

Modul 3

PENGENALAN PEMOGRAMAN PEWAKTU, PENCACAH DAN INTERUPSI (ARDUINO UNO dan ESP32)

3.1 Tujuan

Setelah mengikuti praktikum Modul 3, mahasiswa diharapkan dapat:

1. Praktikan dapat Memahami Konsep Interupsi pada Arduino UNO dan ESP32
2. Praktikan dapat Mengimplementasikan Pewaktu (Timer) pada Arduino UNO dan ESP32
3. Praktikan dapat Mengaplikasikan Pencacah (Counter) pada Arduino UNO dan ESP32
4. Praktikan dapat Mengontrol Perangkat Output (LED) pada Arduino UNO dan ESP32

3.2 Alat dan Bahan

1. Software Arduino IDE
2. Arduino UNO
3. ESP32
4. LED
5. Resistor 220Ω untuk LED
6. Tombol (push button)
7. Resistor $10k\Omega$ untuk pull-down resistor
8. Breadboard
9. kabel jumper
10. Laptop

3.3 Dasar Teori Praktikum Modul 3

Dalam pemrograman mikrokontroler seperti Arduino Uno dan ESP32, konsep pewaktu (timer), pencacah (counter), dan interupsi (interrupt) sering digunakan untuk membuat sistem yang responsif dan efisien. Modul ini akan menjelaskan ketiga konsep tersebut serta cara menerapkannya menggunakan Arduino Uno dan ESP32.

Pengenalan Konsep

- **Pewaktu (Timer):** Digunakan untuk mengukur waktu, menghasilkan sinyal berkala, atau menjalankan suatu tindakan setelah waktu tertentu berlalu.
- **Pencacah (Counter):** Menghitung jumlah kejadian seperti penekanan tombol, sinyal dari sensor, atau putaran motor.

- **Interupsi (Interrupt):** Menghentikan program utama untuk segera merespon suatu kejadian seperti perubahan sinyal pada tombol.

3.3.1 Pewaktu (Timer)

Pewaktu (timer) pada Arduino adalah fitur yang terdapat dalam mikrokontroler AVR, berfungsi untuk mengukur waktu dan menentukan kapan program akan dijalankan dan dihentikan. Timer ini menggunakan fungsi `millis()` untuk menghitung waktu dalam milidetik sejak program dimulai. Selain itu, timer juga membantu mengatasi masalah bouncing pada tombol, yaitu fenomena di mana sinyal tombol bergetar ketika ditekan. Dengan menggunakan timer, kita dapat mencatat berapa lama sejak tombol terakhir ditekan dan memastikan bahwa penekanan tombol dianggap sah jika telah melewati batas waktu tertentu.

❖ Contoh Penggunaan Pewaktu:

1. Ketika tombol ditekan, kita mencatat waktu saat ini.
2. Jika waktu yang berlalu sejak penekanan terakhir lebih dari batas waktu tertentu (misalnya 50 milidetik), maka penekanan dianggap sah.

3.3.2 Pencacah (Counter)

Counter adalah rangkaian logika sekuensial yang berfungsi untuk menghitung jumlah pulsa yang masuk dan dinyatakan dalam bilangan biner. Counter dapat menghitung naik (counter up) atau turun (counter down) dan biasanya terbentuk dari rangkaian flip-flop sesuai kebutuhan. Alat ini sering digunakan dalam perangkat digital untuk mengontrol urutan operasi program.

Pencacah digunakan untuk menghitung berapa kali tombol ditekan. Pada program ini, setiap kali tombol ditekan dengan benar (setelah debouncing), nilai pencacah (count) akan bertambah 1.

3.3.3 Interupsi (Interrupt):

Interupsi adalah mekanisme yang memungkinkan mikrokontroler untuk menghentikan sementara eksekusi program utama dan menjalankan fungsi tertentu (Interrupt Service Routine atau ISR) ketika suatu peristiwa terjadi. Ini memungkinkan respons cepat terhadap peristiwa eksternal atau internal, seperti perubahan pada tombol, tanpa harus terus-menerus memeriksa status dalam loop utama. Dengan

Modul Praktikum

menggunakan interupsi, Arduino dapat segera merespons kejadian tanpa menunggu eksekusi program utama selesai.

Contoh dalam Arduino Uno dan ESP32:

- Dalam modul ini, contoh penggunaan interupsi dapat ditemukan dalam kode yang mengatur tombol dan LED. Saat tombol ditekan, interupsi akan memicu fungsi yang menghitung jumlah tekanan dan mengubah status LED.
- Penggunaan timer juga dapat menjadi contoh interupsi, di mana mikrokontroler mengeksekusi fungsi tertentu setelah periode waktu tertentu.

Mengapa Interupsi Penting:

1. **Responsif Terhadap Peristiwa:** Interupsi memungkinkan sistem untuk merespons peristiwa, seperti penekanan tombol atau sinyal dari sensor, secara cepat. Ini sangat penting dalam aplikasi real-time di mana respons cepat diperlukan.
2. **Efisiensi Pemrosesan:** Dengan menggunakan interupsi, mikrokontroler dapat menghemat waktu dan energi, karena dapat melakukan tugas lain atau berada dalam mode hemat energi (sleep mode) sampai interupsi terjadi.
3. **Pengelolaan Banyak Peristiwa:** Interupsi memungkinkan penanganan beberapa peristiwa secara bersamaan, sehingga meningkatkan efisiensi sistem. Ini memberi kemampuan untuk menangani beberapa tugas tanpa harus menunggu satu tugas selesai sebelum memulai yang lain.

Cara Memasang Interupsi pada Arduino:

```
attachInterrupt(digitalPinToInterrupt(PIN_INT), handleButtonPress, RISING);
```

1. **digitalPinToInterrupt(PIN_INT):** Fungsi ini mengubah nomor pin (PIN_INT) yang kita gunakan menjadi nomor interupsi yang sesuai, sehingga kita bisa mengaitkan pin itu dengan interupsi.
2. **handleButtonPress:** Ini adalah nama fungsi yang akan dijalankan ketika interupsi terjadi. Fungsi ini bertugas untuk menghitung berapa kali tombol ditekan dan mengubah status LED.
3. **RISING:** Ini adalah pengaturan yang berarti interupsi akan aktif saat sinyal pada pin berubah dari rendah (low) menjadi tinggi (high), seperti saat tombol ditekan.

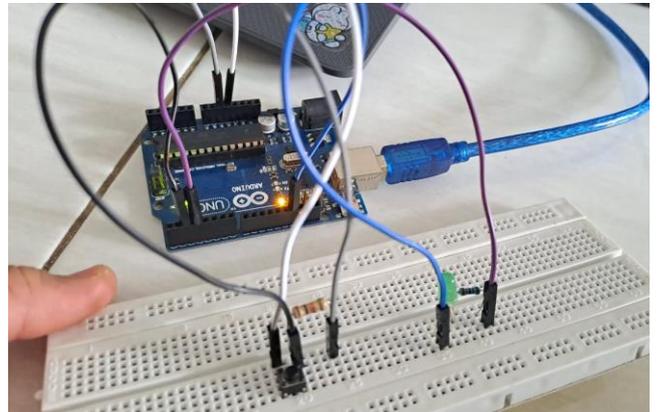
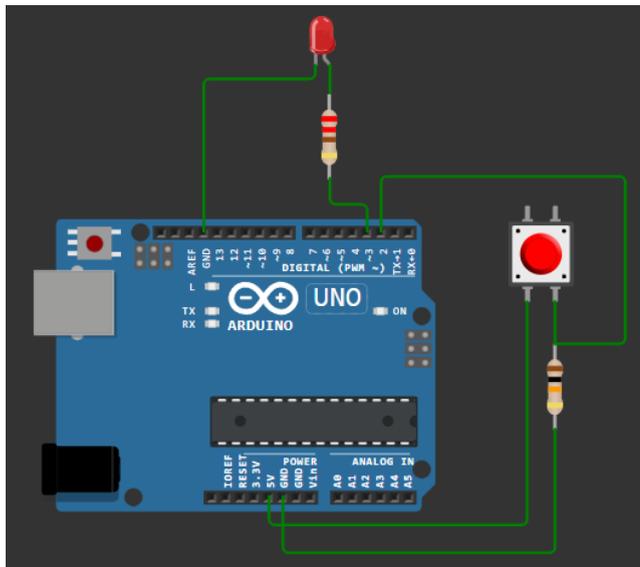
Modul Praktikum

3.4 Modul Praktikum

Mengendalikan LED dengan Tombol Menggunakan Interupsi

➤ Arduino Uno

1. Siapkan alat dan bahan
2. Rangkai Arduino UNO beserta komponennya seperti dibawah ini



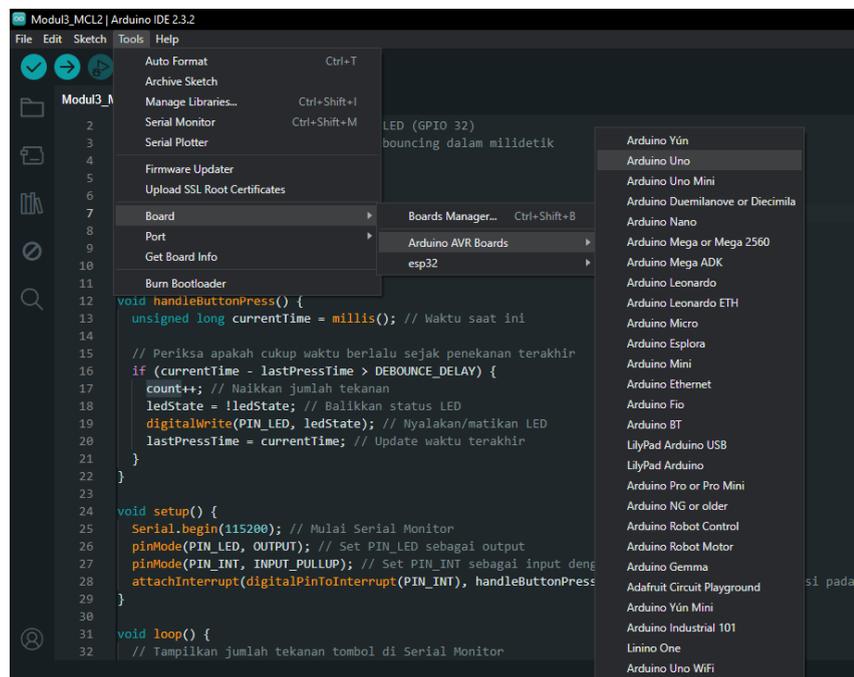
3. Buka Software Arduino IDE pada laptop yang telah terinstal.
4. Ketik script code di bawah ini.

```
1 #define PIN_INT 34 // Pin untuk tombol (GPIO 34)
2 #define PIN_LED 32 // Pin untuk LED (GPIO 32)
3 #define DEBOUNCE_DELAY 50 // Waktu debouncing dalam milidetik
4
5 #include <Arduino.h>
6
7 volatile unsigned long count = 0; // Hitung jumlah tekanan tombol
8 volatile bool ledState = false; // Status LED
9 unsigned long lastPressTime = 0; // Waktu terakhir tombol ditekan
10
11 // Fungsi untuk menangani interupsi
12 void handleButtonPress() {
13     unsigned long currentTime = millis(); // Waktu saat ini
14
15     // Periksa apakah cukup waktu berlalu sejak penekanan terakhir
16     if (currentTime - lastPressTime > DEBOUNCE_DELAY) {
17         count++; // Naikkan jumlah tekanan
18         ledState = !ledState; // Balikkan status LED
19         digitalWrite(PIN_LED, ledState); // Nyalakan/matikan LED
20         lastPressTime = currentTime; // Update waktu terakhir
21     }
22 }
23
24 void setup() {
25     Serial.begin(115200); // Mulai Serial Monitor
26     pinMode(PIN_LED, OUTPUT); // Set PIN_LED sebagai output
27     pinMode(PIN_INT, INPUT_PULLUP); // Set PIN_INT sebagai input dengan pull-up internal
28     attachInterrupt(digitalPinToInterrupt(PIN_INT), handleButtonPress, FALLING); // Pasang interupsi pada FALLING
29 }
```

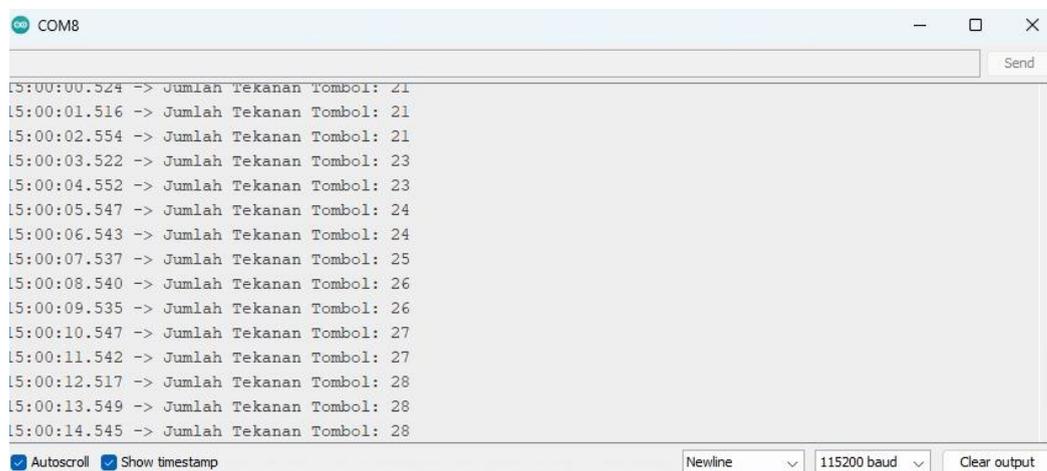
Modul Praktikum

```
30
31 void loop() {
32     // Tampilkan jumlah tekanan tombol di Serial Monitor
33     Serial.print("Jumlah Tekanan Tombol: ");
34     Serial.println(count); // Tampilkan jumlah tekanan terbaru
35     delay(1000); // Tunda 1 detik untuk memperbarui tampilan
36 }
```

5. Kemudian simpan file dengan format **Modul 3 MCL (1)**.
6. Sambungkan USB Arduino UNO ke Laptop kalian.
7. Pilih port dan jenis board yang sesuai di Arduino IDE seperti gambar di bawah ini.



8. kemudian Upload dan tunggu sampai Done
9. Buka Serial Monitor di Arduino IDE dan atur kecepatan dengan baud 115200.



10. Tekan tombol push button dan amati LED apakah sudah ON OFF. Perhatikan juga jumlah tekanan yang ditampilkan di Serial Monitor.

Modul Praktikum

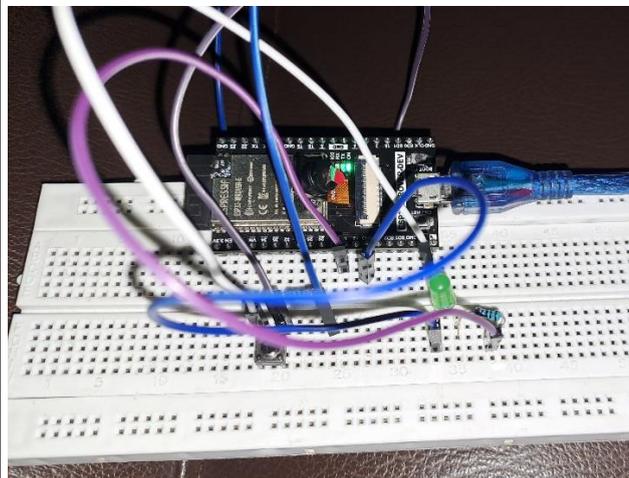
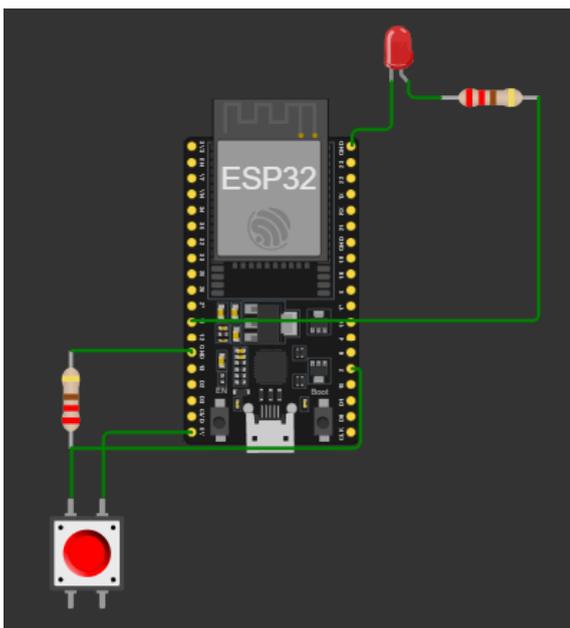
11. Coba tekan tombol beberapa kali untuk memastikan bahwa pembacaan jumlah tekanan berfungsi dengan baik.

Note:

Pastikan semua koneksi pada rangkaian sudah benar sebelum melakukan upload program. Jika LED tidak menyala atau tidak merespons, periksa kembali sambungan dan kode program.

➤ ESP32

1. Siapkan alat dan bahan sesuai dengan gambar di bawah ini.
2. Rangkailah ESP32 beserta komponennya seperti dibawah ini.



3. Buka Software Arduino IDE pada laptop yang telah terinstal.
4. Ketik script code di bawah ini.

```
/* DEKLARASI *****/

#define PIN_INT 2 // Pin untuk tombol (GPIO 2)
#define PIN_LED 14 // Pin untuk LED (GPIO 14)
#define DEBOUNCE_DELAY 50 // Waktu debouncing dalam milidetik

/* IMPLEMENTASI *****/
#include <Arduino.h> // Sertakan library Arduino

volatile unsigned long count = 0; // Hitung jumlah tekanan tombol
volatile bool ledState = false; // Status LED
unsigned long lastPressTime = 0; // Waktu terakhir tombol ditekan

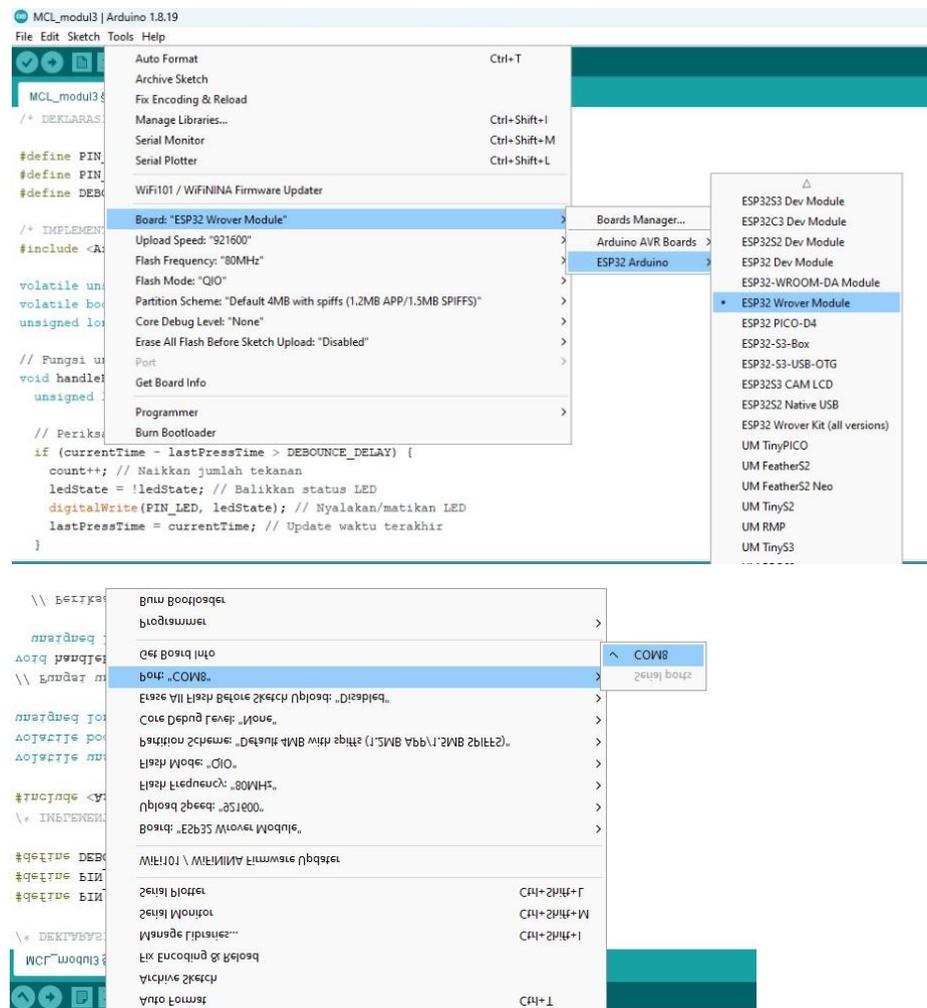
// Fungsi untuk menangani interupsi
void handleButtonPress() {
    unsigned long currentTime = millis(); // Waktu saat ini

    // Periksa apakah cukup waktu berlalu sejak penekanan terakhir
    if (currentTime - lastPressTime > DEBOUNCE_DELAY) {
        count++; // Naikkan jumlah tekanan
        ledState = !ledState; // Balikkan status LED
        digitalWrite(PIN_LED, ledState); // Nyalakan/matikan LED
        lastPressTime = currentTime; // Update waktu terakhir
    }
}
```

Modul Praktikum

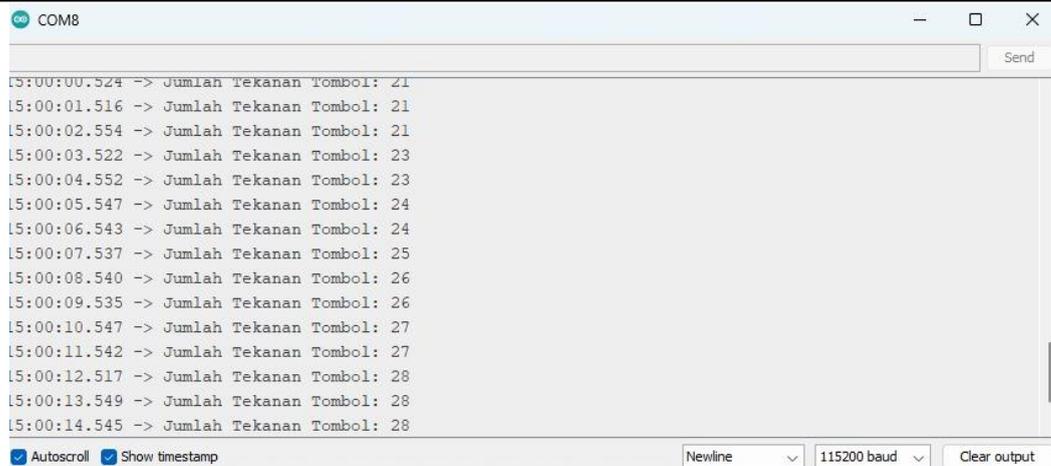
```
}  
  
void setup() {  
  Serial.begin(115200); // Mulai Serial Monitor  
  pinMode(PIN_LED, OUTPUT); // Set PIN_LED sebagai output  
  pinMode(PIN_INT, INPUT_PULLUP); // Set PIN_INT sebagai input dengan pull-up internal  
  attachInterrupt(digitalPinToInterrupt(PIN_INT), handleButtonPress, FALLING); // Pasang interupsi pada FALLING  
}  
  
void loop() {  
  // Tampilkan jumlah tekanan tombol di Serial Monitor  
  Serial.print("Jumlah Tekanan Tombol: ");  
  Serial.println(count); // Tampilkan jumlah tekanan terbaru  
  delay(1000); // Tunda 1 detik untuk memperbarui tampilan  
}
```

5. Kemudian simpan file dengan format **Modul 3 MCL (2)**.
6. Sambungkan ESP32 ke Laptop kalian dengan menggunakan kabel USB micro.
7. Pilih port dan jenis board yang sesuai di Arduino IDE sesuai dengan gambar di bawah ini.



8. kemudian Upload dan tunggu sampai Done.
9. Buka Serial Monitor di Arduino IDE dan atur kecepatan dengan baud 115200.

Modul Praktikum



The screenshot shows a Serial Monitor window titled 'COM8'. The output text is as follows:

```
15:00:00.524 -> Jumlah Tekanan Tombol: 21
15:00:01.516 -> Jumlah Tekanan Tombol: 21
15:00:02.554 -> Jumlah Tekanan Tombol: 21
15:00:03.522 -> Jumlah Tekanan Tombol: 23
15:00:04.552 -> Jumlah Tekanan Tombol: 23
15:00:05.547 -> Jumlah Tekanan Tombol: 24
15:00:06.543 -> Jumlah Tekanan Tombol: 24
15:00:07.537 -> Jumlah Tekanan Tombol: 25
15:00:08.540 -> Jumlah Tekanan Tombol: 26
15:00:09.535 -> Jumlah Tekanan Tombol: 26
15:00:10.547 -> Jumlah Tekanan Tombol: 27
15:00:11.542 -> Jumlah Tekanan Tombol: 27
15:00:12.517 -> Jumlah Tekanan Tombol: 28
15:00:13.549 -> Jumlah Tekanan Tombol: 28
15:00:14.545 -> Jumlah Tekanan Tombol: 28
```

At the bottom of the window, there are checkboxes for 'Autoscroll' and 'Show timestamp', both of which are checked. There are also dropdown menus for 'Newline' and '115200 baud', and a 'Clear output' button.

10. Tekan tombol setiap satu detik satu kali dan amati hasil output di Serial Monitor. Anda akan melihat jumlah tekanan tombol meningkat setiap kali tombol ditekan.

Note:

Jika LED tidak menyala, periksa sambungan kabel dan pastikan resistornya terhubung dengan benar. Pastikan juga bahwa Anda menggunakan resistor yang sesuai untuk LED dan pull-up tombol.

SELAMAT MENGERJAKAN 😊