

**PETUNJUK PELAKSANAAN PRAKTIKUM  
MIKROKONTROLER/MIKROPROSESOR**

**Mhd. Idham Khalif, S.Kom., M.T.**



**PROGRAM STUDI TEKNIK ELEKTRO  
UNIVERSITAS TRISAKTI  
2024**

## PRAKTIKUM 1

### MODUL 1 : PENGENALAN WOKWI, ARDUINO DAN ESP 32

#### A. TUJUAN

1. Memahami langkah-langkah akses simulator dan Install Arduino IDE
2. Memahami penggunaan simulator dan pemrograman Arduino
3. Mampu membuat program sederhana Arduino dan dapat mengimplementasikannya pada simulator

#### B. PERALATAN

1. Simulator Wokwi
2. Arduino IDE

#### C. TEORI

##### 1. PENGENALAN MIKROKONTROLER

Mikrokontroler adalah single chip computer yang memiliki kemampuan untuk diprogram dan digunakan untuk tugas-tugas yang berorientasi kontrol. Mikrokontroler datang dengan dua alasan utama, yang pertama adalah kebutuhan pasar (market need) dan yang kedua adalah perkembangan teknologi baru. Yang dimaksud dengan kebutuhan pasar adalah kebutuhan yang luas dari produk-produk elektronik akan perangkat pintar sebagai pengontrol dan pemroses data. Sedangkan yang dimaksud dengan perkembangan teknologi baru adalah perkembangan teknologi semikonduktor yang memungkinkan pembuatan chip dengan kemampuan komputasi yang sangat cepat, bentuk yang semakin mungil, dan harga yang semakin murah.

Terdapat perbedaan yang signifikan antara mikrokontroler dan mikroprosesor. Perbedaan yang utama antara keduanya dapat dilihat dari dua faktor utama yaitu arsitektur perangkat keras (hardware architecture) dan aplikasi masing-masing.

- ◆ Ditinjau dari segi arsitekturnya, mikroprosesor hanya merupakan single chip CPU, sedangkan mikrokontroler dalam IC-nya selain CPU juga terdapat device lain yang memungkinkan mikrokontroler berfungsi sebagai suatu single chip computer. Dalam sebuah IC mikrokontroler telah terdapat ROM, RAM, EPROM, serial interface dan paralel interface, timer, interrupt controller, konverter Analog ke Digital, dan lainnya (tergantung feature yang melengkapi mikrokontroler tersebut).

- ◆ Sedangkan dari segi aplikasinya, mikroprosesor hanya berfungsi sebagai Central Processing Unit yang menjadi otak komputer, sedangkan mikrokontroler, dalam bentuknya yang mungil, pada umumnya ditujukan untuk melakukan tugas-tugas yang berorientasi kontrol pada rangkaian yang membutuhkan jumlah komponen minimum dan biaya rendah (low cost).

Karena kemampuannya yang tinggi, bentuknya yang kecil, konsumsi dayanya yang rendah, dan harga yang murah maka mikrokontroler begitu banyak digunakan di dunia. Mikrokontroler digunakan mulai dari mainan anak-anak, perangkat elektronik rumah tangga, perangkat pendukung otomotif, peralatan industri, peralatan telekomunikasi, peralatan medis dan kedokteran, sampai dengan pengendali robot serta persenjataan militer. Terdapat beberapa keunggulan yang diharapkan dari alat-alat yang berbasis mikrokontroler (microcontroller-based solutions) yaitu:

- ◆ Kehandalan tinggi (high reliability) dan kemudahan integrasi dengan komponen lain (high degree of integration)
- ◆ Ukuran yang semakin dapat diperkecil (reduced in size)
- ◆ Penggunaan komponen dipersedikit (reduced component count) yang juga akan menyebabkan biaya produksi dapat semakin ditekan (lower manufacturing cost)
- ◆ Waktu pembuatan lebih singkat (shorter development time) sehingga lebih cepat pula dijual ke pasar sesuai kebutuhan (shorter time to market)
- ◆ Konsumsi daya yang rendah (lower power consumption)

Mikrokontroler ini bisa diprogram untuk sebuah perintah tertentu. Mikrokontroler memiliki chip memory untuk penyimpanan data yang digunakan untuk menyimpan data sebuah program. Program ini merupakan sebuah program yang dibuat secara manual oleh manusia dengan susunan dan tujuan tertentu. Agar dapat digabungkan dengan mikrokontroler tersebut, maka program yang dibuat oleh manusia ini harus diubah ke dalam bentuk hexa atau pun binery. Istilah ini bisa dibilang dengan meng-Compile sebuah data yang ada.

Mikrokontroler memiliki keunggulan yaitu tersedianya RAM dan peralatan I/O pendukung yang membuat ukuran board mikrokontroler ini menjadi lebih praktis. Mikrokontroler merupakan sebuah sistem yang memiliki fungsi untuk pengontrol susunan dari alat elektronik dan secara garis besar dapat menyimpan program yang ada. Mikrokontroler terbagi atas Memori, I/O khusus, ROM, Timers (pewaktu), dan unsur pendukung seperti ADC atau (Analog Digital Converter) yang sudah terjamin

pada isi dalamnya. Pada sebuah sistem otomatisasi yang berbasis Internet Of Things, mikrokontroler digunakan sebagai pengendali utama dalam sistem.

## **2. PENGENALAN SIMULASI WOKWI**

Wokwi adalah simulator Elektronik online. Anda dapat menggunakannya untuk mensimulasikan Arduino, ESP32, STM32, dan banyak papan, komponen, dan sensor populer lainnya.

### **Kenapa menggunakan wokwi:**

1. Tidak perlu mengunduh perangkat lunak berukuran besar. Browser Anda memiliki semua yang Anda perlukan untuk mulai mengkode proyek IoT Anda.
2. Anda tidak perlu kawatir menghancurkan/merusak perangkat keras pada saat percobaan. Dapat mengurangi resiko kegagalan atau kerusakan perangkat.
3. Tidak perlu menambahkan library Arduino ke dalam aplikasi, keran ketersedian mikrokontroler sudah tersedia pada aplikasi.
4. Support Wifi simulasi yang dapat menggunakan protokol IoT seperti MQTT dan HTTP
5. Memungkinkan untuk sharing kode dan simulasi secara internet

### **Fitur pada wokwi :**

1. **Simulasi WiFi** - Hubungkan proyek simulasi Anda ke internet. Anda dapat menggunakan MQTT, HTTP, NTP, dan banyak protokol jaringan lainnya.
2. **Virtual Logic Analyzer** - Menangkap sinyal digital dalam simulasi Anda (misalnya UART, I2C, SPI) dan menganalisisnya di komputer Anda.
3. **Simulasi kartu SD** - Simpan dan ambil file dan direktori dari kode Anda.
4. **Chips API** - Buat chip dan suku cadang khusus Anda sendiri.
5. **Integrasi Visual Studio Code** - Simulasikan proyek tersebut langsung dari VS Code.

### **Supported Hardware :**

Family	Microcontrollers
AVR	ATmega328P ( <a href="#">Arduino Uno</a> ), ATmega2560 ( <a href="#">Arduino Mega</a> ), ATtiny85
ESP32	<b>Xtensa:</b> ESP32, ESP32-S2, ESP32-S3 <b>RISC-V:</b> ESP32-C3, ESP32-C6, ESP32-H2*
STM32	<a href="#">STM32C031</a> , <a href="#">STM32L031</a>
Pi Pico	RP2040 (Raspberry Pi Pico), a dual-core ARM Cortex-M0+ microcontroller

Tabel 1.1 Dukungan mikrokontroler pada wokwi

Part	Description
HC-SR04	HC-SR04 Ultrasonic Distance Sensor
DHT22	Digital Humidity and Temperature sensor
DS1307 RTC	RTC (Real Time Clock) module with I2C interface and 56 bytes of NV SRAM
PIR Motion Sensor	Passive Infrared (PIR) motion sensor
Analog Temperature Sensor (NTC)	Analog temperature sensor: NTC (negative temperature coefficient) thermistor
DS18B20 Temperature Sensor	One-Wire digital Temperature Sensor
MPU6050	Integrated sensor with 3-axis accelerometer, 3-axis gyroscope and a temperature sensor with I2C interface
Photoresistor	Photoresistor (LDR) Sensor
HX711 Load Cell	HX711 Load Cell Amplifier with 5kg/50kg/gauge load cell

Tabel 1.2 Dukungan sensor pada wokwi

Part	Description
Pushbutton	12mm Tactile Switch Button (momentary push button)
Slide switch	Standard Single Pole Double Throw (SPDT) slide switch
DIP Switch 8	Set of 8 electrical switches in a single package
Keypad	A standard 4x4 keypad (for numeric input)
Analog Joystick	Analog Joystick with two axes (horizontal/vertical) and an integrated push button
Potentiometer	Knob-controlled variable resistor (linear potentiometer)
Slide Potentiometer	Sliding variable resistor (linear potentiometer)
Rotary Encoder (KY-040)	KY-040 Rotary Encoder module with 20 steps per revolution.

Tabel 1.3 Dukungan perangkat input pada wokwi

Part	Description
LCD 1602	An LCD with 2 lines, 16 characters per line
LCD 2004	An LCD with 4 lines, 20 characters per line
ILI9341 2.8" TFT-LCD display	Full color 240x320 2.8" LCD-TFT display with SPI interface
ILI9341 2.8" Touch Screen LCD	ILI9341 LCD with FT6206 I2C capacitive touch controller
SSD1306 OLED	Monochrome 128x64 OLED display with I2C interface
MAX7219 LED Dot Matrix	8x8 LED Dot Matrix with MAX7219 Controller
Seven Segment Display	Seven segment LED display (1 to 4 digits)
TM1637 Seven Segment Display	4 digit seven segment module with TM1637 4-wire interface
2.9" e-Paper Display	2.9" e-Paper Display
PAL TV	Black and White analog PAL TV screen

Tabel 1.4 Dukungan perangkat layar pada wokwi

Part	Description
Servo	A standard Micro Servo Motor
Bipolar Stepper Motor	A bipolar Stepper Motor
A4988	A4988 Stepper Motor Driver
Biaxial Stepper Motor	A concentric biaxial stepper motor, containing two stepper motors packaged in a single enclosure

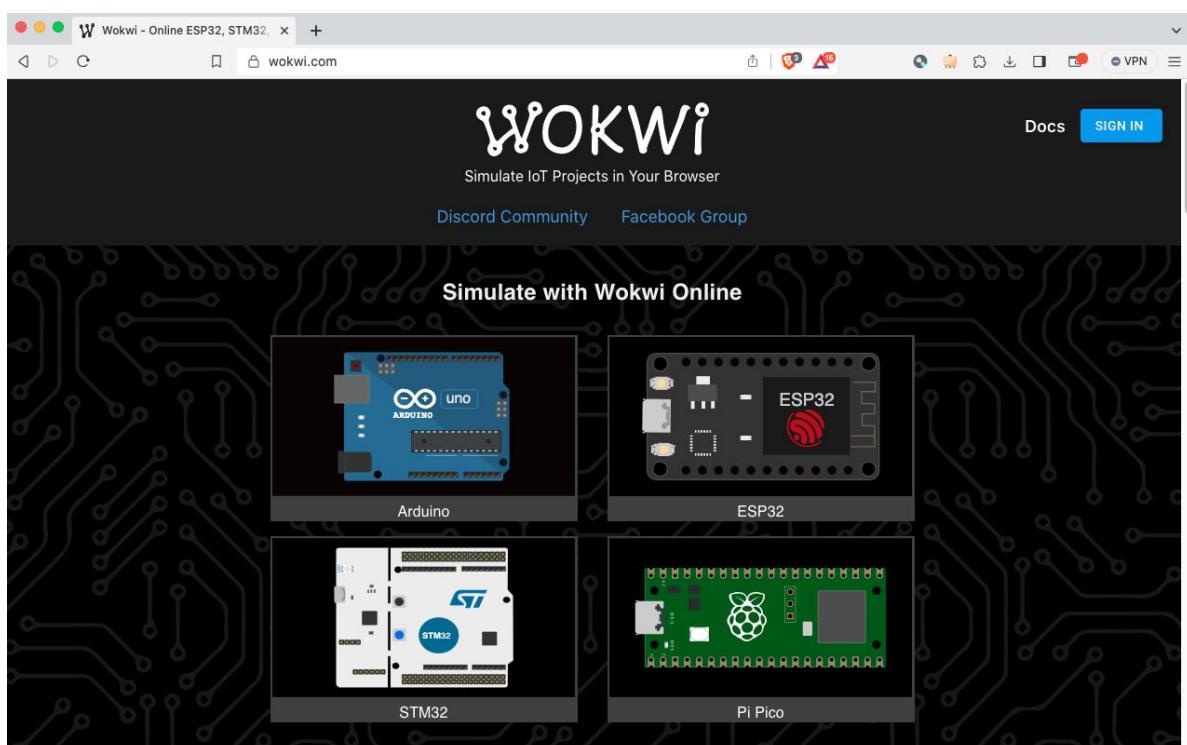
Tabel 1.5 Dukungan motor pada wokwi

Part	Description
NOT gate	NOT gate
AND gate	AND gate
OR gate	OR gate
XOR gate	XOR gate
NAND gate	NAND gate
MUX	MUX
Flip-Flop D	Flip-Flop D
Flip-Flop DSR	Flip-Flop DSR
74HC595 Shift Register	8-bit Serial-In Parallel-Out (SIPO) Shift Register
74HC165 Input Shift Register	8-bit Parallel-In Serial-Out (PISO) Shift Register (Input)

Tabel 1.6 Dukungan gerbang logika pada wokwi

Part	Description
Resistor	Resistor
Buzzer	A piezoelectric buzzer
Relay Module	Relay Module
DPDT Relay	Double Pole Double Throw (DPDT) Relay
Breadboard	Standard, half and mini sizes available
Logic Analyzer	8-Channel Digital Logic Analyzer
microSD card	microSD card with SPI interface

Tabel 1.7 Dukungan perangkat lainnya pada wokwi



Gambar 1.1 Tampilan halaman awal wokwi

### 3. PENGENALAN ARDUINO

Arduino adalah platform open source yang bekerja seperti mikrokontroler sehingga memerlukan kemampuan hardware dan programming. Arduino dapat digunakan untuk pembuatan rangkaian LED simple, aplikasi wearable, IoT, serta 3D printing. Semua bagian Arduino merupakan open source sehingga dapat dikembangkan oleh berbagai kalangan mulai dari akademisi, professional, maupun hobbyist. Arduino sangat memudahkan proses implementasi dibandingkan dengan mikrokontroler biasa. Berikut keunggulan yang ditawarkan oleh Arduino:

- Murah
- Simple, mudah digunakan untuk beginner
- Open source secara software, tersedia extension, bisa menggunakan bahasa Arduino (Processing) maupun AVR-C
- Open source secara hardware, bisa mendesain modul sendiri



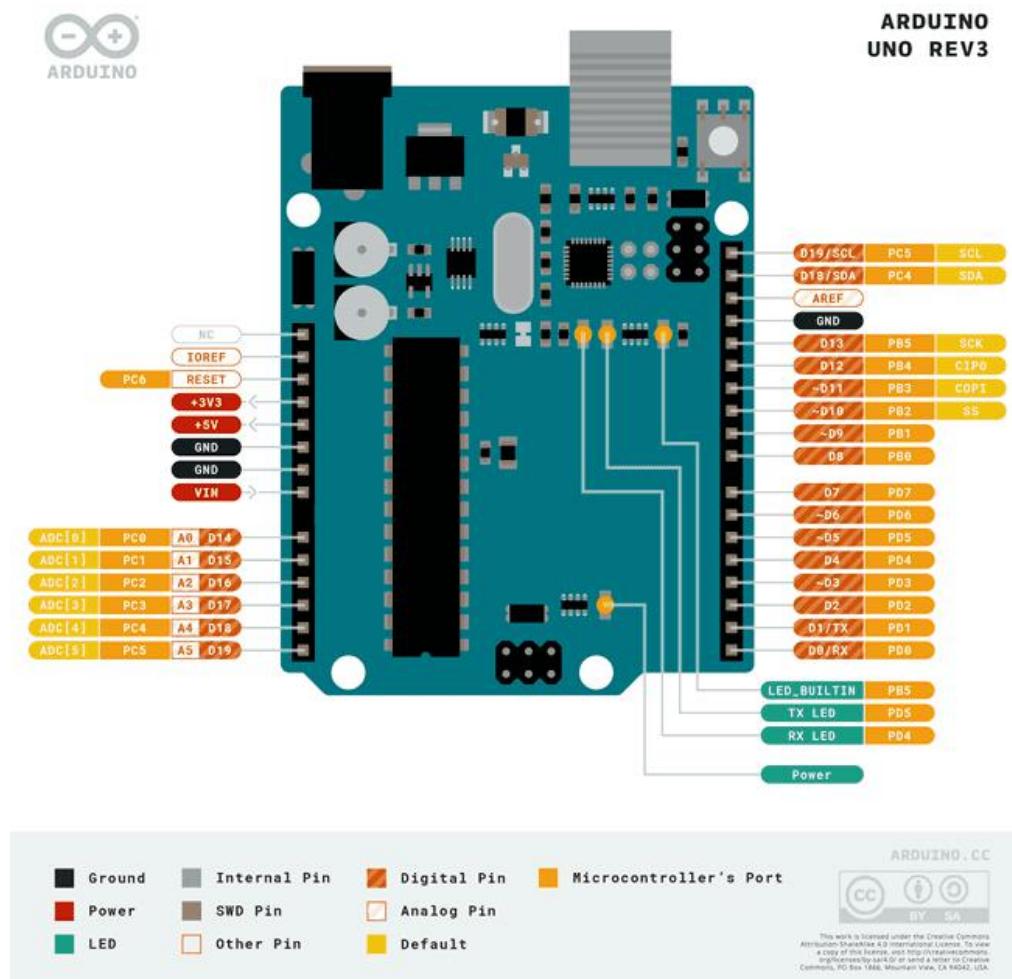
Gambar 1.2 Perangkat Arduino UNO

Arduino memiliki pin analog dan digital. Pin digital pada Arduino dapat dibagi menjadi nilai 1 dan 0. Nilai 1 terjadi saat kondisi pin bertegangan 5 volt sedangkan nilai 0 saat tegangan 0 volt. Pin ini sangat berguna apabila diaplikasikan pada device I/O sehingga dapat diatur behavior I/O yang diinginkan (on/off).

Pin analog pada Arduino menerapkan cara yang sama seperti pin digital yaitu mengubah nilai tegangan 0-5 volt menjadi besaran lain. Akan tetapi, nilai besaran yang dimaksud tidak hanya

berkisar 1 dan 0 saja. Pada pin analog, nilai besaran dapat berkisar antara 0-255 atau 0-1023.

Berikut adalah spesifikasi lengkap Arduino uno yang akan digunakan pada praktikum system mikroprosesor.



Gambar 1.3 PIN pada Arduino UNO

<b>Microcontroller</b>	ATmega328P
<b>Operating Voltage</b>	5V
<b>Input Voltage (recommended)</b>	7-12V
<b>Input Voltage (limit)</b>	6-20V
<b>Digital I/O Pins</b>	14 (of which 6 provide PWM output)
<b>PWM Digital I/O Pins</b>	6
<b>Analog Input Pins</b>	6
<b>DC Current per I/O Pin</b>	20 mA
<b>DC Current for 3.3V Pin</b>	50 mA
<b>Flash Memory</b>	32 KB (ATmega328P) of which 0.5 KB used by bootloader
<b>SRAM</b>	2 KB (ATmega328P)
<b>EEPROM</b>	1 KB (ATmega328P)
<b>Clock Speed</b>	16 MHz
<b>LED_BUILTIN</b>	13
<b>Length</b>	68.6 mm
<b>Width</b>	53.4 mm
<b>Weight</b>	25 g

Tabel 1.8 Spesifikasi Arduino UNO

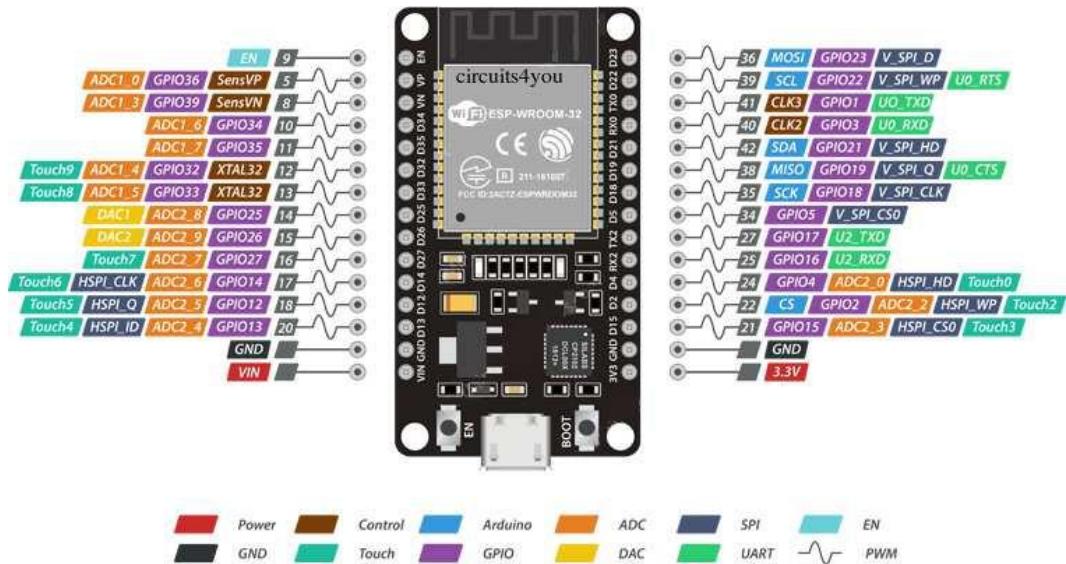
#### 4. PENGENALAN ESP32

Espressif Systems. Ini adalah penerus ESP8266 dan memiliki sejumlah fitur canggih yang menjadikannya pilihan yang sangat baik untuk berbagai proyek IoT (Internet of Things) dan aplikasi nirkabel. Berikut adalah deskripsi singkat tentang ESP32:

- ◆ Mikrokontroler: ESP32 adalah sebuah mikrokontroler yang memiliki unit pemrosesan pusat (CPU), memori, dan sejumlah pin input/output (I/O). Ini memungkinkan Anda untuk mengendalikan berbagai perangkat elektronik dan mengambil keputusan berdasarkan berbagai input sensor.
- ◆ Dual-Core CPU: Salah satu fitur unggulan ESP32 adalah kehadiran dua core CPU Xtensa 32-bit, yang memungkinkan eksekusi tugas-tugas secara paralel. Ini berguna untuk mengatasi tugas-tugas yang memerlukan kinerja tinggi atau untuk memisahkan tugas real-time dan non-real-time.
- ◆ Komunikasi Nirkabel: ESP32 dilengkapi dengan modul Wi-Fi 802.11 b/g/n untuk koneksi ke jaringan nirkabel dan Bluetooth 4.2 / Bluetooth Low Energy (BLE) untuk berkomunikasi dengan perangkat Bluetooth lainnya. Ini membuatnya cocok untuk proyek IoT yang memerlukan konektivitas nirkabel.

- ◆ Pin I/O yang Banyak: ESP32 memiliki banyak pin I/O yang dapat digunakan untuk menghubungkan berbagai perangkat dan sensor. Pin-pin ini dapat digunakan untuk berbagai fungsi seperti PWM, I2C, SPI, dan UART.
- ◆ Memori: ESP32 memiliki memori flash internal yang cukup besar untuk menyimpan program dan data. Ini juga dilengkapi dengan RAM yang memadai untuk menjalankan program yang memerlukan banyak sumber daya.
- ◆ RTC (Real-Time Clock): ESP32 memiliki RTC internal yang memungkinkan perangkat untuk menghitung waktu dengan akurat, bahkan dalam mode tidur. Ini berguna untuk aplikasi yang memerlukan manajemen waktu yang tepat.
- ◆ Konsumsi Daya Rendah: ESP32 dirancang dengan konsumsi daya yang rendah, yang menjadikannya cocok untuk perangkat baterai yang beroperasi dalam waktu lama. Ini memiliki berbagai mode tidur yang dapat diaktifkan untuk mengoptimalkan konsumsi daya.
- ◆ Lingkungan Pengembangan: Anda dapat memprogram ESP32 menggunakan berbagai bahasa pemrograman, termasuk Arduino IDE, MicroPython, dan ESP-IDF (IoT Development Framework) dari Espressif.
- ◆ Komunitas yang Besar: ESP32 memiliki komunitas pengembang yang besar dan aktif, yang berarti Anda dapat dengan mudah menemukan dukungan, dokumentasi, dan berbagai proyek-proyek sumber terbuka dengan komunitas ini.

ESP32 telah digunakan dalam berbagai proyek, seperti kendali perangkat rumah pintar, sistem pengawasan, kendaraan otonom, dan banyak lagi. Kombinasi kemampuan Wi-Fi, Bluetooth, dan kemampuan pemrosesan yang kuat menjadikannya salah satu pilihan utama untuk pengembangan perangkat IoT. Berikut adalah spesifikasi dari ESP32.



ESP32 Dev. Board / Pinout

Gambar 1.4 PIN pada ESP32

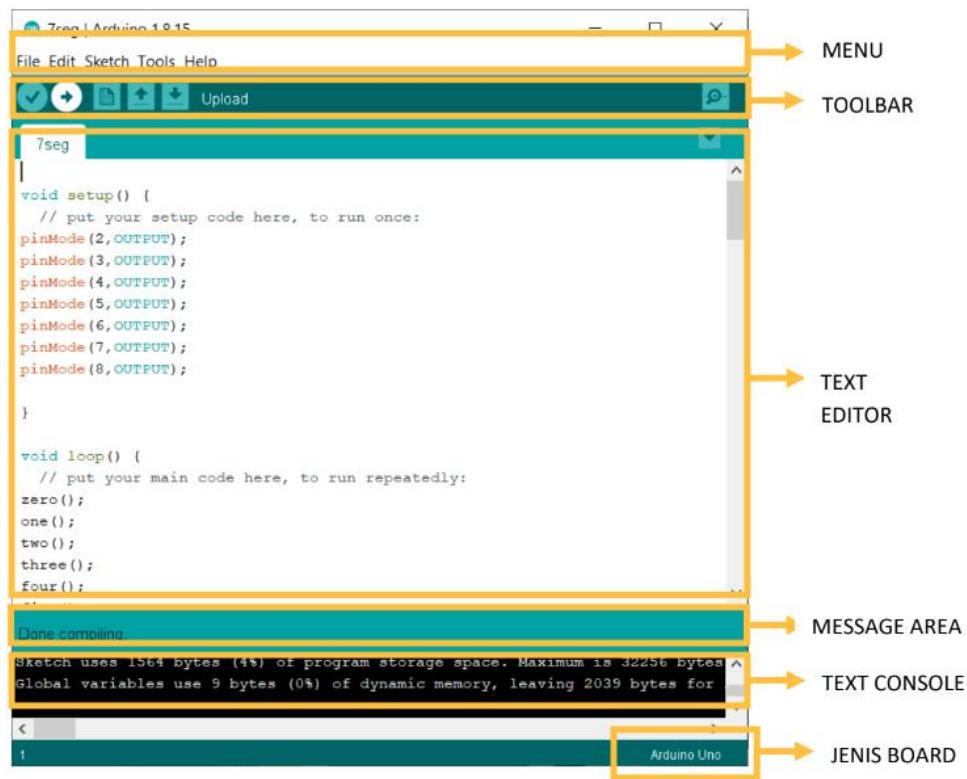
Sumber : [circuits4you.com](http://circuits4you.com)

Fitur	Deskripsi
Mikrokontroler	Tensilica Xtensa Dual-Core 32-bit LX6
Kecepatan CPU	Hingga 240 MHz
Memori Program (Flash)	4 MB (32 Mb) memori flash internal
Memori RAM	520 KB SRAM (tidak termasuk dalam flash)
Wi-Fi	802.11 b/g/n (Wi-Fi 4)
Bluetooth	Bluetooth 4.2 / Bluetooth Low Energy (BLE)
Jumlah Pin I/O	Sekitar 36 pin I/O digital yang dapat dikonfigurasi
Pin I/O Fungsional	GPIO, PWM, I2C, SPI, UART, ADC, DAC, dan banyak lagi
Input ADC	Hingga 18 kanal ADC dengan resolusi 12-bit
Modul RTC (Real-Time Clock)	RTC internal dengan kemampuan manajemen waktu
USB	1x USB-UART untuk pemrograman dan debug
Konsumsi Daya Rendah	Mode tidur yang beragam untuk menghemat daya
Keamanan	Dukungan untuk enkripsi dan tanda tangan perangkat
Sistem Operasi (OS)	Dukungan untuk FreeRTOS
Tegangan Kerja	2.2 V - 3.6 V
Konsumsi Daya (Operasi Normal)	Sekitar 80 mA
Ukuran	Modul ESP32 biasanya sekitar 25 mm x 18 mm
Lingkungan Pengembangan	Dukungan untuk Arduino IDE, MicroPython, ESP-IDF, dll.

Tabel 1.9 Spesifikasi ESP32

## 5. ARDUINO IDE

Arduino IDE adalah open source software yang digunakan untuk menulis kode sehingga dapat diupload pada board Arduino manapun. Arduino IDE berisi text editor untuk menulis code, message area, text console, toolbar, dan beberapa menu.



Gambar 1.5 Aplikasi IDE Arduino

## 6. PEMROGRAMAN DENGAN ARDUINO

Bahasa pemrograman Arduino mengikuti programming language yang simple bernama processing. Bahasa pemrograman ini memiliki struktur yang mirip dengan Bahasa C. Script Bahasa pemrograman Arduino dapat ditulis di Arduino IDE, setelah itu baru dapat di upload ke board Arduino.

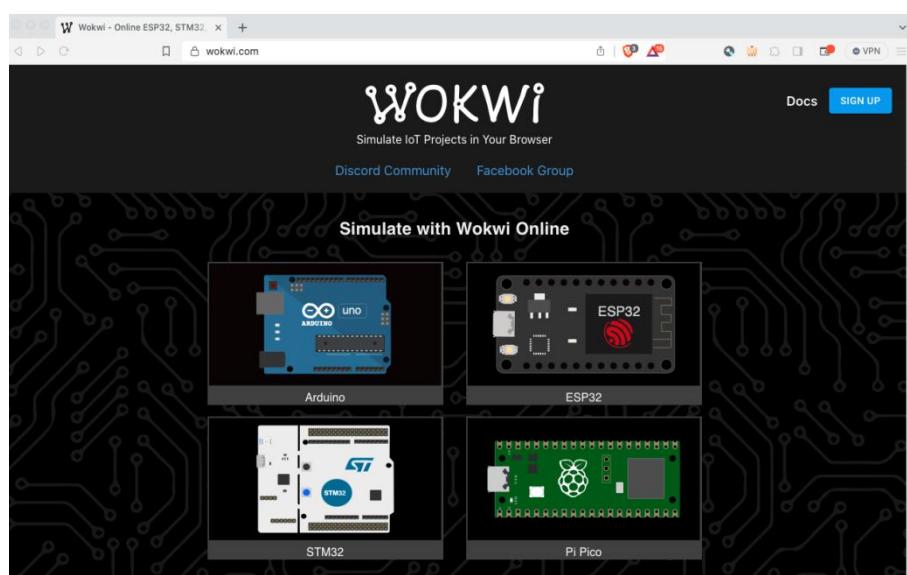
```
void setup() {  
    // put your setup code here, to run once:  
  
}  
  
void loop() {  
    // put your main code here, to run repeatedly:  
  
}
```

Bahasa pemrograman Arduino mempunyai struktur yang terdiri atas preparation dan execution. Preparation adalah bagian yang akan dieksekusi pertama kali dan digunakan untuk mendefinisikan mode pin. Bagian ini juga berfungsi untuk memulai komunikasi serial sehingga apabila tidak dipanggil akan menyebabkan error. Setelah bagian preparation, maka bagian execution akan dijalankan. Execution akan membaca input, mendefinisikan output, serta membuat kondisi tertentu.

## D. PERSIAPAN

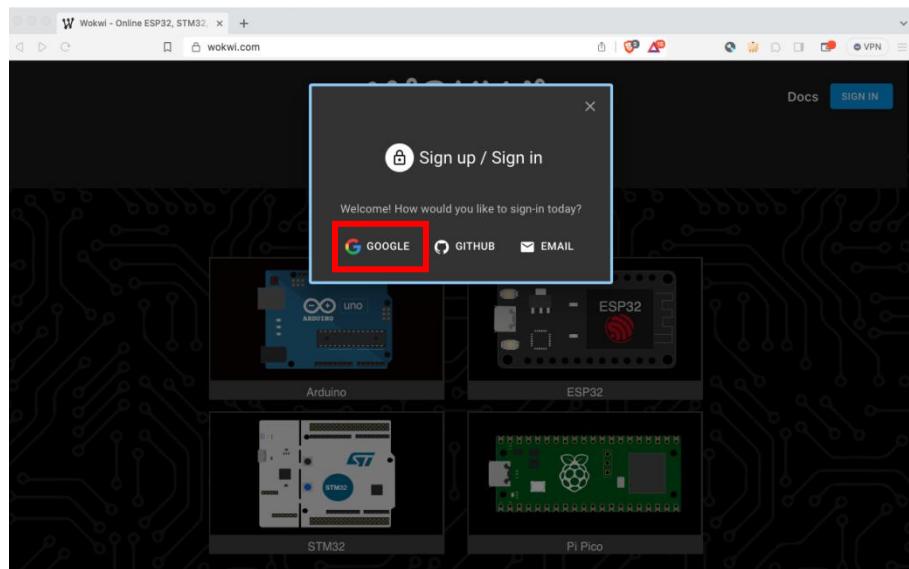
### 1. MEMBUAT AKUN PADA WOKWI

a). Silahkan akses halaman web <https://wokwi.com/> pada browser Anda, maka akan muncul halaman seperti berikut ini.



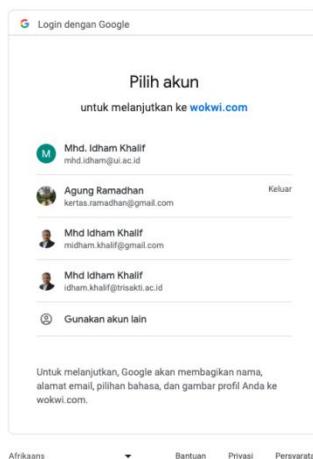
Gambar 1.6 Halaman awal wokwi

b). Pada pojok kanan atas ada tombol SIGN IN/SIGN UP, silahkan pilih SIGN UP/SIGN IN dengan akun gmail dan silahkan pilih akun gmail Anda.



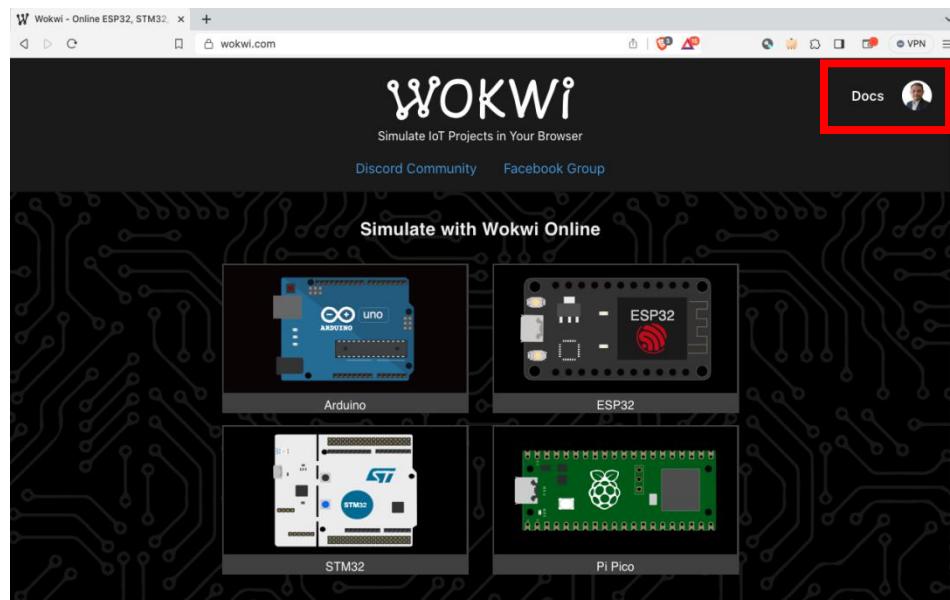
Gambar 1.7 Halaman SIGN UP dan SIGN IN

[accounts.google.com/o/oauth2/auth/oauthchooseaccount?response\\_type=code&client\\_id=706751742332-j19tcjbihgdujmv7q1pc594pq0r7j.apps.googleusercontent.co...](https://accounts.google.com/o/oauth2/auth/oauthchooseaccount?response_type=code&client_id=706751742332-j19tcjbihgdujmv7q1pc594pq0r7j.apps.googleusercontent.co...)



Gambar 1.8 Pilihan akun pada gmail

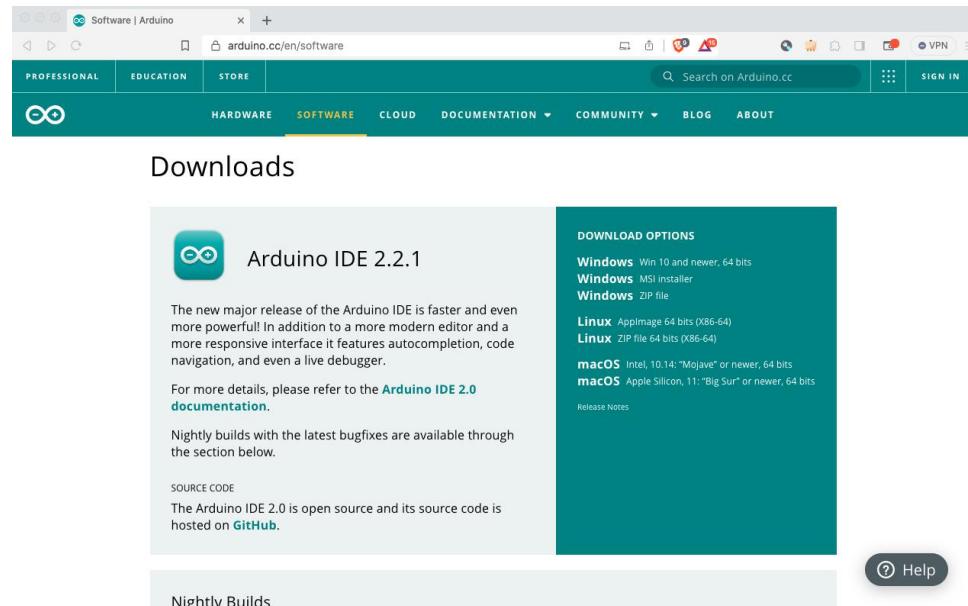
c). Jika sudah berhasil login akan ada tampilan akun Anda pada pojok kanan atas.



Gambar 1.9 Tampilan jika berhasil SIGN IN

## 2. INSTALASI ARDUINO IDE

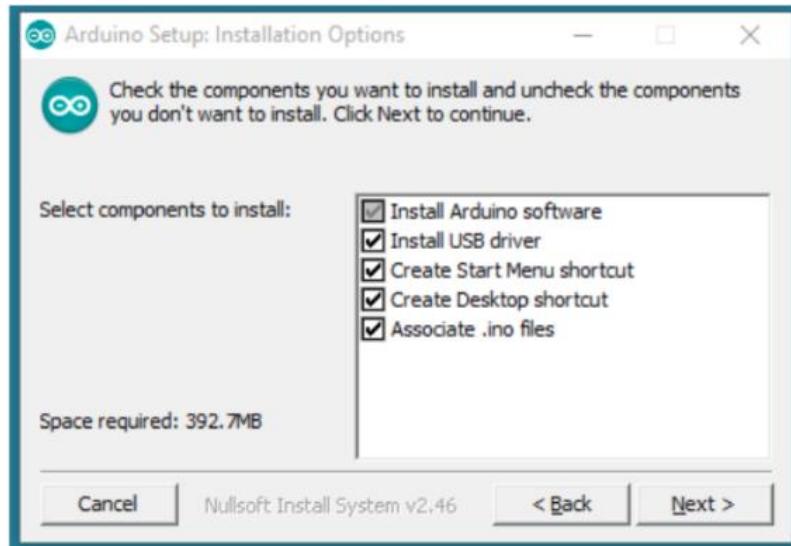
1). Silahkan akses web <https://www.arduino.cc/> pilih menu **Software** dan silahkan pilih sesuai dengan *Operating System* yang digunakan untuk mendownload Aduino IDE dan untuk windows silahkan pilih yang MSI Installer untuk memudahkan menginstal.



Gambar 1.10 Halaman download IDE Arduino

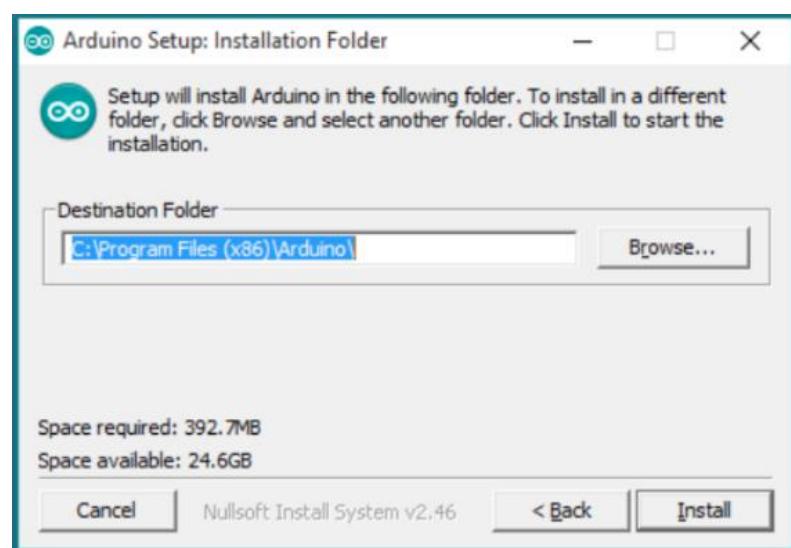
2). Jalankan file instalasi yang telah didownload sehingga terbuka kotak dialog Intallation Options.

Pilihlah komponen yang akan diinstall.



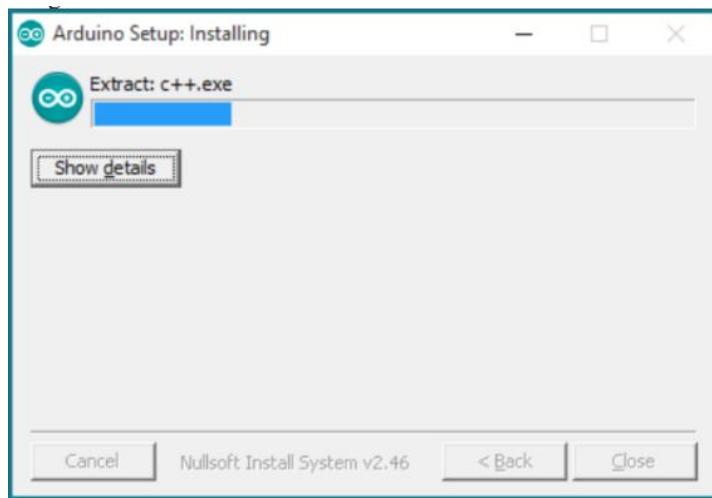
Gambar 1.11 Tampilan Setup Instalasi IDE Arduino

3). Pilih direktori penyimpanan.



Gambar 1.12 Pilih direktori penyimpanan

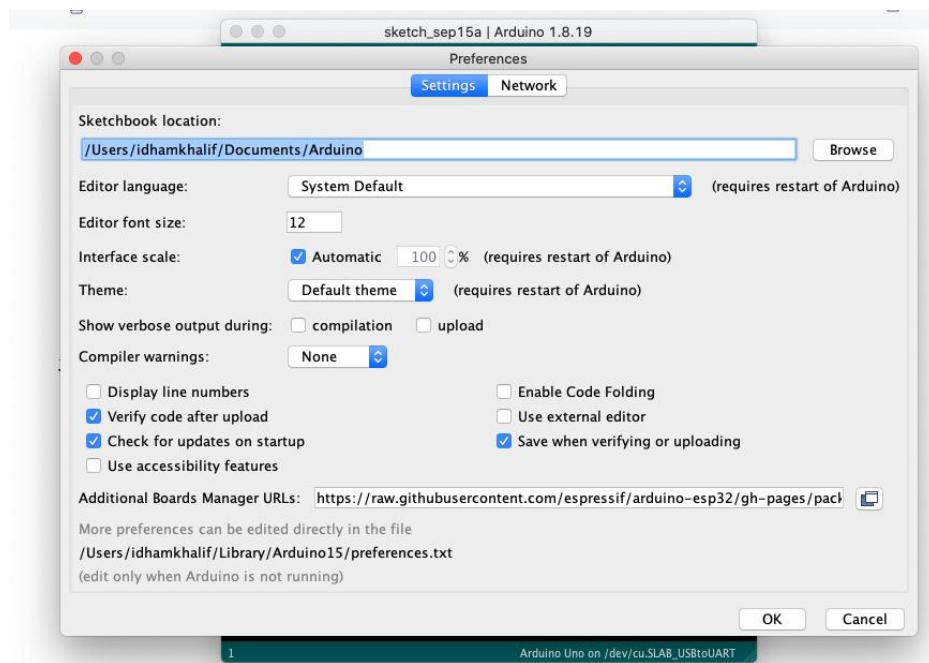
- 4). Proses akan berjalan mengekstrak dan menginstall file yang dibutuhkan untuk mengeksekusi Arduino IDE.



Gambar 1.13 Proses instalasi IDE Arduino

### 3. INSTALLASI *BOARD* ESP32 PADA IDE ARDUINO

- 1). Buka IDE Arduino, silahkan masuk ke menu **File >> Preferences**, seperti tampilan berikut ini.



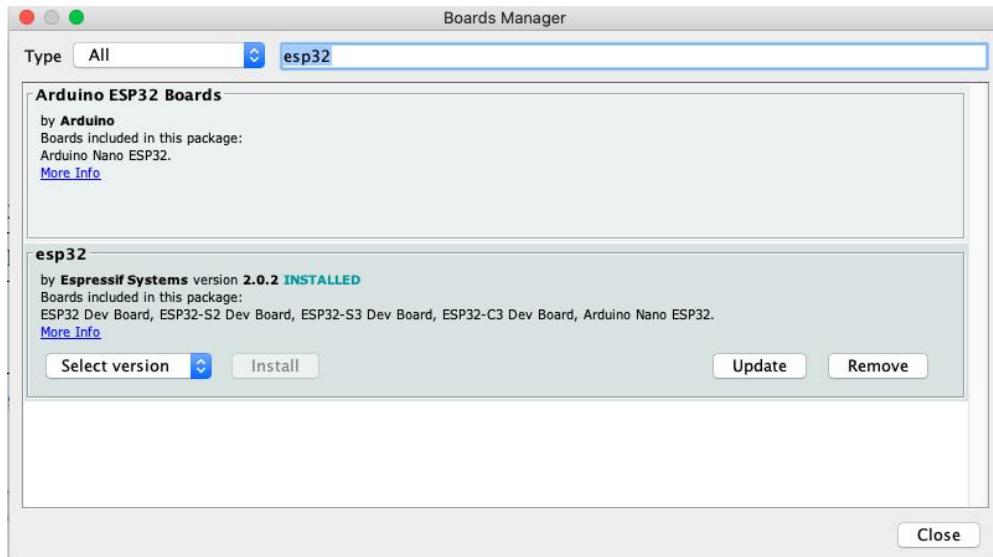
Gambar 1.14 Menu Preferences pada IDE Arduino

- 2). Pada **Additional Boards Manager URLs** isikan link berikut ini dan klik Ok.

[https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package\\_esp32\\_index.json](https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json)

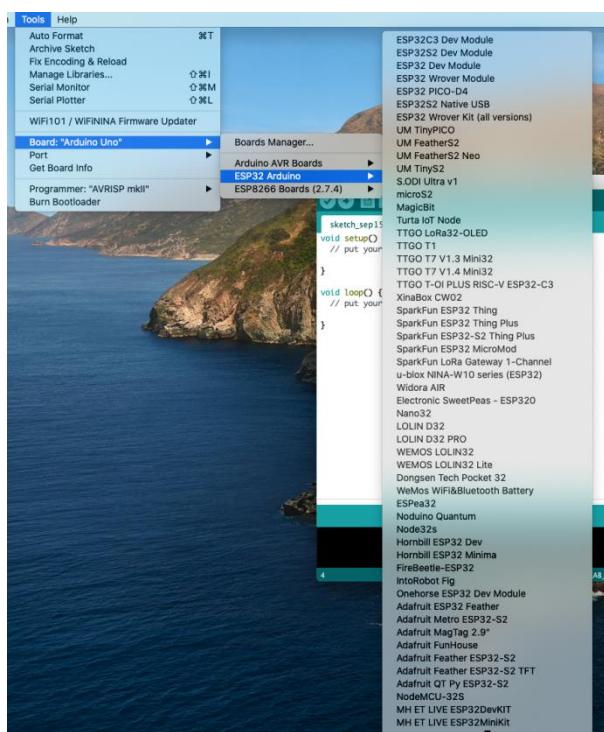
3). Buka Boards Manager: Klik pada menu "Tools" dan pilih "Board" > "Boards Manager". Ini akan membuka jendela Boards Manager.

4). Jika tampilan Boards Manager telah tampil silahkan cari "esp32", pilih versi yang terbaru dan pilih install, seperti pada gambar berikut ini.



Gambar 1.15 Boards Manager IDE Arduino

5). Jika sudah terinstall, pada menu Tools >> Boards, akan ada pilihan ESP32 dan macam - macam boards dari ESP32, seperti pada gambar berikut ini.

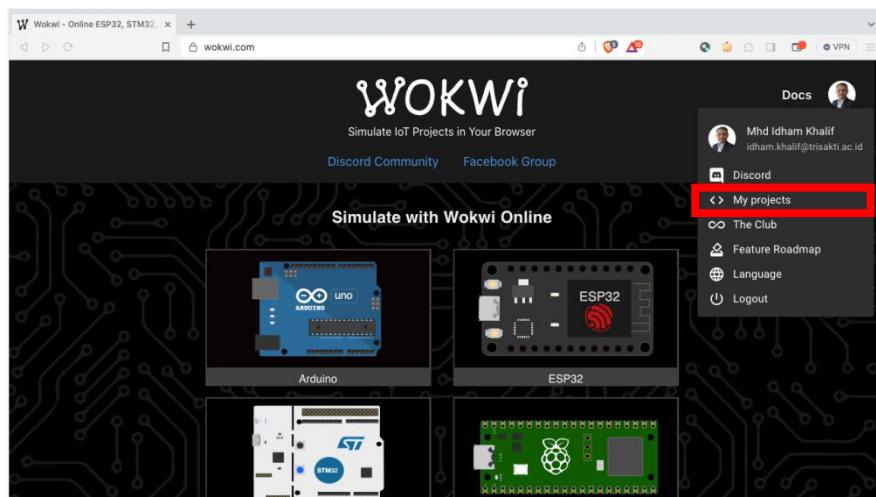


Gambar 1.16 Boards ESP32 pada IDE Arduino

## E. PERCOBAAN

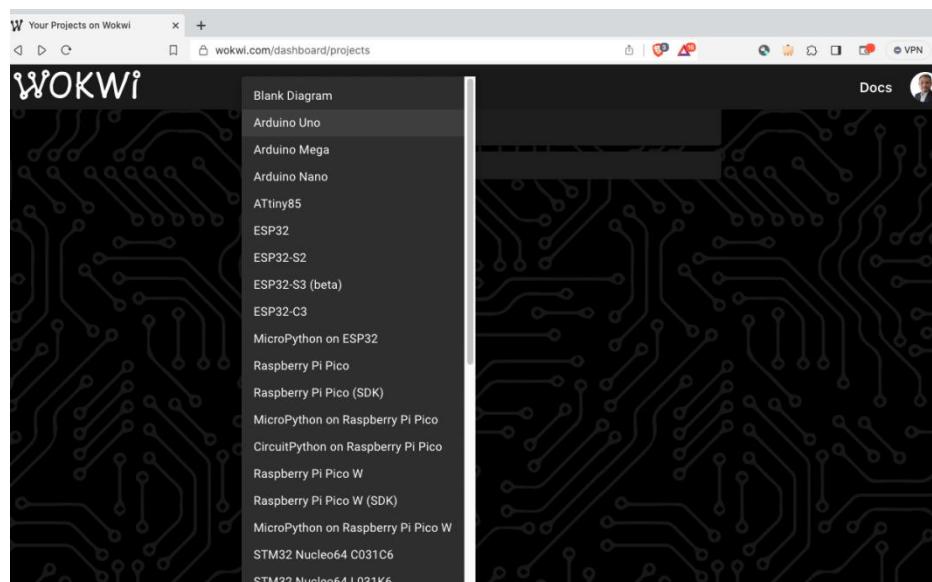
### 1. MENAAMPILKAN KATA PADA SERIAL MONITOR

- 1). Pada halaman awal wokwi yang sudah SIGN IN, silahkan pilih gambar akun Anda >> My projects, seperti gambar berikut ini.

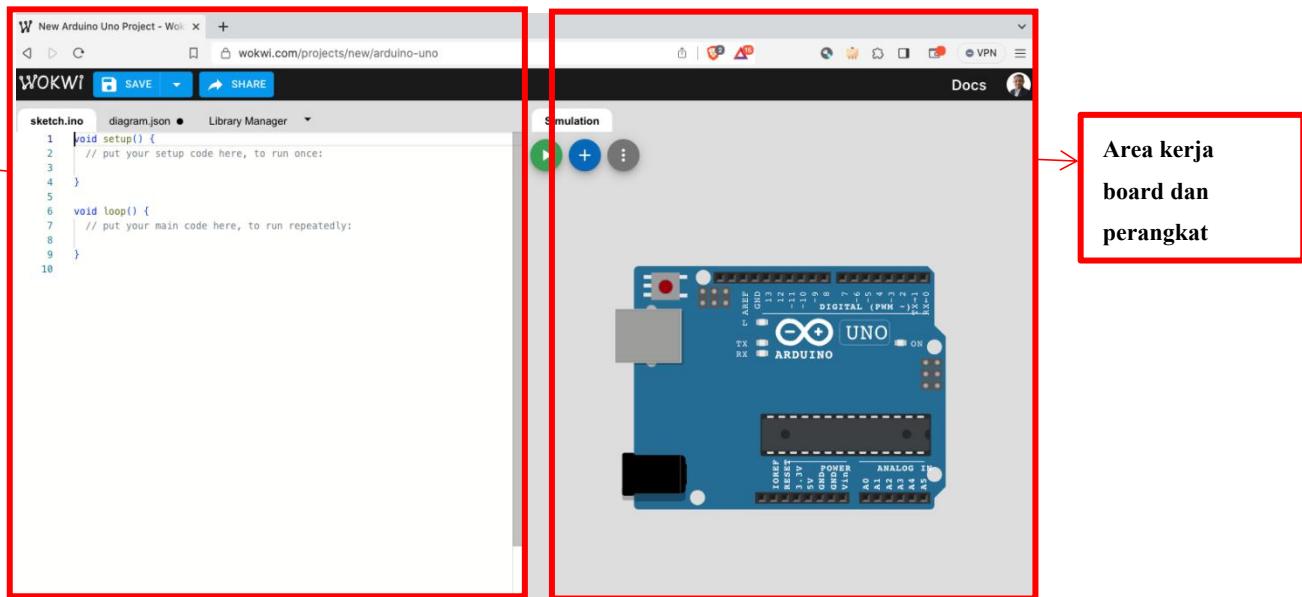


Gambar 1.17 Halaman awal wokwi

- 2). Pada halaman Projects silahkan pilih **New Project >> pilih Arduino Uno** sebagai board yang digunakan, dan nanti akan diarahkan ke halaman kerja project Anda, seperti gambar berikut.



Gambar 1.18 Pilihan board yang digunakan



Gambar 1.19 Area kerja simulasi

3). Pada area kerja kode program silahkan tuliskan kode program seperti berikut ini.

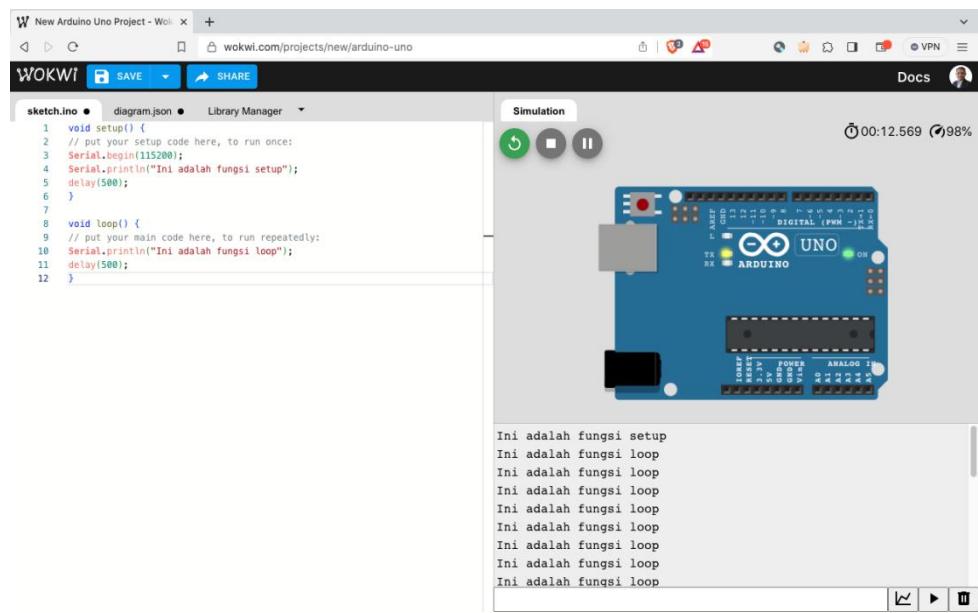
Kode Program 1.1

```

1 void setup() {
2     // put your setup code here, to run once:
3     Serial.begin(115200);
4     Serial.println("Ini adalah fungsi setup");
5     delay(500);
6 }
7
8 void loop() {
9     // put your main code here, to run repeatedly:
10    Serial.println("Ini adalah fungsi loop");
11    delay(500);
12 }
```

Pada kode Program 1.1 terdapat 2 fungsi yaitu fungsi setup dan fungsi loop, yang dimana fungsi setup akan dijalankan 1 kali pada saat board dijalankan sedangkan fungsi loop akan dijalankan secara terus menerus selama simulasi dijalankan atau board masih menyala. Pada baris 4 dan baris 10 adalah kata yang akan ditampilkan pada serial monitor kata "**Ini adalah fungsi setup**" dan "**Ini adalah fungsi loop**" bisa Anda ganti sesuai dengan kebutuhan. Baris 5 dan 11 adalah syntax delay untuk mengatur kecepatan tampilan kata pada serial monitor.

4). Pada area kerja board terdapat tombol Play, untuk menjalankan simulasi, jika tidak ada error hasil dari simmulasi seperti pada gambar berikut ini.



Gambar 1.20 Hasil simulasi yang dijalankan

## PRAKTIKUM 2

### MODUL 2 : PENGGUNAAN LED PADA MIKROKONTROLER

#### A. TUJUAN

1. Mampu membuat berbagai variasi program LED menggunakan mikrokontroler

#### B. PERALATAN

1. Simulasi Wokwi
2. Board Arduino UNO
3. LED

#### C. TEORI

##### 1. LED

LED adalah singkatan dari Light Emitting Diode merupakan semiconductor yang dapat mengeluarkan cahaya jika diberikan arus listrik. LED sering digunakan dalam dunia elektro terutama sebagai indicator. Selain itu, LED dapat juga berfungsi untuk project IoT. Pada LED terdapat 2 kaki yaitu kaki Katoda (-) dan Andoa (+) yang dapat dihubungkan pada pin di mikrokontroler.



Gambar 2.1 LED dan Kaki Katoda dan Anoda

##### 2. LOOP

###### 1). While Loop

While loop merupakan fungsi perulangan yang akan terus berjalan sampai condition bernilai false. Sehingga condition harus diatur titik berhentinya sehingga tidak terjadi infinite loop. Condition dapat berupa variable yang bisa diincrement atau decrement dan variable external (berasal dari sensor).

Kode Program 2.1

```
1 while(condition){  
2 //Statements..  
3 }
```

Contoh penggunaan:

Kode Program 2.2

```
1 while(var < 200){  
2 //Statements..  
3 var++;  
4 }
```

## 2). For Loop

Perulangan for menggunakan 3 parameter dalam pengoperasiannya yaitu initialization, condition, dan increment. Initialization adalah nilai awal variable, condition adalah kondisi berhenti, dan increment adalah besaran perubahan pada suatu variable.

Kode Program 2.3

```
1 for (initialization; condition; increment) {  
2 // statement(s);  
3 }
```

Contoh penggunaan:

Kode Program 2.4

```
1 // Dim an LED using a PWM pin  
2 int PWMpin = 10; // LED in series with 470 ohm resistor on pin 10  
3 void setup() {  
4 // no setup needed  
5 }  
6 void loop() {  
7 for (int i = 0; i <= 255; i++) {  
8 analogWrite(PWMpin, i);  
9 delay(10);  
20 }  
21 }
```

## 3). Do.. While

Do.. While memiliki cara kerja yang sama dengan perulangan while. Akan tetapi Do.. While akan menjalankan loop terlebih dahulu sebelum pengecekan kondisi berhenti. Hal ini mengakibatkan Do.. While akan menjalankan perulangan minimal sekali.

Kode Program 2.5

```
1 do {  
2 // statement block  
3 } while (condition);
```

Contoh penggunaan:

Kode Program 2.6

```

1 int x = 0;
2 do {
3 delay(50); // wait for sensors to stabilize
4 x = readSensors(); // check the sensors
5 } while (x < 100);

```

### 3. CONDITIONAL

#### 1). If Else

If else merupakan tipe conditional yang lazim digunakan. Berawal dari pengecekan if, jika suatu variable tidak memenuhi condition1, maka akan dilakukan pengecekan dengan else if. Jika variable juga tidak memenuhi condition2, maka akan dikategorikan pada bagian else.

Kode Program 2.7

```

1 if (condition1) {
2 // do Thing A
3 }
4 else if (condition2) {
5 // do Thing B
6 }
7 else {
8 // do Thing C
9 }

```

Contoh penggunaan:

Kode Program 2.8

```

1 if (temperature >= 70) {
2 // Danger! Shut down the system.
3 }
4 else if (temperature >= 60) { // 60 <= temperature < 70
5 // Warning! User attention required.
6 }
7 else { // temperature < 60
8 // Safe! Continue usual tasks.
9 }

```

#### 2). Switch Case

Seperti halnya if else, switch case dapat digunakan sebagai conditional untuk mengatur flow program. Pada metode ini, akan dibentuk beberapa case sebagai condition yang digunakan untuk mengecek suatu variable. Jika variable memenuhi suatu case, maka ia akan menjalankan statement

case tersebut. Pada masing-masing case akan diberikan keyword break untuk keluar dari switch case.

Kode Program 2.9

```
1 switch (var) {  
2 case label1:  
3 // statements  
4     break;  
5 case label2:  
6 // statements  
7     break;  
8 default:  
9 // statements  
10    break;  
11 }
```

Contoh penggunaan :

Kode Program 2.10

```
1 int var = 1;  
2 switch (var) {  
3 case 1:  
4     //do something when var equals 1  
5     Serial.print("Satu");  
6     break;  
7 case 2:  
8     //do something when var equals 2  
9     Serial.print("Dua");  
10    break;  
11 default:  
12     // if nothing else matches, do the default  
13     // default is optional  
14     Serial.print("Default");  
15     break;  
16 }
```

## D. PERSIAPAN

1). Bacalah beberapa referensi tutorial project menggunakan Arduino seperti dibawah ini

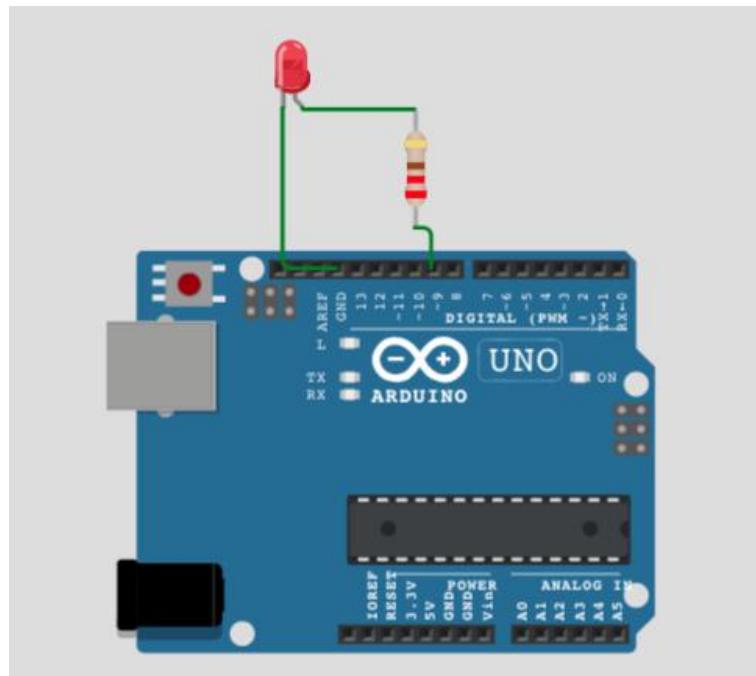
<https://create.arduino.cc/projecthub/projects/tags/beginner>

<https://www.arduino.cc/en/Tutorial/BuiltInExamples>

2). Pada simulasi wokwi buatlah rangkaian seperti berikut ini dengan spesifikasi.

Nama Perangkat	PIN
1. LED	1. Anoda(+) ke resistor 2. Katoda(-) ke GND
2. Resistor 220 ohm	PIN digital 9

Tabel 2.1 Konfigurasi PIN



Gambar 2.2 Konfigurasi rangkaian

## E. PERCOBAAN

### 1. MEMATIKAN DAN MENYALAKAN LED

1). Pada area kerja kode program buatlah kode program seperti berikut ini.

Kode Program 2.11

```

1 int led = 9;
2 void setup() {
3     // put your setup code here, to run once:
4     Serial.begin(115200);
5     pinMode(led, OUTPUT);
6     Serial.println("Program untuk menyalaikan dan mematikan led otomatis");
7 }
8
9 void loop() {
10    // put your main code here, to run repeatedly:

```

```

11  Serial.println("Led Menyal");  

12  digitalWrite(led, HIGH);  

13  delay(1000);  

14  Serial.println("Led Mati");  

15  digitalWrite(led, LOW);  

16  delay(1000);  

17 }

```

Pada baris ke 3 adalah syntax untuk konfigurasi pin yang digunakan yaitu pin 9 (dapat dirubah sesuai kondisi yang dibutuhkan) dan dideklarasikan sebagai komponen output. Pada baris ke 10 adalah syntax yang digunakan untuk menyalakan LED dengan status HIGH, serta pada baris ke 13 adalah untuk mematikan LED pada pin 9 dengan status LOW.

## 2. LED FADE-IN DAN FADE-OUT (CONDITIONAL)

1). Pada area kerja kode program ketiklah kode program seperti berikut ini.

Kode Program 2.12

```

1 int led = 9; // the PWM pin the LED is attached to  

2 int brightness = 0; // how bright the LED is  

3 int fadeAmount = 5; // how many points to fade the LED by  

4  

5 // the setup routine runs once when you press reset:  

6 void setup() {  

7     // declare pin 9 to be an output:  

8     pinMode(led, OUTPUT);  

9 }  

10  

11 // the loop routine runs over and over again forever:  

12 void loop() {  

13     // set the brightness of pin 9:  

14     analogWrite(led, brightness);  

15  

16     // change the brightness for next time through the loop:  

17     brightness = brightness + fadeAmount;  

18  

19     // reverse the direction of the fading at the ends of the fade:  

20     if (brightness <= 0 || brightness >= 255) {  

21         fadeAmount = -fadeAmount;  

22     }  

23     // wait for 30 milliseconds to see the dimming effect  

24     delay(30);  

25 }

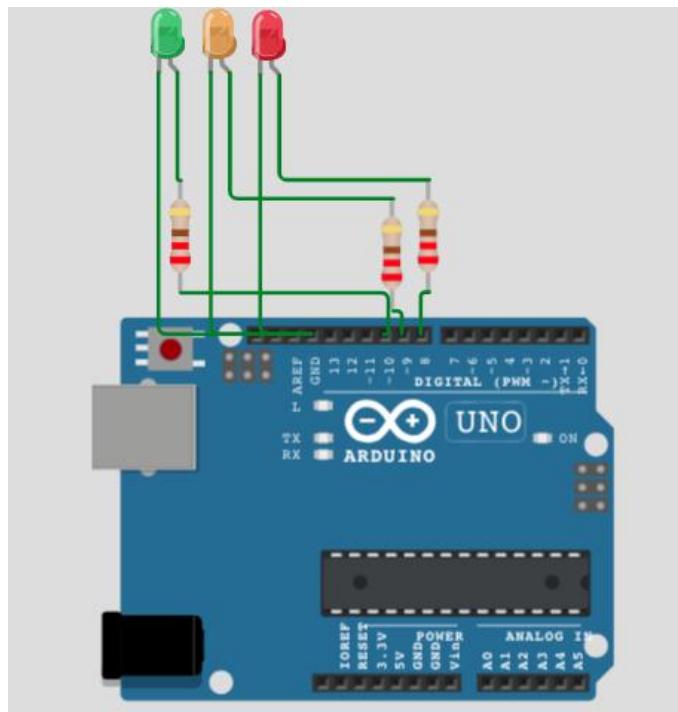
```

### 3. TRAFFIC LIGHTS

1). Buatlah rangkaian seperti berikut ini dengan spesifikasi sebagai berikut.

Nama Perangkat	PIN
1. LED	1. Anoda(+) ke resistor 2. Katoda(-) ke GND
2. Resistor LED merah 220 ohm	PIN digital 8
3. Resistor LED kuning 220 ohm	PIN digital 9
4. Resistor LED hijau 220 ohm	PIN digital 10

Tabel 2.2 Konfigurasi PIN



```
7 // declare pin 9 to be an output:  
8 pinMode(ledMerah, OUTPUT);  
9 pinMode(ledKuning, OUTPUT);  
10 pinMode(ledHijau, OUTPUT);  
11 }  
12  
13 // the loop routine runs over and over again forever:  
14 void loop() {  
15     // set the brightness of pin 9:  
16     digitalWrite(ledMerah, HIGH);  
17     digitalWrite(ledKuning, LOW);  
18     digitalWrite(ledHijau, LOW);  
19     delay(6000);  
20     digitalWrite(ledMerah, LOW);  
21     digitalWrite(ledKuning, HIGH);  
22     digitalWrite(ledHijau, LOW);  
23     delay(2000);  
24     digitalWrite(ledMerah, LOW);  
25     digitalWrite(ledKuning, LOW);  
26     digitalWrite(ledHijau, HIGH);  
27     delay(8000);  
28 }
```

Pada kode program 2.13 jarak waktu yang dibutuhkan dari lampu merah ke kuning adalah 6 detik, berganti ke lampu kuning dengan jarak waktu 2 detik dan lampu hijau menyala selama 8 detik. Untuk jarak waktu yang digunakan dapat disetting pada baris ke 19, 23 dan 27.

## PRAKTIKUM 3

### MODUL 3 : PERCOBAAN INPUT DIGITAL PADA MIKROKONTROLER

#### A. TUJUAN

1. Mampu membuat sistem dengan menggunakan input digital pada Arduino

#### B. PERALATAN

1. Simulator Wokwi
2. Board Arduino UNO
3. Switch
4. Buzzer
5. Keypad 4x4

#### C. TEORI

##### 1. Switch

Switch pada mikrokontroler adalah perangkat input yang digunakan untuk mengendalikan aliran arus listrik ke dalam mikrokontroler. Saat switch ditekan atau dalam posisi tertentu, itu dapat menghasilkan sinyal atau sinyal digital yang dapat digunakan oleh mikrokontroler untuk mengambil tindakan tertentu. Berikut adalah dasar-dasar teori tentang switch pada mikrokontroler:

###### 1). Prinsip Kerja Switch:

Sebuah switch adalah perangkat elektronik yang memiliki dua atau lebih kontak yang dapat dipisahkan atau dihubungkan. Ketika switch dalam posisi tertentu (ditekan atau dalam keadaan tertentu), itu akan menghubungkan atau memutuskan jalur listrik.

Switch pada mikrokontroler berfungsi sebagai input digital. Ketika ditekan atau dalam posisi tertentu, mereka menghasilkan sinyal logika tinggi (1) atau rendah (0) yang dapat dibaca oleh mikrokontroler.

###### 2). Penggunaan Umum:

Switch pada mikrokontroler digunakan untuk mengendalikan berbagai fungsi dalam proyek elektronik. Contoh penggunaan termasuk tombol pintar, saklar lampu, saklar perangkat elektronik, dan banyak lagi.

Mereka juga digunakan dalam proyek-proyek yang melibatkan pemilihan mode atau tindakan tertentu, seperti mengganti layar LCD atau mengubah nilai-nilai variabel dalam program.

### 3). Pull-Up dan Pull-Down Resistor:

Untuk memastikan bacaan yang stabil pada pin mikrokontroler saat switch tidak ditekan, seringkali digunakan resistor pull-up atau pull-down. Resistor pull-up menghubungkan pin input ke VCC (tegangan tinggi) dan resistor pull-down menghubungkan pin input ke GND (tanah).

Dengan menggunakan resistor pull-up atau pull-down, kita dapat menghindari bacaan yang ambigu atau berfluktuasi saat switch tidak ditekan.

### 4). Debouncing:

Ketika switch ditekan atau dilepaskan, bisa terjadi efek bouncing, yaitu sinyal yang tidak stabil dan bisa berubah-ubah secara cepat. Untuk mengatasi ini, biasanya dilakukan debouncing, baik perangkat keras (dengan kapasitor atau resistor tambahan) maupun perangkat lunak (dengan kode program yang memfilter sinyal).

### 5). Interupsi:

Switch juga dapat digunakan untuk memicu interupsi pada mikrokontroler. Ini berarti, ketika switch ditekan, mikrokontroler dapat menjalankan kode tertentu dengan segera tanpa harus terus-menerus memeriksa status switch dalam loop utama program.

Switch adalah salah satu komponen dasar yang digunakan dalam banyak proyek elektronik dan mikrokontroler, dan pemahaman dasar tentang cara mereka bekerja sangat penting dalam pengembangan perangkat elektronik yang melibatkan kontrol dan interaksi dengan pengguna.



Gambar 3.1 Bentuk fisik switch

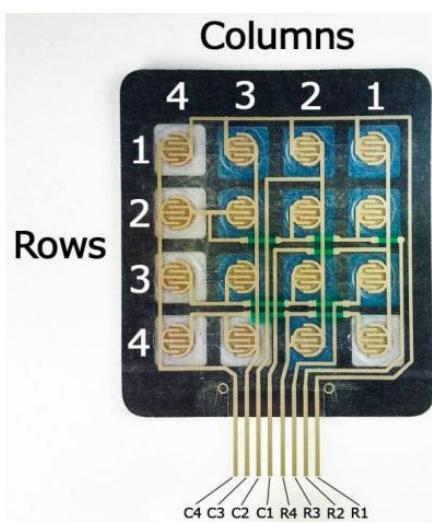
## 2. Keypad

Keypad adalah serangkaian pushbutton yang dapat digunakan untuk berbagai fungsi seperti navigasi, input kata sandi, maupun mengendalikan robot.



Gambar 3.2 Keypad 3X4 dan 4X4

Dipasaran terdapat 2 jenis keypad yaitu keypad  $3 \times 4$  dan  $4 \times 4$ . Tombol pada keypad diatur sedemikian rupa dalam bentuk baris dan kolom. Keypad  $3 \times 4$  berarti keypad mempunyai 3 kolom dan 4 baris. Begitu juga dengan keypad  $4 \times 4$  yang mempunyai 4 kolom dan 4 baris.

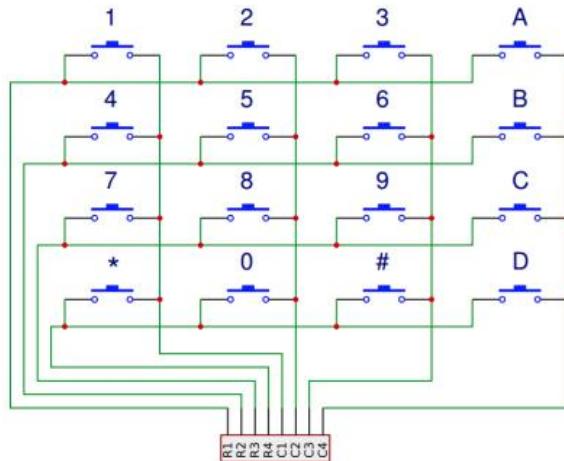


Gambar 3.3 Bentuk komponen dalam keypad

Didalam setiap key pada keypad ada membrane swich. Setiap switch pada 1 row saling berhubungan dengan sebuah konduktor. Begitu juga dengan key dalam 1 kolom, setiap key yang

berada 1 kolom dihubungkan dengan konduktor. Kombinasi behaviour dari konduktor tiap baris dan kolom ini akan mendefinisikan masing-masing key.

Dengan menekan tombol key akan membuat arus mengalir pada kolom pin (C1..C4) dan baris pin (R1..R4). Untuk lebih jelasnya, dapat dilihat skematik dari keypad 4 x 4 dibawah ini.



Gambar 3.4 Skematik pada keypad

### 3. Buzzer

Buzzer adalah perangkat audio yang digunakan untuk menghasilkan bunyi atau nada tertentu ketika diberikan sinyal listrik. Buzzer adalah komponen elektronik yang sering digunakan dalam berbagai aplikasi, seperti perangkat alarm, indikator, atau pengumuman suara pada berbagai perangkat elektronik. Berikut adalah beberapa dasar teori tentang buzzer:

#### 1). Prinsip Kerja Buzzer:

Buzzer adalah jenis transduser, yang berarti ia mengubah energi listrik menjadi energi suara. Ketika diberikan sinyal listrik, buzzer akan menghasilkan gelombang suara yang dapat didengar oleh manusia.

Buzzer bekerja dengan menggunakan vibrasi fisik. Ketika arus listrik mengalir melalui komponen dalam buzzer, itu menyebabkan elemen getar (seperti membran atau pegas) bergetar dengan cepat, menciptakan suara.

#### 2). Jenis Buzzer:

Terdapat dua jenis utama buzzer: Buzzer Aktif dan Buzzer Pasif.

**Buzzer Aktif:** Buzzer aktif memiliki sirkuit internal yang memungkinkannya menghasilkan suara ketika diberikan sinyal listrik. Mereka cukup mudah digunakan dan memerlukan sinyal aktif (biasanya gelombang persegi atau pulsa) untuk menghasilkan suara.

**Buzzer Pasif:** Buzzer pasif tidak memiliki sirkuit penggerak internal. Mereka memerlukan sinyal suara eksternal (misalnya, frekuensi audio) untuk menghasilkan suara. Buzzer pasif umumnya lebih sederhana dan lebih murah, tetapi memerlukan sirkuit penggerak tambahan.

### 3). Frekuensi dan Amplitudo:

Bunyi yang dihasilkan oleh buzzer memiliki frekuensi tertentu, yang menentukan tinggi rendahnya suara. Frekuensi ini tergantung pada desain buzzer dan sinyal listrik yang diberikan.

Amplitudo mengukur tingkat kebisingan atau kekuatan suara yang dihasilkan oleh buzzer. Ini bergantung pada energi sinyal listrik yang diberikan dan karakteristik mekanis buzzer.

### 4). Aplikasi Buzzer:

Buzzer digunakan dalam berbagai aplikasi, termasuk peringatan keamanan, alarm, perangkat medis, permainan, kendaraan, dan banyak lagi.

Mereka juga digunakan sebagai indikator audio dalam banyak perangkat, seperti ponsel, komputer, dan perangkat elektronik konsumen lainnya.

### 5). Pengendalian Buzzer:

Untuk mengendalikan buzzer, Anda perlu memberikan sinyal listrik yang sesuai dengan buzzer yang Anda gunakan. Buzzer aktif dapat dikendalikan langsung dari pin mikrokontroler atau sirkuit elektronik, sedangkan buzzer pasif memerlukan sinyal suara eksternal.

### 6). Durasi Suara:

Durasi suara buzzer dapat dikendalikan dengan memutuskan sinyal listrik yang diberikan ke buzzer. Dalam program mikrokontroler, Anda dapat menggunakan perintah pengendalian waktu untuk mengatur berapa lama buzzer akan aktif.

Buzzer adalah komponen suara yang sederhana namun penting dalam dunia elektronik, dan pemahaman dasarnya sangat berguna saat merancang atau memahami berbagai aplikasi elektronik yang melibatkan suara atau notifikasi audio.



Gambar 3.5 Komponen buzzer

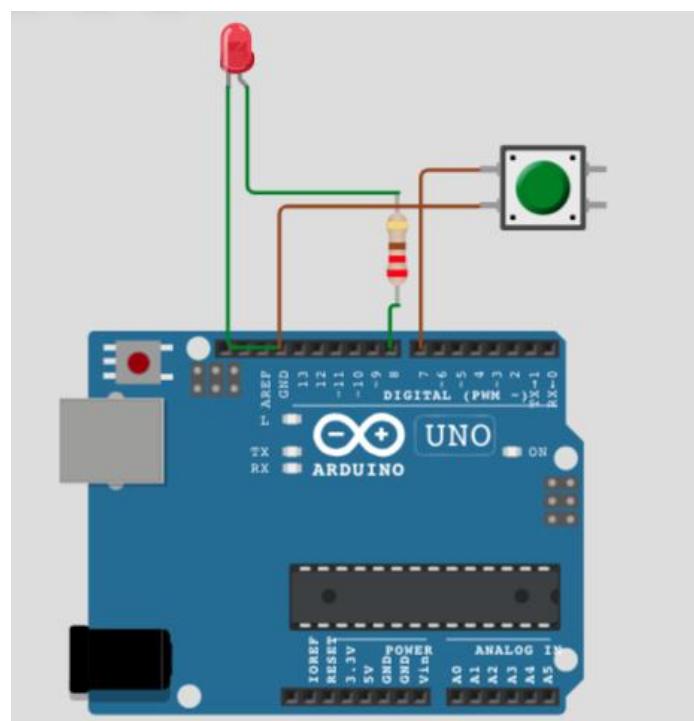
## D. PERCOBAAN

### 1. PERCOBAAN SWITCH DENGAN LED

- 1). Buatlah rangkaian seperti berikut ini dengan spesifikasi sebagai berikut.

Nama Perangkat	PIN
1. LED	1. Anoda(+) ke resistor 2. Katoda(-) ke GND
2. Resistor 220 ohm	PIN digital 8
3. Switch button	PIN digital 7 & Ground

Tabel 3.1 Tabel konfigurasi



Gambar 3.6 Rangkaian yang digunakan

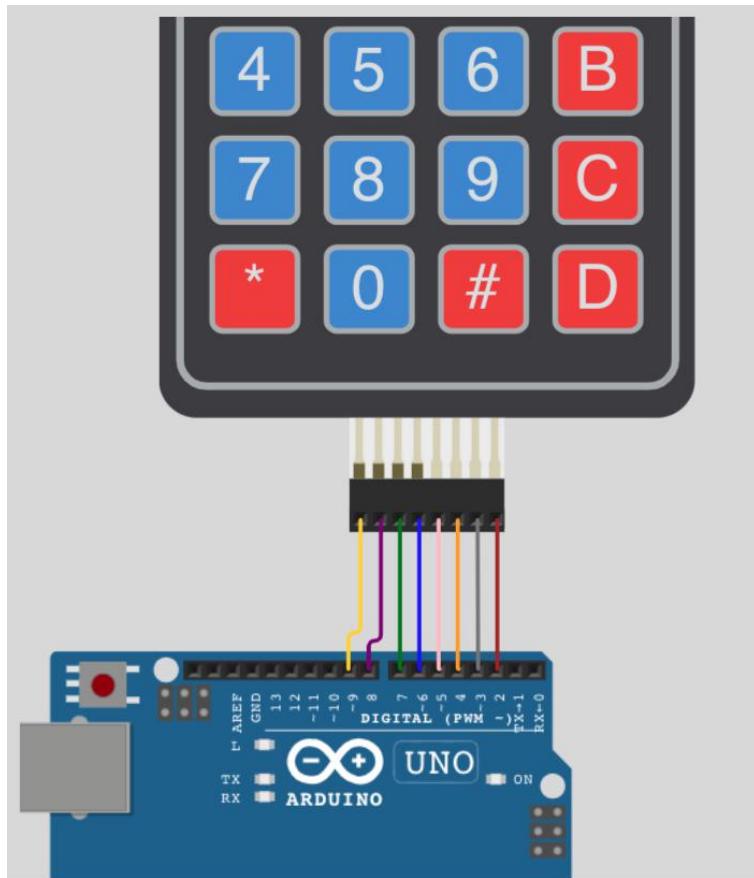
2). Ketiklah kode program seperti berikut ini.

Kode Program 3.1

```
1 int led = 8;
2 int switchButton = 7;
3
4
5 // the setup routine runs once when you press reset:
6 void setup() {
7     // declare pin 9 to be an output:
8     Serial.begin(115200);
9     pinMode(led, OUTPUT);
10    pinMode(switchButton, INPUT_PULLUP);
11 }
12
13 // the loop routine runs over and over again forever:
14 void loop() {
15     // set the brightness of pin 9:
16     int button = digitalRead(switchButton);
17     delay(500);
18     if( button == 1)
19     {
20         digitalWrite(led, HIGH);
21         Serial.println("Led ON");
22     }
23     else
24     {
25         digitalWrite(led,LOW);
26         Serial.println("Led OFF");
27     }
28 }
```

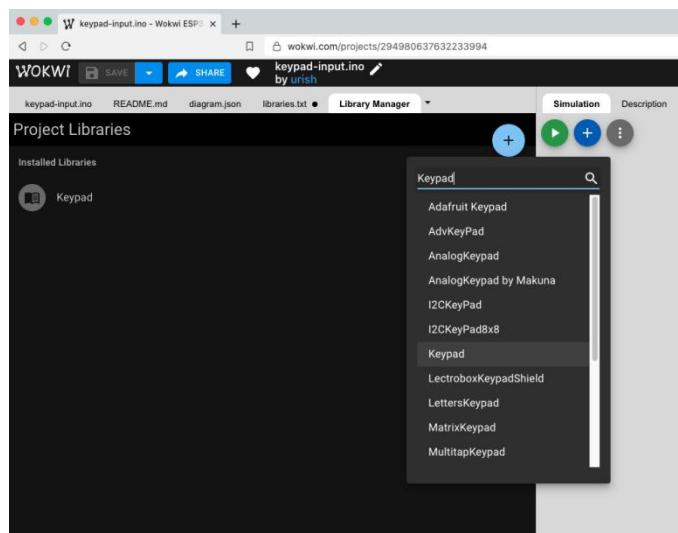
## 2. PERCOBAAN DENGAN KEYPAD

- 1). Buatlah rangkaian dengan spesifikasi sebagai berikut ini.



Gambar 3.7 Konfigurasi PIN keypad dan Arduino

- 2). Pada simulasi terdapat tab library manager >> Add a new Library, seperti pada gambar berikut ini.



Gambar 3.8 Penambahan library pada Library Manager

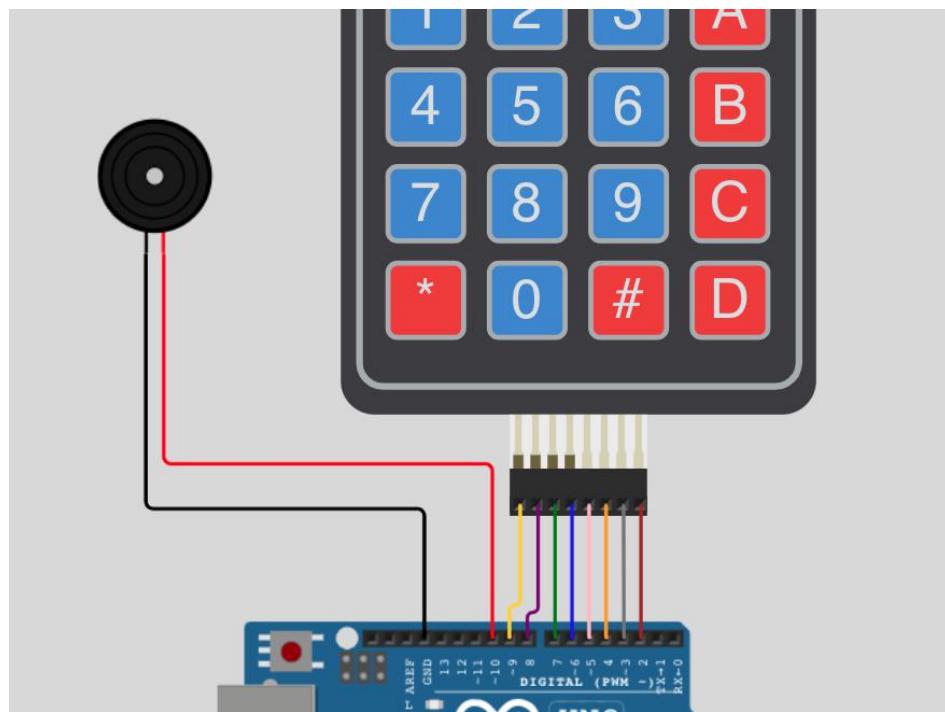
3). Ketiklah kode program berikut ini.

Kode Program 3.2

```
1 #include <Keypad.h>
2
3 const uint8_t ROWS = 4;
4 const uint8_t COLS = 4;
5 char keys[ROWS][COLS] = {
6 { '1', '2', '3', 'A' },
7 { '4', '5', '6', 'B' },
8 { '7', '8', '9', 'C' },
9 { '*', '0', '#', 'D' }
10 };
11
12 uint8_t colPins[COLS] = { 5, 4, 3, 2 }; // Pins connected to C1, C2, C3, C4
13 uint8_t rowPins[ROWS] = { 9, 8, 7, 6 }; // Pins connected to R1, R2, R3, R4
14
15 Keypad keypad = Keypad(makeKeymap(keys), rowPins, colPins, ROWS, COLS);
16
17 void setup() {
18   Serial.begin(115200);
19 }
20
21 void loop() {
22   char key = keypad.getKey();
23
24   if (key != NO_KEY) {
25     Serial.println(key);
26   }
27 }
```

### 3. PERCOBAAN KEYPAD DAN LED

- 1). Buatlah rangkaian dengan spesifikasi sebagai berikut ini.



Gambar 3.9 Konfigurasi PIN keypad dan Buzzer

- 2). Ketiklah kode program seperti berikut ini dan jalankan.

Kode Program 3.3

```

1 #include <Keypad.h>
2
3 int buzzer = 10;
4
5 const uint8_t ROWS = 4;
6 const uint8_t COLS = 4;
7 char keys[ROWS][COLS] = {
8 { '1', '2', '3', 'A' },
9 { '4', '5', '6', 'B' },
10 { '7', '8', '9', 'C' },
11 { '*', '0', '#', 'D' }
12 };
13
14 uint8_t colPins[COLS] = { 5, 4, 3, 2 }; // Pins connected to C1, C2, C3, C4
15 uint8_t rowPins[ROWS] = { 9, 8, 7, 6 }; // Pins connected to R1, R2, R3, R4
16
17 Keypad keypad = Keypad(makeKeymap(keys), rowPins, colPins, ROWS, COLS);
18
19 void setup() {
20     Serial.begin(115200);

```

```
21  pinMode(buzzer, OUTPUT);
22 }
23
24 void loop() {
25     char key = keypad.getKey();
26
27     if (key != NO_KEY) {
28         Serial.println(key);
29         switch (key) {
30             case '1':
31                 tone(buzzer, 100); //membunyikan buzzer sebesar 100Hz, boleh diubah-ubah
32                 sesuai dengan keinginan, semakin tinggi frekuensi semakin tinggi pula suara
33                 buzzernya
34                 break;
35             case '2':
36                 tone(buzzer, 150); //membunyikan buzzer sebesar 150Hz, boleh diubah-ubah
37                 sesuai dengan keinginan, semakin tinggi frekuensi semakin tinggi pula suara
38                 buzzernya
39                 break;
40             case '3':
41                 tone(buzzer, 200); //membunyikan buzzer sebesar 200Hz, boleh diubah-ubah
42                 sesuai dengan keinginan, semakin tinggi frekuensi semakin tinggi pula suara
43                 buzzernya
44                 break;
45             case '4':
46                 tone(buzzer, 250); //membunyikan buzzer sebesar 250Hz, boleh diubah-ubah
47                 sesuai dengan keinginan, semakin tinggi frekuensi semakin tinggi pula suara
48                 buzzernya
49                 break;
50             case '5':
51                 tone(buzzer, 300); //membunyikan buzzer sebesar 300Hz, boleh diubah-ubah
52                 sesuai dengan keinginan, semakin tinggi frekuensi semakin tinggi pula suara
53                 buzzernya
54                 break;
55             case '6':
56                 tone(buzzer, 350); //membunyikan buzzer sebesar 350Hz, boleh diubah-ubah
57                 sesuai dengan keinginan, semakin tinggi frekuensi semakin tinggi pula suara
58                 buzzernya
59                 break;
60             case '7':
61                 tone(buzzer, 400); //membunyikan buzzer sebesar 400Hz, boleh diubah-ubah
62                 sesuai dengan keinginan, semakin tinggi frekuensi semakin tinggi pula suara
63                 buzzernya
64                 break;
```

```
64  case '8':  
65      tone(buzzer, 450); //membunyikan buzzer sebesar 450Hz, boleh diubah-ubah  
66      sesuai dengan keinginan, semakin tinggi frekuensi semakin tinggi pula suara  
67      buzzernya  
68      break;  
69  case '9':  
70      tone(buzzer, 500); //membunyikan buzzer sebesar 500Hz, boleh diubah-ubah  
71      sesuai dengan keinginan, semakin tinggi frekuensi semakin tinggi pula suara  
72      buzzernya  
73      break;  
74  case '0':  
75      noTone(buzzer); //mematikan buzzer  
76      break;  
77  }  
78 }
```

## PRAKTIKUM 4

### MODUL 4 : PERCOBAAN OUTPUT DIGITAL PADA MIKROKONTROLER

#### A. TUJUAN

1. Mampu membuat sistem dengan menggunakan output digital pada Arduino

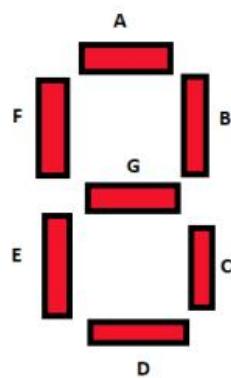
#### B. PERALATAN

1. Simulasi wokwi
2. Board Arduino UNO
3. Seven segment
4. Relay

#### C. TEORI

##### 1. Seven Segment

Seven Segment adalah komponen elektronik yang dapat menghasilkan karakter angka dari 0-9. 7 Segmen ini banyak diaplikasikan keberbagai peralatan elektronik seperti jam digital, multimeter, kalkulator, thermometer, dan sebagianya. Selain angka, 7 segment dapat juga digunakan untuk menampilkan alfabet A..F. Ada beberapa jenis komponen penyusun 7 segmen diantaranya Incandescent bulbs, Fluorescent lamps (FL), Liquid Crystal Display (LCD) dan Light Emitting Diode (LED). Pada umumnya dipasaran menggunakan 7 segment dengan LED.

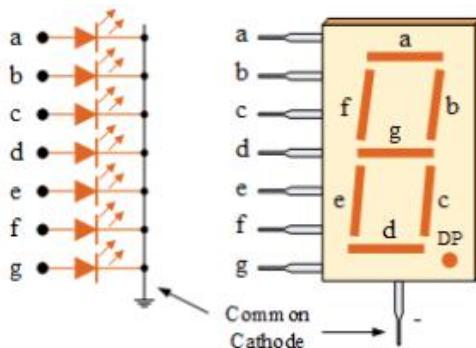


Gambar 4.1 Urutan LED pada seven segment

Terdapat 2 jenis seven segment di pasaran, yaitu:

a. Common Cathode

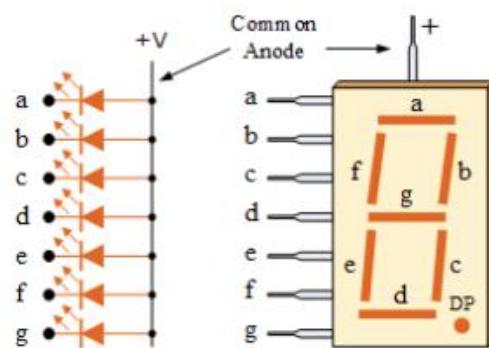
Pada common cathode, semua bagian cathode pada 7 segment dihubungkan dengan ground (0). Sehingga untuk mengaktifkan lampu salah satu segment dapat dilakukan dengan memberikan logic 1 pada suatu segment (a,b,...,g).



Gambar 4.2 Seven segment Common cathode

b. Comon Anode

Pada common anode, semua bagian anode pada 7 segment dihubungkan dengan tegangan (1). Sehingga untuk menyalakan lampu salah satu segment dapat dilakukan dengan memberikan logic 0 pada suatu segment (a,b,...,g).



Gambar 4.3 Seven segment Common anode

## 2. Relay

Relay adalah komponen elektromekanis yang digunakan dalam banyak aplikasi elektronik dan kendali, termasuk dalam mikrokontroler. Relay digunakan untuk mengendalikan perangkat atau sirkuit listrik dengan memanipulasi aliran arus listrik menggunakan sinyal kontrol. Berikut adalah beberapa dasar teori tentang relay pada mikrokontroler:

#### 1). Prinsip Kerja Relay:

Relay bekerja berdasarkan prinsip elektromagnetik. Ini terdiri dari dua bagian utama: kontak elektrik dan elektromagnet.

Ketika arus listrik mengalir melalui kumparan elektromagnet, itu menciptakan medan elektromagnetik yang menggerakkan sebuah saklar atau kontak. Ini mengakibatkan perubahan dalam hubungan kontak listrik dalam relay.

#### 2). Kontak Relay:

Relay memiliki satu atau lebih pasang kontak (biasanya disebut kontak NO/Normally Open dan kontak NC/Normally Closed) yang dapat berfungsi sebagai saklar. Kontak relay dapat membuka atau menutup jalur listrik tergantung pada kondisi elektromagnetiknya.

- Kontak NO (Normally Open): Kontak ini biasanya terbuka ketika relay tidak diaktifkan (tanpa arus pada kumparan elektromagnet) dan tertutup ketika relay diaktifkan.
- Kontak NC (Normally Closed): Kontak ini biasanya tertutup ketika relay tidak diaktifkan dan terbuka ketika relay diaktifkan.

#### 3). Aplikasi Relay:

Relay digunakan dalam berbagai aplikasi, termasuk pengendalian perangkat listrik seperti lampu, motor, pemanas, dan perangkat besar lainnya.

Mereka juga digunakan dalam sistem otomatisasi, peralatan rumah pintar, kendaraan, dan aplikasi industri yang lebih kompleks.

#### 4). Pengendalian Relay oleh Mikrokontroler:

Mikrokontroler biasanya tidak dapat menangani arus listrik besar yang diperlukan untuk mengendalikan perangkat-perangkat seperti lampu atau motor langsung. Oleh karena itu, relay digunakan sebagai perantara.

Mikrokontroler mengirimkan sinyal kontrol (biasanya dalam bentuk sinyal digital) ke relay. Ketika relay diaktifkan, kontak-kontak relay akan berubah sesuai dengan perintah mikrokontroler, yang akan memungkinkan atau menghentikan aliran listrik ke perangkat terkait.

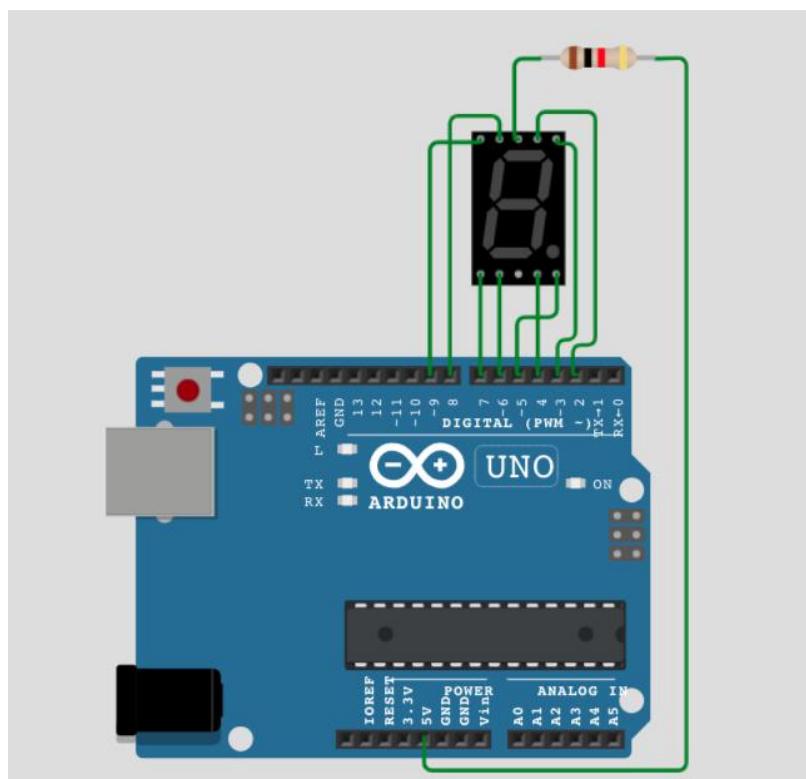


Gambar 4.4 Bentuk fisik komponen relay

## D. PERCOBAAN

### 1. PERCOBAAN SEVEN SEGMENT

- 1). Buatlah rangkaian seperti pada gambar berikut ini.



Gambar 4.5 Rangkaian percobaan seven segment

- 2). Ketiklah kode program seperti pada berikut ini.

Kode Program 4.1

```
1 const byte numeral[10] =  
2 {  
3 // 0 = led dalam seven segment yang tidak diberikan tegangan  
4 // 1 = led dalam seven segment yang diberikan tegangan masukan  
5 B11111101, // angka 0  
6 B01100001, // angka 1
```

```

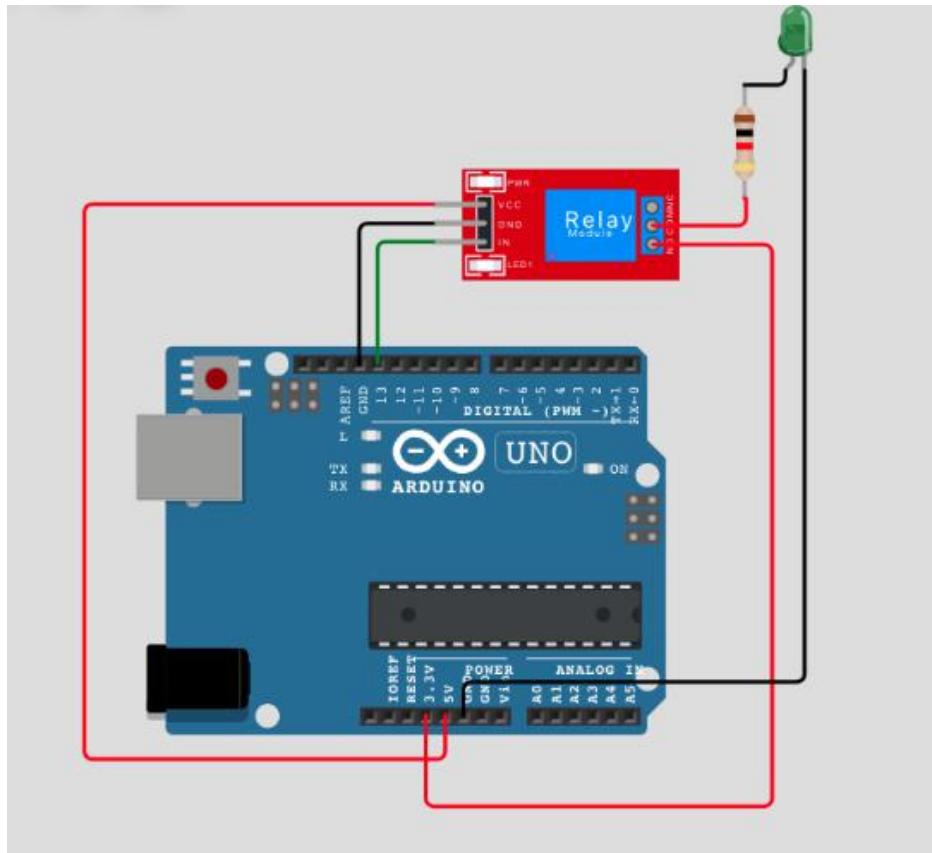
7  B11011011, // angka 2
8  B11110011, // angka 3
9  B01100111, // angka 4
10 B10110111, // angka 5
11 B10111111, // angka 6
12 B11100001, // angka 7
13 B11111111, // angka 8
14 B11110111, // angka 9
15 };
16
17 // pin dari segment dp,G,F,E,D,C,B,A
18 const int segmentPins[8] = { 5,9,8,7,6,4,3,2 };
19
20 void setup() {
21   for(int i=0; i < 8; i++)
22   {
23     pinMode(segmentPins[i], OUTPUT);
24   }
25 }
26
27 void loop() {
28 //Seven segment akan memulai dari 0 sampai 10
29 //cara kerjanya sistem ditambah 1 dengan delay 1 detik dan ditambah 1
30   for(int i=0; i <= 10; i++)
31   {
32     showDigit(i);
33     delay(1000);
34   }
35   delay(2000);
36 }
37
38 void showDigit( int number)
39 {
40   boolean isBitSet;
41   for(int segment = 1; segment < 8; segment++)
42   {
43     if( number < 0 || number > 9)
44     {
45       isBitSet = 0; // turn off all segments
46     }
47     else
48     {
49       // isBitSet will be true if given bit is 1
50       isBitSet = bitRead(numeral[number], segment);

```

```
51     }
52     isBitSet = !isBitSet;
53     digitalWrite( segmentPins[segment], isBitSet);
54 }
55 }
```

## 2. PERCOBAAN RELAY DAN LED

- 1). Buatlah rangkaian seperti pada gambar berikut ini.



Gambar 4.6 Rangkaian relay dengan LED dan Arduino UNO

- 2). Ketiklah kode program seperti perikut ini setelah itu jalankan simulasinya.

Kode Program 4.2

```
1 // Definisikan pin yang akan digunakan untuk mengendalikan relay
2 int relayPin = 13; // Misalnya, hubungkan relay ke pin digital 13 Arduino
3
4 void setup() {
5     // Inisialisasi pin relay sebagai OUTPUT
6     pinMode(relayPin, OUTPUT);
7 }
8
9 void loop() {
10    // Aktifkan relay (tutup kontak) selama 5 detik
```

```
11  digitalWrite(relayPin, HIGH);
12  delay(5000); // Tunggu selama 5 detik
13  // Matikan relay (buka kontak) selama 5 detik
14  digitalWrite(relayPin, LOW);
15  delay(5000); // Tunggu selama 5 detik
16 }
```

## PRAKTIKUM 5

### PRAKTIKUM 5 : PERCOBAAN MOTOR SERVO

#### A. TUJUAN

1. Mampu membuat system dengan menggunakan pengendalian PWM pada motor servo

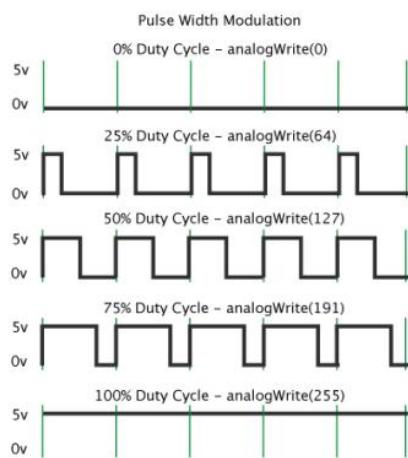
#### B. PERALATAN

1. Simulasi Wokwi
2. Board Arduino UNO
3. Potensiometer
4. Motor servo

#### C. TEORI

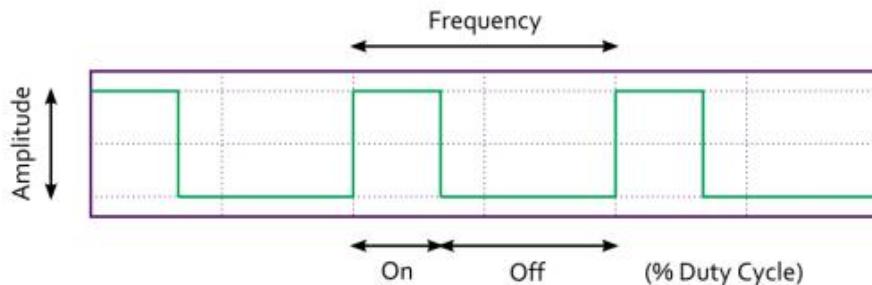
##### 1. PWM

PWM adalah singkatan dari Pulse Width Modulation yang merupakan Teknik untuk mendapatkan sinyal analog dari sebuah sinyal digital. Sinyal digital dapat digunakan untuk membuat sinyal square dengan mengendalikan waktu on dan off sinyal tersebut. Nilai sinyal square yang dihasilkan berada diantara Vcc dan ground. Pada gambar dibawah, terdapat sinyal square dengan kondisi on dan off diatur oleh garis hijau. Garis hijau ini merupakan periode yang diatur oleh frekuensi PWM. Untuk Arduino UNO nilai frekuensi PWM nya adalah 500 Hz. Dengan kata lain, periode nya adalah 1/500 Hz yaitu 2 ms.



Gambar 5.1 Sinyal PWM

analogWrite() mempunyai nilai 0 - 255, ketika analogWrite(255) maka akan menghasilkan 100% duty cycle (always on), dan ketika analogWrite(127) duty cycle nya menjadi 50%. Untuk lebih memahami sinyal square, mari lihat gambar dibawah ini.



Gambar 5.2 Bentuk sinyal square pada PWM

- Amplitude – Jumlah nilai tegangan yang berubah saat on dan off
- Frequency – Jumlah sinyal yang dihasilkan selama 1 second
- Duty Cycle - Proporsi antara on time dan off time (dalam persentase)

Berikut ini referensi frekuensi pada PWM dan PIN PWM di Arduino

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, Yun	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

Gambar 5.3 Referensi PWM pada pin PWM di Arduino

## 2. Motor Servo

Motor servo adalah jenis motor yang digunakan dalam berbagai aplikasi untuk mengontrol posisi sudut tertentu dengan presisi tinggi. Motor ini bekerja dengan menerima sinyal kontrol yang mengatur sudut putaran porosnya. Gambar motor servo tidak dapat ditampilkan dalam teks, tetapi saya dapat memberikan deskripsi visual motor servo:

### Gambaran Motor Servo:

- a. Motor servo umumnya terdiri dari tiga bagian utama: motor DC, gearbox, dan unit umpan balik (seperti potensiometer).

- b. Motor DC digunakan untuk menggerakkan poros utama motor. Ini adalah bagian yang sebenarnya yang menggerakkan sudut motor.
- c. Gearbox adalah mekanisme yang digunakan untuk mengubah kecepatan dan torsi motor sehingga motor dapat menggerakkan sudut dengan presisi tinggi.
- d. Unit umpan balik, seperti potensiometer, digunakan untuk memberikan umpan balik posisi motor ke kendali, sehingga motor servo dapat memastikan bahwa ia berada pada sudut yang benar.

**Motor servo biasanya memiliki tiga kabel:**

- a. **Kabel Listrik:** Kabel ini digunakan untuk memberikan daya ke motor servo. Biasanya, tegangan daya yang digunakan adalah 5V.
- b. **Kabel Ground:** Ini adalah kabel tanah yang terhubung ke ground (GND) pada sumber daya Anda.
- c. **Kabel Kontrol:** Kabel ini digunakan untuk mengirimkan sinyal kontrol ke motor servo. Sinyal ini mengatur posisi atau sudut yang diinginkan.

**Selain itu, ada tiga jenis motor servo utama:**

**Servo Standar:** Digunakan dalam aplikasi umum seperti robotika, kendaraan remote, dan perangkat mekanis lainnya.

**Servo Kontinu:** Jenis servo ini terus berputar dan dapat dikendalikan untuk mengatur kecepatan dan arah putarannya.

**Servo Mikro:** Jenis servo ini lebih kecil dan digunakan dalam aplikasi yang membutuhkan ukuran yang lebih kecil.

Motor servo digunakan dalam berbagai aplikasi, termasuk kendali posisi pada robot, kamera gimbal, pengendalian pintu otomatis, dan banyak lagi. Kelebihan utama motor servo adalah kemampuan untuk mengontrol posisi dengan presisi tinggi dan berulang-ulang.



Gambar 5.4 Bentuk fisik motor servo

### 3. Potensiometer

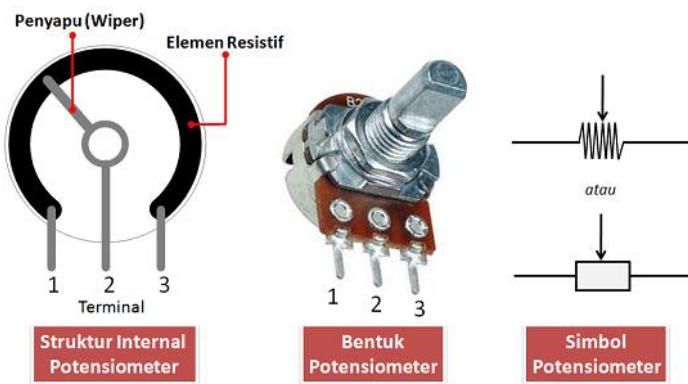
Potensiometer adalah komponen elektronik yang digunakan untuk mengukur atau mengatur resistansi dalam sebuah sirkuit listrik. Ini biasanya digunakan untuk mengendalikan volume dalam perangkat audio, mengukur posisi dalam alat pengukur, atau mengendalikan posisi dalam motor servo. Berikut adalah beberapa informasi dasar tentang potensiometer:

#### Aplikasi potensiometer:

- a. Potensiometer digunakan dalam berbagai aplikasi, termasuk pengendalian volume dalam perangkat audio seperti radio, amplifier, dan mixer audio.
- b. Ini juga digunakan dalam alat pengukur dan instrumen pengukuran untuk mengatur rentang pengukuran atau mengukur posisi.
- c. Dalam motor servo, potensiometer digunakan sebagai elemen umpan balik untuk mengukur posisi relatif dari motor servo.

Potensiometer adalah komponen yang sering digunakan dalam berbagai aplikasi elektronik dan listrik. Kemampuannya untuk mengubah resistansi dalam sirkuit membuatnya sangat berguna dalam mengendalikan dan mengukur berbagai parameter seperti volume, posisi, atau pengukuran rentang.

## POTENSIOMETER

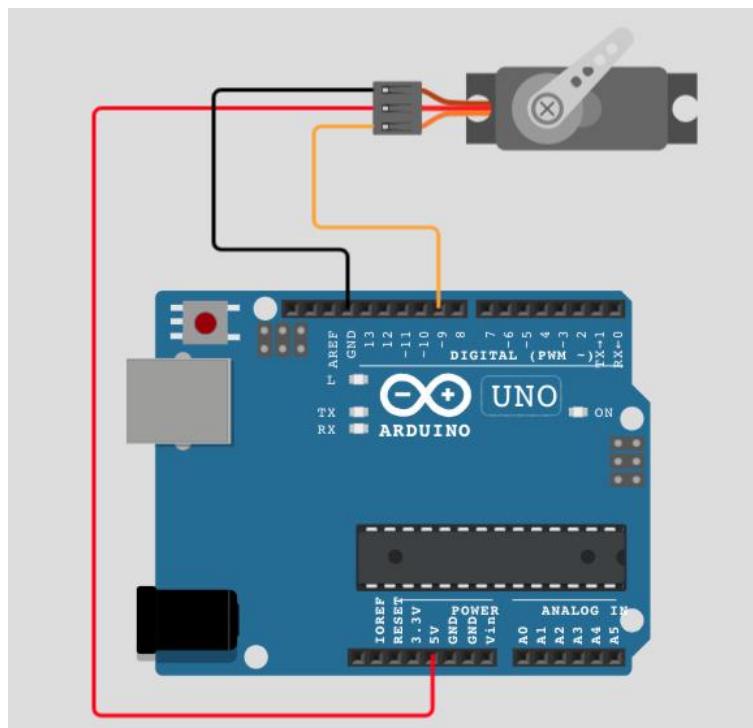


Gambar 5.5 Bentuk fisik potensiometer

## D. PERCOBAAN

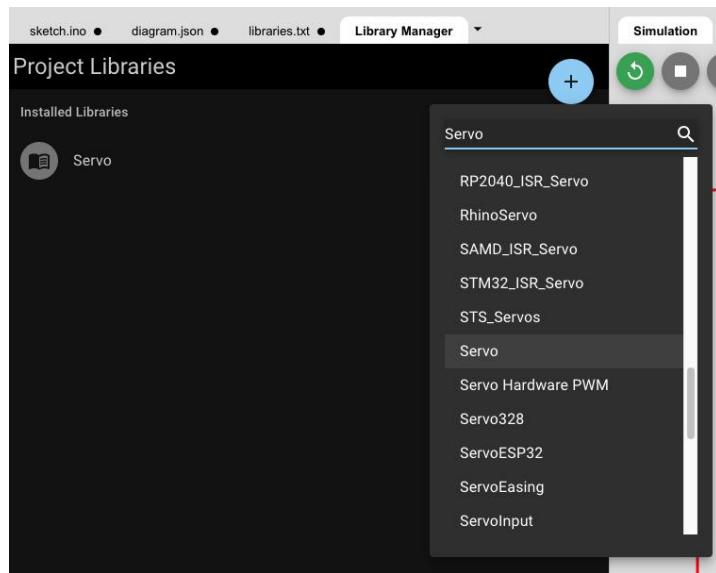
### 1. PERCOBAAN SWEEP MOTOR SERVO

- 1). Buatlah rangkaian seperti pada gambar berikut ini.



Gambar 5.6 Rangkaian motor servo dan Arduino Uno

2). Pada tab Library Manager tambahkan library servo, seperti pada gambar berikut ini.



Gambar 5.7 Menambahkan library servo

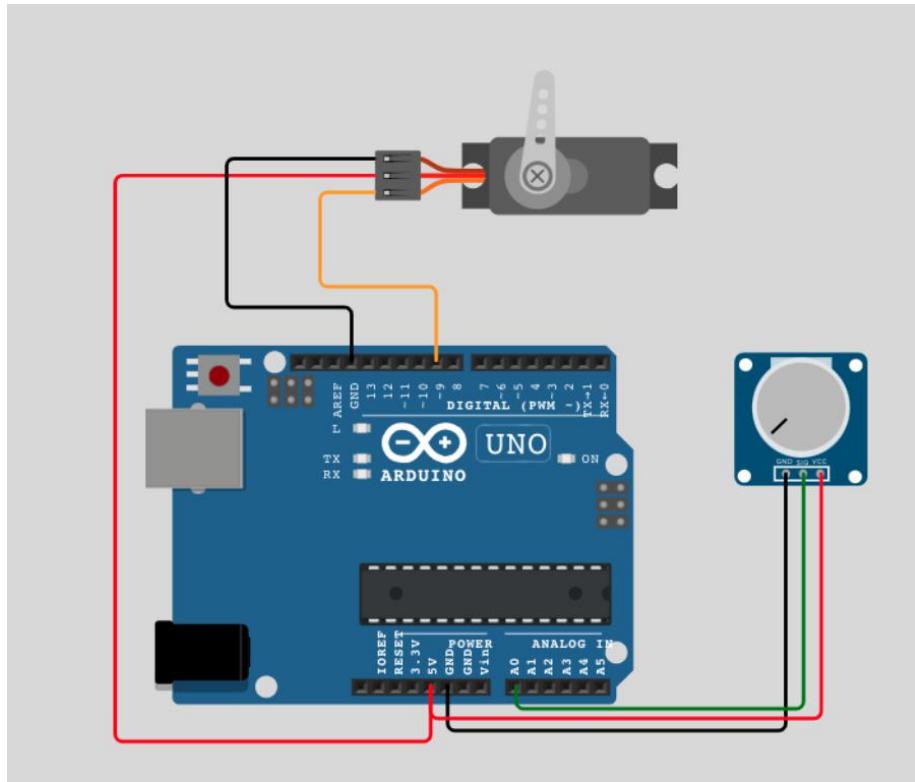
3). Ketikan kode program seperti berikut ini dan jalankan simulasinya.

Kode Program 5.1

```
1 #include <Servo.h>
2
3 Servo myservo; // create servo object to control a servo
4 // twelve servo objects can be created on most boards
5
6 int pos = 0; // variable to store the servo position
7
8 void setup() {
9     myservo.attach(9); // attaches the servo on pin 9 to the servo object
10 }
11
12 void loop() {
13     for (pos = 0; pos <= 180; pos += 1) { // goes from 0 degrees to 180 degrees
14         // in steps of 1 degree
15         myservo.write(pos); // tell servo to go to position in variable 'pos'
16         delay(15); // waits 15ms for the servo to reach the position
17     }
18     for (pos = 180; pos >= 0; pos -= 1) { // goes from 180 degrees to 0 degrees
19         myservo.write(pos); // tell servo to go to position in variable 'pos'
20         delay(15); // waits 15ms for the servo to reach the position
21     }
22 }
```

## 2. PERCOBAAN MENGONTROL SERVO DENGAN POTENSIOMETER

- 1). Buatlah rangkaian seperti pada gambar berikut ini.



Gambar 5.8 Rangkaian potensiometer dengan motor servo

- 2). Ketiklah kode program berikut ini dan jalankan simulasinya

Kode Program 5.2

```
1 #include <Servo.h>
2
3 Servo myservo; // create servo object to control a servo
4 // twelve servo objects can be created on most boards
5
6 int pos = 0; // variable to store the servo position
7
8 void setup() {
9     myservo.attach(9); // attaches the servo on pin 9 to the servo object
10    Serial.begin(115200);
11 }
12
13 void loop() {
14     int potensio = analogRead(A0);
15     int b = constrain(potensio, 0, 180); // nilai yang dihasilkan potensio 1 - 180
16     Serial.println(b);
17     myservo.write(b);
```

```
18   delay(15);  
19 }
```

## PRAKTIKUM 6

### MODUL 6 : PERCOBAAN PENGGUNAAN SENSOR INPUT

#### A. TUJUAN

1. Mampu membuat system analog input dengan menggunakan sensor LDR
2. Mampu membuat system analog input dengan menggunakan sensor DHT 22
3. Mampu membuat system analog input dengan menggunakan sensor PIR

#### B. PERALATAN

1. Simulasi wokwi
2. Board Arduino UNO
3. Sensor LDR
4. Sensor DHT 22
5. Sensor PIR

#### C. TEORI

##### 1. Sensor LDR

Sensor LDR (Light Dependent Resistor) atau kadang-kadang disebut juga sebagai sensor fotoresistor adalah jenis sensor yang digunakan untuk mendeteksi intensitas cahaya di sekitarnya. Sensor ini memiliki resistansi yang berubah-ubah tergantung pada jumlah cahaya yang jatuh padanya. Berikut adalah beberapa informasi dasar tentang sensor LDR:

###### 1. Prinsip Kerja Sensor LDR:

Sensor LDR berdasarkan prinsip perubahan resistansi. Ketika cahaya mengenai permukaan sensor, jumlah elektron di dalam material LDR menjadi lebih aktif, yang mengakibatkan penurunan resistansi sensor. Sebaliknya, ketika sensor berada dalam kegelapan, resistansinya meningkat.

###### 2. Struktur Sensor LDR:

- Sensor LDR biasanya terbuat dari material semikonduktor khusus yang memiliki resistansi yang tinggi dalam kegelapan dan resistansi yang rendah saat terkena cahaya.
- Sensor ini memiliki dua terminal, dan saat dihubungkan dalam sirkuit, resistansinya dapat memengaruhi aliran arus listrik dalam sirkuit tersebut.

###### 3. Aplikasi Sensor LDR:

- Sensor LDR digunakan dalam berbagai aplikasi, termasuk:

- **Lampu Malam Otomatis:** Untuk mengaktifkan lampu ketika cahaya sekitar rendah (misalnya, di malam hari).

- **Kamera Fotografi:** Untuk mengukur cahaya dan mengatur parameter seperti kecepatan rana dan aperture pada kamera.

- **Lampu Jalan Pintar:** Untuk mengontrol pencahayaan jalan berdasarkan kondisi cahaya alami.

- **Sensor Cahaya Dalam Ruangan:** Untuk mengatur pencahayaan dalam ruangan secara otomatis berdasarkan intensitas cahaya yang ada.

4. Keuntungan Sensor LDR:

- Sensor LDR relatif murah dan mudah digunakan.

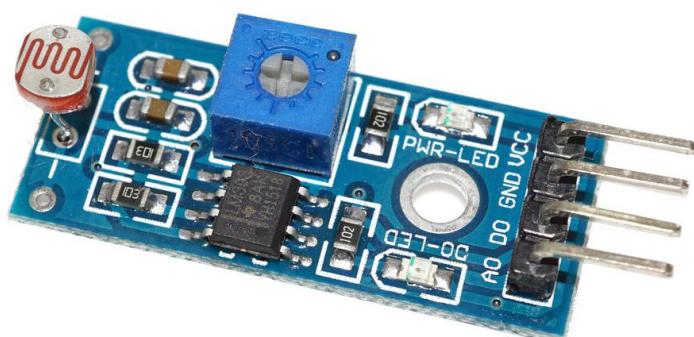
- Mereka memiliki respon yang cukup lambat terhadap perubahan cahaya, sehingga cocok untuk aplikasi di mana perubahan cahaya cepat tidak diperlukan.

5. Kelemahan Sensor LDR:

- Sensor LDR cenderung memiliki sensitivitas yang lebih rendah dalam spektrum cahaya tertentu dan kurang akurat dalam mengukur intensitas cahaya dengan presisi tinggi.

- Mereka kurang cocok untuk aplikasi yang memerlukan deteksi cahaya sangat cepat atau dalam kondisi cahaya yang sangat rendah.

Sensor LDR adalah salah satu jenis sensor cahaya yang paling umum digunakan dalam berbagai aplikasi sehari-hari. Meskipun mereka mungkin kurang akurat dibandingkan dengan sensor cahaya lainnya seperti sensor fotodiode dalam beberapa situasi, mereka tetap menjadi pilihan yang baik untuk banyak aplikasi yang memerlukan deteksi cahaya dasar atau pengaturan otomatis berdasarkan intensitas cahaya.



Gambar 6.1 Bentuk fisik modul sensor LDR

## 2. Sensor DHT 22

Sensor DHT22, juga dikenal sebagai sensor AM2302, adalah sensor suhu dan kelembaban digital yang digunakan untuk mengukur suhu dan kelembaban di lingkungan tertentu. Ini adalah salah satu jenis sensor yang sering digunakan dalam proyek-proyek IoT (Internet of Things) dan pengukuran lingkungan. Berikut adalah beberapa informasi dasar tentang sensor DHT22:

### 1. Prinsip Kerja Sensor DHT22:

- Sensor DHT22 bekerja berdasarkan perubahan resistansi pada elemen termal yang terdapat di dalamnya. Ketika suhu naik, resistansi elemen termal akan berubah, dan perubahan ini digunakan untuk mengukur suhu.
- Selain itu, sensor ini juga memiliki elemen yang dapat menyerap atau melepaskan uap air, yang digunakan untuk mengukur kelembaban.

### 2. Fitur Sensor DHT22:

- Sensor DHT22 adalah sensor digital yang berarti ia mengeluarkan data dalam bentuk digital yang mudah dibaca oleh mikrokontroler.
- Ini memiliki akurasi yang baik untuk mengukur suhu (biasanya  $\pm 0.5^{\circ}\text{C}$ ) dan kelembaban (biasanya  $\pm 2\% \text{ RH}$ ).

Sensor ini memiliki rentang pengukuran suhu sekitar  $-40^{\circ}\text{C}$  hingga  $80^{\circ}\text{C}$  dan kelembaban sekitar 0% hingga 100% RH.

### 3. Koneksi Sensor DHT22:

- Sensor DHT22 biasanya memiliki empat pin: VCC (daya), GND (tanah), Data (keluaran data), dan NC (Not Connected atau tidak terhubung).
- Anda perlu menghubungkan VCC ke tegangan 3.3V atau 5V, GND ke ground, dan Data ke pin input/output digital pada mikrokontroler Anda.

### 4. Penggunaan Sensor DHT22:

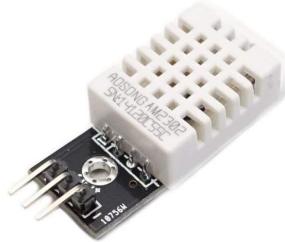
Sensor DHT22 digunakan dalam berbagai aplikasi, termasuk:

- Pemantauan iklim dalam proyek rumah pintar atau stasiun cuaca.
- Pengendalian suhu dan kelembaban dalam perangkat seperti inkubator atau peti penyimpanan makanan.
- Aplikasi IoT yang memerlukan pengukuran suhu dan kelembaban di lokasi yang berbeda.

### 5. Library Arduino untuk Sensor DHT22:

- Untuk menggunakan sensor DHT22 dengan Arduino, Anda dapat menggunakan library seperti "DHT.h" yang memungkinkan Anda membaca data dari sensor dengan mudah.
- Contoh kode Arduino biasanya disertakan dalam dokumentasi atau tutorial yang dapat ditemukan secara online.

Sensor DHT22 adalah pilihan yang baik untuk pengukuran suhu dan kelembaban dengan akurasi yang baik dan mudah digunakan dengan mikrokontroler seperti Arduino. Namun, perlu diingat bahwa sensor ini tidak tahan terhadap kondisi ekstrem seperti paparan air berkepanjangan, jadi perlu perhatian dalam penempatannya.



Gambar 6.2 Bentuk fisik sensor DHT22

### 3. Sensor PIR

Sensor PIR (Passive Infrared Sensor) adalah jenis sensor yang digunakan untuk mendeteksi perubahan suhu dalam bentuk radiasi inframerah yang dipancarkan oleh objek di sekitarnya. Sensor PIR sering digunakan dalam berbagai aplikasi untuk mendeteksi gerakan manusia atau hewan dalam ruangan atau di luar ruangan. Berikut adalah beberapa informasi dasar tentang sensor PIR:

#### 1. Prinsip Kerja Sensor PIR:

Sensor PIR bekerja berdasarkan perubahan perbedaan suhu dalam lingkungannya. Ketika objek bergerak, seperti manusia atau hewan, memasuki ruangan atau area yang diawasi oleh sensor, sensor ini akan mendeteksi perubahan suhu yang dihasilkan oleh objek tersebut.

Sensor ini tidak mengeluarkan radiasi inframerah sendiri, tetapi hanya mendeteksi perubahan suhu dalam lingkungannya.

#### 2. Struktur Sensor PIR:

- Sensor PIR memiliki beberapa elemen deteksi inframerah yang disusun dalam susunan kisi-kisi atau lensa Fresnel.
- Ketika objek bergerak melalui area sensor, elemen deteksi akan menerima radiasi inframerah dari objek tersebut, dan perubahan suhu ini akan diubah menjadi sinyal listrik oleh sensor.

### 3. Aplikasi Sensor PIR:

Sensor PIR digunakan dalam berbagai aplikasi, termasuk:

- **Lampu Sensor Gerak:** Untuk mengaktifkan lampu secara otomatis ketika ada gerakan dalam ruangan atau luar ruangan, sehingga menghemat energi.
- **Sistem Keamanan:** Dalam sistem alarm keamanan, sensor PIR digunakan untuk mendeteksi gerakan intruder.
- **Otomatisasi Rumah:** Dalam rumah pintar, sensor PIR digunakan untuk mengaktifkan perangkat atau sistem berdasarkan kehadiran manusia.
- **Penyiram Taman Otomatis:** Untuk mengaktifkan penyiram taman ketika ada gerakan binatang liar yang mendekat.

### 4. Fitur Sensor PIR:

- Sensor PIR adalah sensor yang pasif, yang berarti mereka tidak mengeluarkan radiasi sendiri dan hanya mendeteksi perubahan dalam radiasi inframerah yang ada di sekitarnya.
- Mereka memiliki rentang deteksi dan jarak operasi yang dapat diatur melalui lensa atau potensiometer yang ada pada sensor.
- Sensor ini umumnya sangat sensitif terhadap perubahan suhu yang signifikan, tetapi kurang sensitif terhadap perubahan suhu yang lebih kecil.

### 5. Koneksi Sensor PIR:

- Sensor PIR biasanya memiliki tiga pin: VCC (daya), GND (tanah), dan OUT (keluaran).
- Anda dapat menghubungkan VCC ke tegangan yang sesuai (biasanya 3.3V atau 5V), GND ke ground, dan OUT ke pin input/output digital pada mikrokontroler Anda.
- Sensor PIR adalah komponen yang sangat berguna dalam aplikasi yang melibatkan pengendalian berdasarkan gerakan atau kehadiran manusia. Mereka memungkinkan otomatisasi dan efisiensi energi dalam berbagai situasi.

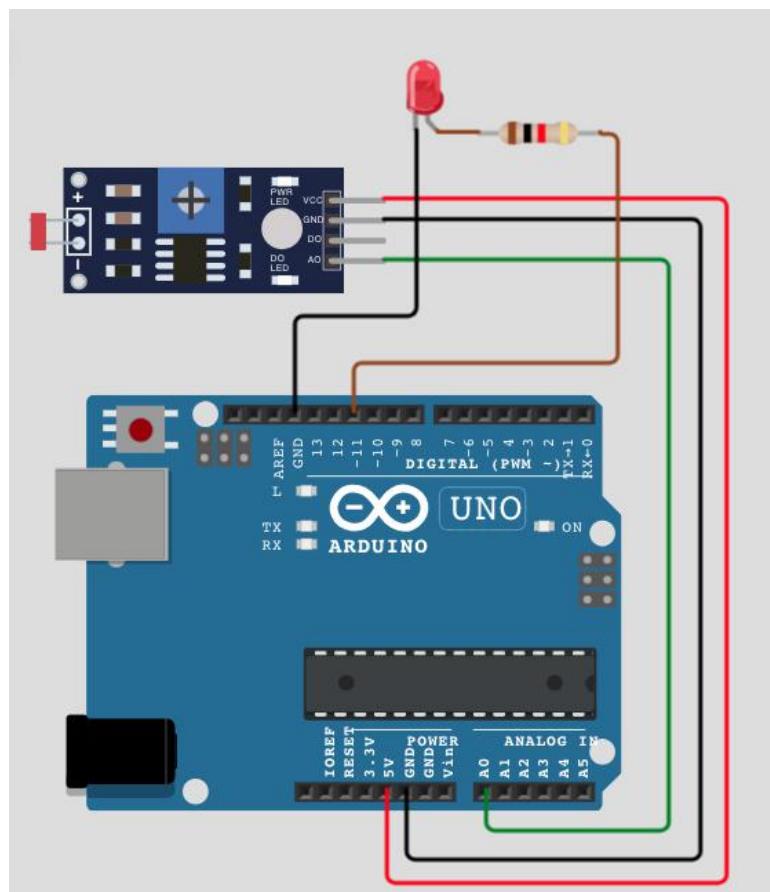


Gambar 6.3 Bentuk fisik sensor PIR

## D. PERCOBAAN

### 1. PERCOBAAN SENSOR LDR DENGAN LED

- 1). Buatlah rangkaian seperti pada gambar berikut ini.



Gambar 6.4 Rangkaian sensor LDR dan LED

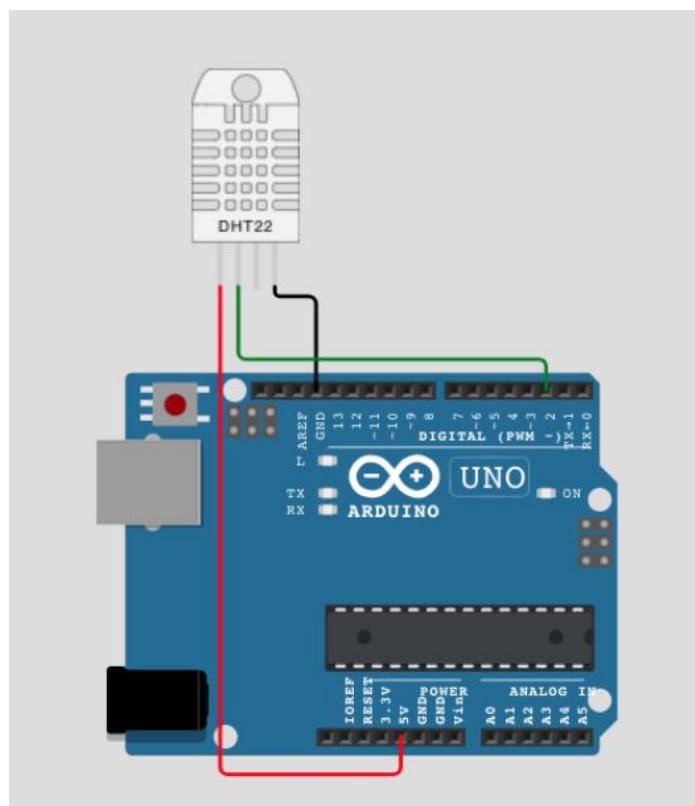
2). Ketikan kode program seperti berikut ini dan jalankan simulasinya.

Kode Program 6.1

```
1 int led = 11;
2
3 void setup() {
4     // put your setup code here, to run once:
5     pinMode(led, OUTPUT);
6 }
7
8 void loop() {
9     // put your main code here, to run repeatedly:
10    int ldr = analogRead(A0);
11    int b = constrain(ldr, 0, 255);
12    analogWrite(led, b);
13    delay(50);
14 }
```

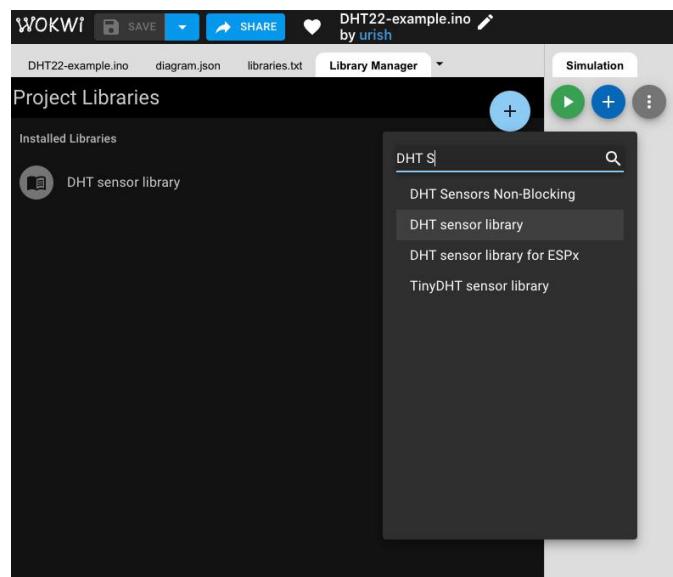
## 2. PERCOBAAN SENSOR DHT 22

1). Buatlah rangkaian seperti pada gambar berikut ini.



Gambar 6.5 Rangkaian sensor DHT 22

2). Pada tab Library Manager instal library silahkan tambahkan DHT sensor library, seperti pada gambar berikut ini.



Gambar 6.6 Instalasi library DHT

3). Ketikan kode program seperti berikut ini dan jalankan simulasinya.

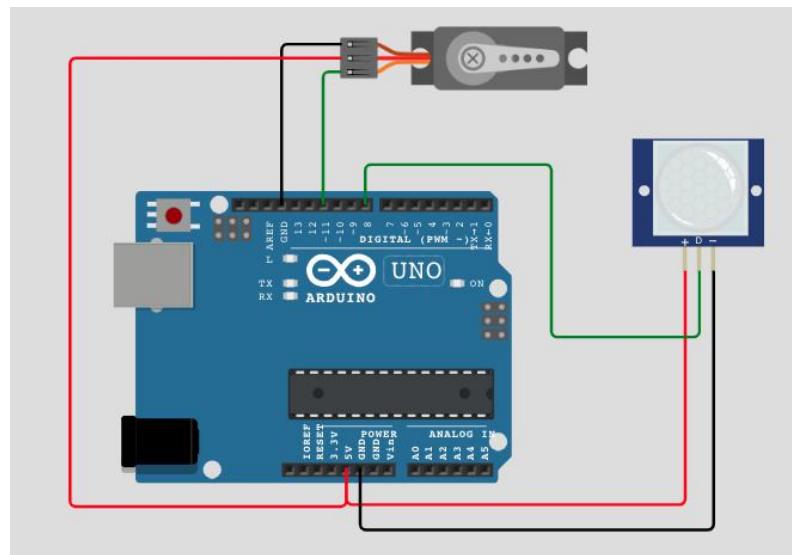
Kode Program 6.2

```
1 #include "DHT.h"
2
3 #define DHTPIN 2
4 #define DHTTYPE DHT22 // DHT 22 (AM2302), AM2321
5
6 DHT dht(DHTPIN, DHTTYPE);
7
8 void setup() {
9     Serial.begin(115200);
10    Serial.println(F("DHT22 example!"));
11    dht.begin();
12 }
13
14 void loop() {
15     float temperature = dht.readTemperature();
16     float humidity = dht.readHumidity();
17
18 // Check if any reads failed and exit early (to try again).
19     if (isnan(temperature) || isnan(humidity)) {
20         Serial.println(F("Failed to read from DHT sensor!"));
21         return;
22     }
23 }
```

```
24  Serial.print(F("Humidity: "));  
25  Serial.print(humidity);  
26  Serial.print(F("% Temperature: "));  
27  Serial.print(temperature);  
28  Serial.println(F("°C "));  
29  
30  // Wait a few seconds between measurements.  
31  delay(2000);  
32 }
```

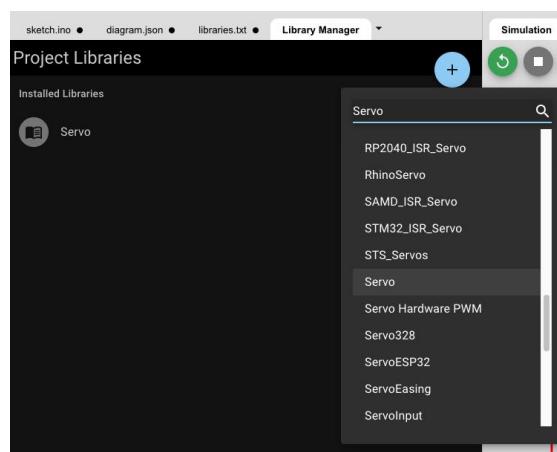
### 3. PERCOBAAN SENSOR PIR DAN SERVO

- 1). Buatlah rangkaian seperti pada gambar berikut ini.



Gambar 6.7 Rangkaian sensor PIR dan motor servo

- 2). Pada tab Library Manager tambahkan library servo, seperti pada gambar berikut ini.



Gambar 6.8 Menambahkan library servo

2). Ketikan kode program seperti berikut ini dan jalankan simulasinya.

Kode Program 6.3

```
1 #include <Servo.h>
2
3 int pir = 8;
4
5 Servo myservo;
6
7 void setup() {
8     // put your setup code here, to run once:
9     pinMode(pir, INPUT);
10    Serial.begin(115200);
11    myservo.attach(11);
12    myservo.write(0);
13 }
14
15 void loop() {
16     // put your main code here, to run repeatedly:
17     int pirVal = digitalRead(pir);
18     if (pirVal == 1)
19     {
20         myservo.write(180);
21         Serial.println("Pintu terbuka");
22         delay(10);
23     }
24     else
25     {
26         myservo.write(0);
27         Serial.println("Pintu tertutup");
28         delay(10);
29     }
30 }
```

## PRAKTIKUM 7

### MODUL 7 : PERCOBAAN PENGGUNAAN LCD

#### A. TUJUAN

1. Mampu menggunakan LCD pada Arduino 16X2
2. Mampu menggunakan layar OLED pada Arduino

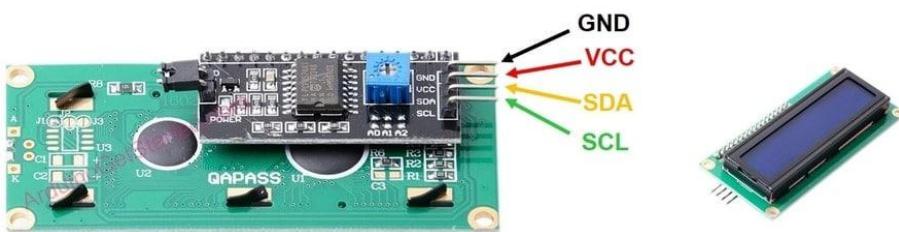
#### B. PERALATAN

1. Simulasi wokwi
2. Board Arduino UNO
3. LCD 16X2
4. Layar OLED

#### C. TEORI

##### 1. LCD

LCD dapat digunakan sebagai output yang menampilkan karakter alphanumeric. Ada beberapa jenis LCD dipasaran. Adapun LCD yang sering digunakan adalah LCD (16,2) yang artinya ada 16 kolom dan 2 baris. LCD ini juga memiliki 16 pin yang dapat dilihat pada gambar dibawah ini. Untuk mempermudah LCD terhubung dengan mikrokontroler sekarang di pasaran telah banyak LCD dengan interface I2C agar tidak banyak pin I/O yang digunakan pada mikrokontroler dan memudahkan wiring.



Gambar 7.1 PIN LCD dengan interface I2C

##### 2. Layar OLED

OLED (Organic Light-Emitting Diode) adalah teknologi tampilan yang semakin populer dalam berbagai aplikasi elektronik, termasuk layar pada perangkat ponsel cerdas, televisi, jam tangan pintar, dan banyak perangkat lainnya. Berikut adalah beberapa informasi dasar tentang layar OLED:

### 1. Prinsip Kerja Layar OLED:

Layar OLED menggunakan lapisan bahan organik yang dapat memancarkan cahaya ketika diberi arus listrik. Setiap piksel pada layar OLED adalah dioda kecil yang memancarkan cahaya sendiri, sehingga tidak memerlukan sumber cahaya luar (seperti latar belakang LED dalam LCD). Ketika arus listrik mengalir melalui piksel OLED, bahan organiknya memancarkan cahaya berdasarkan warna yang diinginkan.

### 2. Keuntungan Layar OLED:

- Kualitas Gambar yang Lebih Baik: Layar OLED memberikan kualitas gambar yang luar biasa dengan kontras yang tinggi, warna yang kaya, dan hitam yang benar-benar hitam karena piksel yang dimatikan tidak memancarkan cahaya.
- Konsumsi Daya yang Efisien: OLED hanya mengkonsumsi daya saat pikselnya dinyalakan, sehingga dapat menghemat daya baterai pada perangkat portabel.
- Fleksibilitas: OLED dapat dibuat dalam bentuk layar fleksibel atau layar yang dapat dilipat, memungkinkan desain perangkat yang lebih inovatif.

### 3. Jenis Layar OLED:

Ada beberapa jenis layar OLED, termasuk:

- Layar OLED Putih (WOLED): Menggunakan substrat warna putih yang menghasilkan semua warna dengan mengendalikan intensitas cahaya pada piksel.
- Layar OLED Berwarna (COLED): Menggunakan substrat warna berwarna merah, hijau, dan biru (RGB) yang menggabungkan tiga warna dasar untuk menghasilkan berbagai warna.
- Layar OLED Penuh (FOLED): Memiliki lapisan organik yang mengisi seluruh permukaan layar, memungkinkan desain yang lebih fleksibel.

### 4. Aplikasi Layar OLED:

Layar OLED digunakan dalam berbagai aplikasi, termasuk pada ponsel cerdas, televisi, jam tangan pintar, kamera digital, perangkat medis, peralatan rumah tangga, dan lainnya. Mereka juga sering digunakan dalam proyek-proyek elektronik hobi dan pengembangan prototipe.

### 5. Koneksi Layar OLED:

- a. Untuk menghubungkan layar OLED ke mikrokontroler atau papan pengembangan, Anda perlu menggunakan interface yang sesuai, seperti I2C (Inter-Integrated Circuit) atau SPI (Serial Peripheral Interface), tergantung pada model layar OLED yang Anda miliki.

b. Kebanyakan layar OLED memiliki pustaka perangkat lunak (library) yang dapat digunakan untuk mengontrol dan menampilkan teks atau grafik pada layar.

Layar OLED adalah teknologi tampilan yang menawarkan kualitas gambar yang luar biasa, konsumsi daya yang efisien, dan fleksibilitas desain. Mereka semakin umum digunakan dalam berbagai aplikasi dan proyek elektronik.

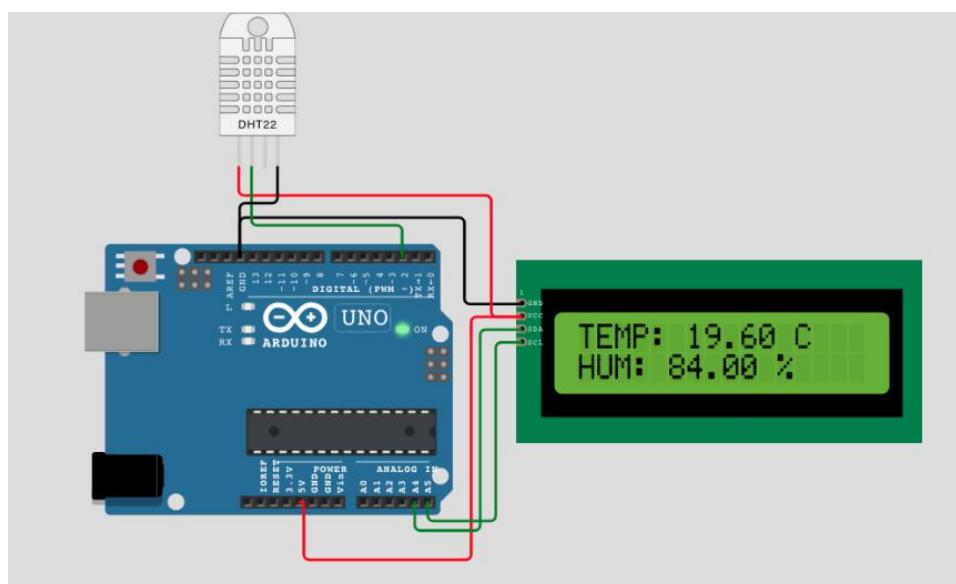


Gambar 7.2 Bentuk fisik layar OLED

## D. PERCOBAAN

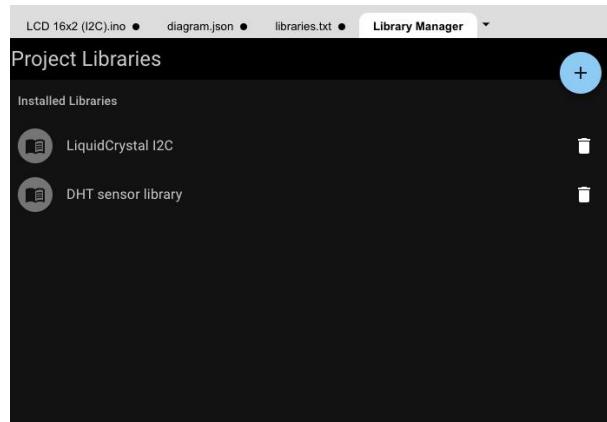
### 1. PERCOBAAN MENAMPILKAN NILAI SENSOR DHT 22 PADA LCD 16X2

1). Buatlah rangkaian seperti pada gambar berikut ini.



Gambar 7.3 Rangkaian sensor DHT 22 dan LCD 16X2

2). Pada tab Library Manager tambahkan library DHT sensor library dan LiquidCrystal I2C, seperti pada gambar berikut ini.



Gambar 7.4 Menambahkan library DHT sensor library dan LiquidCrystal I2C

3). Ketikan kode program seperti berikut ini dan jalankan simulasinya.

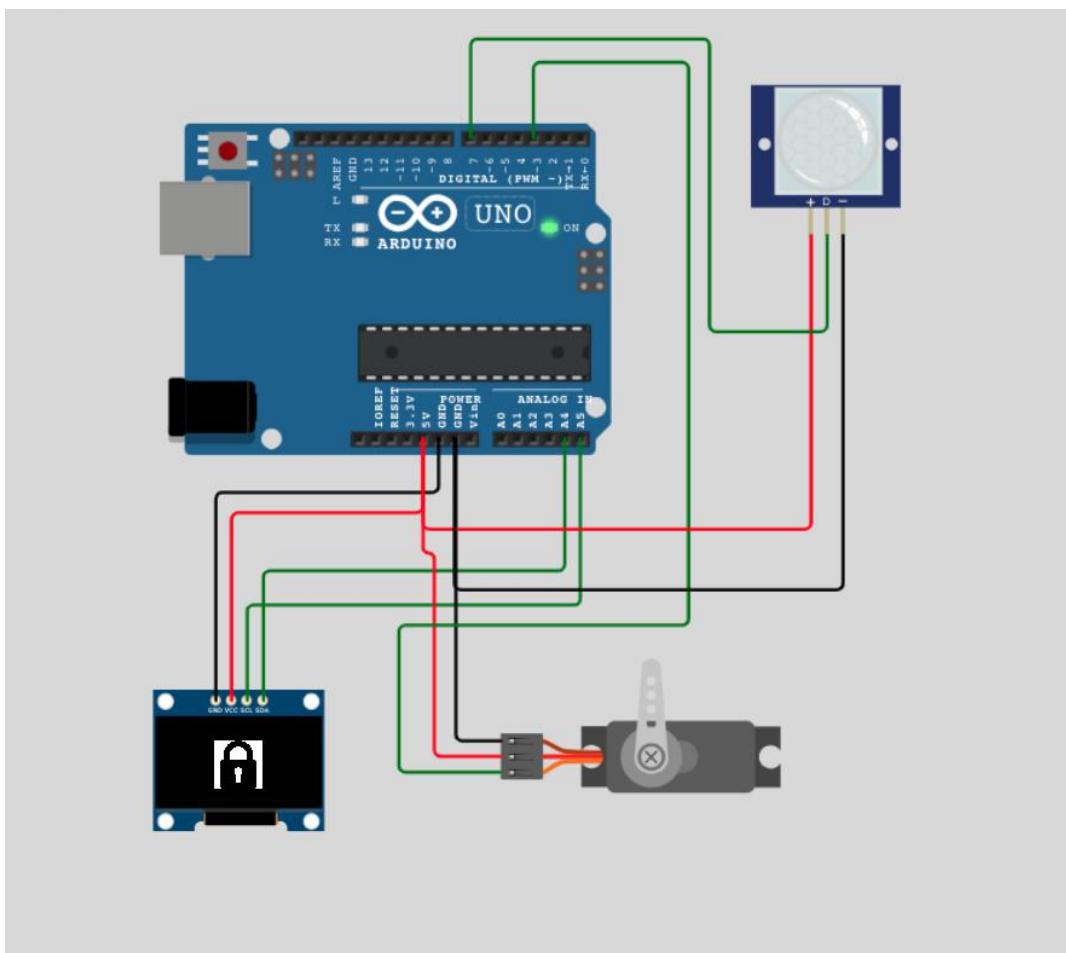
Kode Program 7.1

```
1 #include <LiquidCrystal_I2C.h>
2 #include "DHT.h"
3
4 #define DHTPIN 2
5 #define DHTTYPE DHT22 // DHT 22 (AM2302), AM2321
6
7 LiquidCrystal_I2C lcd(0x27, 16, 2);
8
9 DHT dht(DHTPIN, DHTTYPE);
10
11 void setup() {
12     lcd.init();
13     dht.begin();
14 }
15 void loop() {
16     float temperature = dht.readTemperature();
17     float humidity = dht.readHumidity();
18
19     if (isnan(temperature) || isnan(humidity)) {
20         lcd.backlight();
21         lcd.setCursor(0, 0);
22         lcd.print(F("Failed to read DHT!"));
23         return;
24     }
25
26     lcd.backlight();
```

```
27 lcd.setCursor(0, 0);
28 lcd.print("TEMP: ");
29 lcd.print(temperature);
30 lcd.print(" C");
31
32 lcd.setCursor(0, 1);
33 lcd.print("HUM: ");
34 lcd.print(humidity);
35 lcd.print(" %");
36 }
```

## 2. PERCOBAAN LAYAR OLED, MOTOR SERVO DAN SENSOR PIR

- 1). Buatlah rangkaian seperti pada gambar berikut ini.



Gambar 7.5 Rangkaian layar OLED, motor servo dan sensor PIR

2). Pada tab Library Manager tambahkan library seperti pada gambar berikut ini.



Gambar 7.6 Menambahkan library yang digunakan

4). Ketikan kode program seperti berikut ini dan jalankan simulasinya.

Kode Program 7.2

```

1 #include <SPI.h>
2 #include <Wire.h>
3 #include <Adafruit_GFX.h>
4 #include <Adafruit_SSD1306.h>
5 #include <Servo.h>
6
7 #define SCREEN_WIDTH 128 // OLED display width, in pixels
8 #define SCREEN_HEIGHT 64 // OLED display height, in pixels
9 Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire);
10
11 int pir = 7;
12 Servo myservo;
13
14 const unsigned char unlockIcon [] PROGMEM = {
15 0xff, 0xff, 0x81, 0xff, 0xff, 0xff, 0xfc, 0x00, 0x3f, 0xff, 0xff, 0xf8, 0x00,
16 0x1f, 0xff, 0xff,
17 0xf0, 0x00, 0x0f, 0xff, 0xe0, 0x00, 0x07, 0xff, 0xff, 0xc0, 0xfe, 0x03,
18 0xff, 0xff, 0x81,
19 0xff, 0x81, 0xff, 0x83, 0xff, 0xc1, 0xff, 0x87, 0xff, 0xc1, 0xff,
20 0xff, 0x07, 0xff,
21 0xe0, 0xff, 0x07, 0xff, 0xe0, 0xff, 0xff, 0x0f, 0xff, 0xe0, 0xff, 0xff,
22 0x0f, 0xff, 0xe0,
23 0xff, 0x0f, 0xff, 0xf0, 0xff, 0x0f, 0xff, 0xff, 0xff, 0xff, 0x0f,
24 0xff, 0xff, 0xff,
25 0xff, 0x0f, 0xff, 0xff, 0xff, 0x0f, 0xff, 0xff, 0xff, 0x0f, 0xff,
26 0xff, 0xff, 0xff,
27 0x0f, 0xff, 0xff, 0xfe, 0x00, 0x00, 0x00, 0x7f, 0xfc, 0x00, 0x00, 0x00,
28 0x3f, 0xfc, 0x00,
29 0x00, 0x00, 0x3f, 0xfc, 0x00, 0x00, 0x3f, 0xfc, 0x00, 0x00, 0x00, 0x3f,
```

```

30 0xfc, 0x00, 0x00,
31 0x00, 0x3f, 0xfc, 0x00, 0x00, 0x00, 0x3f, 0xfc, 0x00, 0x00, 0x00, 0x3f, 0xfc,
32 0x00, 0x1c, 0x00,
33 0x3f, 0xfc, 0x00, 0x3e, 0x00, 0x3f, 0xfc, 0x00, 0x3e, 0x00, 0x3f, 0xfc, 0x00,
34 0x3e, 0x00, 0x3f,
35 0xfc, 0x00, 0x1c, 0x00, 0x3f, 0xfc, 0x00, 0x00, 0x00, 0x3f, 0xfc, 0x00, 0x00,
36 0x00, 0x3f, 0xfc,
37 0x00, 0x00, 0x00, 0x3f, 0xfc, 0x00, 0x00, 0x00, 0x3f, 0xfc, 0x00, 0x00, 0x00,
38 0x3f, 0xfc, 0x00,
39 0x00, 0x00, 0x3f, 0xfe, 0x00, 0x00, 0x00, 0x7f
40 };
41
42 const unsigned char lockIcon [] PROGMEM = {
43 0xff, 0xff, 0x81, 0xff, 0xff, 0xff, 0xfc, 0x00, 0x3f, 0xff, 0xff, 0xf8, 0x00,
44 0x1f, 0xff, 0xff,
45 0xe0, 0x00, 0x07, 0xff, 0xff, 0xe0, 0x00, 0x03, 0xff, 0xff, 0xc0, 0x3c, 0x03,
46 0xff, 0xff, 0x80,
47 0xff, 0x01, 0xff, 0x81, 0xff, 0x81, 0xff, 0x03, 0xff, 0xc0, 0x0f, 0x0f,
48 0xff, 0x07, 0xff,
49 0xe0, 0xff, 0x07, 0xff, 0xe0, 0xff, 0xff, 0x07, 0xff, 0xe0, 0xff, 0xff,
50 0x07, 0xff, 0xe0,
51 0xff, 0xff, 0x07, 0xe0, 0xff, 0xff, 0x07, 0xff, 0xe0, 0xff, 0xff, 0x07,
52 0xff, 0xe0, 0xff,
53 0xfc, 0x00, 0x00, 0x00, 0x3f, 0xf8, 0x00, 0x00, 0x00, 0x1f, 0xf8, 0x00, 0x00,
54 0x00, 0x1f, 0xf8,
55 0x00, 0x00, 0x00, 0x1f, 0xf8, 0x00, 0x00, 0x00, 0x1f, 0xf8, 0x00, 0x00, 0x00,
56 0x1f, 0xf8, 0x00,
57 0x18, 0x00, 0x1f, 0xf8, 0x00, 0x7e, 0x00, 0x1f, 0xf8, 0x00, 0x7e, 0x00, 0x1f,
58 0xf8, 0x00, 0x7e,
59 0x00, 0x1f, 0xf8, 0x00, 0x7e, 0x00, 0x1f, 0xf8, 0x00, 0x7e, 0x00, 0x1f, 0xf8,
60 0x00, 0x3c, 0x00,
61 0x1f, 0xf8, 0x00, 0x3c, 0x00, 0x1f, 0xf8, 0x00, 0x3c, 0x00, 0x1f, 0xf8, 0x00,
62 0x3c, 0x00, 0x1f,
63 0xf8, 0x00, 0x3c, 0x00, 0x1f, 0xf8, 0x00, 0x3c, 0x00, 0x1f, 0xf8, 0x00, 0x3c,
64 0x00, 0x1f, 0xf8,
65 0x00, 0x00, 0x00, 0x1f, 0xf8, 0x00, 0x00, 0x00, 0x1f, 0xf8, 0x00, 0x00, 0x00,
66 0x1f, 0xf8, 0x00,
67 0x00, 0x00, 0x1f, 0xfc, 0x00, 0x00, 0x00, 0x3f
68 };
69
70 void setup() {
71   Serial.begin(9600);
72   pinMode(pir, INPUT);
73   // SSD1306_SWITCHCAPVCC = generate display voltage from 3.3V internally

```

```
74  if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
75    Serial.println(F("SSD1306 allocation failed"));
76    for (++);
77  }
78  myservo.attach(3);
79  myservo.write(0);
80 }
81
82 void loop() {
83   int pirVal = digitalRead(pir);
84   if (pirVal == 1)
85   {
86     myservo.write(180);
87     display.clearDisplay(); //clear sebelum tampilan baru
88     display.drawBitmap(44, 15, unlockIcon, 40, 40, WHITE);
89     display.display(); //tampilkan data
90     delay(3000);
91   }
92   else
93   {
94     myservo.write(0);
95     display.clearDisplay(); //clear sebelum tampilan baru
96     display.drawBitmap(44, 15, lockIcon, 40, 40, WHITE);
97     display.display(); //tampilkan data
98   }
99 }
```

## **DAFTAR PUSTAKA**

Arduino.cc. Language Reference. Diakses pada 17 September 2023, dari <https://www.arduino.cc/reference/en/>

Arduino.cc. Built-in Example. Diakses pada 17 September 2023, dari <https://docs.arduino.cc/built-in-examples/>

Arduino.cc. Libraries. Diakses pada 17 September 2023, dari <https://www.arduino.cc/reference/en/libraries/>

Circuits4you.com. 14 Desember 2014. Four Digit 7-Segment Display Interfacing with Arduino. Diakses pada 17 September 2023, dari <https://circuits4you.com/2016/12/14/segment-display-interfacing-arduino/>

Image2cpp. Image2cpp. Diakses 17 September 2023, dari <https://javl.github.io/image2cpp/>

Arduino.cc. Project Hub. Diakses pada 17 September 2023, dari <https://projecthub.arduino.cc/>