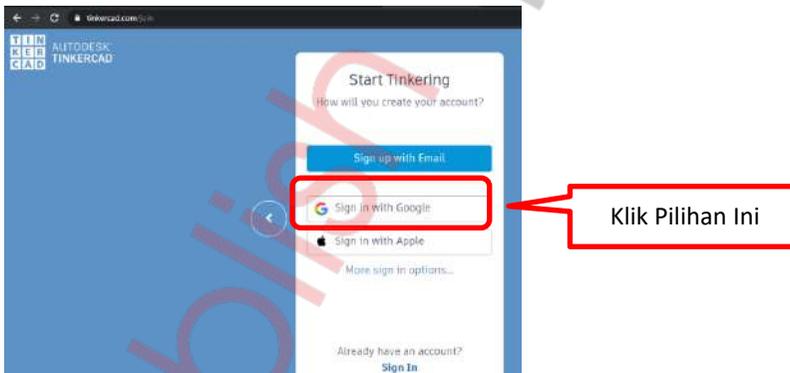
Gambar 2. Halaman Tampilan Awal tinkercad.com

4. Kemudian akan diarahkan pada pilihan berikut. Pilih “Create a personal account”.



Gambar 3. Halaman Tampilan Pilihan Jenis Akun

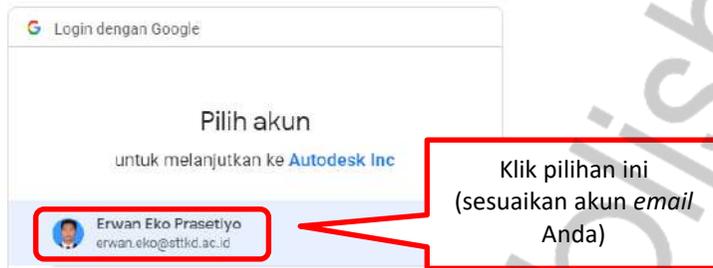
5. Kemudian akan diarahkan pada pilihan berikut. Pilih “Sign in With Google”



Gambar 4. Halaman Tampilan Pilihan Jenis *Email*

6. Pilih akun *email* yang ingin Anda gunakan untuk daftar (gunakan *email* dari sekolah/kampus Anda).

Catatan: Jika Anda belum *login* ke *email* Anda, silakan Anda *login* ke *email* Anda terlebih dahulu.



Gambar 5. Halaman Tampilan Pilihan Akun *Email*

7. Jika berhasil, maka akan diarahkan masuk ke halaman *dashboard* Tinkercad.

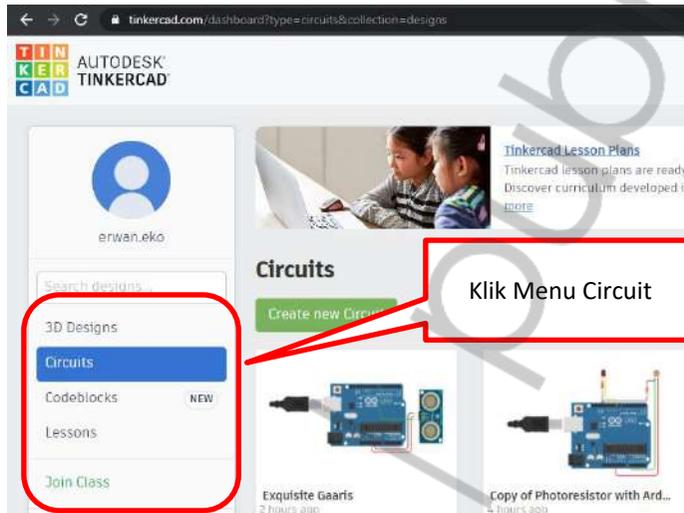


Gambar 6. Halaman *Dashboard*

8. Sampai di sini akun Anda sudah terdaftar di Tinkercad.com dan siap untuk menggunakan layanan simulasi.

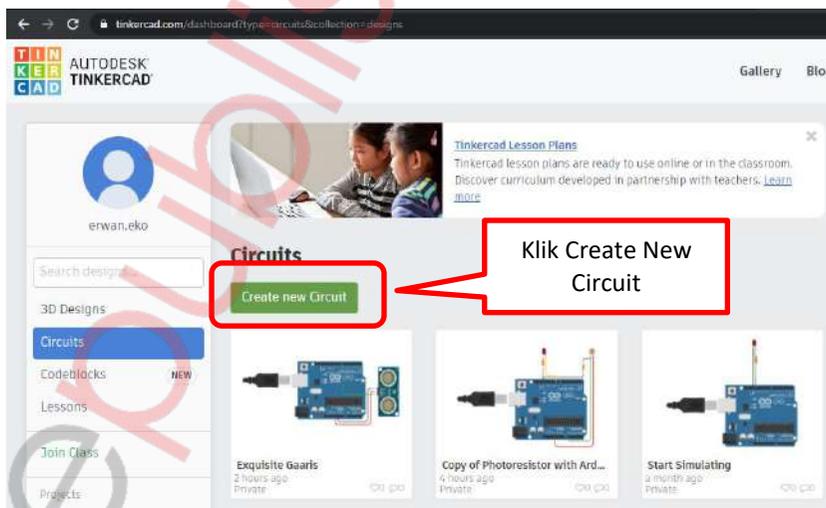
b. Memulai Simulasi Rangkaian Elektronik Arduino dan Pemrogramannya

1. Pada halaman *dashboard* klik menu Circuit pada pilihan menu di sebelah kiri.



Gambar 7. Pilihan Menu pada Halaman *Dashboard*

2. Untuk mulai membuat rangkaian baru klik "Create New Circuit".



Gambar 8. Pilihan Menu untuk Membuat Rangkaian Elektronik

3. Jika berhasil maka akan tampil seperti berikut.



Gambar 9. Tampilan Lembar Kerja Baru untuk Memulai Membuat Rangkaian

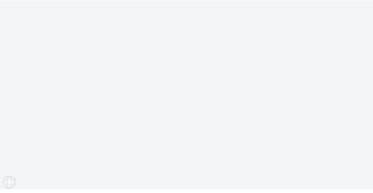
4. Sampai di sini Anda sudah siap untuk memulai membuat rangkaian elektronik sesuai yang Anda kehendaki.

c. Mengenal *Tools* Simulasi Rangkaian di Tinkercad

Tools untuk melakukan simulasi dijelaskan sebagai berikut:

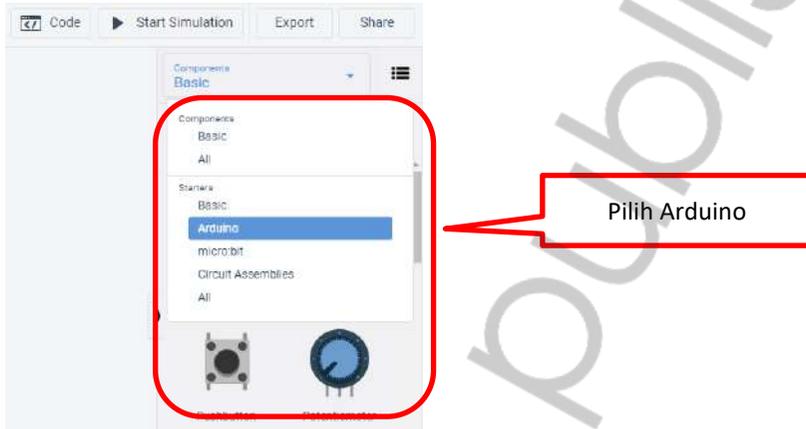


Gambar 10. Tools Simulasi

No	Simbol	Penjelasan
1		<p>Berisi tombol untuk mengatur lembar kerja saat membuat rangkaian yang berisi <i>rotate</i>, <i>delete</i>, <i>undo</i>, <i>redo</i>, <i>annotation</i>, dan <i>view/hide</i></p>
2		<p>Papan kerja/lembar kerja untuk meletakkan rangkaian yang akan dibuat/disusun.</p>
3		<p>Berisi tombol untuk mengatur tampilan simulasi dan informasi akun.</p>
4		<p>Berisi tombol untuk mengatur simulasi yang berupa tombol <i>Code</i>, <i>Start Simulation</i>, <i>Export</i> dan <i>Share</i>.</p>
5		<p>Berisi pilihan daftar komponen yang dapat digunakan untuk simulasi.</p>

d. Menjalankan Contoh Simulasi LED Blink (LED Berkedip)

1. Dari halaman lembar kerja baru, pilih komponen pada menu bagian kanan. Kemudian pilih/klik Arduino.



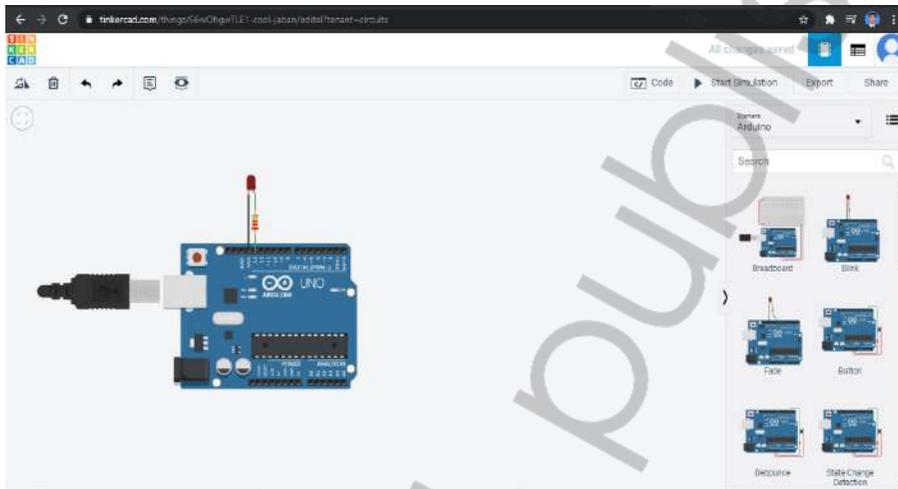
Gambar 11. Memilih Komponen Arduino

2. Jika berhasil maka akan tampil beberapa pilihan contoh rangkaian Arduino. Pilih rangkaian Arduino "Blink".



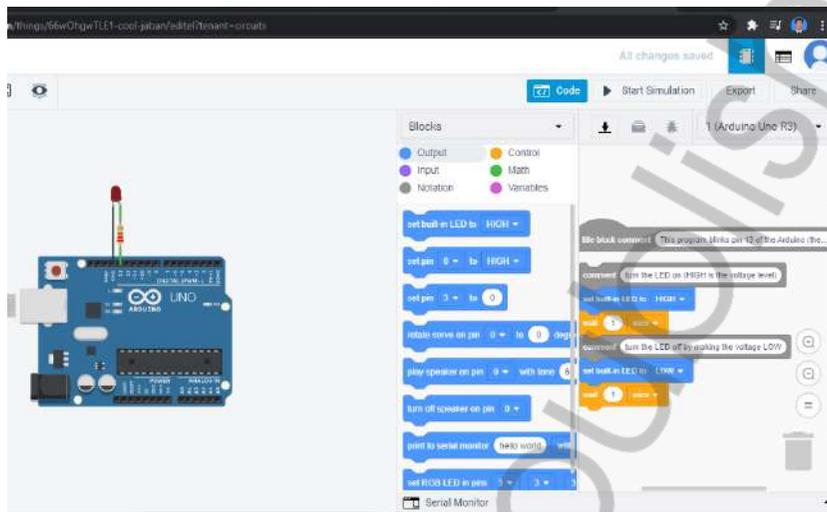
Gambar 12. Pilihan Rangkaian Arduino

3. Klik dan *drop* rangkaian LED Blink pada lembar kerja. Jika berhasil maka akan tampil seperti berikut.



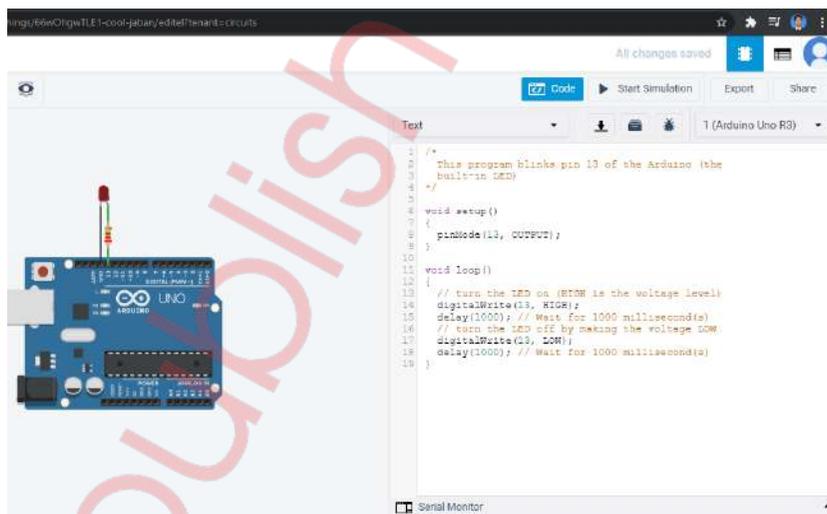
Gambar 13. Contoh Simulasi Rangkaian LED Blink

4. Klik tombol “Start Simulation” pada sisi kanan atas untuk memulai simulasi.
5. Perhatikan kondisi LED. Jika simulasi berhasil maka LED akan berkedip-kedip.
6. Klik “Stop Simulation” untuk menghentikan simulasi.
7. Sampai di sini Anda sudah berhasil mencoba melakukan simulasi rangkaian Arduino LED Blink.
8. Untuk mengubah durasi kedipan LED, maka Anda dapat mengedit kode program dengan cara sebagai berikut.
9. Klik tombol “Code” pada sisi kanan atas. Maka akan tampil seperti berikut.



Gambar 14. Tampilan Saat Klik Code Program

10. Klik “Block” kemudian pilih/klik “Text” kemudian pilih “Continue” untuk masuk ke mode pemrograman menggunakan *text*.



Gambar 14. Tampilan Code Program dalam Bentuk *Text*

11. Lakukan edit *text* kode program untuk mengubah durasi kedipan LED Blink. Misal dengan mengubah *delay* menjadi 500, untuk menghasilkan durasi kedipan yang lebih cepat.

12. Jika sudah selesai mengedit kode program, kemudian klik “Start Simulation” dan amati hasilnya.
13. Jika kedipan LED menjadi lebih cepat dari sebelumnya, maka percobaan Anda berhasil.

E. Latihan

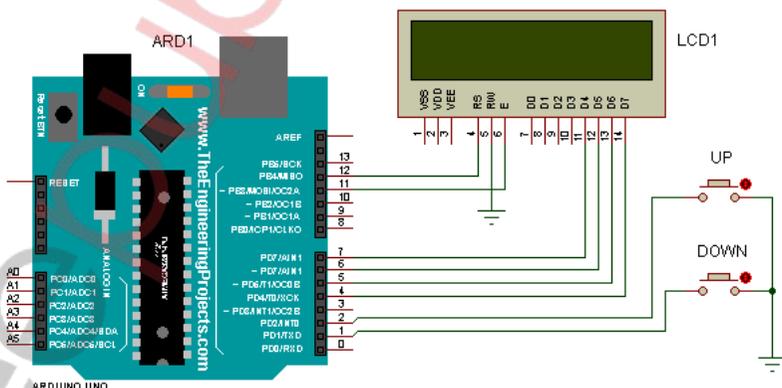
1. Jelaskan langkah-langkah melakukan simulasi pemrograman Arduino menggunakan aplikasi Tinkercad!
2. Jelaskan fungsi tiap baris kode program pada potongan kode program berikut ini!

```

1 // Kode Program Arduino
2 const int buttonPin = 2;
3 const int ledPin = 13;
4 int buttonState = 0;
5 void setup() {
6   pinMode(ledPin, OUTPUT);
7   pinMode(buttonPin, INPUT);
8 }
9 void loop() {
10  buttonState = digitalRead(buttonPin);
11  if (buttonState == HIGH) {
12    digitalWrite(ledPin, HIGH);
13  } else {
14    digitalWrite(ledPin, LOW);
15  }

```

3. Tuliskan kode program rangkaian berikut untuk menampilkan tulisan MIKROKONTROLER. Tulisan akan terhapus (*clear*) jika *push button UP* dan *DOWN* ditekan bersamaan!



F. Hasil Praktik

Tuliskan hasil praktik Anda saat melakukan percobaan ini.

G. Analisis

Tuliskan hasil analisis Anda setelah melakukan percobaan ini.

H. Kesimpulan

Tuliskan kesimpulan yang Anda dapatkan dari praktik ini.

Bab 3

Simulasi Arduino Running LED

A. Tujuan

Praktik ini bertujuan agar praktikan dapat:

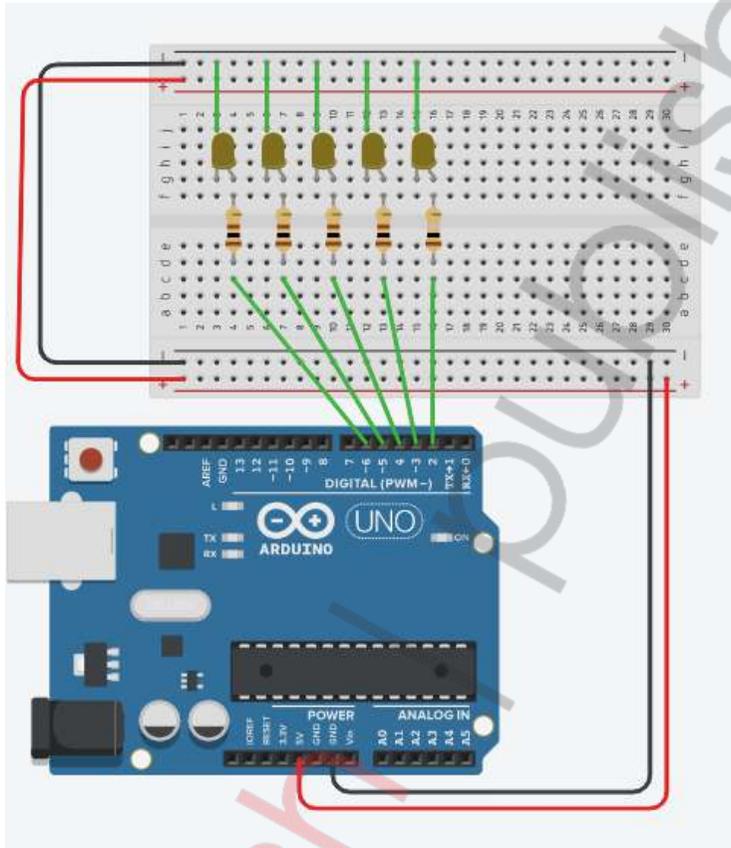
1. Mengetahui dan memahami fungsi Digital Output pada Arduino
2. Melakukan implementasi desain aplikasi Digital Output pada Arduino
3. Merancang aplikasi menggunakan Digital Output pada Arduino

B. Alat dan Bahan

1. PC atau Laptop
2. Koneksi internet

C. Aplikasi Dasar

Aplikasi pada praktik ini merupakan aplikasi dasar Arduino yang memanfaatkan fungsi Digital Output. Aplikasi ini akan memfungsikan Arduino memberikan output digital berupa kondisi 0 atau 1. Kondisi output ini ditunjukkan dengan indikator LED yang mempunyai kondisi “mati dan “nyala”. Berikut ini adalah contoh aplikasi Arduino Uno dengan 5 buah LED sebagai output yang dipasang pada pin 2,3,4,5,6. Program pada aplikasi ini akan membuat LED menyala bergeser.



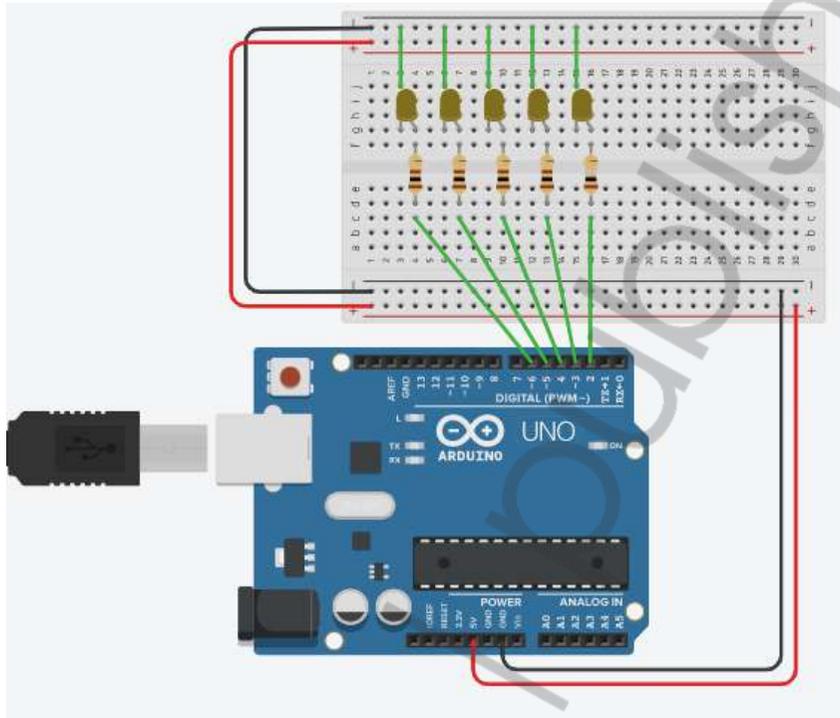
Gambar 1. Rangkaian Aplikasi Digital Output dengan LED

Sumber: https://www.tinkercad.com/things/i4s6DJ1YtOg-sizzling-vihelmowolt/editel?sharecode=P_08aoxH8QTf9arDZ9OSRshFA4WcCEfjJuaCjtNphPs

D. Langkah Kerja

Berikut ini adalah langkah-langkah membuat aplikasi Running LED menggunakan Arduino Uno.

1. Siapkan alat dan bahan yang akan digunakan sesuai kebutuhan.
2. Buka *web browser* Anda dan masuk ke web simulasi *online* Tinkercad.
3. Buat rangkaian seperti gambar berikut (panduan cara membuat rangkaian simulasi di Tinkercad dijelaskan pada *labsheet* praktik 1).



Gambar 2. Skema Rangkaian Running LED dengan Arduino

4. Ketikkan kode program berikut pada halaman Code (pilih mode *text*)

```
//=====
//RUNNING LED ARDUINO
//=====

int i,n; //variable global
void setup() {
  // Mengatur pin sebagai output
  pinMode(2,OUTPUT);
  pinMode(3,OUTPUT);
  pinMode(4,OUTPUT);
  pinMode(5,OUTPUT);
  pinMode(6,OUTPUT);
}

void loop() {
  // Mengatur nilai variabel i=1
  i=1;
  // Perulangan sebanyak 5x dengan FOR
  for (n=1; n<=6; n++){
```

```
digitalWrite(n,LOW);
  if (n>6){n=0;}
}
// Perulangan sebanyak 5x dengan WHILE
while(i<=6)
{
  digitalWrite(i,HIGH);
  delay(1000);
  i++;
}
}
```

5. Jalankan simulasi dengan cara mengklik tombol "Start Simulation".
6. Jika berhasil, maka LED akan menyala dari kanan ke kiri.

E. Tugas Praktik

1. Buatlah rangkaian Running LED tersebut agar LED menyala bergeser dari kiri ke kanan, setelah LED menyala semua kemudian LED padam/mati satu per satu dari kanan ke kiri
2. Simpan hasil simulasi Anda pada akun Tinkercad dengan nama **PRAKTIK 02-NIT-Simulasi Arduino Running LED.**

F. Analisis

Tuliskan hasil analisis Anda berdasarkan hasil simulasi yang telah Anda lakukan. Tulis pada *project* Anda menggunakan *Notes Tool*.

G. Kesimpulan

Tuliskan kesimpulan yang Anda dapatkan dari praktik ini. Tulis pada *project* Anda menggunakan *Notes Tool*.

Bab 4

Simulasi Arduino Digital Input

A. Tujuan

Praktik ini bertujuan agar praktikan dapat:

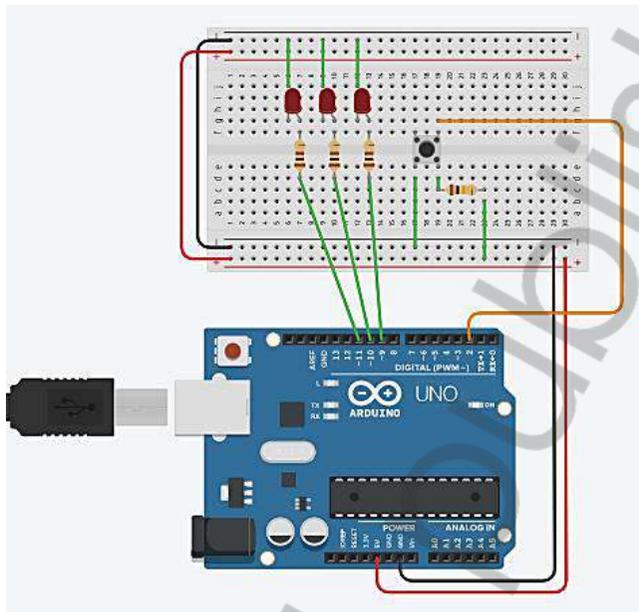
1. Mengetahui dan memahami fungsi digital input pada Arduino
2. Melakukan implementasi desain aplikasi digital input pada Arduino
3. Merancang aplikasi menggunakan digital input pada Arduino

B. Alat dan Bahan

1. PC atau Laptop
2. Koneksi internet

C. Aplikasi Dasar

Aplikasi pada praktik ini merupakan aplikasi dasar Arduino yang memanfaatkan fungsi digital input. Aplikasi ini akan memfungsikan Arduino menerima input digital berupa kondisi 0 atau 1. Kondisi input ini direpresentasikan sebagai kondisi biner yang mempunyai kondisi "0 atau LOW dan "1 atau HIGH". Berikut ini adalah contoh aplikasi Arduino Uno dengan 1 buah *push button* sebagai input yang dipasang pada pin 2. Program pada aplikasi ini akan membaca data digital yang diberikan oleh *push button*.



Gambar 1. Rangkaian Aplikasi Digital Input dengan *Push Button*

Sumber: <https://www.tinkercad.com/things/hQvUgJawSFe-praktik-03-simulasi-arduino-digital-input>

D. Langkah Kerja

Berikut ini adalah langkah-langkah membuat aplikasi Running LED menggunakan Arduino Uno.

1. Siapkan alat dan bahan yang akan digunakan sesuai kebutuhan.
2. Buka *web browser* Anda dan masuk ke web simulasi *online* Tinkercad.
3. Buat rangkaian seperti Gambar 1 (panduan cara membuat rangkaian simulasi di Tinkercad dijelaskan pada *labsheet* praktik 1)
4. Ketikkan kode program berikut pada halaman Code (pilih mode *text*)

```
//=====
//SIMULASI DIGITAL INPUT
//=====

//Deklarasi variabel
const int LED1=9;
const int LED2=10;
const int LED3=11;
const int btnPin=2;

int btnData=0;
```

```

int nyalaLED=0;

void setup()
{
  // Mengatur pin I/O
  pinMode(btnPin, INPUT);
  pinMode(LED1, OUTPUT);
  pinMode(LED2, OUTPUT);
  pinMode(LED3, OUTPUT);
}

void loop()
{
  //Baca Data Button/Tombol
  btnData=digitalRead(btnPin);
  if (btnData==LOW){ //Jika Button ditekan
    nyalaLED++;      //Data variabel nyalaLED +1
    delay(100);      //Tunda waktu 100 ms
    if (nyalaLED>3){nyalaLED=0;}
  }

  //Pilihan Penyalaan LED
  //Saat kondisi nyalaLED==0
  if (nyalaLED==0){
    digitalWrite(LED1, LOW);
    digitalWrite(LED2, LOW);
    digitalWrite(LED3, LOW);
  }
  //Saat kondisi nyalaLED==1
  else if (nyalaLED==1){
    digitalWrite(LED1, HIGH);
    digitalWrite(LED2, LOW);
    digitalWrite(LED3, LOW);
  }

  //Saat kondisi nyalaLED==2
  else if (nyalaLED==2){
    digitalWrite(LED1, HIGH);
    digitalWrite(LED2, HIGH);
    digitalWrite(LED3, LOW);
  }
  //Saat kondisi nyalaLED==3
  else if (nyalaLED==3){
    digitalWrite(LED1, HIGH);
    digitalWrite(LED2, HIGH);
    digitalWrite(LED3, HIGH);
  }
}
}

```

5. Jalankan simulasi dengan cara mengklik tombol “Start Simulation”.
6. Jika berhasil, maka:
Apabila *push button* ditekan 1x maka LED akan menyala 1 buah.
Apabila *push button* ditekan 2x maka LED akan menyala 2 buah.
Apabila *push button* ditekan 3x maka LED akan menyala 3 buah.
Apabila *push button* ditekan 4x maka LED akan padam semua/kembali ke kondisi awal.

E. Tugas Praktik

1. Buatlah rangkaian Running LED dengan ketentuan sebagai berikut:
 - a. Running LED terdiri atas 5 (lima) buah LED, warna bebas.
 - b. LED menyala satu per satu (saat 1 buah LED menyala, maka LED yang lainnya mati)
 - c. Kondisi nyala LED diatur dengan 2 (dua) buah *push button*.
 - d. Apabila *push button* tidak ditekan, maka semua LED mati.
 - e. Apabila *push button* A ditekan maka *running LED* nyalanya bergeser dari kiri ke kanan.
 - f. Apabila *push button* B ditekan maka *running LED* nyalanya bergeser dari kanan ke kiri.
2. Simpan hasil pekerjaan Anda pada akun Tinkercad.

F. Analisis

Tuliskan hasil analisis Anda berdasarkan hasil simulasi yang telah Anda lakukan. (Tulis pada *project* Anda menggunakan *Anotation*)

G. Kesimpulan

Tuliskan kesimpulan yang Anda dapatkan dari praktik ini. (Tulis pada *project* Anda menggunakan *Anotation*)

Bab 5

Simulasi Arduino Analog Input

A. Tujuan

Praktik ini bertujuan agar praktikan dapat:

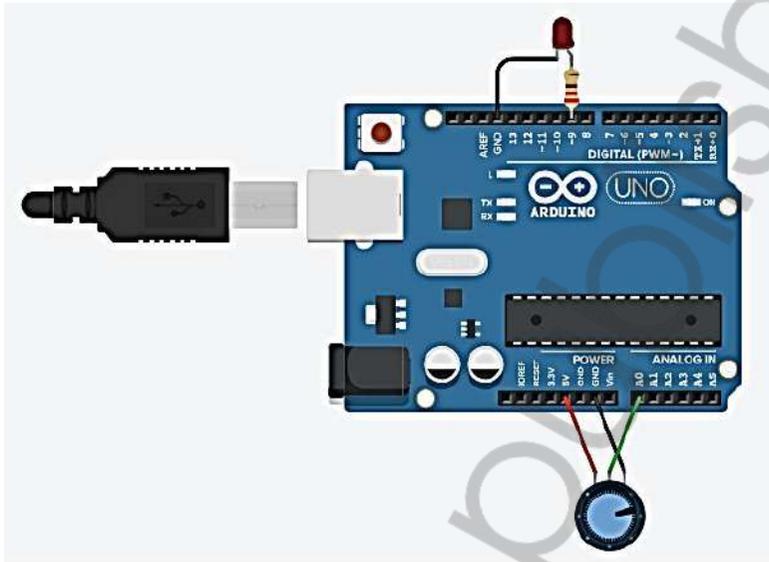
1. Mengenal dan memahami fungsi analog input pada Arduino
2. Melakukan implementasi desain aplikasi analog input pada Arduino
3. Merancang aplikasi menggunakan analog input pada Arduino

B. Alat dan Bahan

1. PC atau Laptop
2. Koneksi internet

C. Aplikasi Dasar

Aplikasi pada praktik ini merupakan aplikasi dasar Arduino yang memanfaatkan fungsi analog input. Aplikasi ini akan membaca kondisi potensiometer sebagai pendeteksi input yang memiliki kondisi data analog antara 0 volt–5 volt. Data analog kemudian diubah oleh Arduino menjadi data digital dengan nilai antara 0 sampai 1023 (karena Arduino menggunakan ADC 10 bit). Berikut ini adalah contoh aplikasi Arduino Uno dengan sebuah potensiometer yang dipasang pada pin A0, LED pada pin 9 dan serial monitor untuk menampilkan data digital hasil pembacaan data.



Gambar 1. Rangkaian Aplikasi Analog Input dengan *Push Button*
 Sumber: <https://www.tinkercad.com/things/1EkN5d18Tx-daring-uusam-juttuli/editel?tenant=circuits>

D. Langkah Kerja

Berikut ini adalah langkah-langkah membuat aplikasi Running LED menggunakan Arduino Uno.

1. Siapkan alat dan bahan yang akan digunakan sesuai kebutuhan.
2. Buka *web browser* Anda dan masuk ke web simulasi *online* Tinkercad.
3. Buat rangkaian seperti Gambar 1. Rangkaian ini untuk mengatur nyala LED dari redup ke terang dengan memutar *potentiometer*.
4. Ketikkan kode program berikut pada halaman Code (pilih mode *text*)

```
//=====
//SIMULASI ANALOG INPUT
//=====

//Deklarasi Variabel
int sensorValue = 0;
int outputValue = 0;

void setup()
{
  //Mengatur pin I/O
  pinMode(A0, INPUT);
  pinMode(9, OUTPUT);
}
```

```

    //Mengaktifkan serial monitor
    Serial.begin(9600);
}

void loop()
{
    // read the analog in value:
    sensorValue = analogRead(A0);
    // map it to the range of the analog out:
    outputValue = map(sensorValue, 0, 1023, 0, 255);
    // change the analog out value:
    analogWrite(9, outputValue);
    // print the results to the serial monitor:
    Serial.print("sensor = ");
    Serial.print(sensorValue);
    Serial.print("\t output = ");
    Serial.println(outputValue);
    // wait 2 milliseconds before the next loop for
    // the analog-to-digital converter to settle
    // after the last reading
    delay(2); // Wait for 2 millisecond(s)
}

```

5. Jalankan simulasi dengan cara mengklik tombol “Start Simulation”.
6. Klik Serial Monitor pada sisi bagian bawah *text* kode program.
7. Putar *potentiometer* dari paling kiri ke paling kanan.
8. Amati nyala LED dan amati perubahan data yang ditampilkan pada Serial Monitor.
9. Jika berhasil, maka akan ada perubahan data yang ditampilkan Serial Monitor dan nyala LED akan berubah dari redup ke terang atau sebaliknya.

E. Tugas Praktik

1. Buatlah rangkaian Running LED dengan ketentuan sebagai berikut:
 - a. Running LED terdiri atas 5 (lima) buah LED, warna bebas.
 - b. LED menyala bergeser dari kiri ke kanan.
 - c. Kecepatan bergesernya LED diatur oleh *potentiometer*.
 - d. Apabila *potentiometer* diputar ke kiri maka *running LED* nyalanya (bergesernya) semakin pelan sampai berhenti.
 - e. Apabila *potentiometer* diputar ke kanan maka *running LED* nyalanya (bergesernya) semakin cepat.
2. Simpan hasil pekerjaan Anda pada akun Tinkercad.

F. Analisis

Tuliskan hasil analisis Anda berdasarkan hasil simulasi yang telah Anda lakukan. (Tulis pada *project* Anda menggunakan *Anotation*)

G. Kesimpulan

Tuliskan kesimpulan yang Anda dapatkan dari praktik ini. (Tulis pada *project* Anda menggunakan *Anotation*)

Bab 6

Simulasi Arduino 7 Segment

A. Tujuan

Praktik ini bertujuan agar praktikan dapat:

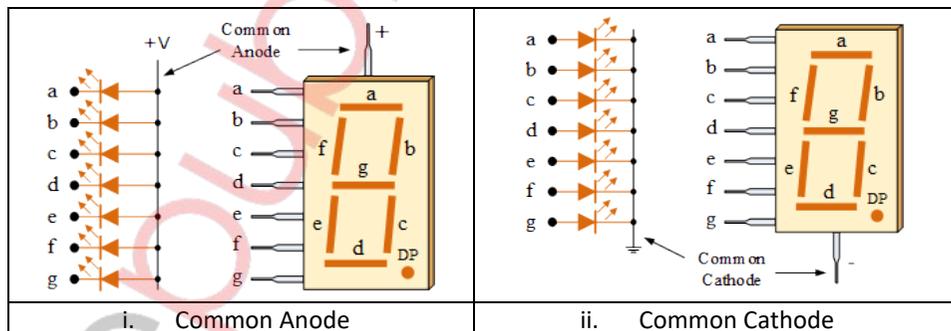
1. Mengenal dan memahami fungsi 7 Segment sebagai *display*
2. Melakukan implementasi desain aplikasi 7 Segment pada Arduino
3. Merancang aplikasi menggunakan 7 Segment pada Arduino

B. Alat dan Bahan

1. PC atau Laptop
2. Koneksi internet

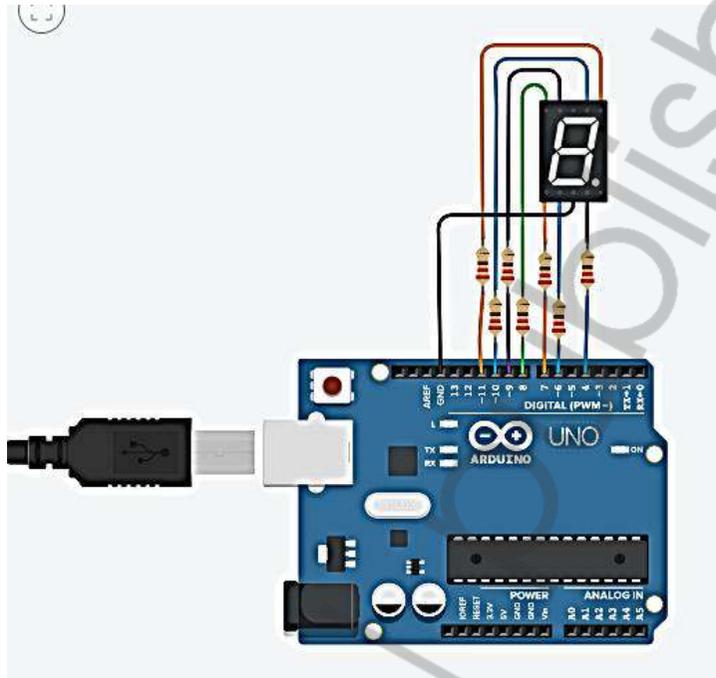
C. Aplikasi Dasar

Aplikasi pada praktik ini merupakan aplikasi dasar Arduino dimanfaatkan untuk menampilkan angka pada 7 Segment. Aplikasi ini dapat menampilkan angka dari 0–9. Ada dua jenis 7 Segment yaitu Common Anode (CA) dan Common Cathode (CC). Berikut ini konfigurasi 7 Segment Common Anode (CA) dan Common Cathode (CC).



Gambar 1. Konfigurasi 7 Segment

Sumber: <https://www.electronics-tutorials.ws/blog/7-segment-display-tutorial.html>



Gambar 1. Rangkaian Aplikasi 7 Segment Counter Up 0-9

Sumber: <https://www.tinkercad.com/things/e2q668FL2IR-exquisite-habbi/editel?tenant=circuits>

D. Langkah Kerja

Berikut ini adalah langkah-langkah membuat aplikasi Counter Up (pencacah naik) menggunakan Arduino Uno.

1. Siapkan alat dan bahan yang akan digunakan sesuai kebutuhan.
2. Buka *web browser* Anda dan masuk ke web simulasi *online* Tinkercad.
3. Buat rangkaian seperti Gambar 1. Rangkaian ini untuk menampilkan angka 0 sampai 9 dengan jeda 1 detik.
4. Ketikkan kode program berikut pada halaman Code (pilih mode *text*)

```
//=====
//SIMULASI 7 Segment C/C
//=====

//Deklarasi Variabel
int pinA = 10;
int pinB = 11;
int pinC = 4;
int pinD = 6;
```

```
int pinE = 7;
int pinF = 9;
int pinG = 8;
```

```
void setup()
{
  // Mengatur pin sebagai Output
  pinMode(pinA, OUTPUT);
  pinMode(pinB, OUTPUT);
  pinMode(pinC, OUTPUT);
  pinMode(pinD, OUTPUT);
  pinMode(pinE, OUTPUT);
  pinMode(pinF, OUTPUT);
  pinMode(pinG, OUTPUT);
}
```

```
void loop()
{
  nol(); //memanggil fungsi nol
  delay(1000);
  satu(); //memanggil fungsi satu
  delay(1000);
  dua(); //memanggil fungsi dua
  delay(1000);
  tiga(); //memanggil fungsi tiga
  delay(1000);
  empat();
  delay(1000);
  lima();
  delay(1000);
  enam();
  delay(1000);
  tujuh();
  delay(1000);
  delapan();
  delay(1000);
  sembilan();
  delay(1000);
}
```

```
void nol()
{
  //Membuat tampilan Angka 0
  digitalWrite(pinA, HIGH);
  digitalWrite(pinB, HIGH);
  digitalWrite(pinC, HIGH);
}
```

```

        digitalWrite(pinD, HIGH);
        digitalWrite(pinE, HIGH);
        digitalWrite(pinF, HIGH);
        digitalWrite(pinG, LOW);
    }

void satu ()
{
    //Membuat tampilan Angka 1
    digitalWrite(pinA, LOW);
    digitalWrite(pinB, HIGH);
    digitalWrite(pinC, HIGH);
    digitalWrite(pinD, LOW);
    digitalWrite(pinE, LOW);
    digitalWrite(pinF, LOW);
    digitalWrite(pinG, LOW);
}

void dua ()
{
    //Membuat tampilan Angka 2
    digitalWrite(pinA, HIGH);
    digitalWrite(pinB, HIGH);
    digitalWrite(pinC, LOW);
    digitalWrite(pinD, HIGH);
    digitalWrite(pinE, HIGH);
    digitalWrite(pinF, LOW);
    digitalWrite(pinG, HIGH);
}

void tiga ()
{
    //Membuat tampilan Angka 3
    digitalWrite(pinA, HIGH);
    digitalWrite(pinB, HIGH);
    digitalWrite(pinC, HIGH);
    digitalWrite(pinD, HIGH);
    digitalWrite(pinE, LOW);
    digitalWrite(pinF, LOW);
    digitalWrite(pinG, HIGH);
}

void empat ()
{
    //Membuat tampilan Angka 4
    digitalWrite(pinA, LOW);
    digitalWrite(pinB, HIGH);
    digitalWrite(pinC, HIGH);
    digitalWrite(pinD, LOW);
}

```

```
    digitalWrite(pinE, LOW);
    digitalWrite(pinF, HIGH);
    digitalWrite(pinG, HIGH);
}
```

```
void lima()
{
    //Membuat tampilan Angka 5
    digitalWrite(pinA, HIGH);
    digitalWrite(pinB, LOW);
    digitalWrite(pinC, HIGH);
    digitalWrite(pinD, HIGH);
    digitalWrite(pinE, LOW);
    digitalWrite(pinF, HIGH);
    digitalWrite(pinG, HIGH);
}
```

```
void enam()
{
    //Membuat tampilan Angka 6
    digitalWrite(pinA, HIGH);
    digitalWrite(pinB, LOW);
    digitalWrite(pinC, HIGH);
    digitalWrite(pinD, HIGH);
    digitalWrite(pinE, HIGH);
    digitalWrite(pinF, HIGH);
    digitalWrite(pinG, HIGH);
}
```

```
void tujuh ()
{
    //Membuat tampilan Angka 7
    digitalWrite(pinA, HIGH);
    digitalWrite(pinB, HIGH);
    digitalWrite(pinC, HIGH);
    digitalWrite(pinD, LOW);
    digitalWrite(pinE, LOW);
    digitalWrite(pinF, LOW);
    digitalWrite(pinG, LOW);
}
```

```
void delapan ()
{
    //Membuat tampilan Angka 8
    digitalWrite(pinA, HIGH);
    digitalWrite(pinB, HIGH);
    digitalWrite(pinC, HIGH);
    digitalWrite(pinD, HIGH);
}
```

```

digitalWrite(pinE, HIGH);
digitalWrite(pinF, HIGH);
digitalWrite(pinG, HIGH);
}

void sembilan()
{
  //Membuat tampilan Angka 9
  digitalWrite(pinA, HIGH);
  digitalWrite(pinB, HIGH);
  digitalWrite(pinC, HIGH);
  digitalWrite(pinD, HIGH);
  digitalWrite(pinE, LOW);
  digitalWrite(pinF, HIGH);
  digitalWrite(pinG, HIGH);
}

```

5. Jalankan simulasi dengan cara mengklik tombol “Start Simulation”.
6. Amati nyala *display 7 Segment* dan amati perubahan angka yang ditampilkan.
7. Jika berhasil, maka 7 Segment akan menampilkan angka 0 sampai 9 dengan jeda 1 detik secara berulang-ulang.

E. Tugas Praktik

1. Buatlah rangkaian *counter down* (pencacah turun) dengan ketentuan sebagai berikut:
 - a. Menampilkan angka 20 sampai 00.
 - b. Menggunakan dua buah 7 Segment untuk menampilkan 2 digit angka.
 - c. Jika *display* menunjukkan angka 00, maka *counter/pencacah* berhenti di tampilan angka 00 dan berkedip-kedip.
2. Simpan hasil simulasi Anda pada akun Tinkercad.

F. Analisis

Tuliskan hasil analisis Anda berdasarkan hasil simulasi yang telah Anda lakukan. (Tulis pada *project* Anda menggunakan *Anotation*)

G. Kesimpulan

Tuliskan kesimpulan yang Anda dapatkan dari praktik ini. (Tulis pada *project* Anda menggunakan *Anotation*)

deepublish / publisher

Bab 7

Simulasi Arduino LCD 16x2

A. Tujuan

Praktik ini bertujuan agar praktikan dapat:

1. Mengenal dan memahami fungsi LCD 16x2 sebagai *display*
2. Melakukan implementasi desain aplikasi LCD 16x2 pada Arduino
3. Merancang aplikasi menggunakan LCD 16x2 pada Arduino

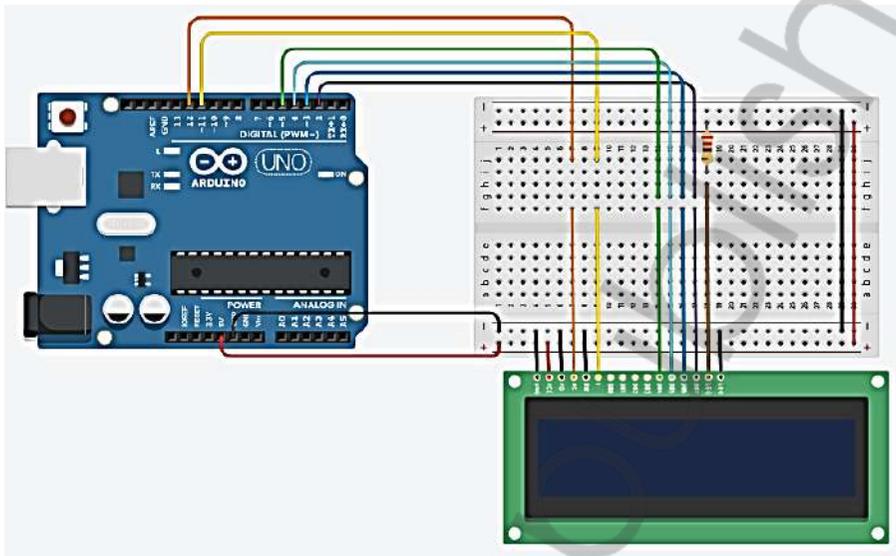
B. Alat dan Bahan

1. PC atau Laptop
2. Koneksi internet

C. Aplikasi Dasar

Aplikasi pada praktik ini merupakan aplikasi dasar Arduino dimanfaatkan untuk menampilkan data teks dan angka pada LCD 16x2. Aplikasi ini dapat menampilkan karakter berupa teks maupun angka sebanyak 16x2 karakter.

LCD (Liquid Crystal Display) 16x2 adalah jenis media tampilan atau *display* dari bahan cairan kristal sebagai penampil utama. LCD 16x2 dapat menampilkan sebanyak 32 karakter yang terdiri dari 2 baris dengan tiap baris menampilkan 16 karakter. Pada Arduino untuk mengendalikan LCD Karakter 16x2 untuk *library*-nya secara default sudah ada *library*-nya yaitu LiquidCrystal.h. LCD ada bermacam-macam ukuran 8x1, 16x1, 16x2, 16x4, 20x4. Untuk mengendalikan atau mengontrol macam-macam LCD karakter pada prinsipnya hampir sama, perbedaannya hanya pada inialisasi jumlah kolom dan baris. (Sumber: <http://www.labelektronika.com/2017/03/cara-program-lcd-karakter-16x2-Arduno-dan-Proteus.html>)



Gambar 1. Aplikasi LCD 16x2 dengan Arduino

Sumber: <https://www.tinkercad.com/things/bNIKDx8fvIM-smooth-snicket-juttuli/editel?tenant=circuits>

D. Langkah Kerja

Berikut ini adalah langkah-langkah membuat aplikasi *display* LCD 16x2 untuk membuat *timer* menggunakan Arduino Uno.

1. Siapkan alat dan bahan yang akan digunakan sesuai kebutuhan.
2. Buka *web browser* Anda dan masuk ke web simulasi *online* Tinkercad.
3. Buat rangkaian seperti Gambar 1. Rangkaian ini untuk menampilkan teks dan angka dari *timer*.
4. Ketikkan kode program berikut pada halaman Code (pilih mode *text*)

```
//=====
//SIMULASI LCD 16x2
//=====

// include the library code:
#include <LiquidCrystal.h>
// initialize the library with the numbers of the
// interface pins
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup() {
  // set up the LCD's number of columns and rows:
  lcd.begin(16, 2);
}
```

```

// set the cursor to column 0, line 0
lcd.setCursor(0, 0);
// Print a message to the LCD.
lcd.print("Praktik 06");
// set the cursor to column 0, line 1
lcd.setCursor(0, 1);
// Print a message to the LCD
lcd.print("Arduino LCD 16x2");
delay (1000);
// Clear LCD Display
lcd.clear();
}

void loop() {
// set the cursor to column 0, line 0
lcd.setCursor(0, 0);
// Print a message to the LCD
lcd.print("STTKD Yogyakarta");
// set the cursor to column 0, line 1
// (note: line 1 is the second row, since
// counting begins with 0):
lcd.setCursor(0, 1);
// Print a message to the LCD
lcd.print("Timer:");
lcd.setCursor(6, 1);
// print the number of seconds since reset:
lcd.print((millis() / 100)-10);
}

```

5. Jalankan simulasi dengan cara mengklik tombol “Start Simulation”.
6. Amati tampilan pada LCD 16x2 dan amati perubahan angka yang ditampilkan.
7. Jika berhasil, maka LCD akan menampilkan teks dan angka *timer*.

E. Tugas Praktik

1. Buatlah desain tampilan LCD 16x2 dengan ketentuan sebagai berikut:
 - Menggunakan 4 buah tombol/*push button* (atas, bawah, kanan, kiri).
 - Jika tombol Atas ditekan maka LCD menampilkan tulisan “ATAS”.
 - Jika tombol Bawah ditekan maka LCD menampilkan tulisan “BAWAH”.
 - Jika tombol Kanan ditekan maka LCD menampilkan tulisan “KANAN”.
 - Jika tombol Kiri ditekan maka LCD menampilkan tulisan “KIRI”.
2. Simpan hasil simulasi Anda pada akun Tinkercad.

F. Analisis

Tuliskan hasil analisis Anda berdasarkan hasil simulasi yang telah Anda lakukan. (Tulis pada *project* Anda menggunakan *Anotation*)

G. Kesimpulan

Tuliskan kesimpulan yang Anda dapatkan dari praktik ini. (Tulis pada *project* Anda menggunakan *Anotation*)

CATATAN

Keterangan Pin LCD 16x2

The circuit:

- * LCD RS pin to digital pin 12
- * LCD Enable pin to digital pin 11
- * LCD D4 pin to digital pin 5
- * LCD D5 pin to digital pin 4
- * LCD D6 pin to digital pin 3
- * LCD D7 pin to digital pin 2
- * LCD R/W pin to ground
- * LCD VSS pin to ground
- * LCD VCC pin to 5V
- * 10K resistor:
 - * ends to +5V and ground
 - * wiper to LCD VO pin (pin 3)

Bab 8

Simulasi Arduino Motor Servo

A. Tujuan

Praktik ini bertujuan agar praktikan dapat:

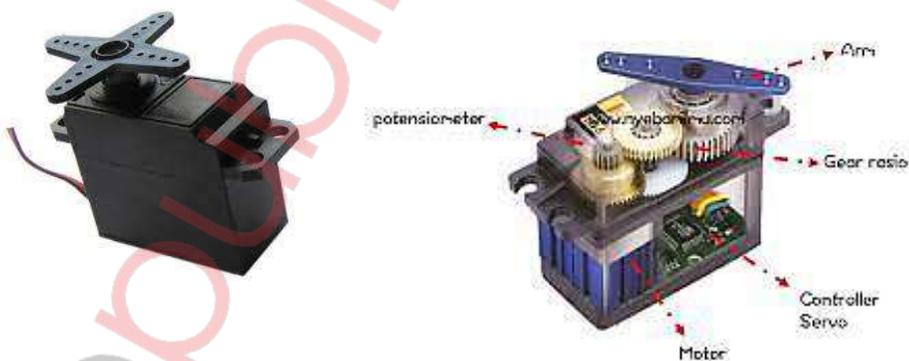
1. Mengetahui dan memahami fungsi motor servo sebagai output
2. Melakukan implementasi desain aplikasi motor servo pada Arduino
3. Merancang aplikasi menggunakan motor servo pada Arduino

B. Alat dan Bahan

1. PC atau Laptop
2. Koneksi internet

C. Teori dan Aplikasi Dasar

Aplikasi pada praktik ini merupakan aplikasi dasar Arduino dimanfaatkan untuk mengendalikan motor servo. Aplikasi ini dapat mengendalikan putaran motor servo dengan sudut dan arah sesuai keinginan.



Gambar 1. Motor Servo dan Konfigurasinya

Servo Motor adalah sebuah motor dengan sistem umpan balik tertutup di mana posisi dari motor akan diinformasikan kembali ke rangkaian kontrol yang ada di dalam servo motor. Motor ini terdiri dari sebuah motor DC, serangkaian *gear*, potensiometer dan rangkaian kontrol. Potensiometer berfungsi untuk menentukan batas sudut dari putaran servo. Sedangkan sudut dari sumbu servo motor diatur berdasarkan lebar pulsa yang dikirim melalui kaki sinyal dari kabel motor.

Motor DC servo merupakan alat untuk mengubah energi listrik menjadi energi mekanik, maka magnet permanen motor DC servolah yang mengubah energi listrik ke dalam energi mekanik melalui interaksi dari dua medan magnet.

Salah satu medan dihasilkan oleh magnet permanen dan yang satunya dihasilkan oleh arus yang mengalir dalam kumparan motor. Resultan dari dua medan magnet tersebut menghasilkan torsi yang membangkitkan putaran motor tersebut. Saat motor berputar, arus pada kumparan motor menghasilkan torsi yang nilainya konstan.

Secara umum terdapat 2 jenis servo motor:

1. Servo motor *standard*, servo motor tipe *standard* hanya mampu berputar 180 derajat. Servo motor *standard* sering dipakai pada sistem robotika misalnya untuk membuat "Robot Arm" (Robot Lengan).
2. Servo motor *continuous*, servo motor *continuous* dapat berputar sebesar 360 derajat. Servo motor *continuous* sering dipakai untuk Mobile Robot. Pada badan servo tertulis tipe servo yang bersangkutan.

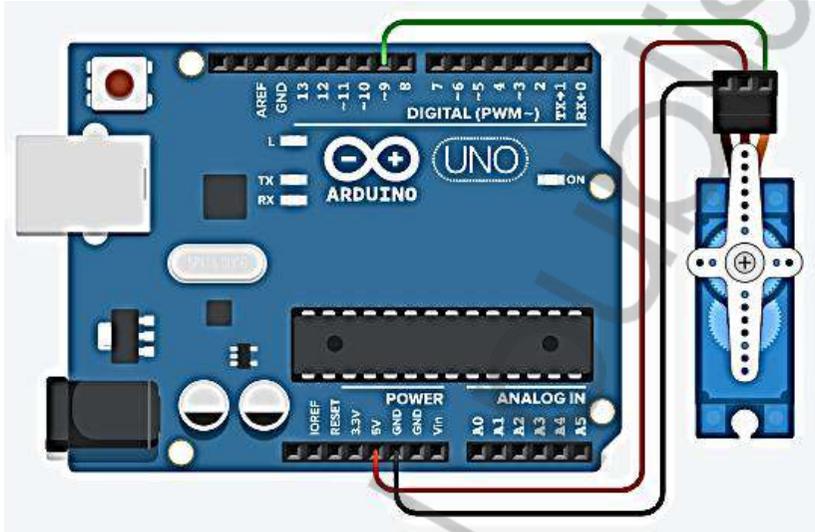
Pengendalian gerakan batang servo motor dapat dilakukan dengan menggunakan metode PWM (*Pulse Width Modulation*). Teknik ini menggunakan sistem lebar pulsa untuk mengemudikan putaran motor. Sudut dari sumbu servo motor diatur berdasarkan lebar pulsa yang dikirim melalui kaki sinyal dari kabel motor. (Sumber: <http://www.kitomaindonesia.com/kategori/4/servo-motor-drive>)

D. Langkah Kerja

Berikut ini adalah langkah-langkah membuat aplikasi motor servo untuk mengendalikan putaran menggunakan Arduino Uno.

1. Siapkan alat dan bahan yang akan digunakan sesuai kebutuhan.
2. Buka *web browser* Anda dan masuk ke web simulasi *online* Tinkercad.

3. Buat rangkaian seperti Gambar 2. Rangkaian ini untuk mengendalikan putaran motor servo.



Gambar 1. Aplikasi Motor Servo dengan Arduino

Sumber: <https://www.tinkercad.com/things/500zJRKi9IQ-surprising-maimu/editel?tenant=circuits>

4. Ketikkan kode program berikut pada halaman Code (pilih mode *text*)

```
//=====
//SIMULASI MOTOR SERVO
//=====

//Menambahkan library Servo.h
#include <Servo.h>
//Deklarasi variabel
int pos = 0;
Servo servo_9;

void setup()
{
  //Mengatur pin arduino untuk pin data pada motor servo
  servo_9.attach(9);
}

void loop()
{
  // sweep the servo from 0 to 180 degrees in steps
  // of 1 degrees
```

```

for (pos = 0; pos <= 180; pos += 1) {
  // tell servo to go to position in variable 'pos'
  servo_9.write(pos);
  // wait 15 ms for servo to reach the position
  delay(15); // Wait for 15 millisecond(s)
}
for (pos = 180; pos >= 0; pos -= 1) {
  // tell servo to go to position in variable 'pos'
  servo_9.write(pos);
  // wait 15 ms for servo to reach the position
  delay(15); // Wait for 15 millisecond(s)
}
}

```

5. Jalankan simulasi dengan cara mengklik tombol “Start Simulation”.
6. Amati arah putaran motor servo.
7. Jika berhasil, maka motor servo akan berputar ke kiri dan ke kanan membentuk sapuan (*sweep*).

E. Tugas Praktik

1. Buatlah desain pengendali motor servo untuk mengatur posisi dengan ketentuan sebagai berikut:
 - Motor servo yang dikendalikan sebanyak 1 buah motor
 - Menggunakan 2 buah tombol/*push button* (kanan, kiri).
 - Jika tombol kanan ditekan maka motor servo berputar ke kanan sebesar 90 derajat.
 - Jika tombol kiri ditekan maka motor servo berputar ke kiri sebesar 90 derajat.
2. Simpan hasil simulasi Anda pada akun Tinkercad.

F. Analisis

Tuliskan hasil analisis Anda berdasarkan hasil simulasi yang telah Anda lakukan. (Tulis pada *project* Anda menggunakan *Anotation*)

G. Kesimpulan

Tuliskan kesimpulan yang Anda dapatkan dari praktik ini. (Tulis pada *project* Anda menggunakan *Anotation*)

CATATAN

Berikut ini penjelasan dan contoh *syntax* pada motor servo. Untuk lebih jelasnya lihat pada <https://www.arduino.cc/en/Reference/ServoAttach>

`attach()`

Description

Attach the Servo variable to a pin. Note that in Arduino 0016 and earlier, the Servo library supports only servos on only two pins: 9 and 10.

Syntax

`servo.attach(pin)`

`servo.attach(pin, min, max)`

Parameters

`servo`: a variable of type Servo

`pin`: the number of the pin that the servo is attached to

`min` (optional): the pulse width, in microseconds, corresponding to the minimum (0-degree) angle on the servo (defaults to 544)

`max` (optional): the pulse width, in microseconds, corresponding to the maximum (180-degree) angle on the servo (defaults to 2400)

Bab 9

Simulasi Arduino Pengukur Jarak

A. Tujuan

Praktik ini bertujuan agar praktikan dapat:

1. Mengenal dan memahami fungsi sensor ultrasonik
2. Melakukan implementasi desain aplikasi pengukur jarak dengan sensor ultrasonik
3. Merancang aplikasi pengukur jarak menggunakan sensor ultrasonik dan LCD 16x2 pada Arduino

B. Alat dan Bahan

1. PC atau Laptop
2. Koneksi internet

C. Teori dan Aplikasi Dasar

Aplikasi pada praktik ini merupakan aplikasi dasar Arduino yang dimanfaatkan untuk membaca data jarak dari sensor ultrasonik PING kemudian menampilkan data hasil pembacaan pada LCD 16x2. Aplikasi ini dapat digunakan sebagai alat ukur digital untuk mengetahui jarak suatu benda. Berikut ini adalah sensor ultrasonik PING.



Gambar 1. Sensor Ultrasonik PING

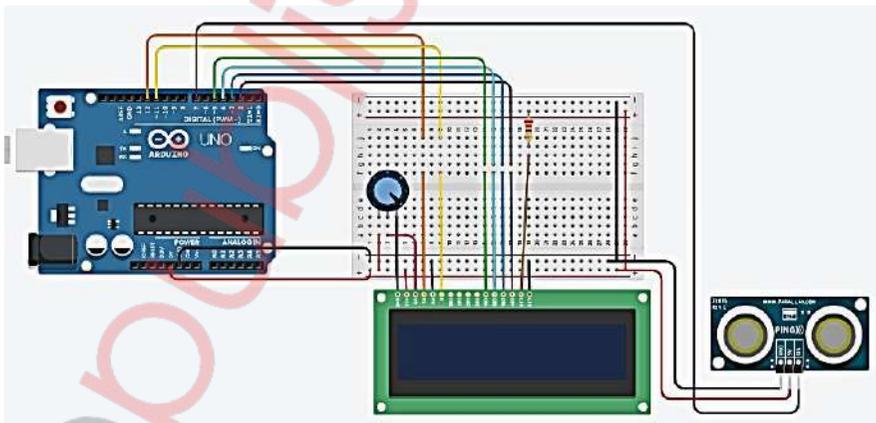
Sumber: https://www.immersa-lab.com/wp-content/uploads/2018/02/Pengertian-Sensor-Ultrasonik-PING-Dan-Jenis-Jenisnya_Featured-Image.jpg

Sensor ini merupakan sensor ultrasonik yang berfungsi untuk mengubah besaran bunyi menjadi besaran listrik dan sebaliknya. Sensor ping ini dapat mendeteksi jarak dari suatu objek dengan cara memancarkan gelombang ultrasonik dengan frekuensi 40 KHz dan kemudian mendeteksi pantulannya. Sensor ini dapat mengukur jarak antara 3 cm sampai 300 cm. Keluaran dari sensor ini berupa pulsa yang lebarnya merepresentasikan jarak. Lebar pulsanya bervariasi dari 115 us sampai 18,5 ms. Pada dasarnya, sensor PING terdiri dari sebuah *chip* pembangkit sinyal 40 KHz, sebuah speaker ultrasonik dan mikrofon ultrasonik. Speaker ultrasonik akan berfungsi sebagai pengubah sinyal 40 KHz menjadi besaran bunyi/suara dan mikrofon ultrasonik akan berfungsi untuk mendeteksi pantulan suaranya. (Sumber: <https://www.immersa-lab.com/pengertian-sensor-ultrasonik-ping-dan-jenis-jenisnya.htm>)

D. Langkah Kerja

Berikut ini adalah langkah-langkah membuat aplikasi *display* LCD 16x2 untuk membuat *timer* menggunakan Arduino Uno.

1. Siapkan alat dan bahan yang akan digunakan sesuai kebutuhan.
2. Buka *web browser* Anda dan masuk ke web simulasi *online* Tinkercad.
3. Buat rangkaian seperti Gambar 2. Rangkaian ini untuk membaca data sensor ultrasonik PING dan menampilkan data jarak pada LCD 16x2.



4. Ketikkan kode program berikut pada halaman Code (pilih mode text)

```
//=====
//SIMULASI PENGUKUR JARAK
//=====

int cm = 0;
#include <LiquidCrystal.h>
// initialize the library with the numbers of the
// interface pins
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup() {
  // set up the LCD's number of columns and rows:
  lcd.begin(16, 2);
  // set the cursor to column 0, line 0
  lcd.setCursor(0, 0);
  // Print a message to the LCD.
  lcd.print("Praktik 08");
  // set the cursor to column 0, line 1
  lcd.setCursor(0, 1);
  // Print a message to the LCD
  lcd.print("Pengukur Jarak");
  delay (1000);
  // Clear LCD Display
  lcd.clear();
}

void loop() {
  // measure the ping time in cm
  cm = 0.01723 * readUltrasonicDistance(7, 7);
  lcd.setCursor(0,0);
  lcd.print("Jarak:  ");
  lcd.setCursor(6,0);
  lcd.print(cm);
  lcd.setCursor(9,0);
  lcd.print("cm");
  delay(100); // Wait for 100 millisecond(s)
}

long readUltrasonicDistance(int triggerPin, int
echoPin)
{
  pinMode(triggerPin, OUTPUT); // Clear the trigger
  digitalWrite(triggerPin, LOW);
  delayMicroseconds(2);
  // Sets the trigger pin to HIGH state for 10
  microseconds
  digitalWrite(triggerPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(triggerPin, LOW);
```

```
pinMode(echoPin, INPUT);  
// Reads the echo pin, and returns the sound wave  
travel  
// time in microseconds  
return pulseIn(echoPin, HIGH);  
}
```

5. Jalankan simulasi dengan cara mengklik tombol “Start Simulation”.
6. Klik sensor ultrasonik PING sampai muncul lingkaran warna hijau, ini sebagai representasi benda yang akan dibaca jaraknya oleh sensor ultrasonik PING.
7. Klik lingkaran tersebut (seolah-olah ini sebagai benda yang akan diukur jaraknya). Kemudian geser-geser lingkaran tersebut dan amatilah tampilan pada LCD 16x2. Amati perubahan angka yang ditampilkan. Angka yang ditampilkan adalah data hasil pembacaan sensor.

E. Tugas Praktik

1. Buatlah desain kendali motor servo sederhana dengan ketentuan sebagai berikut:
 - Menggunakan 1 buah motor servo (seperti yang digunakan pada praktik 7)
 - Menggunakan 1 buah sensor ultrasonik PING.
 - Jika jarak yang terbaca sensor kurang dari sama dengan 50 cm maka motor berputar sebesar 90 derajat ke kanan.
 - Jika jarak yang terbaca sensor lebih dari 50 meter maka motor Kembali pada posisi 0 derajat.
2. Simpan hasil simulasi Anda pada akun Tinkercad.

F. Analisis

Tuliskan hasil analisis Anda berdasarkan hasil simulasi yang telah Anda lakukan. (Tulis pada *project* Anda menggunakan *Anotation*)

G. Kesimpulan

Tuliskan kesimpulan yang Anda dapatkan dari praktik ini. (Tulis pada *project* Anda menggunakan *Anotation*)

Bab 10

Rancang Bangun *Embedded System* Berbasis Arduino

A. Tujuan

Praktik ini bertujuan agar praktikan dapat:

1. Mengetahui dan memahami proses perancangan *embedded system*
2. Merancang suatu *embedded system* berbasis Arduino
3. Mengimplementasikan rancangan *embedded system* berbasis Arduino dalam bentuk simulasi
4. Menganalisis hasil implementasi rancangan *embedded system* berbasis Arduino

B. Alat dan Bahan

1. PC atau Laptop
2. Koneksi internet

C. Teori dan Aplikasi Dasar

Embedded system adalah sistem elektronika yang di dalamnya terdapat mikroprosesor sebagai pengendali kerja sistem. Desain *embedded system* menyebabkan dalam sistem elektronika yang didesain telah terintegrasi perangkat keras dan perangkat lunak. Perangkat keras terkait dengan komponen elektronika dan interkoneksinya dalam rangkaian. Sedangkan perangkat lunak bekerja untuk mengatur kerja bagian-bagian yang terdapat dalam rangkaian dengan komponen mikroprosesor sebagai otak dari sistem. (Sumber: <https://mce.telkomuniversity.ac.id/embedded-system/>)

DAFTAR PUSTAKA

- Arduino. (2018). *What is Arduino?* <https://www.arduino.cc/en/Guide/Introduction>
- Arduino. (2022). *Language Reference*. <https://www.arduino.cc/reference/en/>
- Elektronika, L. (2017). *CARA PROGRAM LCD KARAKTER 16x2 MENGGUNAKAN ARDUINO DAN SIMULASI PROTEUS*. <http://www.labelektronika.com/2017/03/cara-program-lcd-karakter-16x2-Arduno-dan-Proteus.html>
- Gunther Gridling, Bettina Weiss. (2007). *Introduction to Microcontroller*. Vienna Institute of Technology
- Immersia Lab. (2018). *Pengertian Sensor Ultrasonik PING dan Jenis-jenisnya*. <https://www.immersa-lab.com/pengertian-sensor-ultrasonik-ping-dan-jenis-jenisnya.htm>
- Power Up Your Imagination. (2022). <https://www.tinkercad.com/circuits>
- Rahmat, A. (2019). *Belajar Pemrograman Dasar Arduino untuk Pemula*.
- Servo Motor & Drive*. (2022). <http://www.kitomaindonesia.com/kategori/4/servo-motor-drive>
- Steven F. Barrett. (2010). *Arduino Microcontroller: Processing for Everyone! Part I*. Morgan & Claypool Publishers
- Steven F. Barrett. (2020). *Arduino I: Getting Started*. Morgan & Claypool Publishers
- Sutarsi Suhaeb, dkk. (2017). *Mikrokontroler dan Interface*. Universitas Negeri Makasar
- Team, P. R. (2022). *USB Pinout*. https://pinoutguide.com/Slots/USB_pinout.shtml
- Tutorial, E. (n.d.). *Seven Segment Display*. Retrieved October 17, 2022, from <https://www.electronics-tutorials.ws/blog/7-segment-display-tutorial.html>