

### Modul 1 Pengenalan Editor dan Diagram Alir Pemrograman (Arduino & Thonny IDE) Mikrokontroler

#### 1.1 Tujuan Praktikum Modul

Setelah mempraktekkan topik ini, praktikan diharapkan dapat :

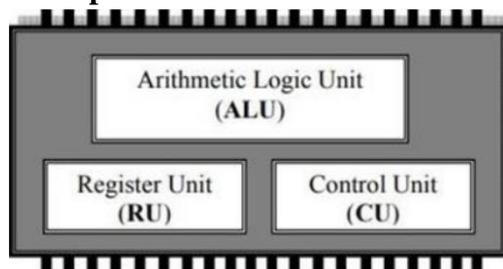
1. Praktikan dapat memahami tentang definisi Arduino & Thonny IDE
2. Praktikan dapat memahami alur kerja pemrograman pada platform Arduino & Thonny IDE
3. Mengimplementasikan kode sederhana pada Arduino dan ESP32 menggunakan Arduino IDE & Thonny IDE.

#### 1.2 Alat dan Bahan

1. Software Arduino IDE
2. Software Thonny
3. Laptop
4. ESP32
5. Arduino UNO

#### 1.3 Dasar Teori Praktikum Modul 1

##### 1.3.1 Microprocessor



Mikroprosesor, sering disebut sebagai Central Processing Unit (CPU), adalah unit utama yang mengendalikan proses komputasi. Beberapa contoh mikroprosesor yang umum di pasaran meliputi:

- Intel Core (i3, i5, i7)
- AMD Ryzen, Athlon, Phenom
- Qualcomm Snapdragon

Seiring dengan perkembangan teknologi, mikroprosesor juga dikenal sebagai MPU (Microprocessor Unit). Komponen utama dalam mikroprosesor meliputi:

1. **ALU (Arithmetic Logic Unit):** Melakukan operasi aritmetika (seperti penjumlahan) dan operasi logika (seperti AND, OR).
2. **Register Unit (RU):** Menyimpan data sementara dan hasil operasi ALU. Register utama disebut akumulator.
3. **Control Unit (CU):** Mengendalikan aliran data melalui bus data dan bus alamat.

## Modul Praktikum

### 1.3.2 Arduino IDE



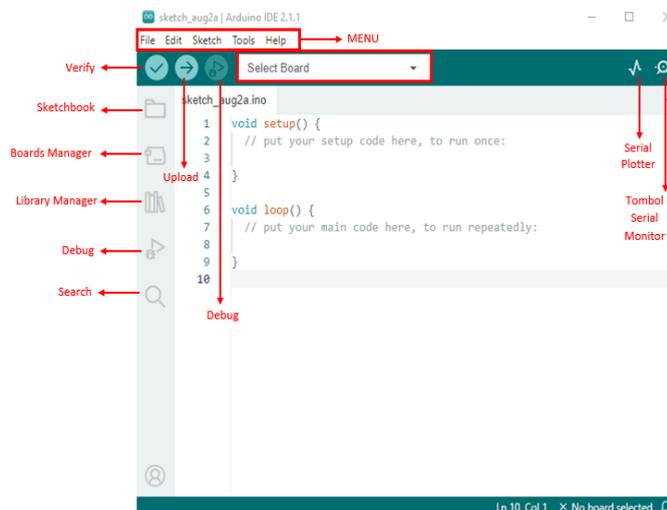
Arduino Integrated Development Environment (IDE) adalah perangkat lunak yang dirancang khusus untuk pemrograman dan pengembangan proyek dengan platform Arduino.

#### Fungsi dan Syntax Dasar pada Arduino

Sintaks/Fungsi	Deskripsi	Contoh Penggunaan
<code>setup()</code>	Fungsi ini digunakan untuk inisialisasi. Kode di dalam <code>setup()</code> hanya dijalankan sekali saat perangkat dimulai.	<pre>void setup() { pinMode(13, OUTPUT); }</pre>
<code>loop()</code>	Fungsi yang dijalankan terus-menerus setelah <code>setup()</code> selesai. Ideal untuk menjalankan program utama.	<pre>void loop() { digitalWrite(13, HIGH); delay(1000); }</pre>
<code>pinMode(pin, mode)</code>	Mengatur fungsi dari pin Arduino (input atau output).	<pre>pinMode(13, OUTPUT);</pre>
<code>digitalWrite(pin, val)</code>	Mengatur nilai pin digital (HIGH atau LOW).	<pre>digitalWrite(13, HIGH);</pre>
<code>digitalRead(pin)</code>	Membaca nilai pin digital (HIGH atau LOW).	<pre>int val = digitalRead(2);</pre>
<code>analogWrite(pin, val)</code>	Menulis nilai analog (PWM) pada pin output tertentu (0 hingga 255).	<pre>analogWrite(9, 128);</pre>
<code>analogRead(pin)</code>	Membaca nilai analog dari pin input tertentu (0 hingga 1023).	<pre>int sensorValue = analogRead(A0);</pre>
<code>delay(ms)</code>	Memberikan jeda selama beberapa milidetik.	<pre>delay(1000);</pre>
<code>Serial.begin(baudrate)</code>	Menginisialisasi komunikasi serial pada kecepatan baud tertentu.	<pre>Serial.begin(9600);</pre>
<code>Serial.print(val)</code>	Mencetak data ke monitor serial.	<pre>Serial.print("Hello World");</pre>
<code>Serial.println(val)</code>	Mencetak data ke monitor serial dengan pindah baris baru.	<pre>Serial.println("Data Received");</pre>

## Modul Praktikum Tools pada Arduino IDE

Nama Tool	Deskripsi	Fungsi
Verify (Centang)	Memeriksa apakah ada kesalahan dalam kode (kompilasi).	Memeriksa kesalahan sintaks dalam program.
Upload (Panah)	Mengunggah kode ke mikrokontroler Arduino setelah kompilasi berhasil.	Mengunggah kode ke perangkat Arduino.
New (Kertas)	Membuat sketch (proyek) baru.	Membuat file proyek baru.
Open (Folder)	Membuka sketch yang telah disimpan sebelumnya.	Membuka file proyek yang sudah ada.
Save (Disk)	Menyimpan sketch yang sedang dikerjakan.	Menyimpan proyek yang sedang dibuat.
Serial Monitor (Layar)	Menampilkan data yang dikirim dari dan ke Arduino melalui komunikasi serial.	Melihat hasil atau data dari komunikasi serial.
Board Manager	Mengelola jenis board yang digunakan.	Memilih board yang sesuai dengan perangkat.
Port	Menentukan port serial tempat Arduino terhubung ke komputer.	Memilih port tempat perangkat terhubung.



### Format Angka dalam Arduino

Format	Deskripsi	Contoh Deklarasi	Output
Binary	Angka biner diawali dengan <code>0b</code> .	<code>int varBiner = 0b1010;</code>	1010 (biner)
Hexadecimal	Angka heksadesimal diawali dengan <code>0x</code> .	<code>int varHex = 0xA;</code>	A (heksadesimal)
Octal	Angka oktal diawali dengan <code>0</code> .	<code>int varOkta1 = 012;</code>	12 (oktal)
Decimal	Angka desimal biasa.	<code>int var = 10;</code>	10 (desimal)

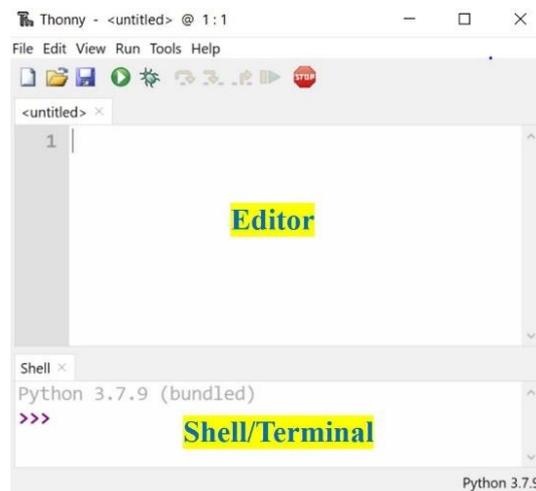
### Baudrate yang sering digunakan:

- 9600
- 115200
- 57600
- 38400

## Modul Praktikum

### 1.3.3 Thonny IDE

Thonny IDE adalah lingkungan pengembangan terintegrasi (IDE) untuk Python yang dirancang untuk kemudahan belajar, terutama bagi pemula. Thonny juga mendukung pemrograman mikrokontroler dengan MicroPython, seperti ESP32 dan Raspberry Pi Pico.



### 1.3.4 Perbandingan Arduino IDE dan Thonny IDE

Aspek	Arduino IDE	Thonny IDE
Bahasa Pemrograman	C/C++ (sederhana)	Python
Dukungan Platform	Arduino Boards	Mikrokontroler dengan MicroPython
Debugging	Tidak ada debugging terintegrasi	Debugging terintegrasi
Pengguna Sasaran	Pemula dan pengguna Arduino	Pemula Python dan MicroPython
Kelebihan	Mudah digunakan, komunitas besar	Mudah dipahami, mendukung MicroPython
Keterbatasan	Fitur terbatas untuk proyek kompleks	Terbatas pada Python

### 1.3.5 Micropython

MicroPython adalah implementasi bahasa pemrograman Python yang dirancang untuk mikrokontroler. Memiliki sintaks yang hampir sama dengan Python 3 dan banyak digunakan dalam pengembangan sistem tertanam.

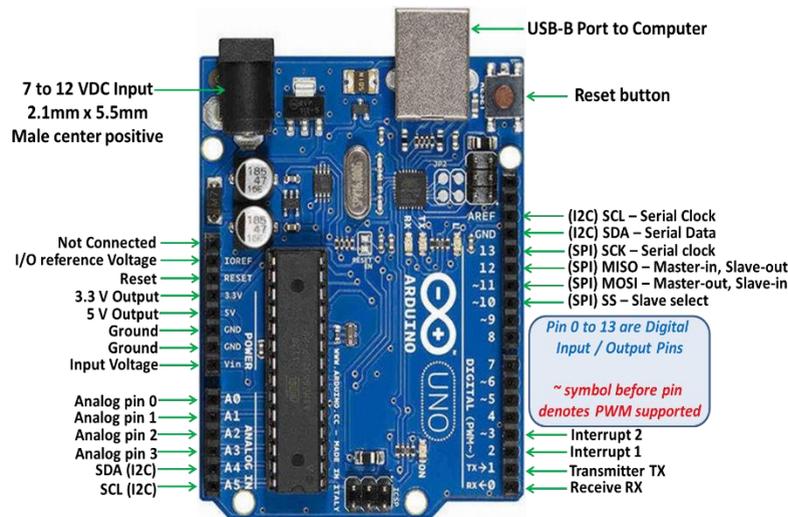
#### Syntax dasar

Sintaks	Deskripsi	Contoh Penggunaan
<code>print()</code>	Mencetak teks atau nilai ke terminal.	<code>print("Hello, MicroPython!")</code>
<code>time.sleep(seconds)</code>	Menunda program dalam detik.	<code>time.sleep(1)</code>
<code>time.sleep_ms(milliseconds)</code>	Menunda program dalam milidetik.	<code>time.sleep_ms(500)</code>
<code>machine.Pin(pin, mode)</code>	Mengatur pin sebagai input atau output.	<code>led = machine.Pin(2, machine.Pin.OUT)</code>
<code>led.value(1)</code>	Menyalakan pin output (misalnya LED).	<code>led.value(1)</code>
<code>led.value(0)</code>	Mematikan pin output.	<code>led.value(0)</code>

## Modul Praktikum

### 1.3.6 Arduino UNO

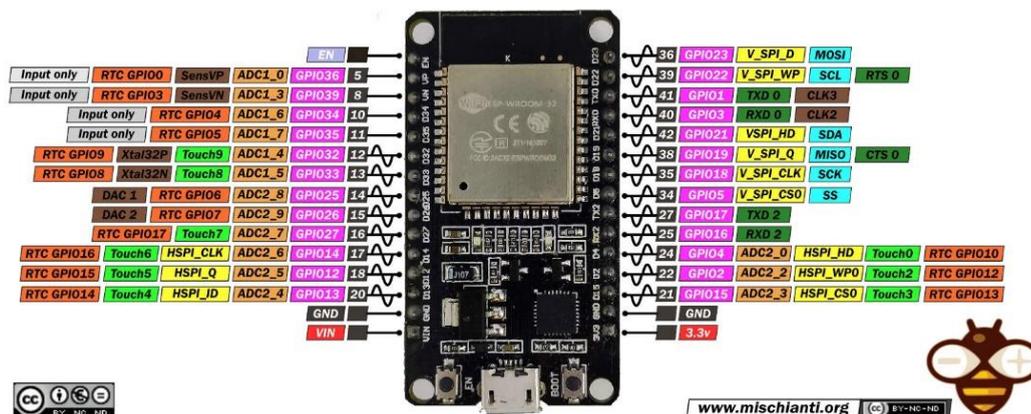
Arduino UNO adalah papan mikrokontroler berbasis sistem open hardware yang menggunakan mikrokontroler Atmel AVR. Arduino UNO telah dilengkapi dengan prosesor, memori, dan input/output (I/O) yang memungkinkan pengguna untuk mengendalikan berbagai perangkat elektronik.



### 1.3.7 ESP32

ESP32 adalah mikrokontroler yang dikembangkan oleh Espressif System, sebagai penerus dari ESP8266. Mikrokontroler ini kompatibel dengan Arduino IDE dan dilengkapi dengan modul WiFi serta Bluetooth Low Energy (BLE) dalam satu chip. Hal ini menjadikannya pilihan yang ideal untuk pengembangan aplikasi IoT.

#### ESP32 DEV KIT V1 PINOUT

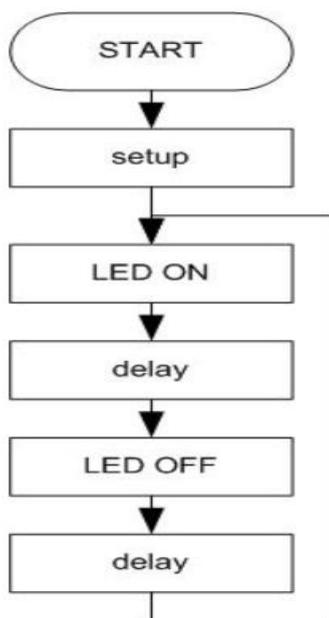


# Modul Praktikum

## 1.3.8 Simbol Flowchart

	<b>Flow</b> Simbol yang digunakan untuk menggabungkan antara simbol yang satu dengan simbol yang lain. Simbol ini disebut juga dengan Connecting Line.		<b>Input/output</b> Simbol yang menyatakan proses input atau output tanpa tergantung peralatan.
	<b>On-Page Reference</b> Simbol untuk keluar - masuk atau penyambungan proses dalam lembar kerja yang sama.		<b>Manual Operation</b> Simbol yang menyatakan proses yang tidak dilakukan oleh komputer.
	<b>Off-Page Reference</b> Simbol untuk keluar - masuk atau penyambungan proses dalam lembar kerja yang berbeda.		<b>Document</b> Simbol yang menyatakan bahwa input berasal dari dokumen dalam bentuk fisik, atau output yang perlu dicetak.
	<b>Terminator</b> Simbol yang menyatakan awal atau akhir suatu program.		<b>Predefine Proses</b> Simbol untuk pelaksanaan suatu bagian (sub-program) atau prosedur.
	<b>Process</b> Simbol yang menyatakan suatu proses yang dilakukan komputer.		<b>Display</b> Simbol yang menyatakan peralatan output yang digunakan.
	<b>Decision</b> Simbol yang menunjukan kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban, yaitu ya dan tidak.		<b>Preparation</b> Simbol yang menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberikan nilai awal.

## 1.3.9 Flowchart Program Blink



Pada Flowchart tersebut menggambarkan alur kerja program **blink** untuk mikrokontroler, seperti pada Arduino, yang bertujuan menyalakan dan mematikan LED secara bergantian dengan jeda waktu tertentu. Proses dimulai dengan inisialisasi di bagian **setup**, di mana pin yang terhubung ke LED diatur sebagai output.

Setelah inisialisasi, LED dinyalakan dengan memberikan sinyal HIGH pada pin tersebut, diikuti oleh periode tunggu atau **delay** selama satu detik untuk menjaga LED tetap menyala. Setelah periode ini, LED dimatikan dengan mengirimkan sinyal LOW ke pin yang sama, dan program kembali menunggu selama satu detik lagi.

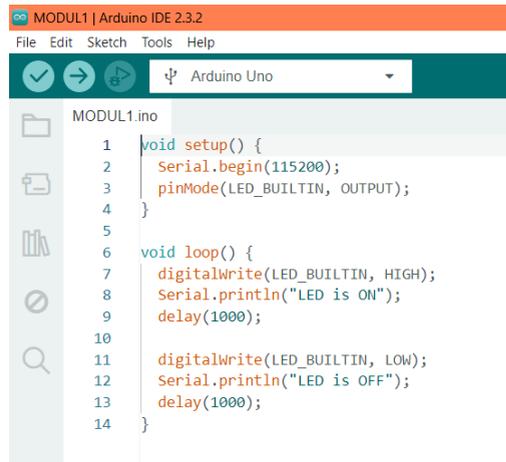
Siklus ini kemudian diulang secara terus-menerus, memastikan bahwa LED menyala dan mati secara bergantian dalam interval waktu yang sama, menciptakan efek kedip (blink) pada LED.

# Modul Praktikum

## 1.4 Langkah-langkah Praktikum Modul 1:

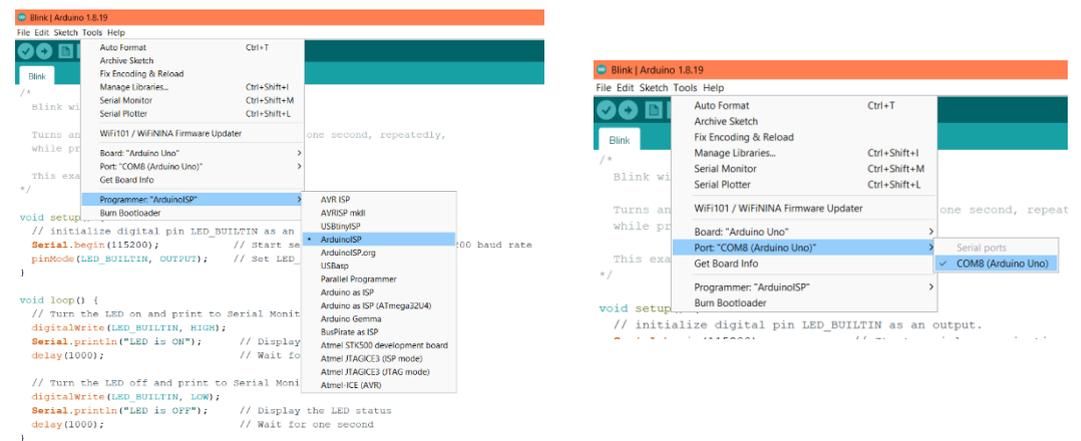
### 1.4.1 Menampilkan Blink ON/OFF pada Arduino

1. Buka Software Arduino IDE pada laptop yang telah terinstal.
2. Sambungkan USB Arduino UNO ke Laptop kalian.
3. Buat pemrograman seperti di bawah ini:

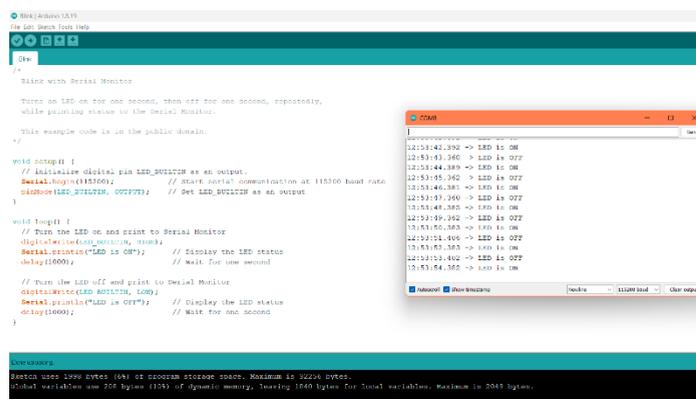


```
MODUL1 | Arduino IDE 2.3.2
File Edit Sketch Tools Help
Arduino Uno
MODUL1.ino
1 void setup() {
2   Serial.begin(115200);
3   pinMode(LED_BUILTIN, OUTPUT);
4 }
5
6 void loop() {
7   digitalWrite(LED_BUILTIN, HIGH);
8   Serial.println("LED is ON");
9   delay(1000);
10
11  digitalWrite(LED_BUILTIN, LOW);
12  Serial.println("LED is OFF");
13  delay(1000);
14 }
```

4. Pada tools, ubah board menjadi “Arduino UNO”, lalu pilih port yang tersambung pada Arduino IDE.



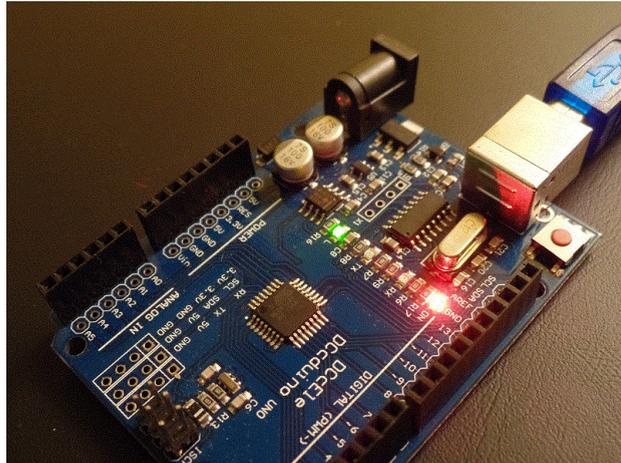
5. Lalu, compile dan upload program tersebut. Tunggu hingga proses upload program selesai hingga bertuliskan “Done Uploading”. Selanjutnya buka serial monitor dan akan menampilkan hasil seperti dibawah ini



6. Perhatikan Arduino saat selesai compile dan upload, dimana jika LED menyala

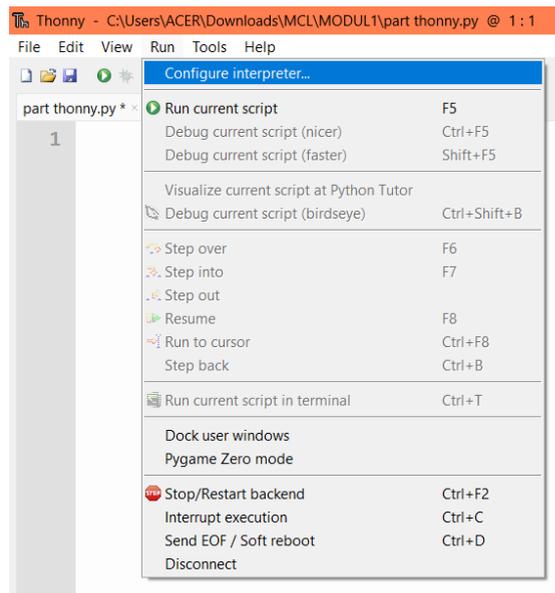
## Modul Praktikum

maka akan menampilkan output ON pada serial monitor begitupun sebaliknya



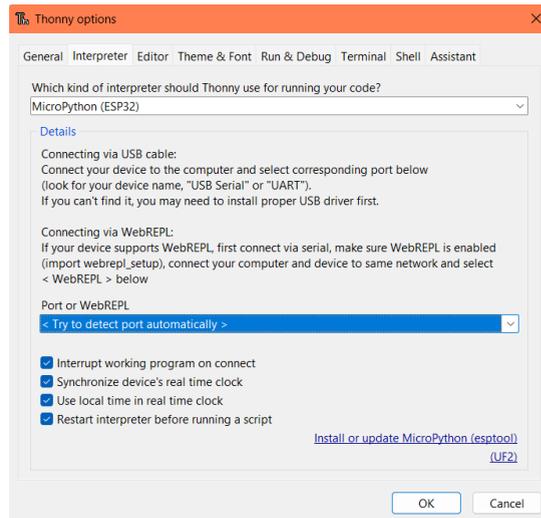
### 1.4.2 Langkah Menampilkan blink ON/OFF dengan Micropython

1. Download bahan yang telah disediakan (Firmware Micropython, Firmware blink, ESP32CAM, Thonny IDE)
2. Buka Thonny IDE
3. Hubungkan ESP32
4. Klik Run > Configure Interpreter

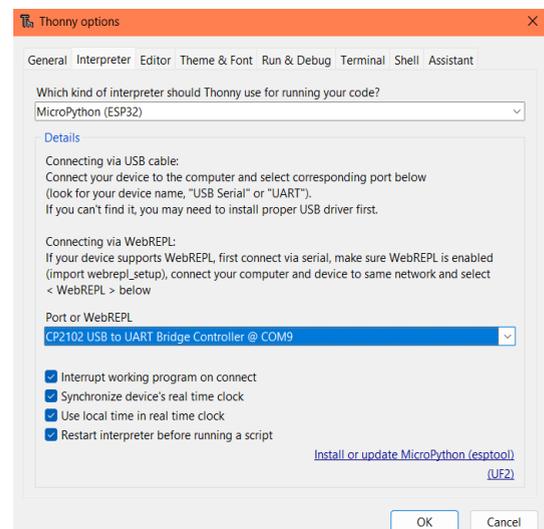
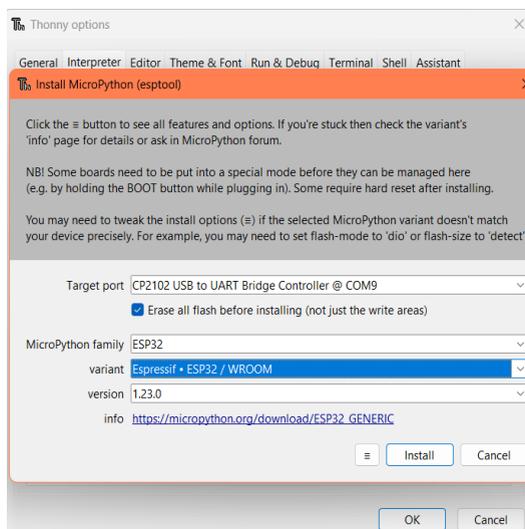


5. Pada which kind of interpreter should Thonny ..., pilih MicroPython (ESP32) lalu, klik Install or Update Micropython

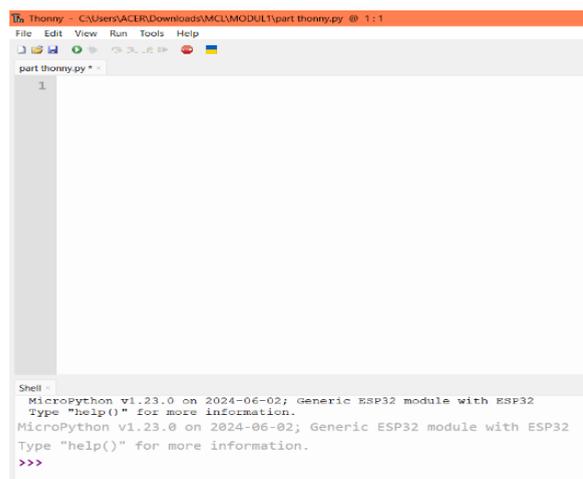
## Modul Praktikum



6. Pilih Port yang terbaca dan pilih file firmware Micropython yang disediakan. Lalu klik Install. Tunggu sampai selesai. lalu pada Which Kind of interpreter should ..., pilih MicroPython (ESP32) dan pada Port or WebREPL pilih USB-Serial sesuai port masing masing. Lalu klik ok

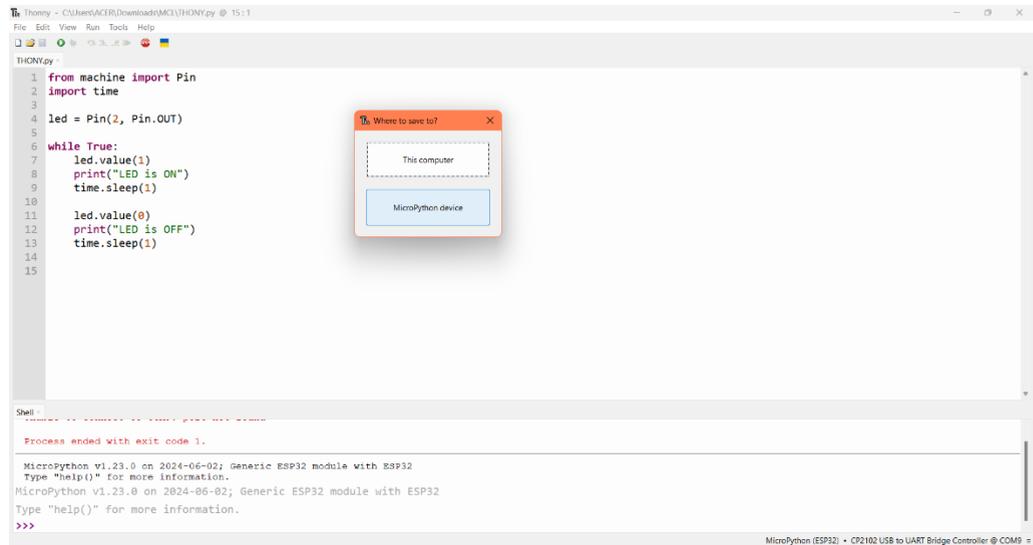


7. Jika pada Shell terdapat tulisan seperti berikut berarti micropython telah terinstall dengan baik



## Modul Praktikum

- Untuk mencetak melakukan blink ON/OFF, pada shell ketikkan perintah seperti dibawah dan save as MicroPython device dengan format file .py



- Lalu copy-kan perintah yang telah dibuat pada editor ke shell seperti dibawah ini, lalu enter



