



# Fundamental Programming & Pin I/O

[Short Bootcamp]

# Komponen Dasar [IoT]



Microcontroller



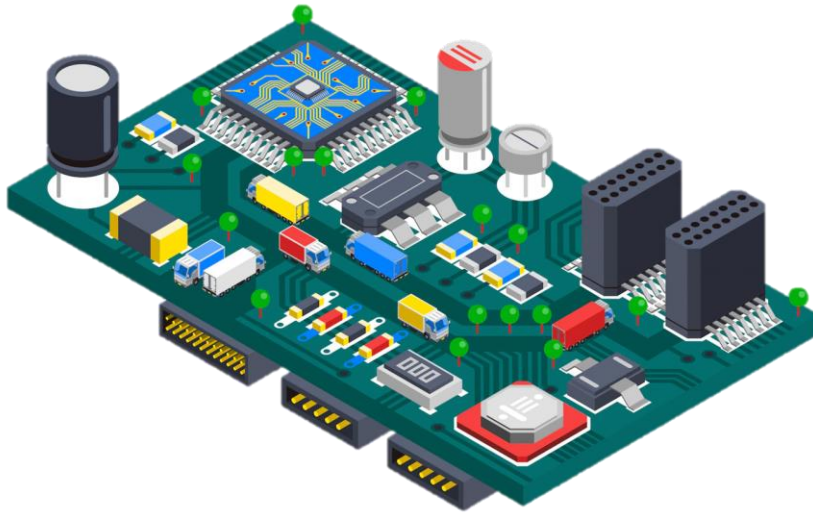
Sensor & Actuator



Communication  
Module



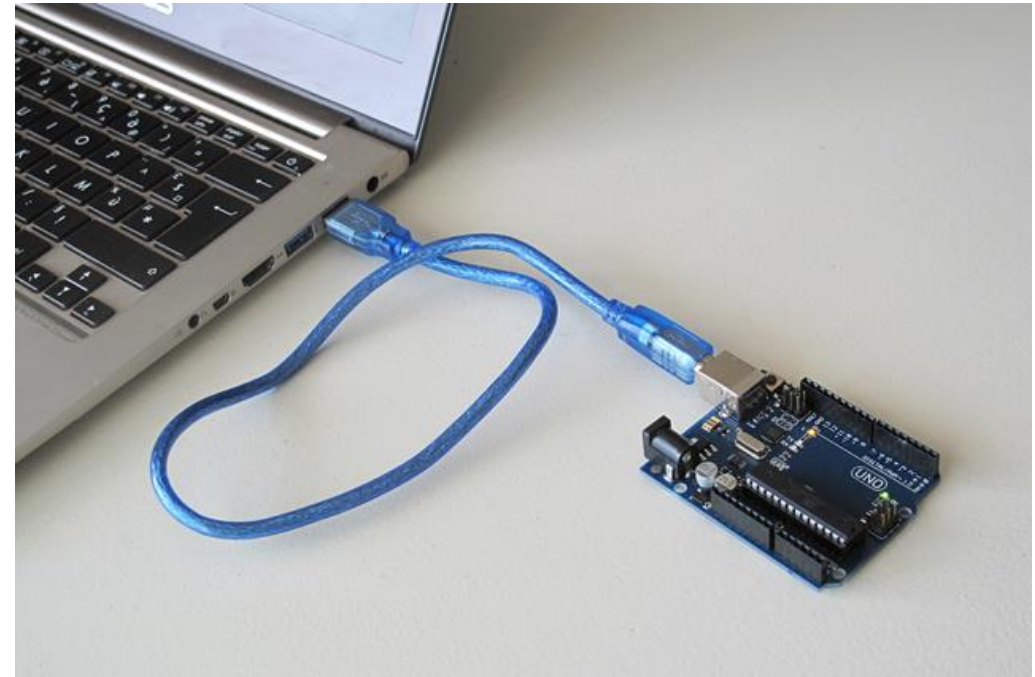
Power source



# Fundamental of Arduino Programming

# Arduino ↔ Computer

<b>Serial.begin(baudrate)</b>	digunakan untuk menentukan kecepatan dan penerimaan data melalui port serial. Kecepatan yang umumnya digunakan adalah 9600 bit per detik (9600 bps).
<b>Serial.print(data) atau Serial.println(data)</b>	digunakan untuk mengirimkan data ke port serial.



# Tipe Data

Data Type	Value Range
<b>boolean</b>	0 (false) atau 1 (true)
<b>int</b>	-32.768 hingga 32.767
<b>float</b>	3.4028235E+38 hingga -3.4028235E+38
<b>double</b>	1.7976931348623157E+308 hingga -1.7976931348623157E+308
<b>String</b>	"Statement"

# Variabel & Kontanta

## Variabel

**Variabel** adalah nama yang dibuat dan **disimpan di dalam memori** mikrokontroller yang **nilainya dapat diubah** sewaktu-waktu pada saat program dijalankan.

Syntax `TIPE_DATA NAMA_VARIABEL;`

```
int hitung, penjumlahan, perkalian;
```

## Konstanta

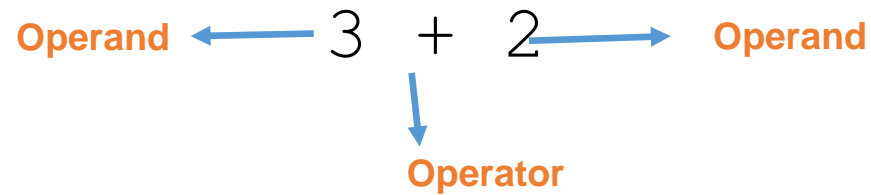
**Konstanta** adalah nilai suatu besaran yang tidak berubah besarnya atau memiliki **nilai yang tetap**.

Syntax `CONST TIPE_DATA NAMA_VARIABEL;`

```
const int hargabaju = 12;
```

# Operator

Bahasa C sebagai basis pemrograman Arduino merupakan bahasa yang kaya akan operator bawaan (built-in) dan menyediakan jenis-jenis operator.



**Jenis-jenis operator berikut ini:**

- Arithmetic Operators
- Comparison Operators
- Boolean Operators
- Bitwise Operators
- Compound Operators

# Arithmetic Operator

Operator name	Operator simple	Description	Example
assignment operator	=	Stores the value to the right of the equal sign in the variable to the left of the equal sign.	A = B
addition	+	Adds two operands	A + B will give 30
subtraction	-	Subtracts second operand from the first	A - B will give -10
multiplication	*	Multiply both operands	A * B will give 200
division	/	Divide numerator by denominator	B / A will give 2
modulo	%	Modulus Operator and remainder of after an integer division	B % A will give 0



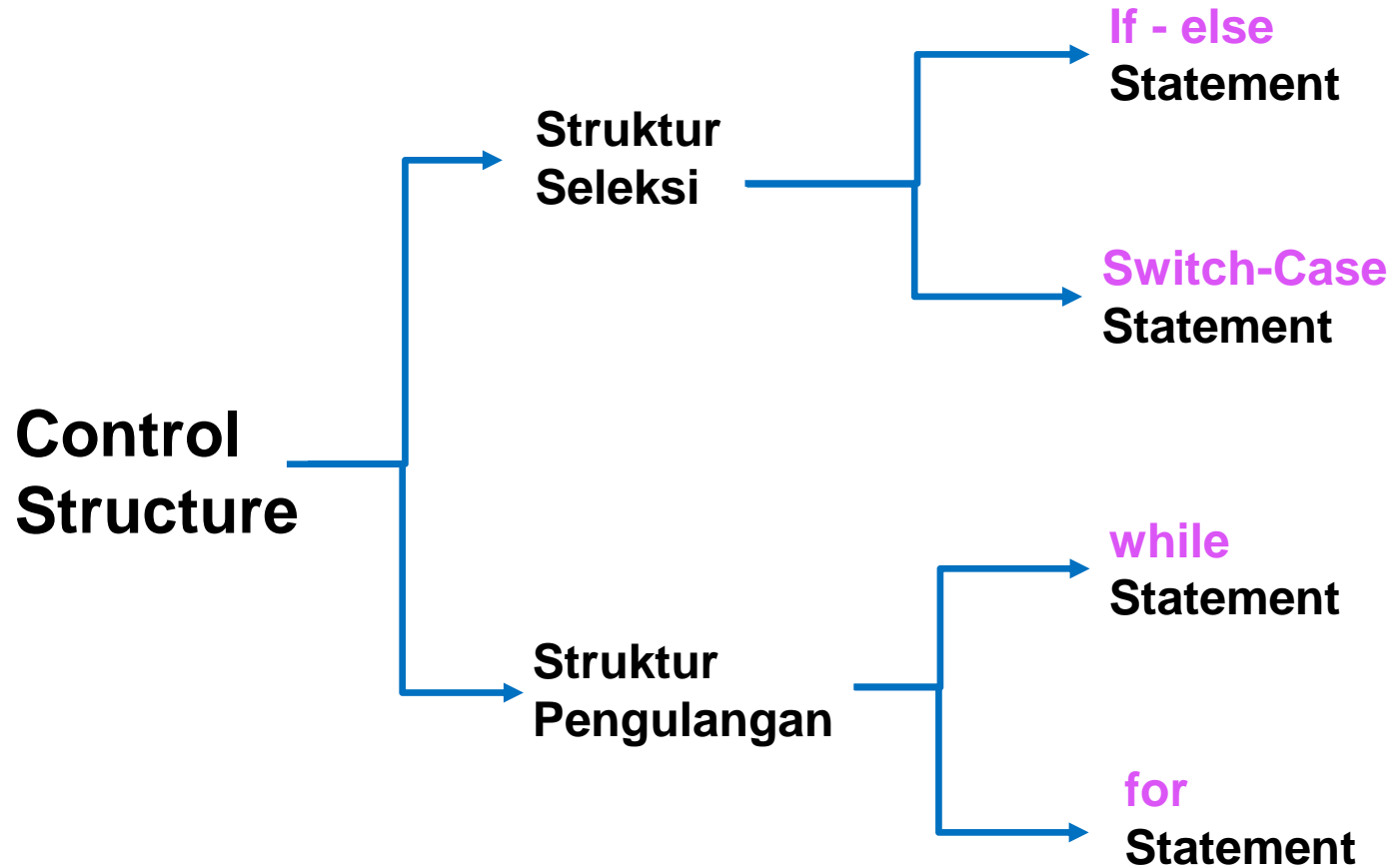
# Comparison Operator

Operator name	Operator simple	Description	Example
equal to	==	Checks if the value of two operands is equal or not, if yes then condition becomes true.	(A == B) is not true
not equal to	!=	Checks if the value of two operands is equal or not, if values are not equal then condition becomes true.	(A != B) is true
less than	<	Checks if the value of left operand is less than the value of right operand, if yes then condition becomes true.	(A < B) is true
greater than	>	Checks if the value of left operand is greater than the value of right operand, if yes then condition becomes true.	(A > B) is not true
less than or equal to	<=	Checks if the value of left operand is less than or equal to the value of right operand, if yes then condition becomes true.	(A <= B) is true
greater than or equal to	>=	Checks if the value of left operand is greater than or equal to the value of right operand, if yes then condition becomes true.	(A >= B) is not true

# Boolean Operator

Operator name	Operator simple	Description	Example
and	&&	Called Logical AND operator. If both the operands are non-zero then then condition becomes true.	(A && B) is true
or		Called Logical OR Operator. If any of the two operands is non-zero then then condition becomes true.	(A    B) is true
not	!	Called Logical NOT Operator. Use to reverses the logical state of its operand. If a condition is true then Logical NOT operator will make false.	!(A && B) is false

# Control Structure

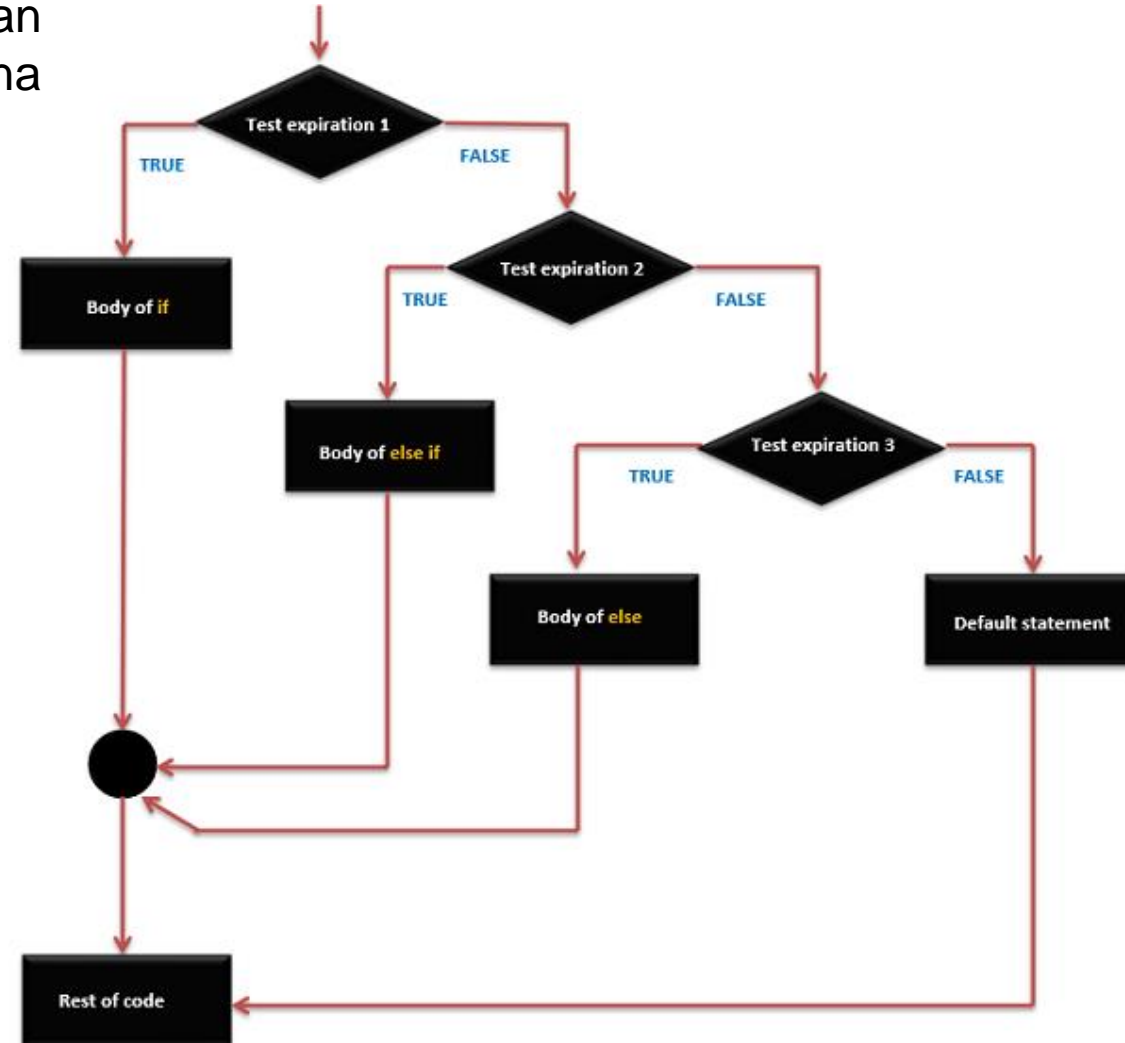


# If ... else if ... else Statement

Pernyataan if dapat diikuti oleh pernyataan if...else opsional lainnya, yang sangat berguna untuk menguji berbagai kondisi.

## Syntax

```
if (Expresion_1){  
    // Statements block of if;  
}  
else if(Expresion_2){  
    // Statements block of else-if;  
}  
else{  
    // Statements block of else;  
}
```

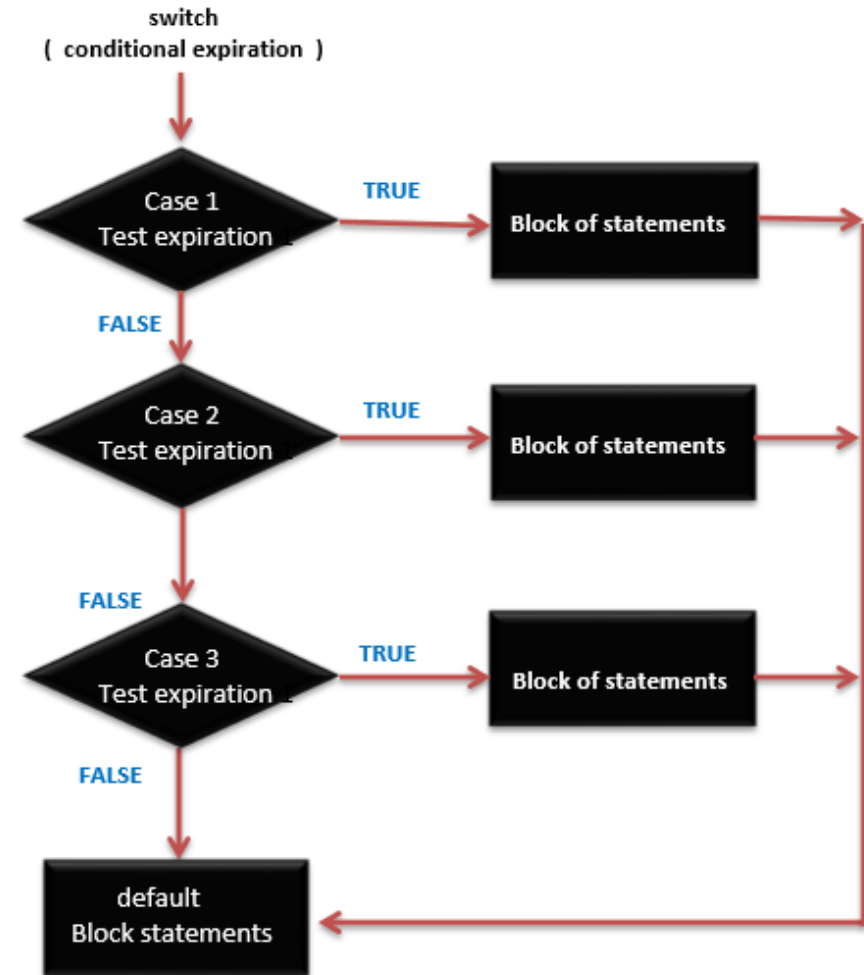


# Switch Case Statement

Mirip dengan pernyataan if, switch...case mengontrol aliran program dengan mengizinkan pemrogram untuk menentukan kode berbeda yang harus dijalankan dalam berbagai kondisi.

## Syntax

```
switch (variable) {  
  case label_1:{  
    //Statements block of case with label_1;  
    break;  
  }  
  case label_2:{  
    //Statements block of case with label_2;  
    break;  
  }  
  default:{  
    //Statements block of default;  
    break;  
  }  
}
```

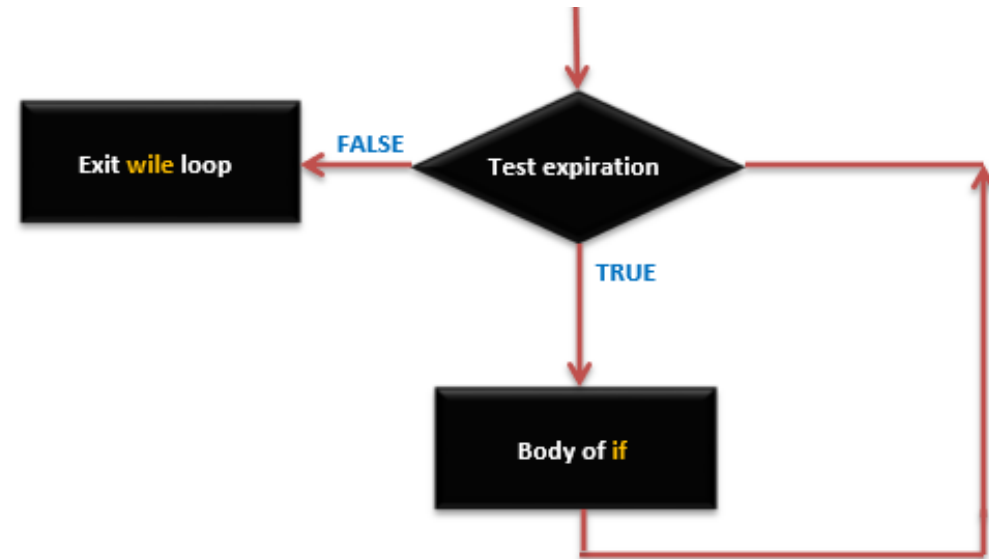


# While()

**while loop** akan berulang terus menerus, dan tak terbatas, sampai ekspresi di dalam tanda kurung, () menjadi salah. Sesuatu harus mengubah variabel yang diuji, atau loop while tidak akan pernah keluar.

## Syntax

```
while (Expresion) {  
    //Block of Statements;  
}
```

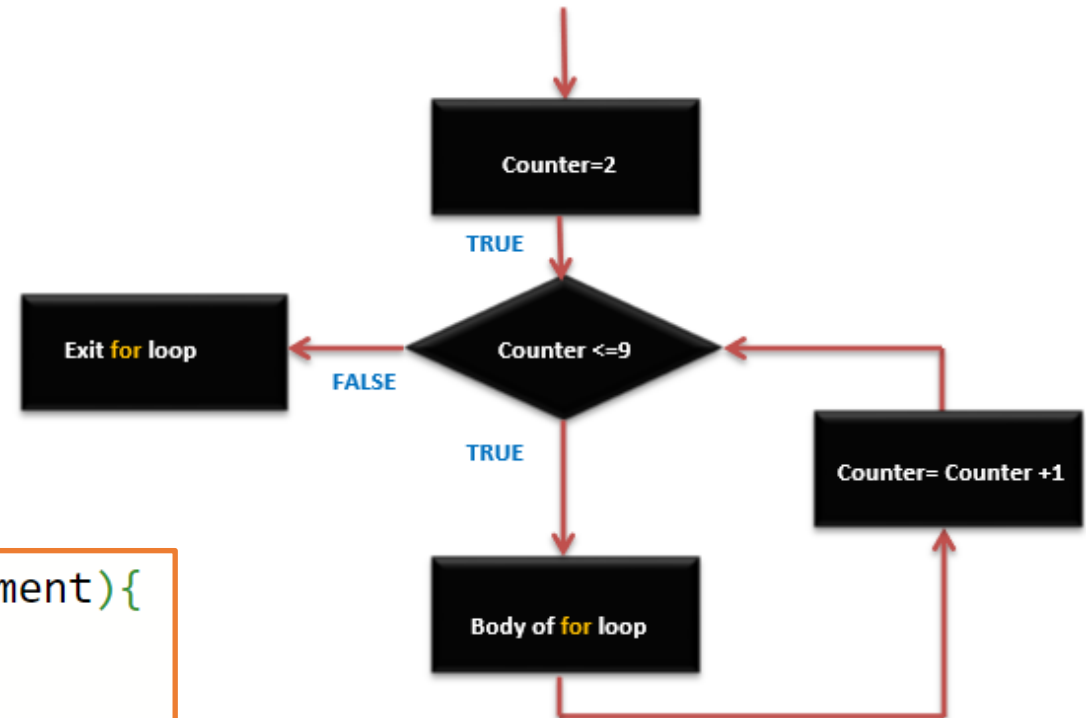


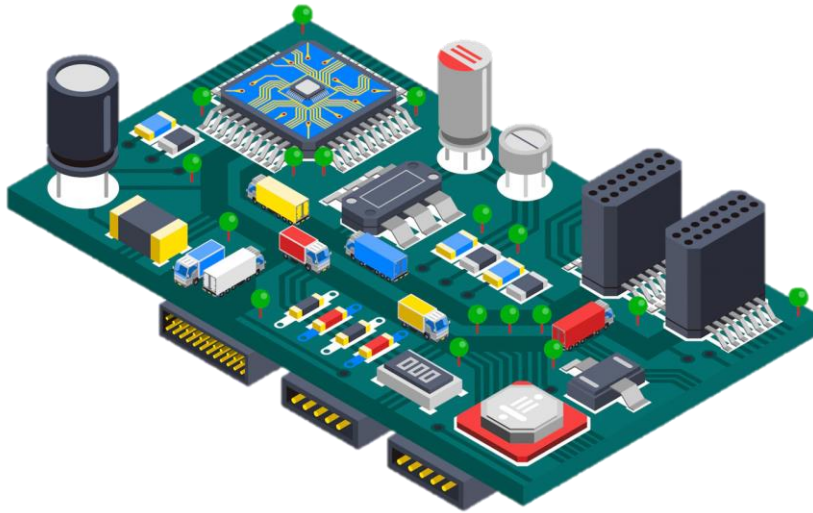
# For()

**for loop** mengeksekusi pernyataan beberapa kali **yang telah ditentukan**. Ekspresi kontrol untuk loop diinisialisasi, diuji, dan dimanipulasi seluruhnya dalam tanda kurung for loop.

## Syntax

```
for (initial; control; increment_or_decrement){  
    //Block of Statements;  
}
```





# Arduino Uno Peripherals



# IoT?



DIP



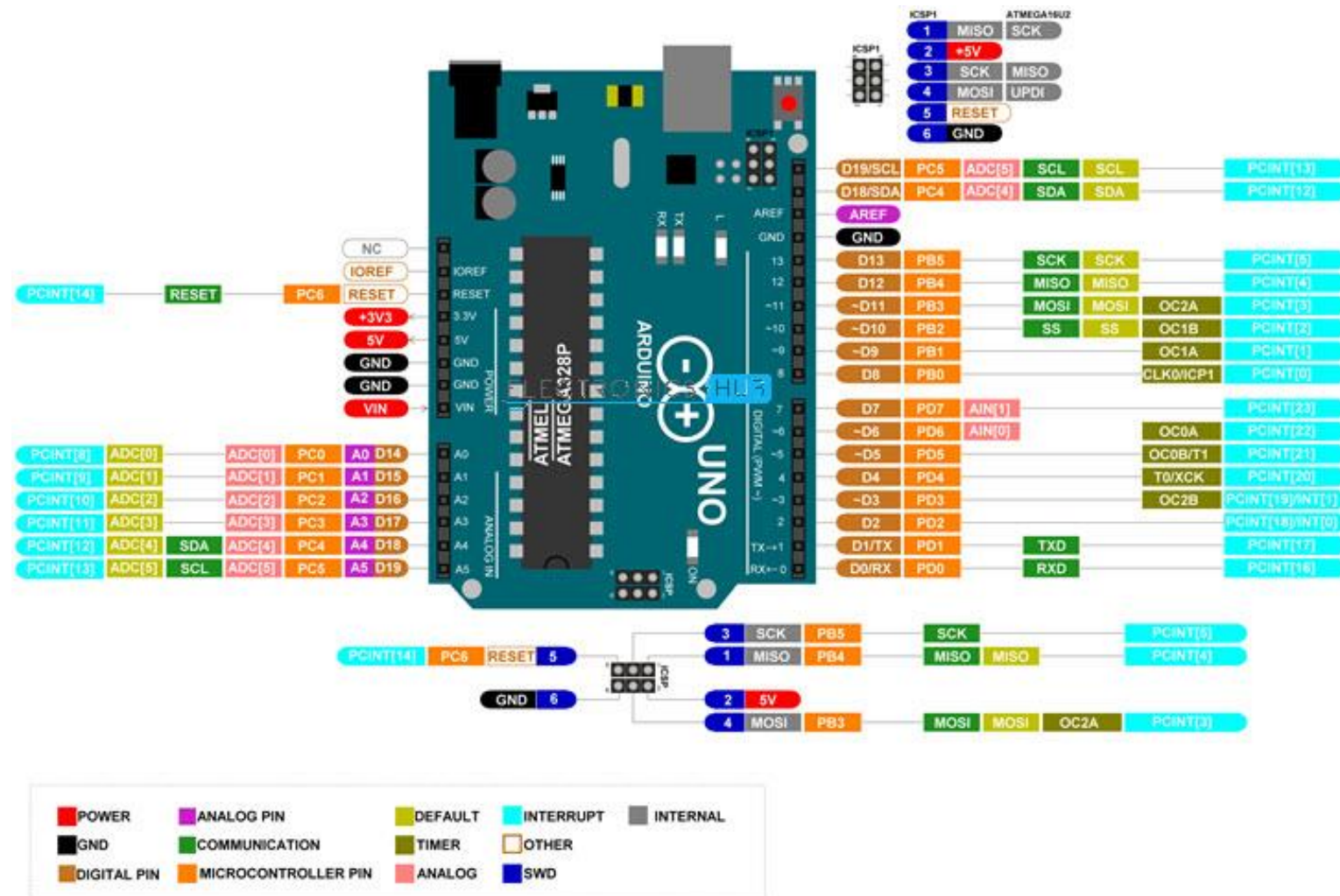
SMD

## Common Arduino Peripherals:

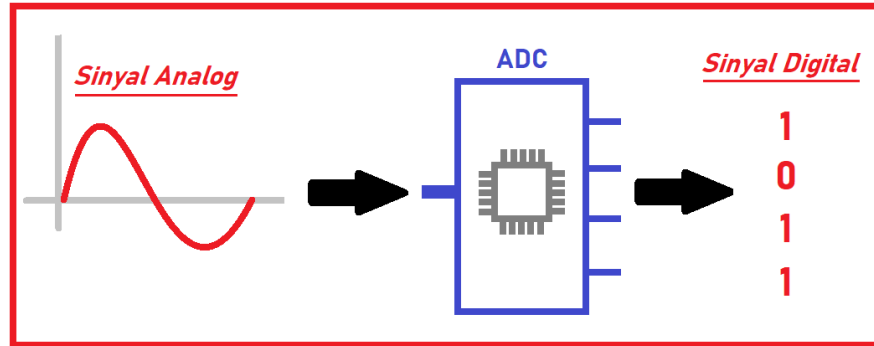
- ADC (Analog to Digital Converter)
- Digital I/O
- PWM (Pulse Width Modulation)
- UART (Universal Asynchronous Receiver/Transmitter)
- I2C (Inter-Integrated Circuit)
- SPI (Serial Peripheral Interface)

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

# IoT?



# ADC (Analog to Digital Converter)



ADC adalah sebuah metode untuk mengubah sinyal analog menjadi sinyal digital.

VRH = Tegangan referensi high ADC

VRL = Tegangan Referensi rendah ADC

b = jumlah bit dalam sistem digital

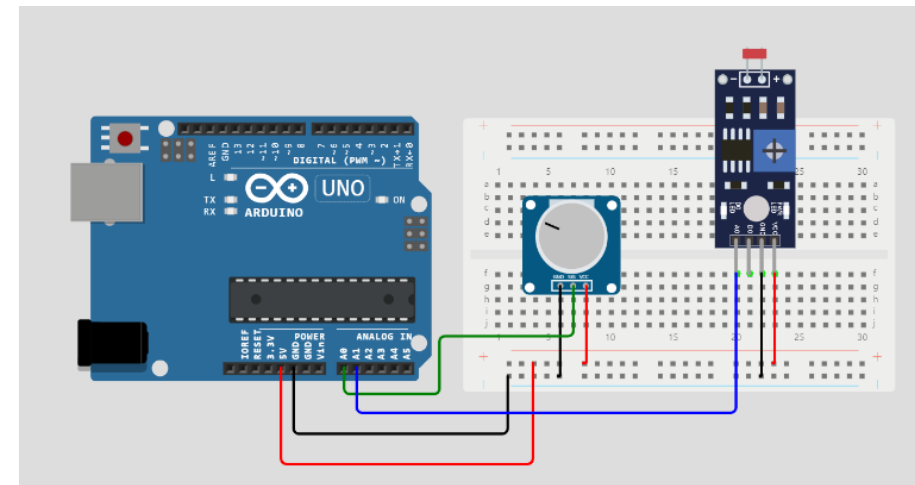
$$\text{Step Size} = (VRH - VRL) / 2^b$$

$$\text{Digital Output Code} = \frac{\text{Analog Input}}{\text{Reference Input}} (2^n - 1)$$

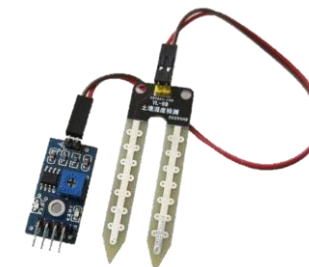
## Syntax

`analogRead(PIN)`

untuk membaca data dari pin analog kemudian disimpan dalam suatu variabel



Sensor Arus



Soil Mousture



Sensor Cahaya

# Digital I/O

**pinMode(PIN,MODE)**

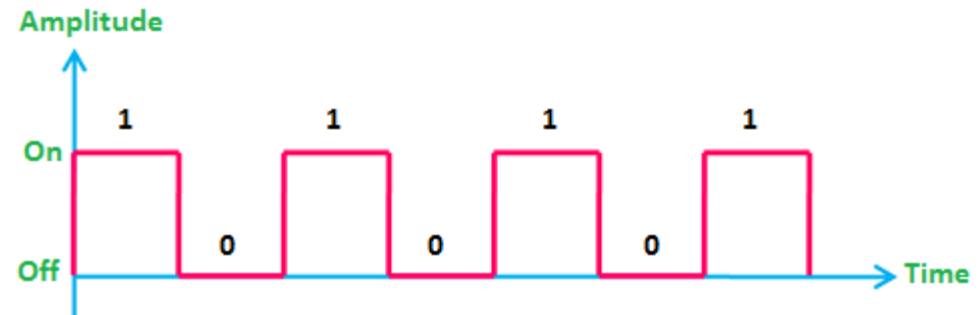
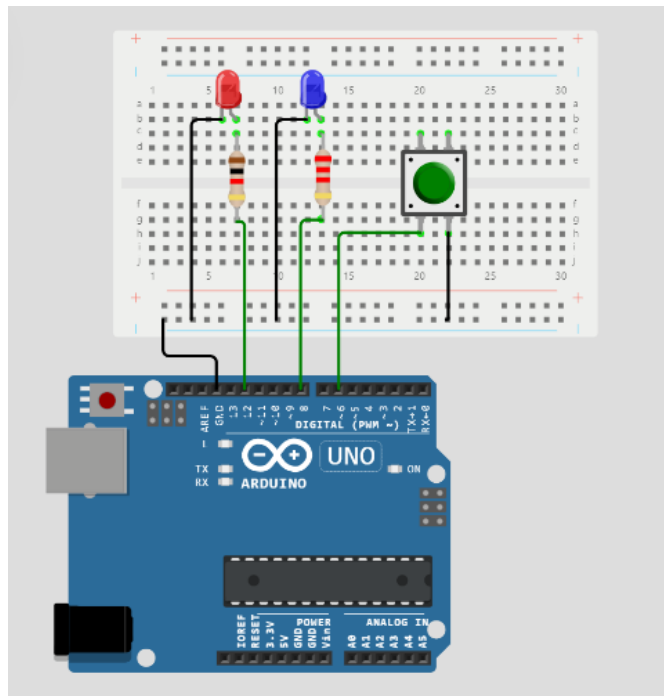
untuk mengkonfigurasi pin tertentu agar bekerja menjadi sebuah input atau output

**digitalRead(PIN)**

digunakan untuk membaca nilai dari suatu inputan yang berasal dari pin digital Arduino baik nilai HIGH maupun LOW

**digitalWrite(PIN,VALUE)**

berfungsi untuk memberi nilai 1 atau 0 kepada pin digital yang ada pada arduino



# PWM

`analogWrite(PIN, VALUE)`

untuk mengirimkan nilai analog dengan metode PWM (Pulse Width Modulation) pada pin khusus yang berlabel PWM,

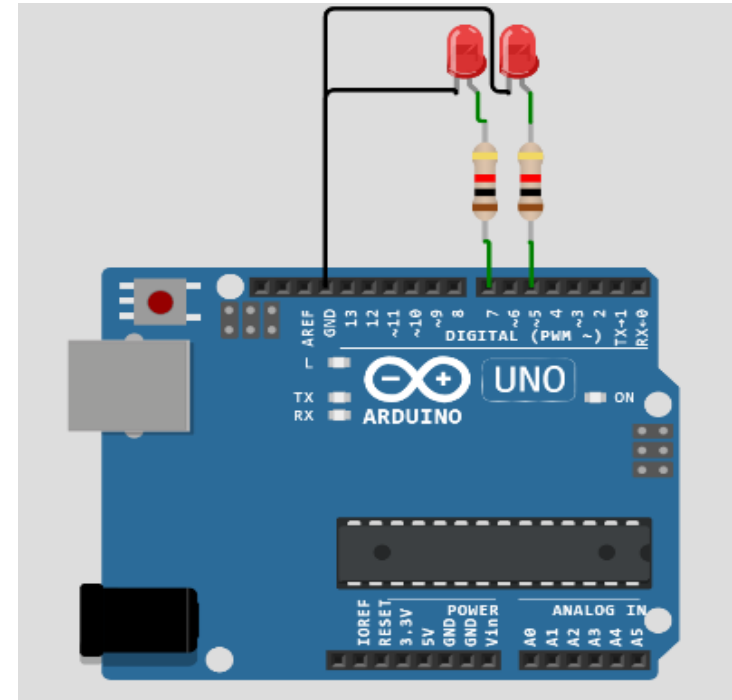
50% duty cycle



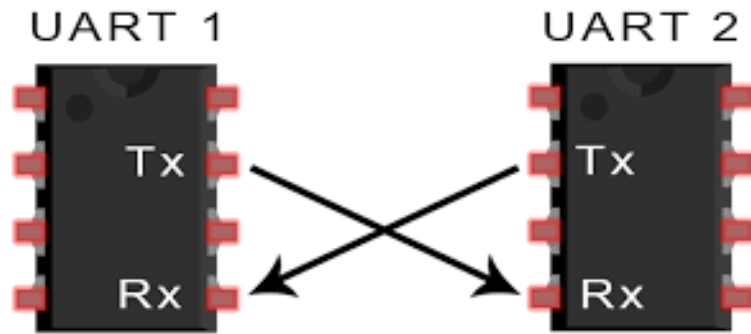
75% duty cycle



25% duty cycle



# UART



UART mengirimkan data secara *asinkron* , yang berarti tidak ada sinyal clock untuk menyinkronkan keluaran bit dari UART pengirim ke pengambilan sampel bit oleh UART penerima. Alih-alih sinyal clock, UART transmisi menambahkan bit start dan stop ke paket data yang ditransfer. Bit-bit ini menentukan awal dan akhir paket data sehingga UART penerima tahu kapan harus mulai membaca bit.



GPS Module



Nextion LCD



PZEM Sensor



## GPS

**#include <SoftwareSerial.h>**

Library untuk membuat pin I/O digital menjadi UART

**SoftwareSerial ss(4, 3);**

Membuat object ss

**ss.begin(baudrate)**

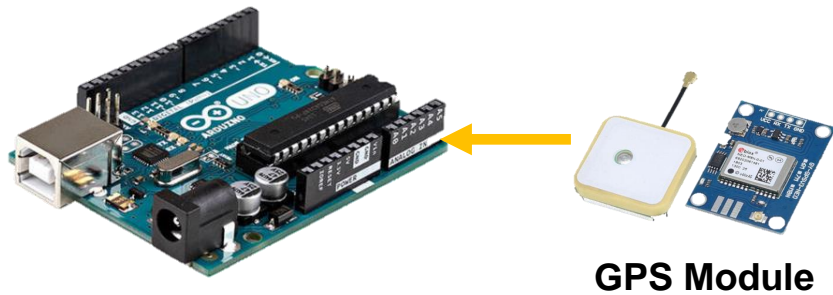
digunakan untuk menentukan kecepatan dan penerimaan data melalui port serial. Kecepatan yang umumnya digunakan adalah 9600 bit per detik (9600 bps).

**ss.available()**

untuk mengetahui jumlah byte data yang berada pada port serial.

**ss.readString()**

untuk membaca data yang terdapat di port serial dengan format String.



**GPS Module**

Arduino	GPS Module
5V	VCC
TX pin defined in the software serial	RX
RX pin defined in the software serial	TX
GND	GND

THANK  
YOU