6.1 Praktikum Pemrograman Mikrokontroler Wemos D1 dan Optimasinya

Indobot Academy 21 November 2022

1. Dasar Teori

1.1. Kurung Kurawal

Kurung kurawal atau { } merupakan salah satu tanda (sintaks) yang bertujuan untuk membatasi awal dan akhir program yang ada di suatu method atau fungsi.

Contoh cara penggunaan :

```
void loop()
{
  int jumlahsiswa_masuk = 10;
  int jumlahsiswa_bolos = 2;
  int jumlahtotal_siswa = jumlahsiswa_masuk +
jumlahsiswa_bolos;
}
```

1.2. Titik Koma

Titik koma atau ; merupakan salah satu tanda (sintaks) yang bertujuan untuk mengakhiri baris kode.

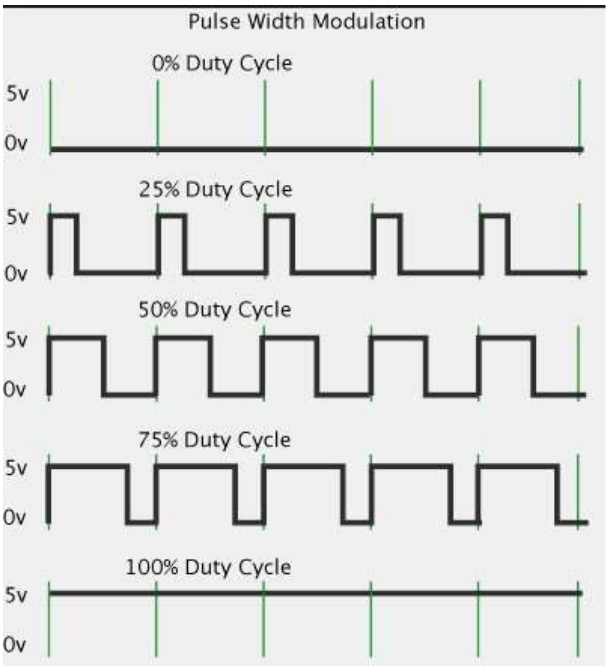
Contoh cara penggunaan :

```
int jumlah_siswa;
String nama_siswa;
```

1.3. PWM (Pulse Width Modulation)

PWM adalah metode yang sering digunakan untuk mengontrol daya. Selain sebagai pengatur daya, PWM juga berfungsi sebagai pengatur gerak dalam sebuah perangkat elektronika. Sesuai namanya yakni Pulse Width Modulation, maka dalam sistemnya, PWM digunakan untuk mengubah lebar pulsa. Hal ini karena pada umumnya, sinyal PWM memiliki frekuensi dasar dan juga amplitudo yang terbilang tetap. Dalam perhitungannya, lebar pulsa dalam PWM dibuat berbanding lurus dengan amplitudo. Artinya, sinyal PWM ini memiliki frekuensi gelombang yang tetap. Namun tetap saja memiliki nilai dutycycle yang berbeda, yaitu dengan digit nilai antara 0 sampai dengan 100%.

Cara Kerja PWM



Gambar 1. Duty Cycle PWM

Metode PWM memang dibuat dengan tujuan untuk mendapatkan sinyal analog dari piranti digital. Untuk membangkitkan sinyal analog pada PWM, kita dapat melakukan berbagai cara. Salah satunya dengan memanfaatkan metode analog dan digital. Ketika menggunakan metode analog, perubahan PWM terjadi dengan sangat halus. Namun ketika kita menggunakan metode digital, maka perubahan pada PWM akan di pengaruhi oleh resolusi dari alat itu sendiri. Untuk menghitung resolusinya dari PWM, kita dapat menggunakan rumus sederhana. Misalnya sebuah PWM yang memiliki resolusi 8 bit, maka nilai PWM tersebut memiliki perubahan variasi sebanyak 0 sampai dengan 225. Nilai ini mewakili dutycycle yang dikeluarkan oleh PWM tersebut. Yang mana PWM memiliki nilai antara 0 sampai dengan 100 %.

Kita dapat mengirimkan sinyal HIGH (ON) dan LOW (OFF) pada Wemos D1 untuk menyalakan dan mematikan suatu perangkat menggunakan digitalWrite(). Bagaimana kalo kita ingin mengaktifkan sesuatu dengan kekuatan tertentu dan bisa diubah-ubah sesuai keperluan, seperti misalnya mengatur intensitas cahaya LED atau mengatur kecepatan putaran motor? Dalam hal ini kita punya analog output. Analog output pada Arduino IDE berarti kita mengirimkan sinyal analog dengan intensitas yang ditentukan sesuai kebutuhan.

Analog input dihasilkan oleh teknik yang dikenal dengan istilah PWM atau Pulse Width Modulation. PWM memanipulasi keluaran digital sedemikian rupa sehingga menghasilkan sinyal analog. Mikrokontroler mengeset output digital ke HIGH dan LOW bergantian dengan porsi waktu tertentu untuk setiap nilai keluarannya. Durasi waktu untuk nilai HIGH disebut pulse width atau panjang pulsa. Variasi nilai output analog didapatkan dari perubahan panjang pulsa yang diberikan pada satu periode waktu dan dilakukan secara berulang kali.

Untuk mengatur nilai duty cycle, kita gunakan fungsi analogWrite([nomorPin], [nilai]). Nilai pada parameter, tersedia berkisar antara 0 hingga 255. Bila kita hendak mengeset duty cycle ke 0%, maka kita set nilai parameter ke 0, dan untuk duty cycle 100%, maka kita set nilai parameter ke 255. Jadi bila misalkan kita hendak mengeset duty cycle ke 50%, berarti nilai yang harus kita set adalah 127 (50% x 255).

```
analogWrite(D2, 127);
```

Sebenarnya berdasarkan konsep PWM di atas, kita dapat mensimulasikan PWM pada semua pin digital. Tapi khusus penggunaan fungsi digitalWrite() kita hanya bisa menggunakannya pada pin-pin PWM. Biasanya pin PWM disimbolkan dengan karakter '~'.

Demikian sekilas tentang teori PWM. Supaya lebih paham tentang analog output, kita coba implementasikan menggunakan Wemos.

2. Alat/Instrumen/Media/Aparatus

- Laptop /PC.
- Wemos D1 R1/R2.
- Resistor 10K Ohm.
- Resistor 470 Ohm.
- Sensor LDR.
- LED.

3. Keselamatan Kerja

3.1. Pemasangan Komponen

Perhatikan bagian pin yang digunakan. Terutama bagian komponen yang memiliki polaritas, jangan sampai terbalik antara kaki positif dan negatif.

3.2. Penggunaan Ukuran Resistor

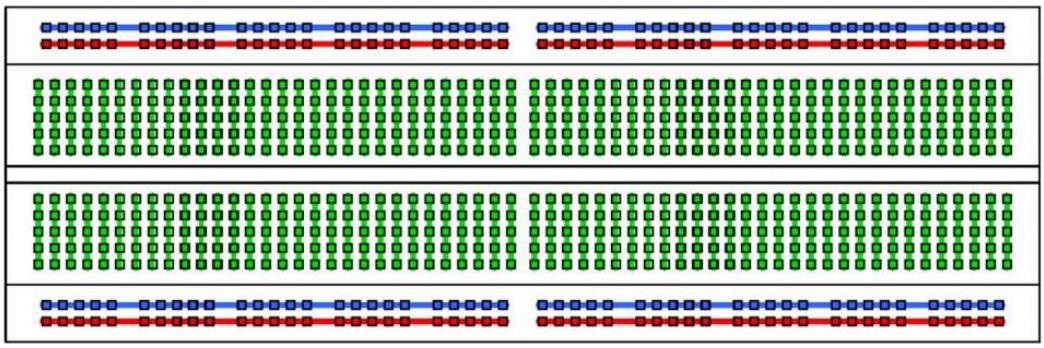
Hal yang perlu diperhatikan lainnya adalah mengenai ukuran resistor. Ukuran resistor dapat menyesuaikan dengan gambar rangkaiannya.

3.3. Perhatikan pin

Selanjutnya kita juga perlu memperhatikan pin yang ada dalam wemos D1 maupun sensor.

3.4. Pemahaman Jalur Project Board

Agar kita mengetahui tentang jalur yang ada pada project board, kita bisa melihat gambar skema dalam project board seri MB-102 berikut ini.



Gambar 2. Jalur Project Board

- Bagian tengah project board akan saling terhubung secara vertikal setiap 5 pin. Kemudian akan ada celah, nah celah ini bisa gunakan untuk meletakkan push button atau komponen lainnya.

- Untuk bagian atas dan bawah ini terhubung secara horizontal, dengan celah juga di bagian tengah dari project board.

Dikarenakan ada 2 versi Wemos D1, Rekan-rekan bisa memperhatikan device yang digunakan kemudian disesuaikan untuk pin-pinnya.

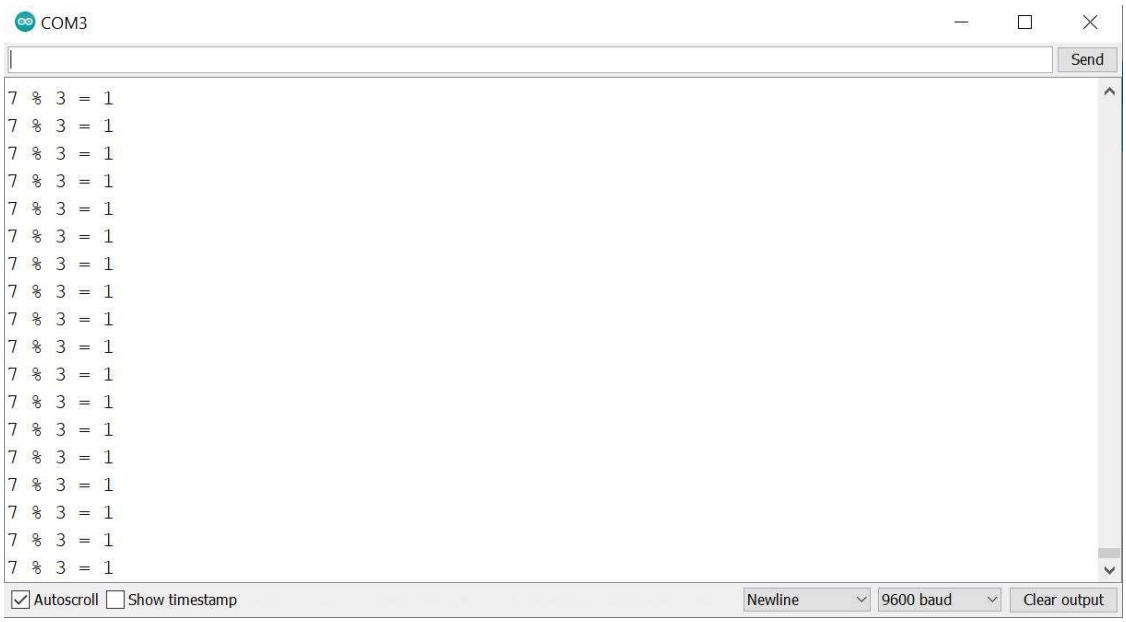
4. Langkah Praktikum 1 – Operator Aritmetika

Praktikum 4.1

- **Modulus (%)**
  Operator jenis ini digunakan untuk mendapatkan hasil perhitungan sisa bagi dari dua bilangan atau lebih, contohnya : 7 % 3 = 1.

```
int x = 7;
int y = 3;
int z = x % y;
void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
}

void loop() {
  // put your main code here, to run repeatedly:
  Serial.println(String(x)+" % "+String(y)+" =
"+String(z));
}
```

Hasil program pada Serial Monitor :

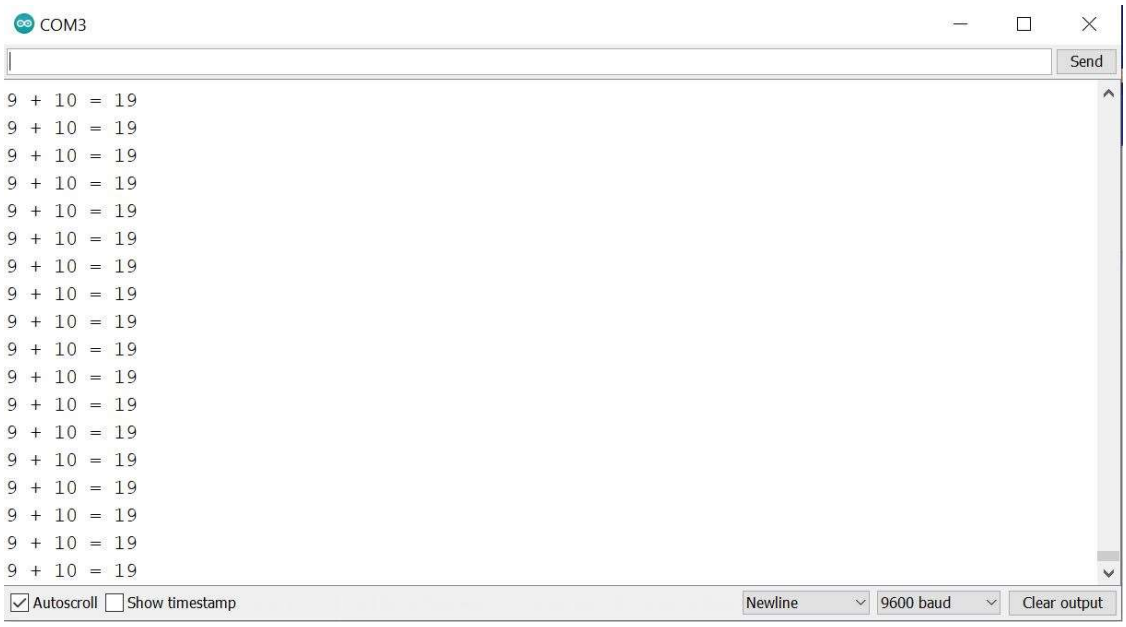


Praktikum 4.2

- **Penjumlahan (+)**
  Operator jenis ini digunakan untuk mendapatkan hasil penjumlahan dari dua bilangan atau lebih, contohnya : 9 + 10 = 19.

```
int x = 9;
int y = 10;
int z = x + y;
void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
}

void loop() {
  // put your main code here, to run repeatedly:
  Serial.println(String(x)+" + "+String(y)+" =
"+String(z));
}
```

Hasil program pada serial monitor :

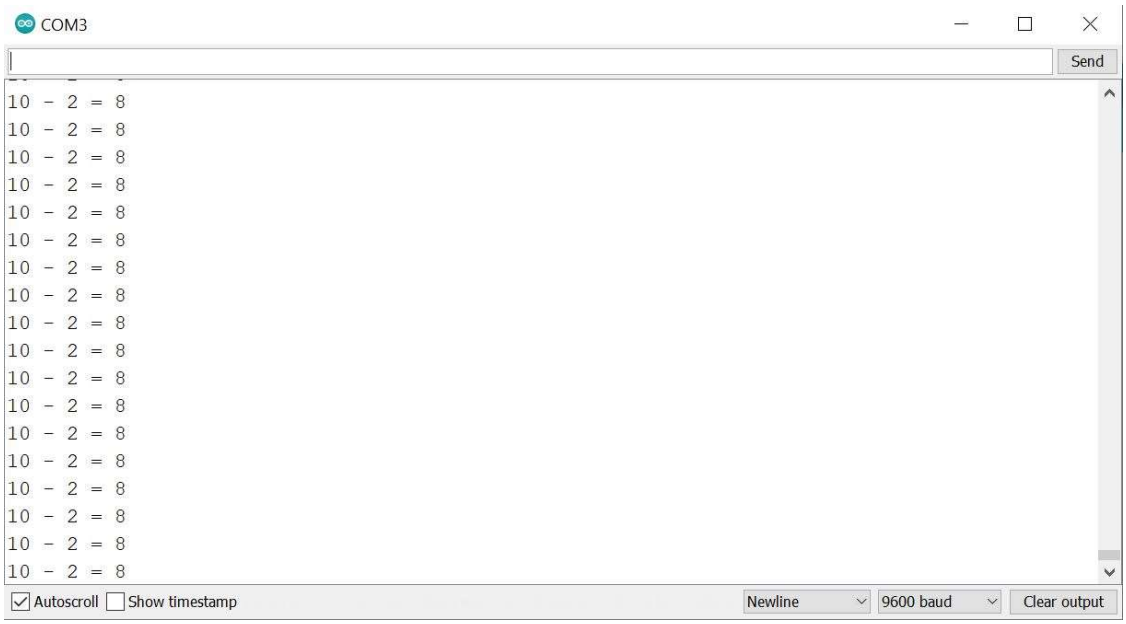


Praktikum 4.3

- **Pengurangan (-)**
  Operator jenis ini digunakan untuk mendapatkan hasil pengurangan dari dua bilangan atau lebih, contohnya : 10 - 2 = 8.

```
int x = 10;
int y = 2;
int z = x - y;
void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
}

void loop() {
  // put your main code here, to run repeatedly:
  Serial.println(String(x)+" - "+String(y)+" =
"+String(z));
}
```

Hasil program pada serial monitor

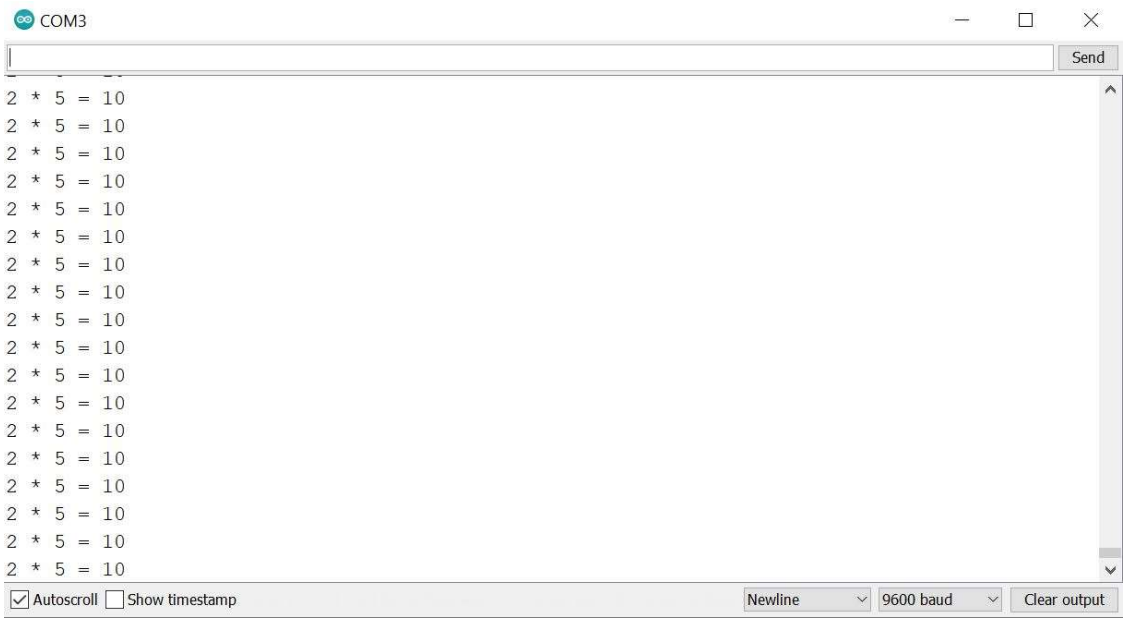


Praktikum 4.4

- **Perkalian (*)**
  Operator jenis ini digunakan untuk mendapatkan hasil perkalian dari dua bilangan atau lebih, contohnya : 2 * 5 = 10.

```
int x = 2;
int y = 5;
int z = x * y;
void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
}

void loop() {
  // put your main code here, to run repeatedly:
  Serial.println(String(x)+" * "+String(y)+" =
"+String(z));
}
```

Hasil program serial monitor:

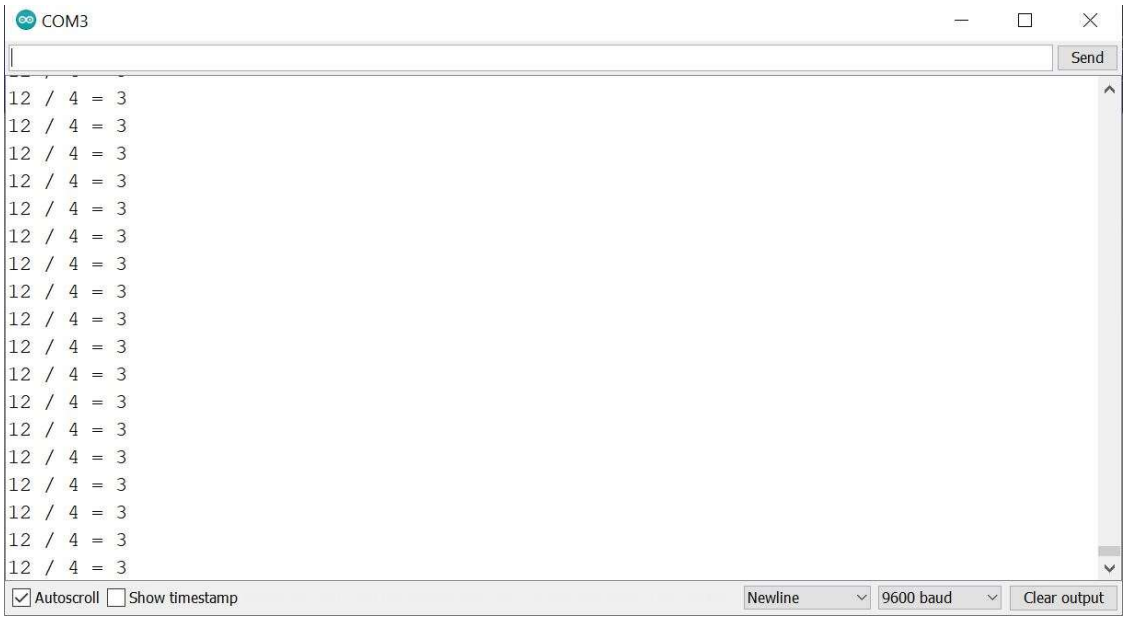


Praktikum 4.5

- **Pembagian (/)**
  Operator jenis ini digunakan untuk mendapatkan hasil pembagian dari dua bilangan atau lebih, contohnya : 12 / 4 = 3.

```
int x = 12;
int y = 4;
int z = x / y;
void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
}

void loop() {
  // put your main code here, to run repeatedly:
  Serial.println(String(x)+" / "+String(y)+" =
"+String(z));
}
```

Hasil program pada serial monitor



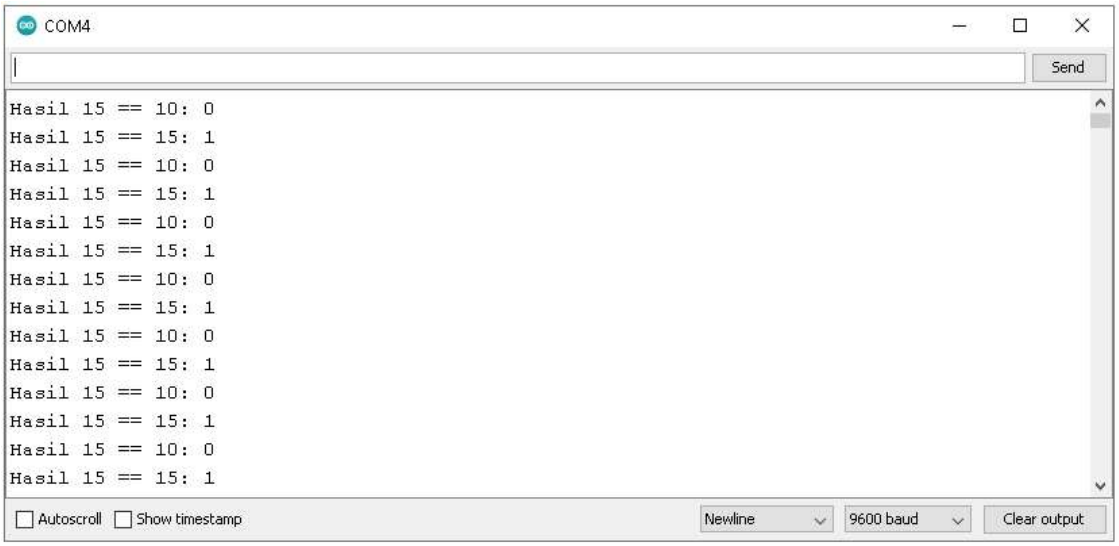
5. Langkah Praktikum 2 – Operator Perbandingan

Praktikum 5.1

- **Sama Dengan (==)**
  Operator jenis ini digunakan untuk mendapatkan hasil sama dengan dari perbandingan dua bilangan, contohnya : 15 == 10 **is FALSE** atau 15 == 15 **is TRUE**.

```
int x = 15;
int y = 10;
bool hasil1 = 15 == 10;
bool hasil2 = 15 == 15;
void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
}

void loop() {
  // put your main code here, to run repeatedly:
  Serial.print("Hasil 15 == 10: ");
  Serial.println(hasil1);
  Serial.print("Hasil 15 == 15: ");
  Serial.println(hasil2);
}
```

Hasil program pada serial monitor :

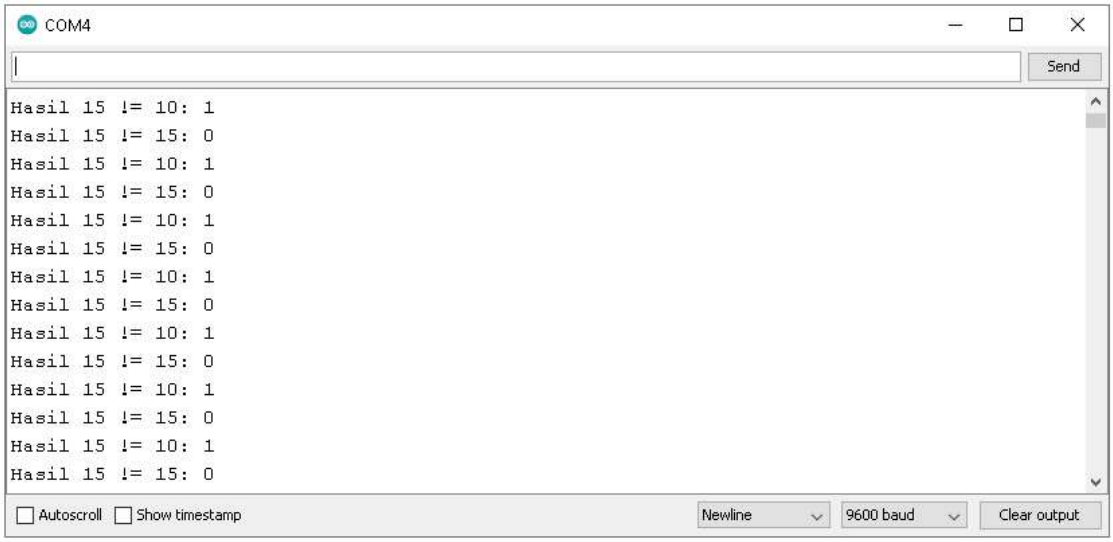


Praktikum 5.2

- **Tidak Sama Dengan (!=)**
  Operator jenis ini digunakan untuk mendapatkan hasil tidak sama dengan dari perbandingan dua bilangan, contohnya : 15 != 10 **is TRUE** atau 15 != 15 **is FALSE**.

```
int hasil1 = 15 != 10;
int hasil2 = 15 != 15;
void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
}

void loop() {
  // put your main code here, to run repeatedly:
  Serial.print("Hasil 15 != 10: ");
  Serial.println(hasil1);
  Serial.print("Hasil 15 != 15: ");
  Serial.println(hasil2);
}
```

Hasil program pada serial monitor :

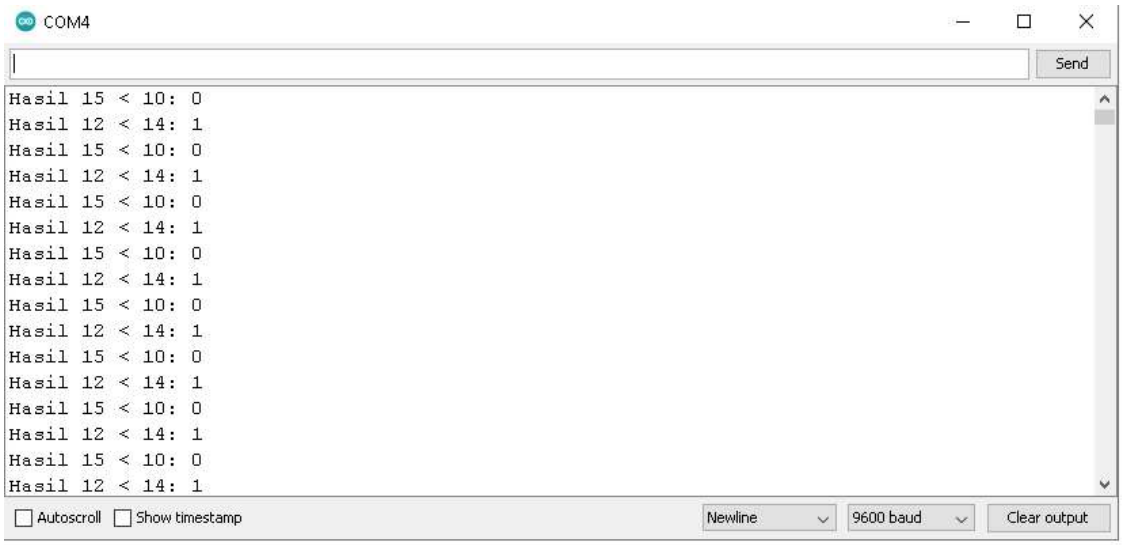


Praktikum 5.3

- **Lebih Kecil Dari (<)**
  Operator jenis ini digunakan untuk mendapatkan hasil lebih kecil dari perbandingan dua bilangan, contohnya : 15 < 10 **is FALSE** atau 12 < 14 **is TRUE**.

```
int hasil1 = 15 < 10;
int hasil2 = 12 < 14;
void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
}

void loop() {
  // put your main code here, to run repeatedly:
  Serial.print("Hasil 15 < 10: ");
  Serial.println(hasil1);
  Serial.print("Hasil 12 < 14: ");
  Serial.println(hasil2);
}
```

Hasil program pada Serial monitor :

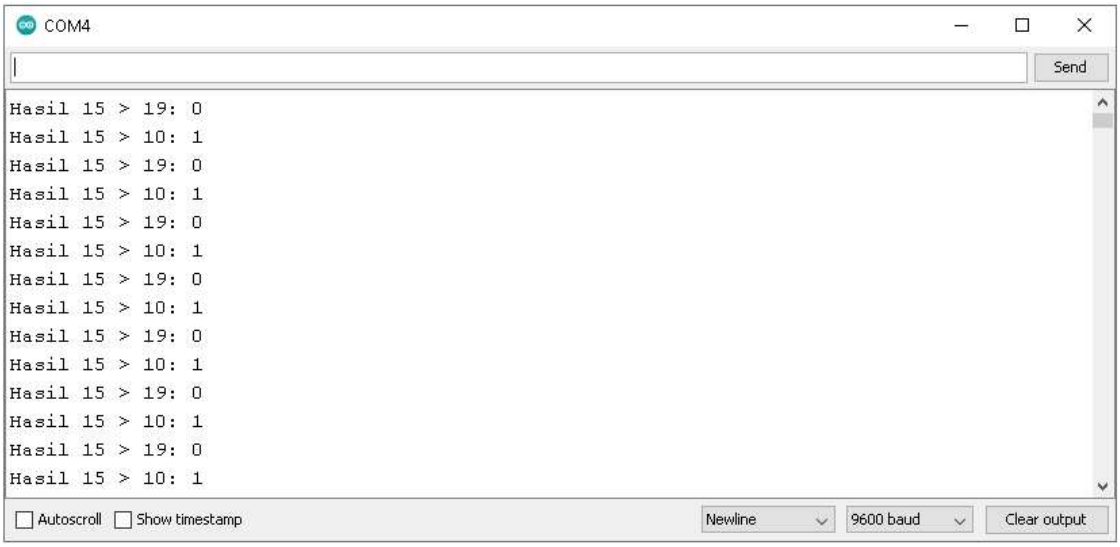


Praktikum 5.4

- **Lebih Besar Dari (>)**
  Operator jenis ini digunakan untuk mendapatkan hasil lebih besar dari perbandingan dua bilangan, contohnya : 15 > 19 **is FALSE** atau 15 > 10 **is TRUE**.

```
int hasil1 = 15 > 19;
int hasil2 = 15 > 10;
void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
}

void loop() {
  // put your main code here, to run repeatedly:
  Serial.print("Hasil 15 > 19: ");
  Serial.println(hasil1);
  Serial.print("Hasil 15 > 10: ");
  Serial.println(hasil2);
}
```

Hasil program pada Serial monitor :

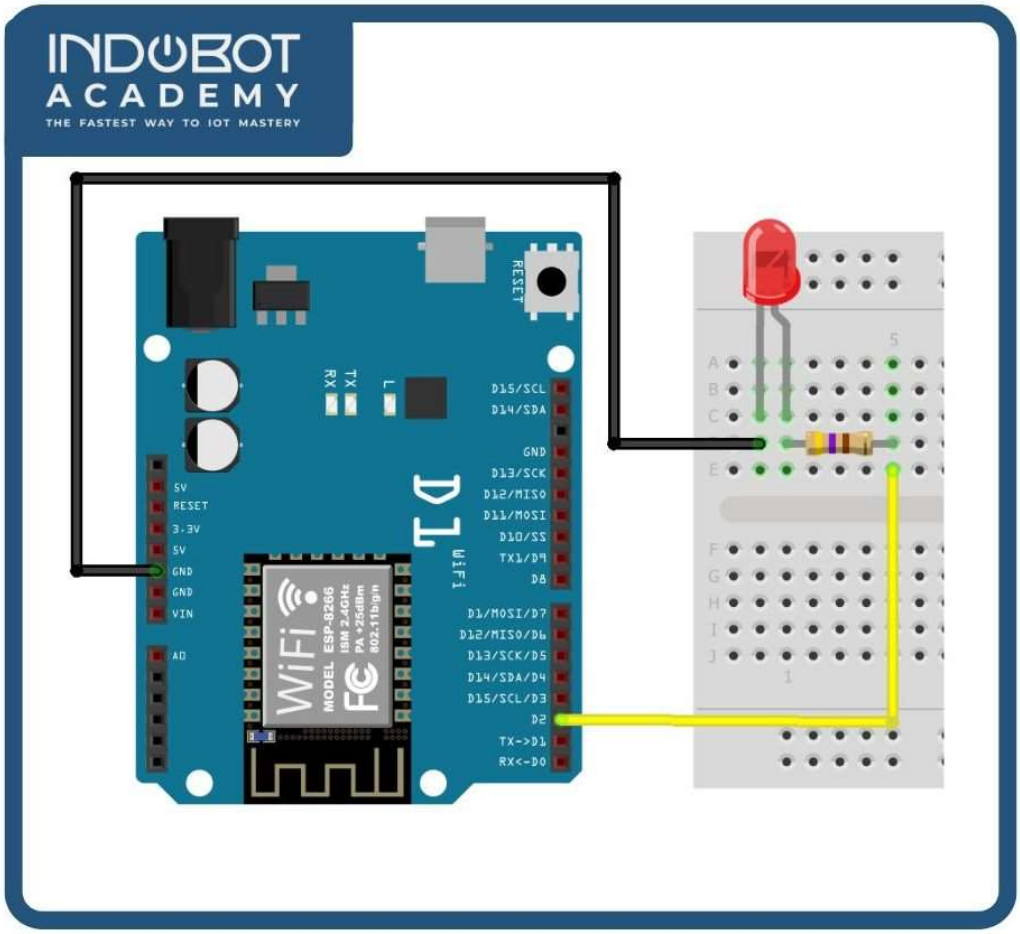


6. Langkah Praktikum 3 – DigitalWrite

6.1. Penjelasan Praktikum

Pada praktikum ini, anda akan belajar bagaimana cara menggunakan sintaks output digital `digitalWrite` untuk menyalakan LED pada rangkaian.

6.2. Skema Rangkaian



Keterangan :

- Hubungkan PIN Positif LED dan resistor 470 Pada PIN D2 Wemos D1.
- Hubungkan PIN Negatif LED pada PIN GND Wemos D1.

6.3. Coding

Program :

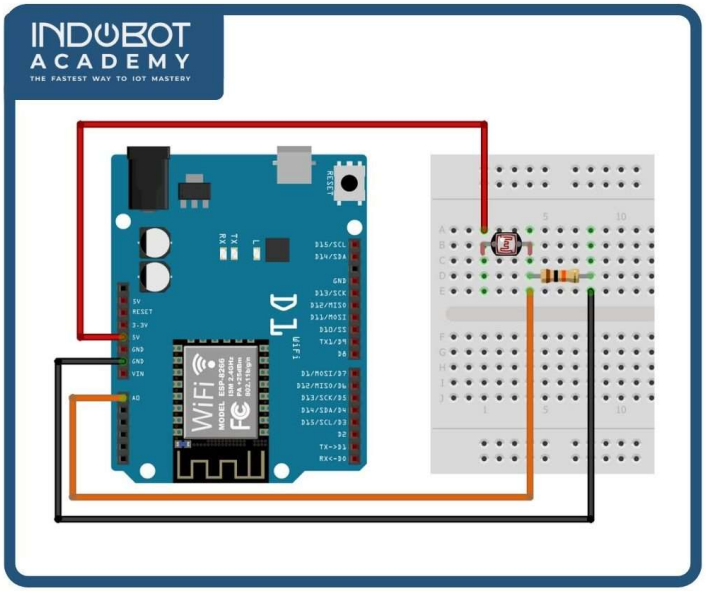
```
int ledPin = D2;

void setup()
{
  pinMode(ledPin, OUTPUT);
}

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

7. Langkah Praktikum 4 – Pin Analog

7.1. Penjelasan Praktikum

Pada praktikum ini, anda akan belajar bagaimana cara menerima input analog dari sensor analog seperti misalnya sensor LDR yang digunakan pada praktikum ini.

7.2. Skema Rangkaian



Keterangan PIN :

- Hubungkan satu Kaki LDR pada 5v Wemos D1.
- Hubungkan Kaki LDR yang terhubung ke resistor ke A0 Wemos D1.
- Hubungkan Kaki Resistor lainnya ke Pin GND Wemos D1.

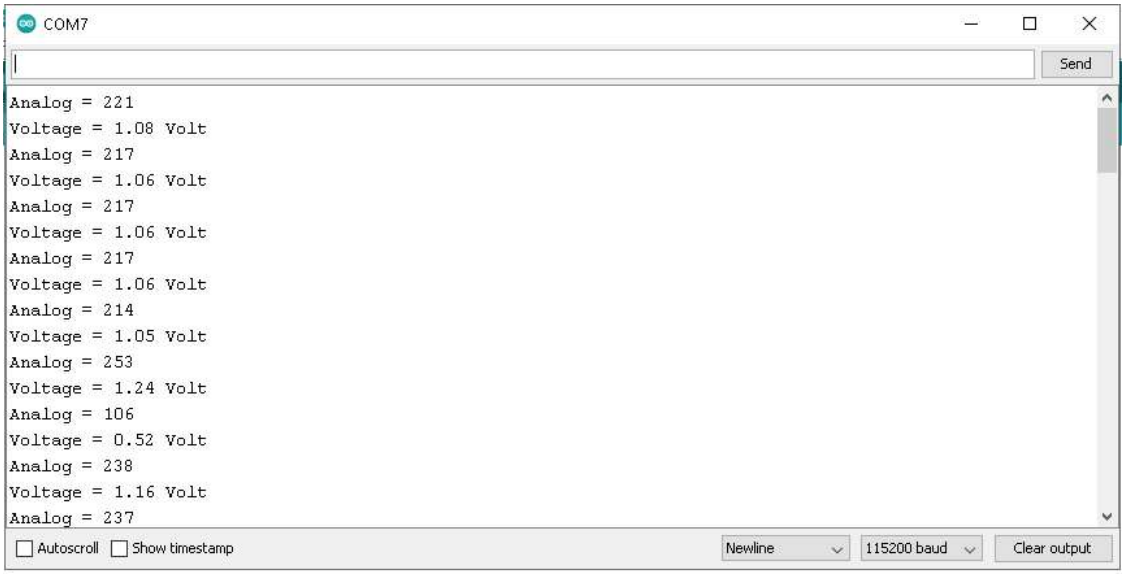
7.3. Coding

Program :

```
void setup() {
  Serial.begin(115200);
}

void loop() {
  int sensorValue = analogRead(A0);
  float voltage = sensorValue * (5.0 / 1023.0);
  Serial.print("Analog = ");
  Serial.println(sensorValue);
  Serial.print("Voltage = ");
  Serial.print(voltage);
  Serial.println(" Volt");
  delay(1000);
}
```

7.4. Hasil

Hasil pembacan sensor analog pada serial monitor :



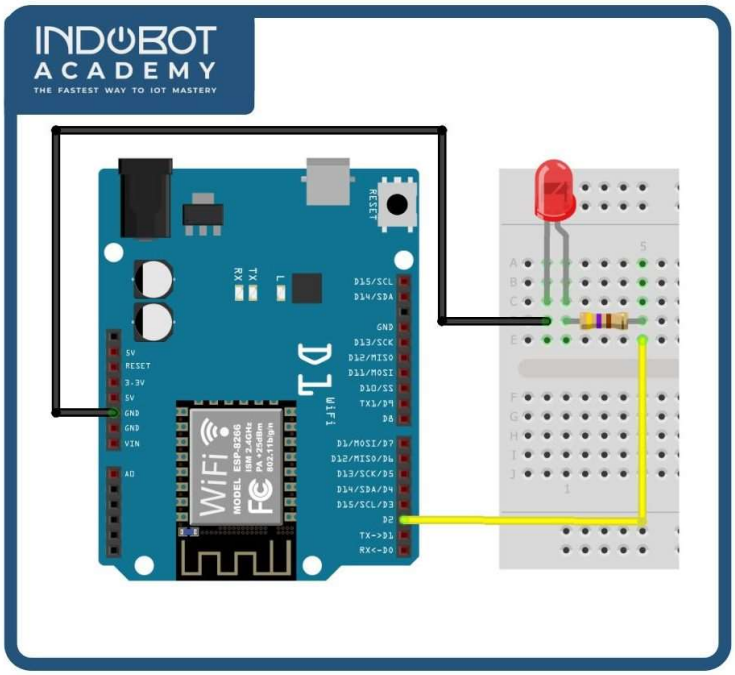
8. Langkah Praktikum 6 – PWM

8.1. Penjelasan Praktikum

Pada praktikum ini anda akan belajar bagaimana menggunakan PWM pada board Wemos D1 untuk mengendalikan lampu LED.

8.2. Skema Rangkaian

Cobalah praktikum di bawah ini dengan menggunakan hardware yang telah disediakan!



Keterangan PIN :

- Hubungkan Kaki Positive pada Pin D2 Wemos D1.
- Hubungkan kaki Negative pada Pin GND Wemos D1.

8.3. Coding

Program:

```
int led_pin = D2;

void setup() {
  //Declaring LED pin as output
  pinMode(led_pin, OUTPUT);
}

void loop() {
  for(int i=0; i<255; i++)
  {
    analogWrite(led_pin, i);
    delay(5);
  }

  for(int i=255; i>0; i--)
  {
    analogWrite(led_pin, i);
    delay(5);
  }
}
```

Upload program diatas, setelah itu lampu LED yang terpasang akan menyala perlahan lalu meredup perlahan.

9. Langkah Praktikum 7 – Kontrol LED Builtin dengan Serial Monitor

9.1. Penjelasan Praktikum

Pada praktikum ini anda akan belajar mengendalikan LED builtin atau LED yang ada pada board Wemos D1 menggunakan input dari serial monitor.

9.2. Coding

Kode Program :

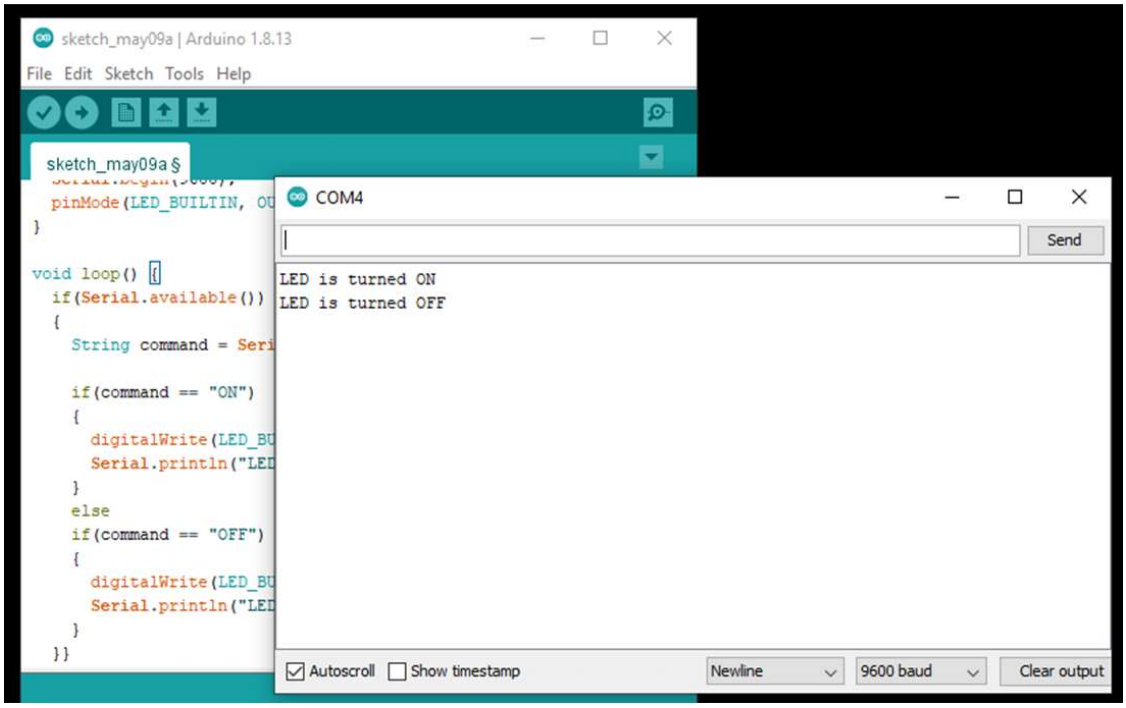
```
void setup() {
  Serial.begin(9600);
  pinMode(LED_BUILTIN, OUTPUT);
}
void loop() {
  if(Serial.available())
  {
    String command = Serial.readStringUntil('\n');

    if(command == "ON")
    {
      digitalWrite(LED_BUILTIN, HIGH);
      Serial.println("LED is turned ON");
    }
    else
    if(command == "OFF")
    {
      digitalWrite(LED_BUILTIN, LOW);
      Serial.println("LED is turned OFF");
    }
  }
}
```

Selanjutnya lakukan hal ini :

- Buka serial monitor.
- Setting Baud Rate ke 9600 dan newline option.
- Ketikan “ON” atau “OFF” pada text box lalu kirim.

9.3. Hasil

Jika berhasil maka tampilan dari Serial Monitor akan sebagai berikut.



Note : Jika menggunakan Wemos D1 R2 maka perintah akan terbalik. Jika ditulis ON maka LED akan mati, kemudian jika ditulis OFF maka LED akan menyala. Ini terjadi hanya ketika menggunakan LED Builtin Wemos.

10. Tugas dan Tantangan

- Buatlah sebuah coding dengan gabungan dari void setup dan void loop sehingga menjadi sebuah coding yang utuh.
- Atur kecerahan LED builtin menjadi 50% dan 75% menggunakan prinsip kerja PWM.