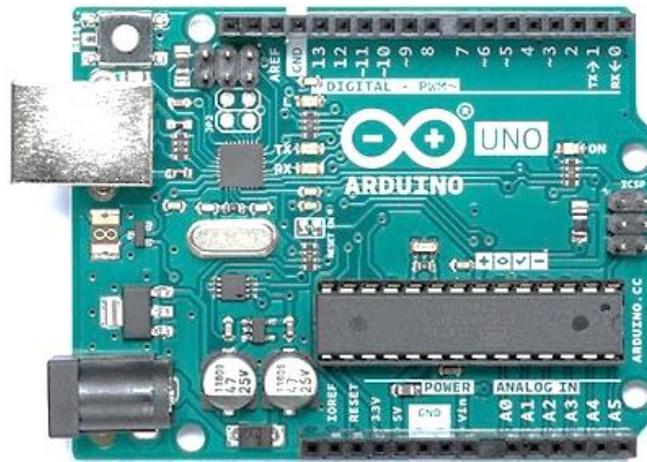


MODUL PRAKTIKUM 1 PENGENALAN ARDUINO UNO

A. Arduino UNO

Arduino itu sebuah board mikrokontroler yang merupakan “*sebuah sistem komputer yang fungsional dalam sebuah chip*”. Dalam bangunan Arduino telah tersedia prosesor, memory, input output, dan bisa dibilang bahwa mikrokontroler ini adalah komputer dalam versi mini yang disertai perangkat lunak pendukung untuk melakukan pemrograman yang disebut dengan Arduino IDE (*Integrated Development Environment*). Arduino menganut sistem *open hardware*, menggunakan Atmel AVR processor dan memiliki I/O onboard. Contoh sederhana apa yang dapat kita lakukan terhadap Arduino, kita dapat mengatur kedipan LED setiap 1 detik sekali atau melakukan pengendalian terhadap putaran rotasi motor servo dan lainnya.

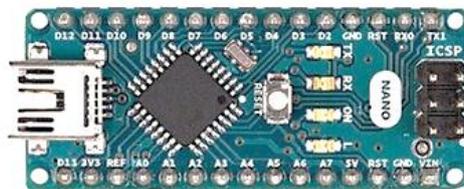
Arduino muncul dengan banyak sekali variannya, mulai dari Arduino Uno, Arduino MEGA 2560, Arduino Nano, dan sebagainya. Semua varian Arduino tersebut dibangun dengan dasar yang sama yaitu menggunakan mikrokontroler Atmel AVR yang memiliki perbedaan di banyaknya pin yang bisa digunakan.



(a)

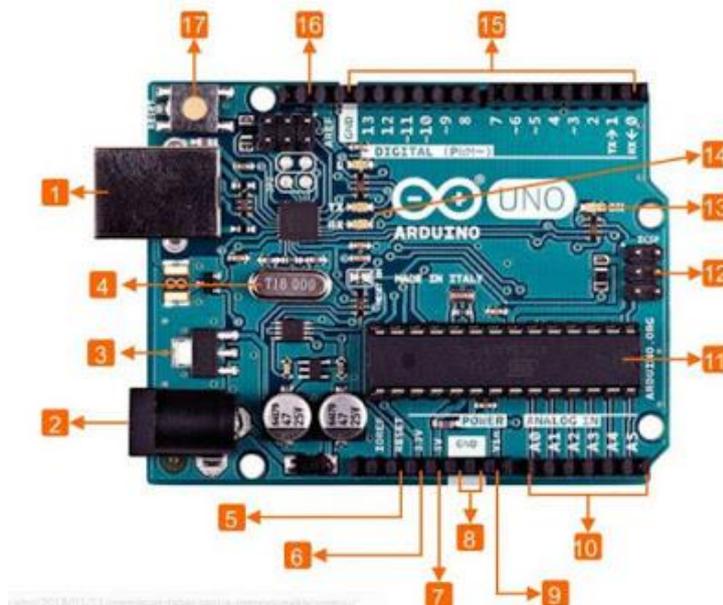


(b)



(c)

Gambar 1. (a) Arduino UNO; (b) Arduino MEGA2560; (3) Arduino Nano



Gambar 2. Bagian-bagian Arduino UNO

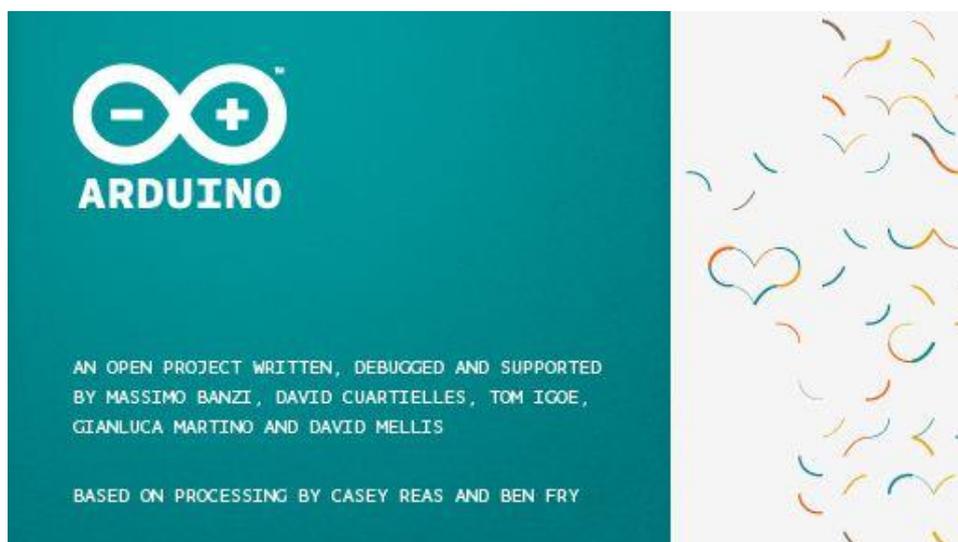
Sebelum menggunakan modul Arduino UNO lebih lanjut, tabel di bawah ini menunjukkan bagian-bagian dari Arduino.

Tabel 1. Fungsi dan kegunaan bagian-bagian Arduino UNO

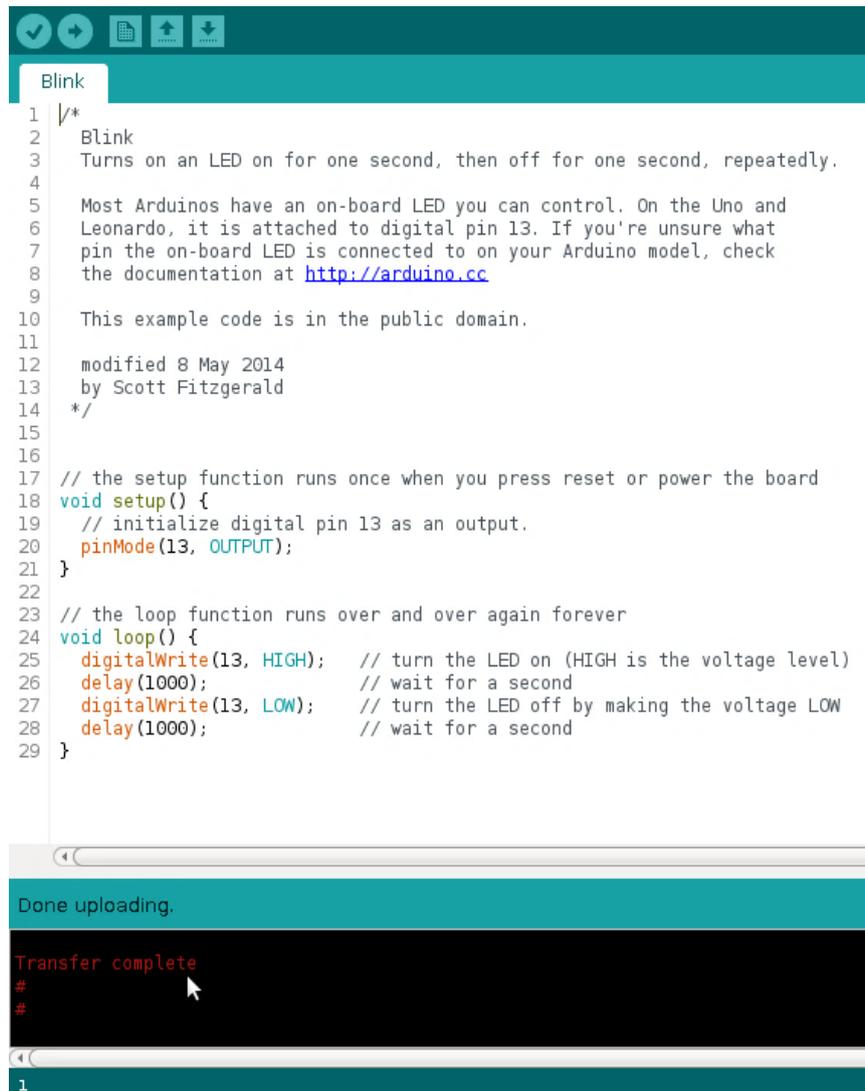
| No. | Fungsi | Kegunaan |
|----------------|-------------------------|---|
| 1 | Power USB | Memberikan catu daya ke Papan Arduino dan untuk upload sketch/kode program menggunakan kabel USB dari komputer. |
| 2 | Power (Barrel Jack) | Memberikan catu daya ke Papan Arduino langsung dari sumber daya AC |
| 3 | Voltage Regulator | Mengendalikan tegangan yang diberikan ke papan Arduino dan menstabilkan tegangan DC yang digunakan oleh prosesor dan elemen-elemen lain. |
| 4 | Crystal Oscillator | Membantu Arduino dalam hal yang berhubungan dengan waktu. |
| 5 dan 17 | Arduino Reset | Melakukan reset proses Arduino dari awal. Terdapat dua cara untuk mereset Arduino Uno. Pertama, dengan menggunakan reset button (17) pada papan arduino. Kedua, dengan menambahkan reset eksternal ke pin Arduino yang berlabel RESET (5). |
| 6, 7, 8, dan 9 | Pin 3.3V, 5V, GND, Vin | <ul style="list-style-type: none"> • 3.3V (6) – Supply 3.3 output volt • 5V (7) – Supply 5 output volt • Arduino bekerja dengan baik pada tegangan 3.3 volt dan 5 volt. • GND (8) – Ground • Vin (9) – Digunakan untuk memberi daya ke papan Arduino dari sumber daya eksternal, seperti sumber daya AC. |
| 10 | Input/output Analog | Pin Analog |
| 11 | Main microcontroller/IC | Sebagai pusat pengolahan Arduino. |
| 12 | Pin ICSP | ICSP (12) adalah AVR, suatu programming header kecil untuk Arduino yang berisi MOSI, MISO, SCK, RESET, VCC, dan GND. Hal ini sering dirujuk sebagai SPI (Serial Peripheral Interface), yang dapat dipertimbangkan sebagai “expansion” dari output. Sebenarnya, kita memasang perangkat output ke master bus SPI. |
| 13 | Power LED Indicator | LED ini harus menyala jika menghubungkan Arduino ke sumber daya. Jika LED tidak menyala, maka terdapat sesuatu yang salah dengan sambungannya. |
| 14 | TX dan RX | Pin yang digunakan untuk melakuakn komunikasi serial, dimana TX (transmit) dan RX (receive). TX dan RX muncul di dua tempat pada papan Arduino UNO. |

| | | |
|----|---------------------|--|
| 15 | Input/Ouput Digital | Pin Digital |
| 16 | AREF | AREF merupakan singkatan dari <i>Analog Reference</i> . AREF terkadang kadang digunakan untuk mengatur tegangan referensi eksternal (antar 0 dan 5 Volts) sebagai batas atas untuk pin input analog input. |

B. Arduino IDE (Integrated Development Environment)



Arduino Integrated Development Environment - atau Arduino Software (IDE) - berisi editor teks untuk menulis kode, area pesan, konsol teks, toolbar dengan tombol untuk fungsi-fungsi umum dan serangkaian menu. Menghubungkan ke perangkat keras Arduino dan Genuino untuk mengunggah program dan berkomunikasi dengan papan sirkuit Arduino. Program yang ditulis menggunakan Arduino Software (IDE) disebut *sketches*. *Sketches* ini ditulis dalam editor teks dan disimpan dengan ekstensi file *.ino*. Editor ini memiliki fitur untuk memotong (*cut*), menempelkan (*paste*), dan pencarian atau mengganti teks. Pada bagian pesan berisikan umpan balik saat menyimpan dan mengeksport dan juga menampilkan kesalahan. Konsol menampilkan output teks dengan Arduino Software (IDE), termasuk pesan kesalahan yang lengkap dan informasi lainnya. Sudut kanan bawah jendela menampilkan papan dikonfigurasi dan port serial. Tombol *toolbar* memungkinkan untuk memverifikasi dan mengunggah program, membuat, membuka, dan menyimpan *sketches*, serta membuka monitor serial.



```

1  /*
2  Blink
3  Turns on an LED on for one second, then off for one second, repeatedly.
4
5  Most Arduinos have an on-board LED you can control. On the Uno and
6  Leonardo, it is attached to digital pin 13. If you're unsure what
7  pin the on-board LED is connected to on your Arduino model, check
8  the documentation at http://arduino.cc
9
10 This example code is in the public domain.
11
12 modified 8 May 2014
13 by Scott Fitzgerald
14 */
15
16
17 // the setup function runs once when you press reset or power the board
18 void setup() {
19   // initialize digital pin 13 as an output.
20   pinMode(13, OUTPUT);
21 }
22
23 // the loop function runs over and over again forever
24 void loop() {
25   digitalWrite(13, HIGH); // turn the LED on (HIGH is the voltage level)
26   delay(1000);           // wait for a second
27   digitalWrite(13, LOW); // turn the LED off by making the voltage LOW
28   delay(1000);           // wait for a second
29 }

```

Done uploading.

Transfer complete

#

Gambar 3. Contoh *sketches* program yang ditulis dengan Arduino IDE (Sumber: https://www.arduino.cc/en/uploads/Guide/Edison_img15.png, 2016)

Gambar 8 di atas merupakan contoh program *sketches* yang ditulis menggunakan Arduino IDE yang pada dasarnya menggunakan bahasa C.

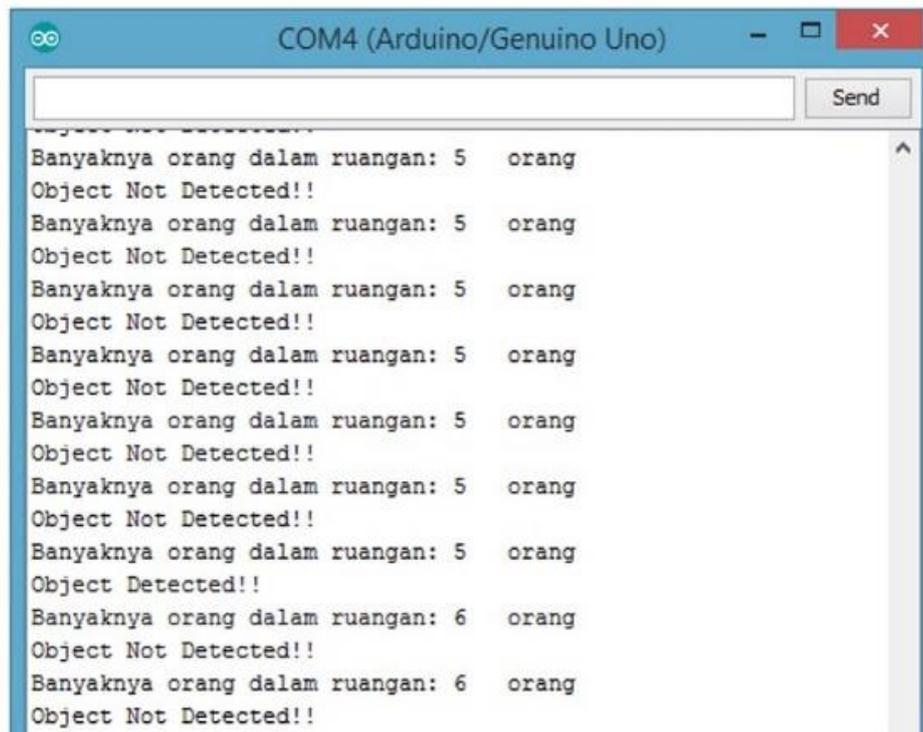
Pembuatan program yang dilakukan pada Arduino IDE atau *sketch* pada umumnya akan dituliskan pada dua bagian utama Arduino IDE, yaitu pada bagian:

1. **void setup()**

Bagian utama pertama adalah bagian yang biasanya digunakan oleh pemrogram untuk melakukan penulisan perintah proses inisialisasi program, seperti `pinMode()`, `Serial.begin()`, `LCD.begin()`, `LCD.clear` dan perintah lainnya.

2. `void loop()`

Bagian utama kedua adalah bagian yang berisikan program utama yang nantinya akan dieksekusi secara berulang oleh pemroses yang terdapat dalam papan sirkuit Arduino UNO, yaitu mikrokontroler Atmega328P. Bagian yang tak kalah penting dari proses pemrograman *sketches* adalah bagaimana menampilkan hasil operasi/proses pengolahan yang dilakukan oleh mikrokontroler Atmega328P yang terdapat dalam modul Arduino UNO. Terkait hal ini, maka Arduino IDE menyertakan sebuah fitur yang dinamakan dengan *serial monitor* yang berfungsi untuk menunjukkan hasil pengolahan mikrokontroler Atmega328P yang pada umumnya digunakan sebagai alat bantu pengujian validitas hasil pengolahan. Gambar di bawah ini menunjukkan hasil pengolahan mikrokontroler Atmega328P pada *serial monitor*.



Gambar 4. Tampilan hasil pengolahan pada serial monitor

Arduino adalah sebuah mikrokontroler yang didesain khusus sebagai pembelajaran mikrokontroler dengan menggunakan processor ATMEL dan bahasa pemrograman sendiri. Arduino bersifat open-source sehingga tidak ada biaya dalam perangkat lunaknya yang mana membuat para pengguna bisa leluasa melakukan eksperimen terhadap perangkat Arduino ini

tanpa terkendala pernakat lunak percobaan (*trial*) atau berbayar. Selain itu tersedia bagi para pemrogram berpengalaman untuk pengembangan lebih lanjut. Bahasanya bisa dikembangkan lebih lanjut melalui pustaka-pustaka C++ yang berbasis pada Bahasa C untuk AVR. Perangkat keras Arduino berbasis mikrokontroler ATMEGA8, ATMEGA168, ATMEGA328 dan ATMEGA1280 (yang terbaru ATMEGA2560). Dengan demikian siapapun bisa membuatnya dan kemudian bisa menjual perangkat keras Arduino ini, Bootloader Arduino tersedia langsung dari perangkat lunak Arduino IDE-nya. Bisa juga menggunakan breadboard untuk membuat perangkat Arduino beserta periferan-periferan lain yang dibutuhkan.

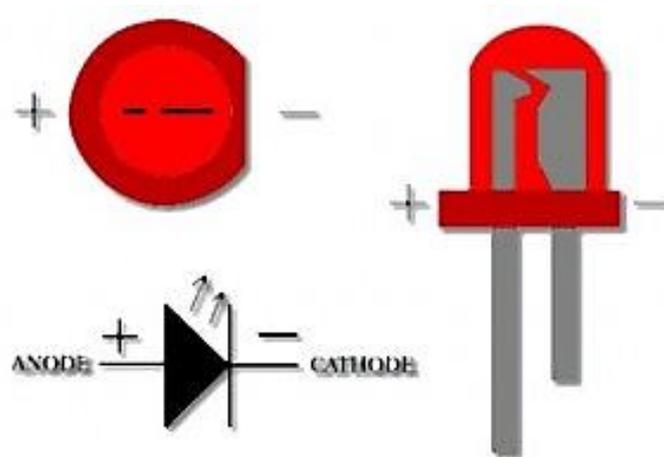
Pembuatan Arduino dimulai pada tahun 2005, dimana sebuah situs perusahaan komputer Olivetti di Ivrea, Italia membuat perangkat untuk mengendalikan proyek desain interaksi siswa supaya lebih murah dibandingkan sistem yang ada pada saat itu. Dilanjutkan pada bulan Mei 2011, dimana sudah lebih dari 300.000 unit Arduino terjual. Pendiri dari Arduino itu sendiri adalah Massimo Banzi dan David Cuartielles. Awalnya mereka memberi nama proyek itu dengan sebutan Arduin dari Ivrea tetapi seturut dengan perkembangan zaman, maka nama proyek itu diubah menjadi Arduino yang berarti “teman yang kuat” atau dalam versi bahasa Inggrisnya dikenal dengan sebutan “Hardwin”. Awalnya Massimo Banzi membangun disekitar proyek Pengkabelan dari Hernando Barragan. Pengkabelan ini adalah proyek tesis Hernando di Desain Interaksi Institute Ivrea. Hal tersebut dimaksudkan untuk menjadi versi elektronik pengolahan yang digunakan di lingkungan pemrograman dan mengambil pola sintaks processing tetapi dengan berkembangnya teknologi maka Arduino menjadi sangat populer dikalangan mahasiswa dan pelajar saat ini. Mereka mengembangkan Arduino dengan bootloader dan software yang user friendly sehingga menghasilkan sebuah board mikrokontroler yang bersifat open source yang bisa dipelajari dan dikembangkan oleh mahasiswa, pelajar, profesional, pemula, dan penggemar elektronika maupun robotik di seluruh dunia.

C. Pengenalan Komponen dan Perangkat Dasar

❖ Light Emitting Dioda (LED)

LED (*Light Emitting Dioda*) adalah dioda yang dapat memancarkan cahaya pada saat mendapat arus bias maju (*forward bias*). LED (*Light Emitting Dioda*) dapat memancarkan

cahaya karena menggunakan doping galium, arsenic dan phosphorus. Jenis doping yang berbeda diata dapat menghasilkan cahaya dengan warna yang berbeda. LED (*Light Emitting Dioda*) merupakan salah satu jenis dioda, sehingga hanya akan mengalirkan arus listrik satu arah saja. LED akan memancarkan cahaya apabila diberikan tegangan listrik dengan konfigurasi forward bias. Berbeda dengan dioda pada umumnya, kemampuan mengalirkan arus pada LED (*Light Emitting Dioda*) cukup rendah yaitu maksimal 20 mA. Apabila LED (*Light Emitting Dioda*) dialiri arus lebih besar dari 20 mA maka LED akan rusak, sehingga pada rangkaian LED dipasang sebuah resistor sebagai pembatas arus. Simbol dan bentuk fisik dari LED (*Light Emitting Dioda*) dapat dilihat pada gambar berikut.



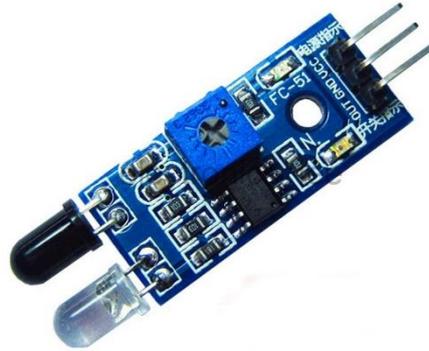
Gambar 5. Simbol dan bentuk fisik LED

Gambar memperlihatkan bahwa LED memiliki kaki 2 buah seperti dengan dioda yaitu kaki anoda dan kaki katoda. Pada gambar diatas kaki anoda memiliki ciri fisik lebih panjang dari kaki katoda pada saat masih baru, kemudian kaki katoda pada LED (*Light Emitting Dioda*) ditandai dengan bagian body LED yang di papas rata. Kaki anoda dan kaki katoda pada LED (*Light Emitting Dioda*) disimbolkan seperti pada gambar diatas. Pemasangan LED (*Light Emitting Dioda*) agar dapat menyala adalah dengan memberikan tegangan bias maju yaitu dengan memberikan tegangan positif ke kaki anoda dan tegangan negatif ke kaki katoda.

❖ **Infra Red Sensor**

Modul infrared (IR) sensor ini memiliki sepasang pemancar dan penerima inframerah. Frekwensi inframerah yang dipancarkan mengenai permukaan halangan/rintangan (objek terdeteksi) akan dipantulkan kembali dan diterima oleh bagian penerima inframerah. Setelah

diproses oleh rangkaian pembanding (*comparator*), lampu hijau akan menyala dan mengeluarkan sinyal digital (*digital output*) rendah. Jarak deteksi dapat diatur dengan potensiometer, dengan jarak efektif 2-30cm, tegangan kerja 3.3V - 5V. Mudah dipasang, mudah digunakan, banyak dipakai pada robot penghindar rintangan, penghindar halangan pada mobil, penghitung garis dan pelacak garis hitam putih dan banyak kegunaan lainnya.



Gambar 6. Rangkaian modul IR

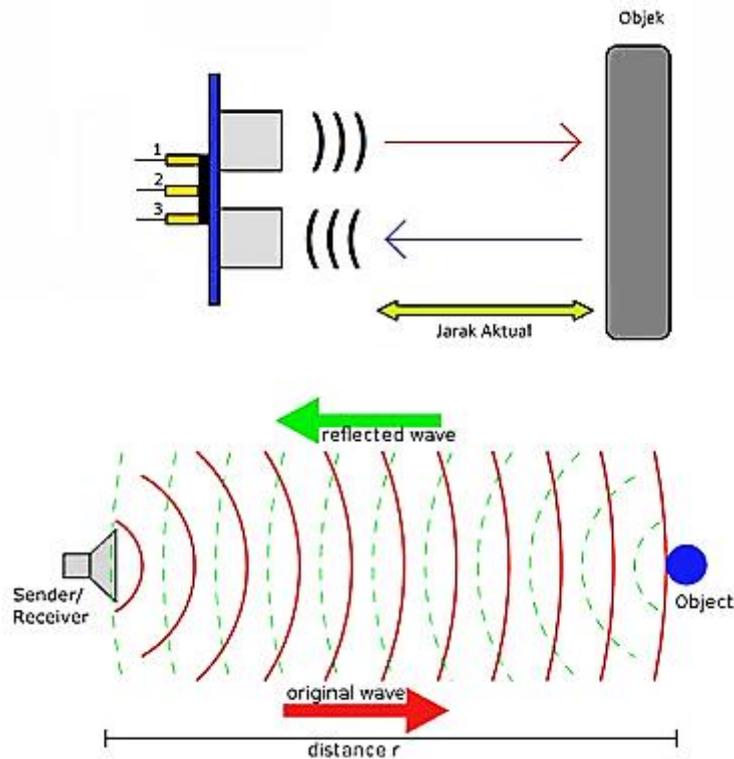
Prinsip kerja dari modul ini adalah sebagai berikut:

1. ketika modul ini mendeteksi halangan di depan sinyal inframerah, lampu indikator warna hijau akan menyala dan port output mengeluarkan sinyal rendah secara menerus. Module ini dapat mendeteksi jarak 2 - 30cm dengan sudut deteksi 35 derajat. Jarak deteksi dapat dinaikkan dengan memutar potensio searah jarum jam dan untuk mengurangi jarak deteksi diputar berlawanan arah jarum jam;
2. sensor aktif inframerah mendeteksi pantulan, oleh karenanya bentuk pantulan dari objek sangat penting. Permukaan warna hitam memiliki permukaan pantulan yang paling kecil dan permukaan putih memiliki pantulan yang paling besar;
3. port output dapat dihubungkan langsung dengan IO port pada mikrokontroler atau dapat juga langsung dihubungkan dengan relay 5V. Memiliki spesifikasi teknis dimana tegangan external (VCC) berkisar antara 3,3V hingga 5V, GND (*ground*) dengan ouput digital 0 dan 1.
4. menggunakan pembanding LM393 comparator yang stabil; dan dapat digunakan pada tegangan 3-5V DC dan ketika diaktifkan, lampu indikator warna merah menyala.

❖ **Ultrasonic Sensor**

Sensor ultrasonik adalah sebuah sensor yang berfungsi untuk mengubah besaran fisis (bunyi) menjadi besaran listrik dan sebaliknya. Cara kerja sensor ini didasarkan pada prinsip dari pantulan suatu gelombang suara sehingga dapat dipakai untuk menafsirkan eksistensi (jarak) suatu benda dengan frekuensi tertentu. Disebut sebagai sensor ultrasonik karena sensor ini menggunakan gelombang ultrasonik (bunyi ultrasonik).

Gelombang ultrasonik adalah gelombang bunyi yang mempunyai frekuensi sangat tinggi yaitu 20.000 Hz. Bunyi ultrasonik tidak dapat di dengar oleh telinga manusia. Bunyi ultrasonik dapat didengar oleh anjing, kucing, kelelawar, dan lumba-lumba. Bunyi ultrasonik nisa merambat melalui zat padat, cair dan gas. Reflektivitas bunyi ultrasonik di permukaan zat padat hampir sama dengan reflektivitas bunyi ultrasonik di permukaan zat cair. Akan tetapi, gelombang bunyi ultrasonik akan diserap oleh tekstil dan busa. Pada sensor ultrasonik, gelombang ultrasonik dibangkitkan melalui sebuah alat yang disebut dengan piezoelektrik dengan frekuensi tertentu. Piezoelektrik ini akan menghasilkan gelombang ultrasonik (umumnya berfrekuensi 40kHz) ketika sebuah osilator diterapkan pada benda tersebut. Secara umum, alat ini akan menembakkan gelombang ultrasonik menuju suatu area atau suatu target. Setelah gelombang menyentuh permukaan target, maka target akan memantulkan kembali gelombang tersebut. Gelombang pantulan dari target akan ditangkap oleh sensor, kemudian sensor menghitung selisih antara waktu pengiriman gelombang dan waktu gelombang pantul diterima.



Gambar 7. Cara kerja sensor ultrasonik dengan transmitter dan receiver (atas), sensor ultrasonik dengan single sensor yang berfungsi sebagai transmitter dan receiver sekaligus

Secara detail, cara kerja sensor ultrasonik adalah sebagai berikut:

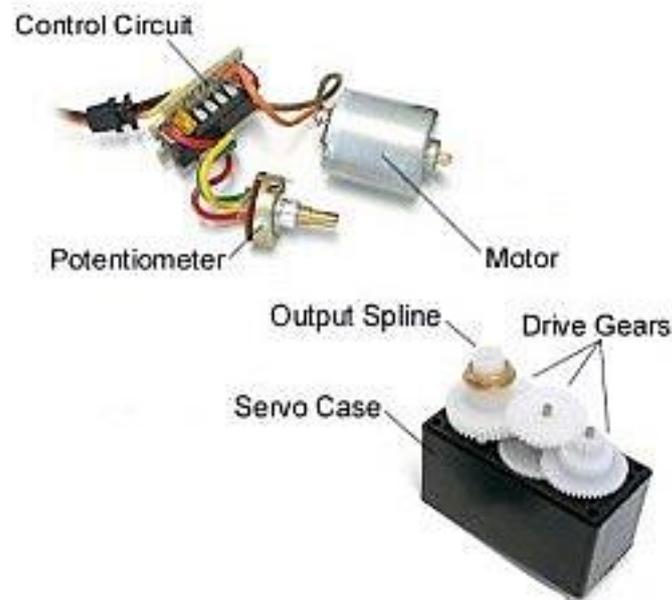
- Sinyal dipancarkan oleh pemancar ultrasonik dengan frekuensi tertentu dan dengan durasi waktu tertentu. Sinyal tersebut berfrekuensi diatas 20kHz. Untuk mengukur jarak benda (sensor jarak), frekuensi yang umum digunakan adalah 40kHz.
- Sinyal yang dipancarkan akan merambat sebagai gelombang bunyi dengan kecepatan sekitar 340 m/s. Ketika menumbuk suatu benda, maka sinyal tersebut akan dipantulkan oleh benda tersebut.
- Setelah gelombang pantulan sampai di alat penerima, maka sinyal tersebut akan diproses untuk menghitung jarak benda tersebut. Jarak benda dihitung berdasarkan rumus:

$$S = 0,034.t/2$$

dimana S merupakan jarak antara sensor ultrasonik dengan benda (bidang pantul), dan t adalah selisih antara waktu (durasi) pemancaran gelombang oleh transmitter dan waktu ketika gelombang pantul diterima *receiver*.

❖ Motor Servo

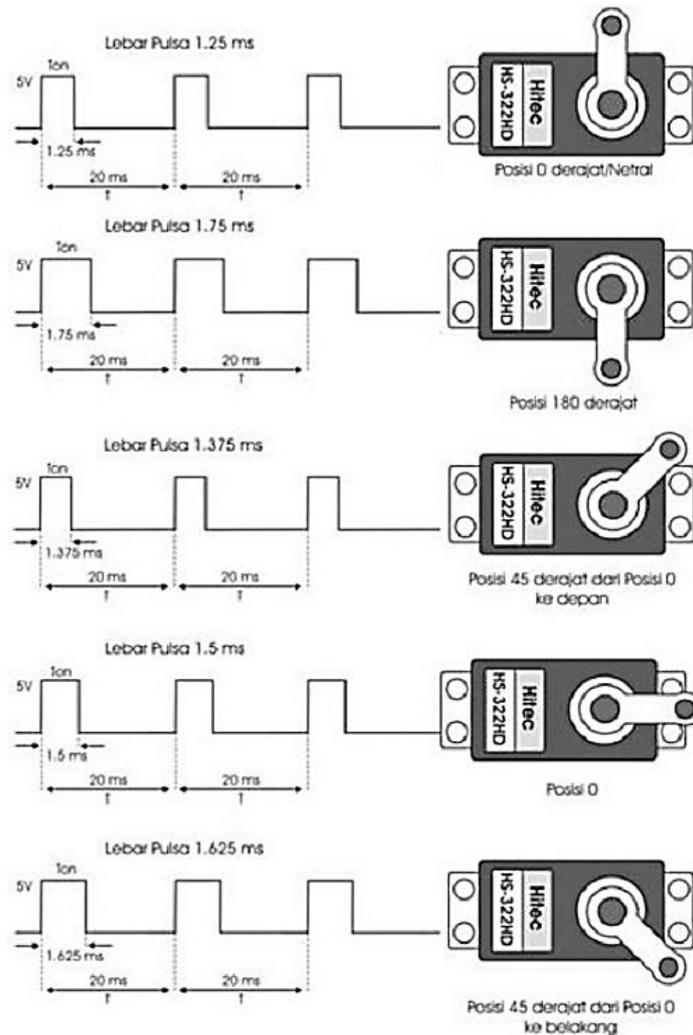
Motor servo adalah jenis motor DC dengan sistem umpan balik tertutup yang terdiri dari sebuah motor DC, serangkaian gear, rangkaian kontrol, dan juga potensiometer. Jadi motor servo sebenarnya tak berdiri sendiri, melainkan didukung oleh komponen-komponen lain yang berada dalam satu paket. Sedangkan fungsi potensiometer dalam motor servo adalah untuk menentukan batas sudut dari putaran servo. Sementara sudut sumbu motor servo dapat diatur berdasarkan lebar pulsa yang dikirim melalui kaki sinyal dari kabel servo itu sendiri. Oleh karena itu motor servo dapat berputar searah dan berlawanan arah jarum jam. Motor servo dapat menampilkan gerakan 0 derajat, 90 derajat, 180 derajat, hingga 360 derajat. Tak heran jika motor ini banyak diaplikasikan untuk penggerak kaki dan juga lengan robot. Selain itu motor servo juga memiliki torsi yang besar sehingga mampu menopang beban cukup berat.



Gambar 8. Bagian-bagian motor servo

Prinsip kerja dari motor servo tak jauh berbeda dibanding dengan motor DC yang lain. Hanya saja motor ini dapat bekerja searah maupun berlawanan jarum jam. Derajat putaran dari motor servo juga dapat dikontrol dengan mengatur pulsa yang masuk ke dalam motor tersebut. Motor servo akan bekerja dengan baik bila pin kontrolnya diberikan sinyal PWM dengan frekwensi 50 Hz. Frekwensi tersebut dapat diperoleh ketika kondisi Ton *duty cycle* berada di angka 1,5 ms. Dalam posisi tersebut rotor dari motor berhenti tepat di tengah-tengah alias sudut

nor derajat atau netral. Pada saat kondisi Ton *duty cycle* kurang dari angka 1,5 ms, maka rotor akan berputar berlawanan arah jarum jam. Sebaliknya pada saat kondisi Ton *duty cycle* lebih dari angka 1,5 ms, maka rotor akan berputar searah jarum jam.

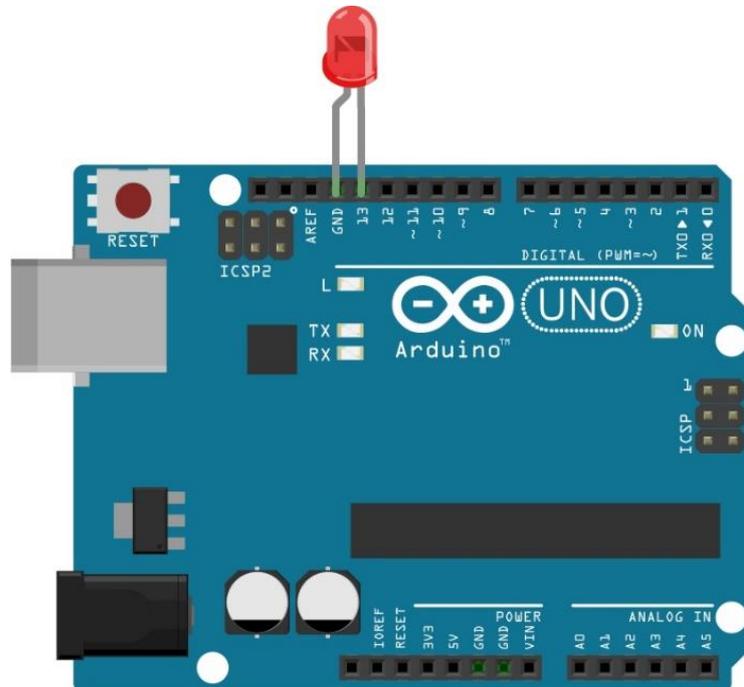


Gambar 9. Skema pulsa kendali motor servo

D. Perakitan dan Pemrograman Dasar

❖ Blinking LED

Skema



Koneksi Pin:

- Kaki positif LED dihubungkan dengan pin digital 13
- Kaki positif LED dihubungkan dengan pin GND

Program

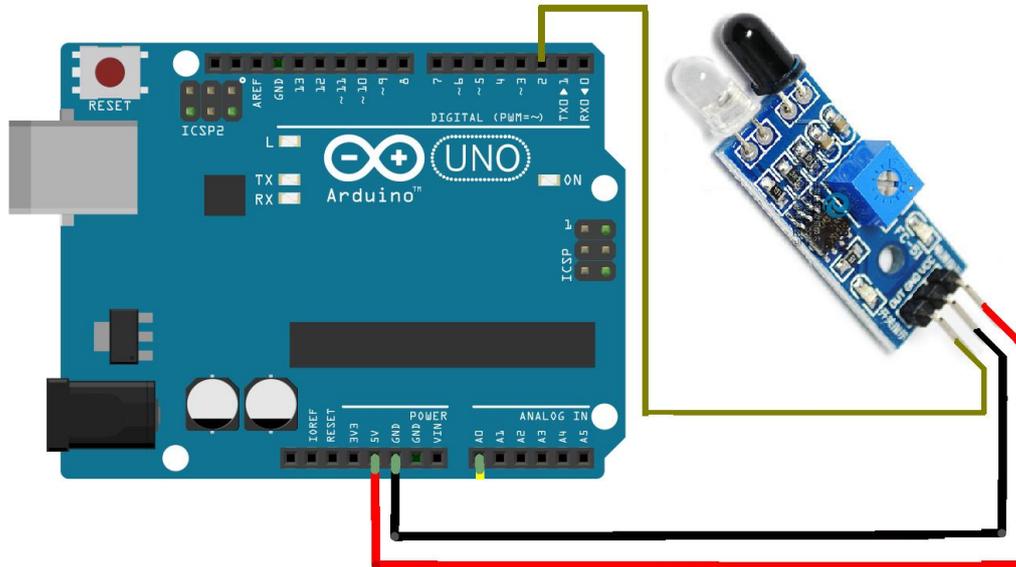
```
int ledPin = 13;

void setup()
{
  PinMode(ledPin, OUTPUT);
  digitalWrite(ledPin, LOW);
}

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

❖ Object Detection

Skema



Koneksi Pin:

- Kaki OUT sensor dihubungkan dengan pin digital 2
- Kaki GND sensor dihubungkan dengan pin GND
- Kaki VCC (tegangan) sensor dihubungkan dengan pin 5V

Program

```
int sensorPin = 2;

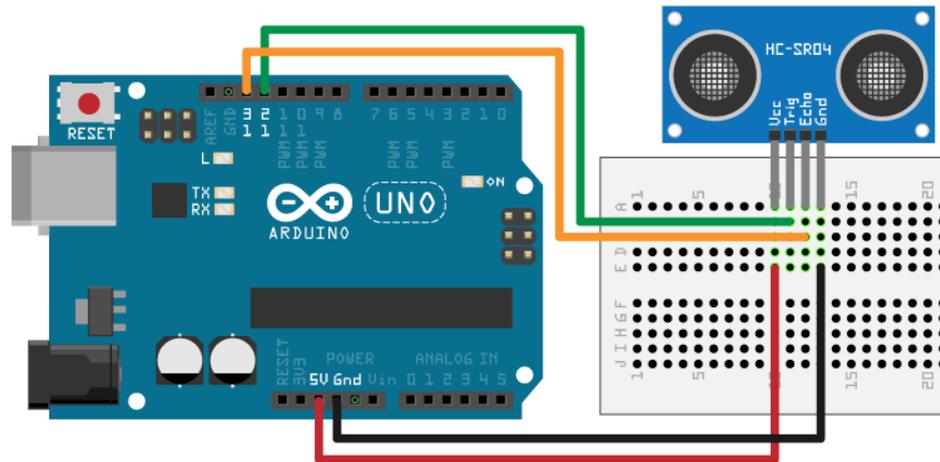
void setup()
{
  PinMode(sensorPin, INPUT);
  Serial.begin(9600);
}

void loop()
{
  int sensorNilai = digitalRead(sensorPin);

  if (sensorNilai == 0)
  {
    Serial.println("Objek Terdeteksi!");
    delay(1000);
  } else
  {
    Serial.println("Objek Tidak Terdeteksi!");
    Delay(1000);
  }
}
```

❖ **Measure Object Distance**

Skema



Koneksi Pin:

- Kaki TRIG sensor dihubungkan dengan pin digital 12 (OUTPUT)
- Kaki ECHO sensor dihubungkan dengan pin digital 13 (INPUT)
- Kaki GND sensor dihubungkan dengan pin GND
- Kaki VCC (tegangan) sensor dihubungkan dengan pin 5V

Program

Pengecekan PulseIn

```
const int trigPin = 12;
const int echoPin = 13;
long durasi;

void setup()
{
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(ledPin, OUTPUT);
  digitalWrite(ledPin, LOW);
  Serial.begin(9600);
}
```

```

void loop()
{
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  durasi = pulseIn(echoPin, HIGH);

  Serial.print("Pulse In: ");
  Serial.println(durasi);
  delay(1000);
  Serial.flush();
}

```

Konversi Durasi ke Jarak dalam Cm

```

const int trigPin = 12;
const int echoPin = 13;
int ledPin = 7;

long durasi;
int jarak;

void setup()
{
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(ledPin, OUTPUT);
  digitalWrite(ledPin, LOW);
  Serial.begin(9600);
}

void loop()
{
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  durasi = pulseIn(echoPin, HIGH);
  jarak = 0.034*durasi/2;

  Serial.print("Jarak = ");
  Serial.print(jarak);
  Serial.println(" Cm");
  delay(100);

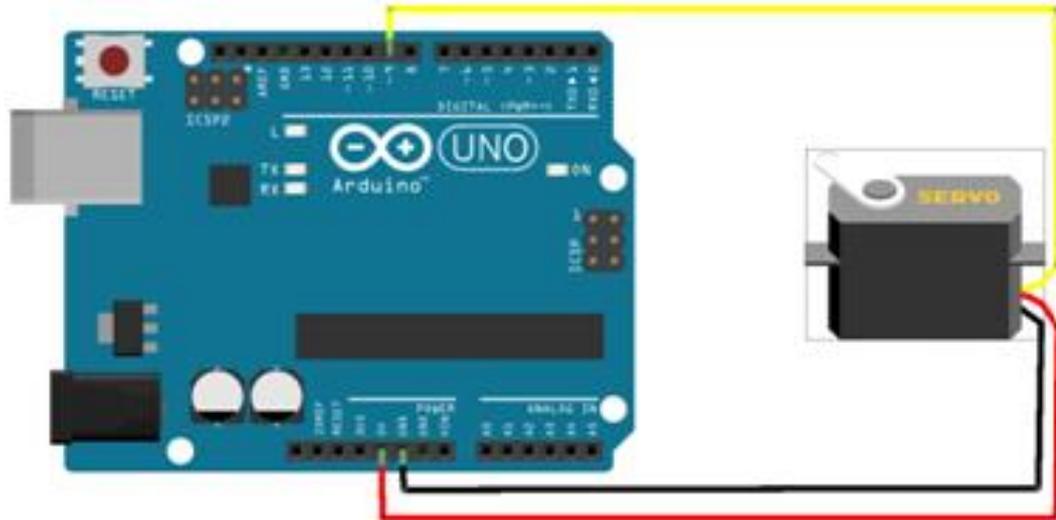
  if(jarak < 10)
    digitalWrite(ledPin, HIGH);
  else
    digitalWrite(ledPin, LOW);

  Serial.flush();
}

```

❖ Servo Rotation Setting

Skema



Koneksi Pin:

- Kaki DATA servo dihubungkan dengan pin digital 9 (OUTPUT)
- Kaki GND servo dihubungkan dengan pin GND
- Kaki VCC (tegangan) servo dihubungkan dengan pin 5V

Program

```
#include <Servo.h>

Servo servo1;

void setup()
{
  servo1.attach(9);
  servo1.write(90);
  delay(3000);
}

void loop()
{
  servo1.write(0);
  delay(1000);
  servo1.write(90);
  delay(1000);
  servo1.write(180);
  delay(1000);
  servo1.write(90);
  delay(1000);
}
```

E. Latihan

KASUS #1

Rangkai dan buatlah sebuah program lampu flip flop menggunakan 3 buah LED yang dapat membuat LED akan menyala secara bergantian dari satu LED ke LED yang lain.

KASUS #2

Rangkai dan buatlah sebuah program menggunakan sensor IR dengan tujuan apabila sensor mendeteksi adanya objek, maka LED akan menyala dan LED akan mati jika sensor tidak mendeteksi adanya objek.

KASUS #3

Rangkai dan buatlah sebuah program menggunakan sensor *ultrasonic* dengan tujuan apabila sensor mendeteksi adanya objek, maka LED akan menyala dan LED akan mati jika sensor tidak mendeteksi adanya objek.

KASUS #4

Rangkai dan buatlah sebuah program menggunakan servo dengan tujuan dimana:

1. Jika servo berputar sebesar 90° , maka hanya 1 LED yang menyala.
2. Jika servo berputar sebesar 180° , maka 2 LED yang menyala.
3. Jika servo berputar sebesar 270° , maka 3 LED yang menyala.

KASUS #5

Buatlah sebuah project semi kompleks berikut programnya dengan aturan sebagai berikut:

1. Rangkaian menggunakan sebuah LED, servo, sensor ultrasonic dan sensor IR
2. Mekanisme kerja alat adalah sebagai berikut:
 - a. jika sensor infra red mendeteksi adanya objek, maka LED akan menyala dan servo akan berputar sejauh 90° ,
 - b. apabila objek telah bergerak melewati sensor ultrasonic, maka servo akan kembali ke posisi semula.