# 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



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.

#### 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
<pre>setup()</pre>	Fungsi ini digunakan untuk inisialisasi. Kode di dalam setup() hanya dijalankan sekali saat perangkat dimulai.	<pre>void setup() { pinMode(13, OUTPUT); }</pre>
loop()	Fungsi yang dijalankan terus-menerus setelah setup() selesai. Ideal untuk menjalankan program utama.	<pre>void loop() { digitalWrite(13, HIGH); delay(1000); }</pre>
<pre>pinMode(pin, mode)</pre>	Mengatur fungsi dari pin Arduino (input atau output).	<pre>pinMode(13, OUTPUT);</pre>
<pre>digitalWrite(pin, val)</pre>	Mengatur nilai pin digital (HIGH atau LOW).	<pre>digitalWrite(13, HIGH);</pre>
digitalRead(pin)	Membaca nilai pin digital (HIGH atau LOW).	<pre>int val = digitalRead(2);</pre>
analogWrite(pin, val)	Menulis nilai analog (PWM) pada pin output tertentu (0 hingga 255).	analogWrite(9, 128);
analogRead(pin)	Membaca nilai analog dari pin input tertentu (0 hingga 1023).	<pre>int sensorValue = analogRead(A0);</pre>
delay(ms)	Memberikan jeda selama beberapa milidetik.	delay(1000);
Serial.begin(baudrate)	Menginisialisasi komunikasi serial pada kecepatan baud tertentu.	Serial.begin(9600);
Serial.print(val)	Mencetak data ke monitor serial.	<pre>Serial.print("Hello World");</pre>
Serial.println(val)	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 0b .	<pre>int varBiner = 0b1010;</pre>	1010 (biner)
Hexadecimal	Angka heksadesimal diawali dengan 0x .	<pre>int varHex = 0xA;</pre>	A (heksadesimal)
Octal	Angka oktal diawali dengan 0.	<pre>int varOktal = 012;</pre>	12 (oktal)
Decimal	Angka desimal biasa.	int var = 10;	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
<pre>print()</pre>	Mencetak teks atau nilai ke terminal.	<pre>print("Hello, MicroPython!")</pre>
<pre>time.sleep(seconds)</pre>	Menunda program dalam detik.	<pre>time.sleep(1)</pre>
<pre>time.sleep_ms(milliseconds)</pre>	Menunda program dalam milidetik.	<pre>time.sleep_ms(500)</pre>
<pre>machine.Pin(pin, mode)</pre>	Mengatur pin sebagai input atau output.	<pre>led = machine.Pin(2, machine.Pin.OUT)</pre>
<pre>led.value(1)</pre>	Menyalakan pin output (misalnya LED).	<pre>led.value(1)</pre>
<pre>led.value(0)</pre>	Mematikan pin output.	<pre>led.value(0)</pre>

#### 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.



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

#### 1.3.8 Simbol Flowchart

### **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.

### 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:



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

Blink   Arduino 1.8.19							
File Edit Sketch Tools Help							
Auto Format Archive Sketch Fix Encoding & Reload	Ctrl+T			Blink   Arduino 1.	8.19		
/* Manage Libraries Blink wi Serial Monitor Serial Plotter	Ctrl+Shift+I Ctrl+Shift+M Ctrl+Shift+L			File Edit Sketch To	ols Help Auto Format	Ctrl+T	
Turns an while pr Board: "Arduino Uno" Port "COM8 (Arduino Uno)" This exa	Updater > >	one second, repeatedly,		Blink /* Blink wi	Fix Encoding & Reload Manage Libraries Serial Monitor Serial Plotter	Ctrl+Shift+I Ctrl+Shift+M Ctrl+Shift+L	
Programmen "ArduinoISP"     void setup Burn Bootloader     // initialize digital pin LED	BUILTIN as an	AVR ISP AVRISP mkll USBtinyISP		Turns an while pr	WiFi101 / WiFiNINA Firmware U Board: "Arduino Uno"	pdater one seco	ond, re
<pre>Serial.begin(115200); pinMode(LED_BUILTIN, OUTPUT); }</pre>	// Start se // Set LED_	ArduinolSP     ArduinolSP.org     USBarp     Protilial Deservations	00 baud rate	This exa */	Port: "COM8 (Arduino Uno)" Get Board Info	> Serial po COM8 (A	rts <b>rduino Ur</b>
<pre>void loop() {     // Turn the LED on and print t</pre>	to Serial Monit	Arduino as ISP Arduino as ISP Arduino as ISP (ATmega32U4) Arduino Gemma		void setup	Programmer: "ArduinoISP" Burn Bootloader	>	
<pre>digitalwrite(LED_BUILTIN, HIGH Serial.println("LED is ON"); delay(1000);</pre>	// Display // Wait fo	BusPirate as ISP Atmel STK500 development board Atmel JTAGICE3 (ISP mode)		,, Initia	· ····	II a	
<pre>// Turn the LED off and print digitalWrite(LED_BUILTIN, LOW) Serial println("LED is OPP");</pre>	to Serial Moni	Atmel-ICE (AVR)					
<pre>delay(1000);</pre>	// Wait for	one second					

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

for one second, repeatedly,			
	COM8	-	
	1		
	12-52-41 202 -> TED is ON		
	12-53-43,360 > LED is OF		
	12:53:44.389 -> LED is ON		
an output.	12:53:45.362 -> LED is OFF		
Servial companies tim at 115200 baud Pate	12:53:46.381 -> LED is ON		
ab_boasian as an output	12:53:47.360 -> LED is OFF		
	12:53:48.385 -> 18D is ON		
	12:53:49.362 -> LED is OFF		
nitor	12:53:50.383 -> LED is ON		
	12:53:51.406 -> LED is OFF		
lay the LED status	12:53:52.303 -> LED is ON		
for one second	12:53:53.402 -> LED 15 OFF		
	12:53:54.382 -> LED 18 ON		
onitor			1
	Averagion a turn surgrams	100.0re 0 115200 5510	V Cear of
lay the LED status			
for one second			
	the can assess, repartedly,       	constraint, representing,           etc.           etc.	ext         2000           external         2000      2000         2000

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

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 Intrepeter

ኼ Thonny - C:\	Users\ACER\Downloads\MCL\MODUL1\part	thonny.py @ 1:1
File Edit Vie	w Run Tools Help	
🗋 💕 🛃 🛛 🖓	Configure interpreter	
part thonny.py	* × 💽 Run current script	F5
1	Debug current script (nicer)	Ctrl+F5
-	Debug current script (faster)	Shift+F5
	Visualize current script at Python Tutor	
	🐚 Debug current script (birdseye)	Ctrl+Shift+B
	🖘 Step over	F6
	🚴 Step into	F7
	🦂 Step out	
	🖙 Resume	F8
	Nun to cursor	Ctrl+F8
	Step back	Ctrl+B
	Run current script in terminal	Ctrl+T
	Dock user windows	
	Pygame Zero mode	
	Stop/Restart backend	Ctrl+F2
	Interrupt execution	Ctrl+C
	Send EOF / Soft reboot	Ctrl+D
	Disconnect	

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

eneral	Interpreter	Editor	Theme & Font	Run & Debug	Terminal	Shell	Assistant	
Which	kind of inter	preter sh	iould Thonny us	e for running yo	our code?			
MicroP	ython (ESP3	2)						~
Detai	ils							
(look If you	for your dev a can't find it	vice nam , you ma	e, "USB Serial" y need to instal	or "UART"). I proper USB dr	iver first.			
Conn If you (impo < We	ecting via W ur device sup ort webrepl_ ebREPL > bel	ebREPL: ports W setup), c ow	ebREPL, first co onnect your cor	nnect via serial, nputer and dev	make sure ' ice to same	WebRE netwo	PL is enab ork and sele	led ect
Conn If you (impo < We Port o	ecting via W ur device sup ort webrepl_ ebREPL > bel or WebREPL	ebREPL: ports W setup), c ow	ebREPL, first cor onnect your cor	nnect via serial, nputer and dev	make sure ' ice to same	WebRE netwo	PL is enab ork and sele	ect
Conn If you (impo < We Port o < Try	ecting via W ur device sup ort webrepl_ ebREPL > bel or WebREPL to detect po	ebREPL: ports W setup), c ow ort autor	ebREPL, first coi onnect your cor natically >	nnect via serial, nputer and dev	make sure ice to same	WebRE netwo	PL is enab ork and sele	ect

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

The Thonny options	$\times$ $~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~$
General Interpreter Editor Theme & Font Run & Debug Terminal Shell Assistant	General Interpreter Editor Theme & Font Run & Debug Terminal Shell Assistant X Which kind of interpreter should Thomps use for running your code?
Click the ± button to see all features and options. If you're stuck then check the variant's "info' page for details or ask in MicroPython forum. NBI Some boards need to be put into a special mode before they can be managed here (e.g. by holding the BOOT button while plugging in). Some require hard reset after installing. You may need to tweak the install options (=) if the selected MicroPython variant doesn't match your device precisely. For example, you may need to set flash-mode to 'dio' or flash-size to 'de	MicroPython (ESP32)
Target port (CP2102 USB to UART Bridge Controller @ COM9 C Erase all flash before installing (not just the write areas)	
MicroPython family [ESP32 variant Espressif • ESP32 / WROOM version [123.0 info_https://micromython.org/download/ESP32 / EENEBIC	Control and the second se
	Restart interpreter before running a script      Install or update MicroPython (esptool)      (UF2)
OK Cance	OK Cancel

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



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

Re Thom	ny - C.\Lisens\A.CER.Downloads\MCL\THONY.py @ 15:1	0	$\times$
File E	Sit View Run Tools Help		
0 😂 🛙	🛛 🛈 🕸 🗇 3 😂 🧧		
THON	ey ·		
1 2 3	from machine import Pin import time		^
4	led = Pin(2, Pin.OUT)		
6 7 8 9 10 11 12 13 14 15	<pre>while True: led.value(1) print("LED is ON") time.sleep(1) led.value(0) print("LED is OFF") time.sleep(1)</pre>		
Shell ×			,
Pro	cess ended with exit code 1.		
Mic Typ Micr	roPython v1.23.0 on 2024-06-02; Generic ESP32 module with ESP32 • "halp!)* for more information. Python v1.23.5 on 2024-06-02; Generic ESP32 module with ESP32		
Туре	"help()" for more information.		1
>>>			

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

Thonny - C:\Users\ACER\Do	wnloads/MCL/MODUL1/part thonny.py @ 1:1
File Edit View Run Tool	s Help
🗋 🐸 🖬 🛛 🕸 🔅 🕫 🔍 (e)	> 🐵 💻
part thonny.py * ×	
1	
4	
Shell ×	
Type "help()" for >>> from machine import time led = Pin(2,	more information. import Pin Pin.OUT)
while True: led.value print("LE time.slee led.value print("LE time.slee	(1) D is ON") p(1) (0) D is OFF") p(1)

10. Maka akan menampilkan hasil seperti dibawah ini pada shell dan ESP32





# SELAMAT MENGERJAKAN 😊