

17 PROYEK BELAJAR ARDUINO



Ir. ILMANZA R.K., M.InfoTech.

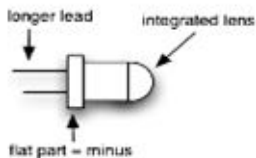
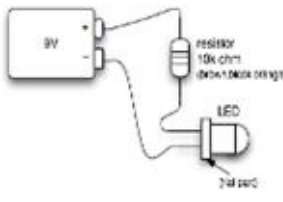
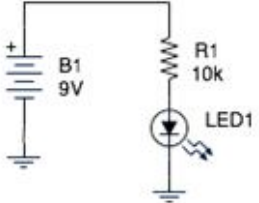
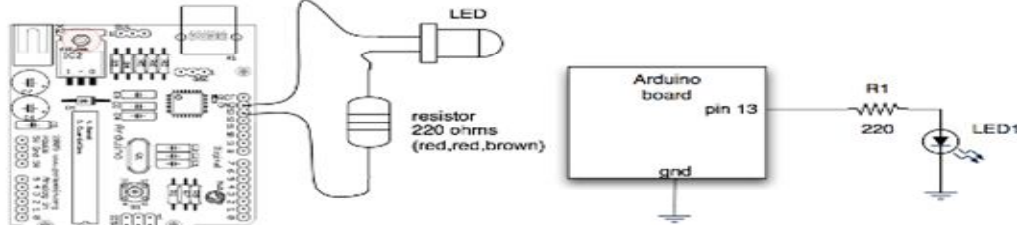
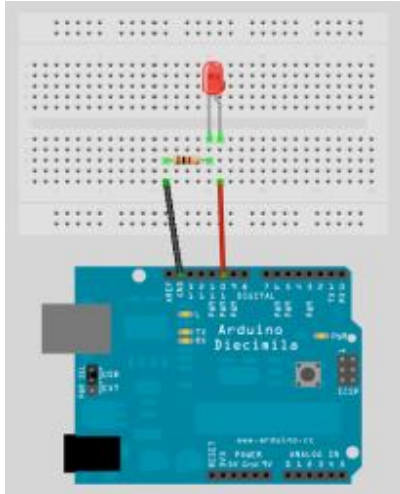
[HTTP://ROBOMANIA.TOKOBAGUS.COM](http://robomania.tokobagus.com)

DAFTAR ISI

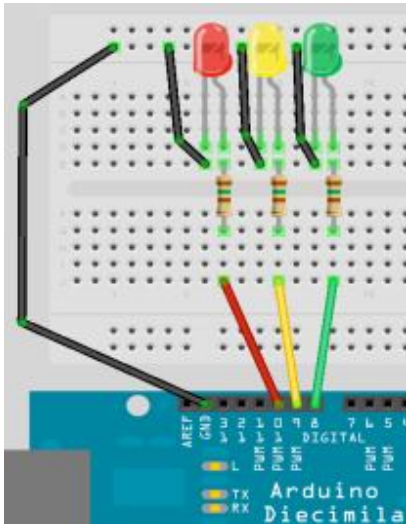
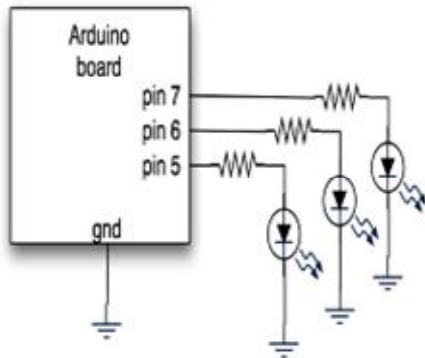
1. BLINKING LED (LAMPU KEDIP)	3
2. TRAFFICT LIGHT (LAMPU LALU LINTAS)	4
3. SWITCH ON/OFF SEBAGAI INPUT MIKROKONTROLER	6
4. POTENTIOMETER SEBAGAI PENGATUR WAKTU	9
5. GENERATOR SUARA / ALARM	11
6. RINGTONE	12
7. PENGATURAN OUTPUT DENGAN PULSE WITH MODULATION (PWM)	14
8. DISPLAY BANNER, BLINKING, SCROLLING & RUNNING TEXT DI LAYAR LCD ..	17
9. SENSOR CAHAYA PHOTOCCELL / LDR	21
10. TERMOMETER DIGITAL	23
11. PENGUKUR JARAK	27
12. PENDETEKSI GERAK	30
13. PENGENDALI PERALATAN LISTRIK DENGAN REMOTE CONTROL TV	31
14. SENSOR GARIS / DETEKTOR WARNA	33
15. SWITCH ON/OFF PERALATAN LISTRIK DENGAN RELAY	35
16. MENGERAKKAN LENGAN ROBOT DENGAN SERVO	36
17. H-BRIDGE MOTOR DRIVER	38

Project 1. BLINKING LED (LAMPU KEDIP)

Langkah-langkah untuk membuat Blinking LED :

 <p>longer lead integrated lens flat part = minus</p> <p>1</p>	 <p>9V resistor 10k ohm (brown, black, orange) LED 3141 00000</p> <p>2</p>	 <p>B1 9V R1 10k LED1</p> <p>3</p>
 <p>LED resistor 220 ohms (red, red, brown) Arduino board pin 13 gnd R1 220 LED1</p> <p>4</p>		
 <p>Pemasangan komponen pada breadboard agar disesuaikan skema.</p>	<p>Sketch:</p> <pre>int led1 = 13; // lampu LED internal pada pin 13 Arduino void setup() { pinMode(led1, OUTPUT); // pin 13 sebagai output } void loop() { digitalWrite(led1, HIGH); // led On delay(1000); // tunggu 1 detik digitalWrite(led1, LOW); // led Off delay(1000); // tunggu 1 detik }</pre>	
<p>Catatan:</p> <ul style="list-style-type: none">- Pemasangan LED harus diperhatikan polaritasnya. Kaki yang panjang mengarah ke kutub Positif (+), sedangkan kaki yang pendek atau kaki pada permukaan LED yg datar, mengarah ke kutub negative (-) atau Ground (Gnd).		

Project 2. TRAFFICT LIGHT (LAMPU LALU LINTAS)



Contoh pemasangan LED menggunakan breadboard pada pin 8,9,10 Arduino

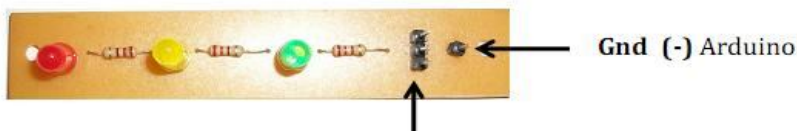
Sketch:

```
int redPin = 7;           // Lampu Merah
int yellowPin = 6;        // Lampu Kuning
int greenPin = 5;         // Lampu Hijau

void setup()
{
  pinMode(redPin, OUTPUT);
  pinMode(yellowPin, OUTPUT);
  pinMode(greenPin, OUTPUT);
}

void loop()
{
  digitalWrite(redPin, HIGH);    // Lampu Merah On
  digitalWrite(yellowPin, LOW);
  digitalWrite(greenPin, LOW);
  delay(3000);                   // Jeda 3 detik
  digitalWrite(redPin, LOW);     // Lampu Kuning On
  digitalWrite(yellowPin, HIGH);
  digitalWrite(greenPin, LOW);
  delay(3000);                   // Jeda 3 detik
  digitalWrite(redPin, LOW);     // Lampu Hijau On
  digitalWrite(yellowPin, LOW);
  digitalWrite(greenPin, HIGH);
  delay(3000);                   // Jeda 3 detik
}
```

MODUL LED 3 WARNA



Signal 1,2,3 (pin output Arduino diberi logika 'HIGH')

Catatan:

- Buatlah lampu lalu lintas dengan urutan: Merah - Kuning (berkedip 3x) - Hijau

Petunjuk:

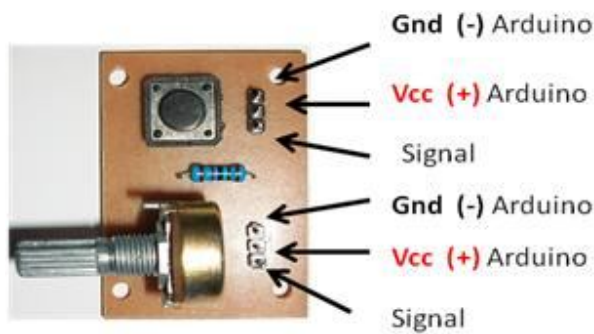
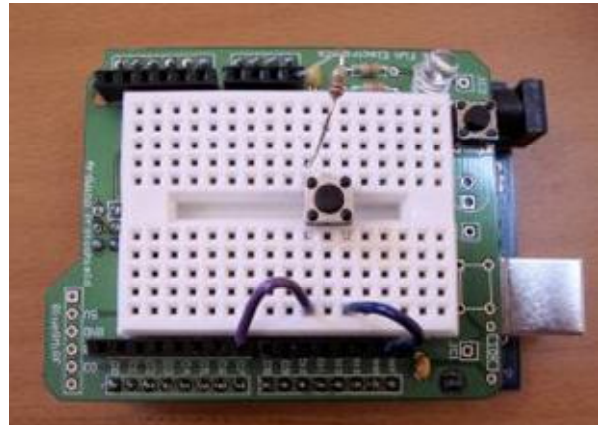
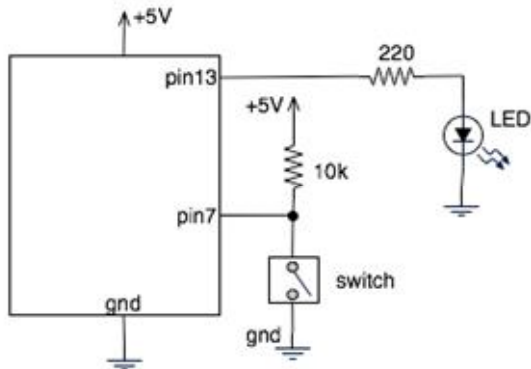
Baris program untuk Lampu kuning:

```
digitalWrite(redPin, LOW);    // Lampu Kuning On  
digitalWrite(yellowPin, HIGH);  
digitalWrite(greenPin, LOW);  
delay(3000);                  // Jeda 3 detik
```

diganti:

```
for( int i=1; i<=3; i++ )  
{  
    digitalWrite(yellowPin, HIGH);  
    delay(1000);  
    digitalWrite(yellowPin, LOW);  
    delay(1000);  
}
```

Project 3. SWITCH ON/OFF SEBAGAI INPUT MIKROKONTROLER



Sketch 1 : Program push button Switch untuk tombol Bell pintu

```
int buttonPin = 7;    // pin signal modul Switch dihubungkan ke pin 7 Arduino
int ledPin = 13;      // nomor pin LED internal Arduino
int buttonState = 0;  // nama variable status switch

void setup() {
  pinMode(buttonPin, INPUT);    // tombol tekan (push button) sebagai input:
  pinMode(ledPin, OUTPUT);      // LED pin sebagai output:
  Serial.begin(9600);           // untuk membaca data pada serial port di layar monitor
}
```

```

void loop() {
  buttonState = digitalRead(buttonPin);    // membaca nilai tombol tekan (sensor digital)
  Serial.println(sensorValue);             // menulis nilai sensor di layar monitor

  if (buttonState == HIGH)                 // check apakah tombol pushbutton ditekan
  {
    digitalWrite(ledPin, HIGH);            // nyalakan LED (on)
  }
  else
  {
    digitalWrite(ledPin, LOW);             // matikan LED (off)
  }
}

```

Catatan: Ganti LED dengan Buzzer (perhatikan polaritas kaki positif / negatif)

Sketch 2: Program push button Switch untuk saklar ON / OFF Lampu

// Program toggle Switch On - Switch Off

```

int buttonPin = 7; // the pin that the pushbutton is attached to
int ledPin = 13;   // the pin that the LED is attached to
int buttonState = 0; // current state of the button

int lastButtonState = 0; // previous state of the button
int buttonPushCounter = 0; // counter for the number of button presses

```

```

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

```

```

void loop() {

  buttonState = digitalRead(buttonPin); // membaca nilai tombol tekan (sensor digital)

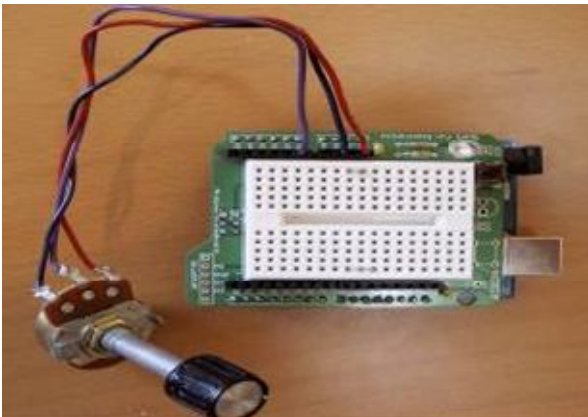
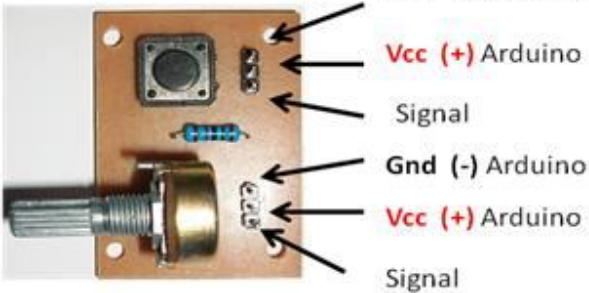
  if (buttonState != lastButtonState) { // jika nilai sekarang tidak sama dengan nilai terakhir

    if (buttonState == HIGH) {

```

```
        buttonPushCounter++;  
    }  
}  
  
lastButtonState = buttonState;  
  
if (buttonPushCounter % 2 == 0) {  
    digitalWrite(ledPin, HIGH);  
}  
else {  
    digitalWrite(ledPin, LOW);  
}  
}
```


Project 4. POTENTIOMETER SEBAGAI PENGATUR WAKTU

<p>turned anti-clockwise</p> <p>0 volts →</p> <p>+5V</p> <p>gnd</p> <p>turned clockwise</p> <p>5 volts →</p> <p>+5V</p> <p>gnd</p> <p>somewhere in the middle</p> <p>2.3 volts →</p> <p>+5V</p> <p>gnd</p>	
<p>Kabel Merah ke + 5v</p> <p>Kabel Ungu (tengah) ke A0 (port analog 0 , bukan port digital)</p> <p>Kabel biru ke Ground (-)</p>	
	
<p>Sketch:</p> <p>// Program pengaturan lamanya lampu On/Off</p> <pre>int sensorPin = 0; // pin signal potentiometer dihubungkan ke port analog 0 int ledPin = 13; // nomor pin LED internal Arduino int sensorValue = 0; // variable nilai awal yg dihasilkan sensor</pre>	

```

void setup()
{
  pinMode(ledPin, OUTPUT);
  Serial.begin(9600);          // untuk membaca data pada serial port di layar monitor
}
void loop()
{
  sensorValue = analogRead(sensorPin);  // membaca nilai dari sensor Analog
  Serial.println(sensorValue);          // menulis nilai sensor di layar monitor

  digitalWrite(ledPin, HIGH);
  delay(sensorValue);                  // lamanya lampu On
  digitalWrite(ledPin, LOW);
  delay(sensorValue);                  // lamanya lampu Off
}

```

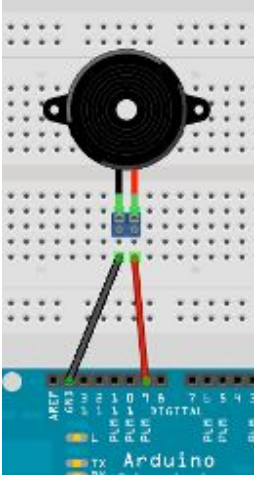
Catatan: Pin tengah potentiometer (juga keluarga resistor yg lain, misalnya LDR, Thermistor) dihubungkan dengan port Analog Arduino, bukan port digital.

Arduino memiliki 5 port Analog yang letaknya bersebelahan dengan pin power supply

Bedakan:

sensorValue = analogRead(sensorPin); // membaca nilai dari sensor Analog
buttonState = digitalRead(buttonPin); // membaca nilai dari sensor Digital (hal 7)

Project 5. GENERATOR SUARA / ALARM

	<p>Sketch:</p> <pre>// Program untuk membunyikan buzzer / alarm int buzz = 9; // BUZZER kutub positive (+) di pin 9 void setup() { pinMode(buzz, OUTPUT); } void loop() { digitalWrite(buzz, HIGH); delay (1000); digitalWrite(buzz, LOW); delay (1000); }</pre>
	<pre>// Alarm dengan pengaturan volume int buzz = 9; // BUZZER void setup() { pinMode(buzz, OUTPUT); } void loop() { analogWrite(buzz, 255); // nilai AnalogWrite antara 0 - 255 delay (1000); analogWrite(buzz, 0); delay (1000); }</pre>
<p>Catatan:</p> <ul style="list-style-type: none">- digitalWrite(buzz, HIGH); -> Akan memberikan tegangan 5v kepada buzzer- analogWrite(buzz, 255); -> Akan memberikan tegangan 5v kepada buzzer- Nilai AnalogWrite berkisar 0 – 255 (diaplikasikan untuk pengaturan volume, cahaya, dll)	

Project 6. RINGTONE

// Program Ringtone 1

```
int speakerPin = 9; // kutup + buzzer atau speaker dipasang pada pin 9 Arduino
```

```
//LAGU 1 (DO RE MI FA SOL ...)
```

```
int length = 16; // ini panjang lagu
```

```
char notes[] = "cdefgabCCbagfedc"; // ini lagunya
```

```
int beats[] = { 1, 1, 1, 1, 1, 1, 1, 4, 1, 1, 1, 1, 1, 1, 4 }; // ini ketukannya
```

```
/*
```

```
// LAGU 2 (Twinkle twinkle)
```

```
int length = 15; // ini panjang lagu
```

```
char notes[] = "ccggaagffeeddc "; // ini lagunya
```

```
int beats[] = { 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 2, 4 }; // ini ketukannya
```

```
*/
```

```
/*
```

```
//LAGU 3 (do mi sol Do ...Do sol mi do)
```

```
int length = 8 ; // ini panjang lagu
```

```
char notes[] = "cegCCgec"; // ini lagunya
```

```
int beats[] = { 1, 1, 1, 4, 1, 1, 1, 4 }; // ini ketukannya
```

```
*/
```

```
int tempo = 300;
```

```
void playTone(int tone, int duration) {
```

```
  for (long i = 0; i < duration * 1000L; i += tone * 2) {
```

```
    digitalWrite(speakerPin, HIGH);
```

```
    delayMicroseconds(tone);
```

```
    digitalWrite(speakerPin, LOW);
```

```
    delayMicroseconds(tone);
```

```
  }
```

```
}
```

```
void playNote(char note, int duration) {
```

```
  char names[] = { 'c', 'd', 'e', 'f', 'g', 'a', 'b', 'C' };
```

```
  int tones[] = { 1915, 1700, 1519, 1432, 1275, 1136, 1014, 956 };
```

```

// play the tone corresponding to the note name
for (int i = 0; i < 8; i++) {
  if (names[i] == note) {
    playTone(tones[i], duration);
  }
}

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

void loop() {
  for (int i = 0; i < length; i++) {
    if (notes[i] == ' ') {
      delay(beats[i] * tempo); // rest
    } else {
      playNote(notes[i], beats[i] * tempo);
    }

    // pause between notes
    delay(tempo / 2);
  }
}

```


Catatan:

Suara yang dihasilkan buzzer kualitasnya kurang baik, karena buzzer diperuntukkan sebagai alarm. Untuk kualitas yang lebih baik, sebaiknya buzzer diganti dengan speaker kecil 8 Ohm.

Project 7. PENGATURAN OUTPUT DENGAN PULSE WITH MODULATION (PWM)

PWM adalah singkatan dari Pulse With Modulation, yang fungsinya untuk mengatur besaran output digital dalam range tertentu (0 – 255). Konsep PWM ini banyak diimplementasikan untuk pengaturan kecepatan motor, besaran intensitas cahaya, dan keperluan lainnya.

Arduino memiliki **6 pin PWM**, yaitu **pin 3, 5, 6, 9, 10, 11**. Oleh karena itu, motor ataupun lampu yang akan dikontrol harus dihubungkan dengan pin PWM tersebut.

 <p>PWM 220R LED</p>	<p>Sketch: // Program pengaturan intensitas cahaya</p> <pre>int ledPin = 9; // PWM pin for the LED void setup() {} // no setup needed void loop() { for (int i=0; i<=255; i++) // ascending value for i { analogWrite(ledPin, i); // sets brightness level to i delay(100); // pauses for 100ms } for (int i=255; i>=0; i--) // descending value for i { analogWrite(ledPin, i); // sets brightness level to i delay(100); // pauses for 100ms } }</pre>
<p>Catatan:</p> <p>for (int i=0; i<=255; i++) = naik +1 secara berulang . for (int i=0; i<=255; i+=5) = naik +5 secara berulang</p> <p>Cobalah memodifikasi program diatas dengan menggunakan 2 LED :</p> <p>// dual fade</p> <pre>int redPin = 9; // Red LED connected to digital pin 9 int yellowPin = 10; // Yellow LED connected to digital pin 10 void setup() { // nothing happens in setup }</pre>	

```

void loop() {

    // fade in from min to max in increments of 5 points:

    for(int fadeValue = 0 ; fadeValue <= 255; fadeValue +=5) {
        // sets the value (range from 0 to 255):
        analogWrite(redPin, fadeValue);
        analogWrite(yellowPin, (255 - fadeValue));
        // wait for 30 milliseconds to see the dimming effect
        delay(30);
    }

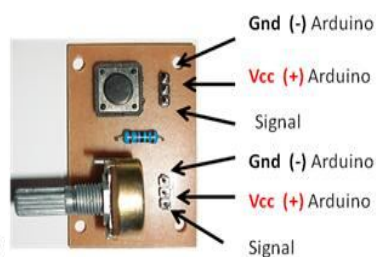
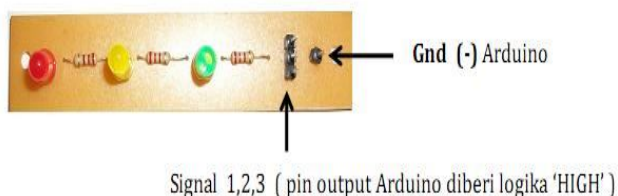
    // fade out from max to min in increments of 5 points:

    for(int fadeValue = 255 ; fadeValue >= 0; fadeValue -=5) {
        // sets the value (range from 0 to 255):
        analogWrite(redPin, fadeValue);
        analogWrite(yellowPin, (255 - fadeValue));
        // wait for 30 milliseconds to see the dimming effect
        delay(30);
    }
}

```

Sketch diatas dapat dikembangkan menjadi lebih dinamis dengan menambah potentiometer sebagai pengatur intensitas cahaya. Cara pemasangan potentiometer sama dengan yang telah dipraktikkan pada project 4, yaitu satu kaki ke kutub positif (+), kaki tengah ke pin A0 (Analog 0), dan kaki satunya ke kutub negatif (-). Adapun sketchnya dapat dimodifikasi menjadi sbb:

MODUL LED 3 WARNA



// Program pengaturan intensitas cahaya menggunakan potentiometer

```

int potPin = 0;
int ledPin = 9;
int potValue = 0;

```

```
void setup()
{
  //pinMode(int ledPin, OUTPUT);
}

void loop() {
  potValue = analogRead(potPin) / 4; // Input analog nilainya 1024. Harus / 4 agar jadi (0- 255)
  //potValue = map(val, 0, 1023, 0, 254); // cara lain mengkonversi nilai 0-1024 menjadi 0-255
  analogWrite(ledPin, potValue);
}

//potValue pada analogWrite nilainya 0-255.
```

Catatan:

Selanjutnya, project tersebut diatas dapat dimodifikasi menjadi pengendali putaran motor dengan potentiometer. Pada prinsipnya, hanya perlu mengganti LED dengan motor DC.

Sketch tetap menggunakan pengaturan output dengan PWM, sebagaimana contoh pengaturan intensitas cahaya dengan potensiometer diatas.

Arus output dari Arduino tidaklah cukup untuk menggerakkan motor jenis TAMIYA, sehingga diperlukan rangkaian penguat untuk menggerakkannya (H-Bridge Motor Driver)

Project 8. DISPLAY BANNER, BLINKING, SCROLLING & RUNNING TEXT DI LAYAR LCD

```
// Program menulis Banner text di layar LCD

// include the library code:
#include <LiquidCrystal.h>

// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup() {
  // set up the LCD's number of rows and columns:
  lcd.begin(16, 2);
  lcd.clear();           // start with a blank screen
  lcd.setCursor(0,0);    // set cursor to column 0, row 0 (the first row)
  lcd.print("BLUES LAND"); // change this text to whatever you like. keep it clean.
  lcd.setCursor(0,1);    // set cursor to column 0, row 1
  lcd.print("Arifal Akmal");
}

void loop()
{
}
```

```
// Program Blinking Banner text

// include the library code:
#include <LiquidCrystal.h>

// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup() {
  // set up the LCD's number of rows and columns:
  lcd.begin(16, 2);
  lcd.clear();           // start with a blank screen
  lcd.setCursor(0,0);    // set cursor to column 0, row 0 (the first row)
  lcd.print("BLUES LAND"); // change this text to whatever you like. keep it clean.
  lcd.setCursor(0,1);    // set cursor to column 0, row 1
  lcd.print("Arifal Akmal");
```

```
}

void loop() {
  // Turn off the blinking cursor:
  lcd.noDisplay();
  delay(500);
  // Turn on the display:
  lcd.display();
  delay(500);
}
```

```
// Program Banner scroll text 1234567890

// include the library code:
#include <LiquidCrystal.h>

// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup() {
  // set up the LCD's number of columns and rows:
  lcd.begin(16,2);
}

void loop() {
  // set the cursor to (0,0):
  lcd.setCursor(0, 0);
  // print from 0 to 9:
  for (int thisChar = 0; thisChar < 10; thisChar++) {
    lcd.print(thisChar);
    delay(500);
  }

  // set the cursor to (16,1):
  lcd.setCursor(16,1);
  // set the display to automatically scroll:
  lcd.autoscroll();
  // print from 0 to 9:
  for (int thisChar = 0; thisChar < 10; thisChar++) {
    lcd.print(thisChar);
    delay(500);
  }
  // turn off automatic scrolling
}
```

```
lcd.noAutoscroll();
```

```
// clear screen for the next loop:  
lcd.clear();  
}
```

```
// Program Baner running text
```

```
// include the library code:  
#include <LiquidCrystal.h>
```

```
// initialize the library with the numbers of the interface pins  
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
```

```
void setup() {  
  // set up the LCD's number of rows and columns:  
  lcd.begin(16, 2);  
  // Print a message to the LCD.  
  lcd.print("ROBOT KREATIF"); // 12 huruf  
  delay(1000);  
}
```

```
void loop() {  
  // scroll 12 positions (string length) to the left  
  // to move it offscreen left:  
  for (int positionCounter = 0; positionCounter < 12; positionCounter++) {  
    // scroll one position left:  
    lcd.scrollDisplayLeft();  
    // wait a bit:  
    delay(300);  
  }
```

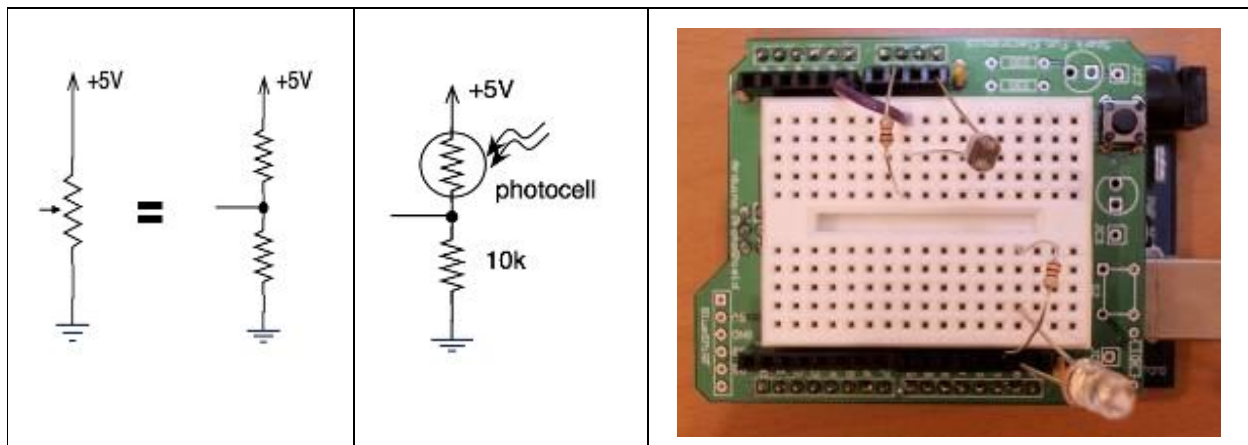
```
  // scroll 28 positions (string length + display length) to the right  
  // to move it offscreen right: 12+16 = 28  
  for (int positionCounter = 0; positionCounter < 28; positionCounter++) {  
    // scroll one position right:  
    lcd.scrollDisplayRight();  
    // wait a bit:  
    delay(300);  
  }
```

```
  // scroll 16 positions (display length + string length) to the left  
  // to move it back to center:
```

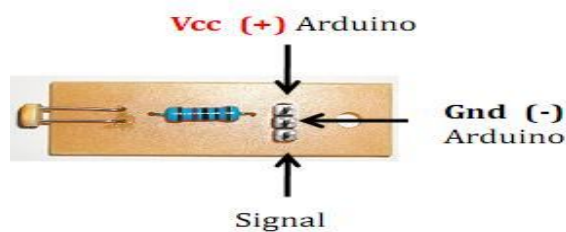
```
for (int positionCounter = 0; positionCounter < 16; positionCounter++) {  
    // scroll one position left:  
    lcd.scrollDisplayLeft();  
    // wait a bit:  
    delay(300);  
}  
  
// delay at the end of the full loop:  
delay(1000);  
}
```

Project 9. SENSOR CAHAYA PHOTOCELL / LDR

Dengan sensor cahaya LDR (Light Dependent Resistor), Lampu akan otomatis menyala (On) jika sensor tidak terkena cahaya (gelap). Sebaliknya, lampu akan mati (Off) apabila sensor terkena cahaya terang.



SENSOR CAHAYA



Sketch:

// Program Lampu otomatis gelap/terang

Int sensorPin = 0; // pin signal LDR dihubungkan ke Port Analog 0 Arduino

Int ledPin = 13; // pin untuk LED

int sensorValue = 0; // variable nilai yg dihasilkan sensor

void setup()

{

pinMode(ledPin, OUTPUT);

Serial.begin(9600); // untuk membaca data pada serial port di layar monitor

}

```
void loop()
{
    sensorValue = analogRead(sensorPin);    // membaca nilai dari sensor:

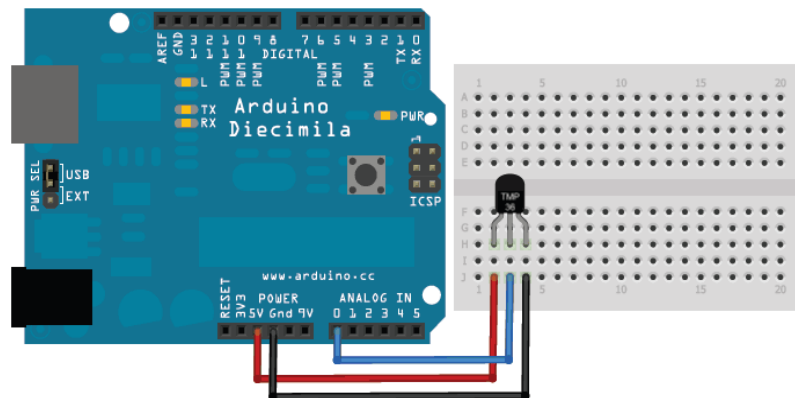
    Serial.println(sensorValue);             // menulis nilai sensor di layar monitor

    if (sensorValue <= 500 )                 // tentukan batas intensitas cahaya 0 - 1024
    {
        digitalWrite(ledPin, HIGH);         // menyalakan lampu LED (on)
    }
    else
    {
        digitalWrite(ledPin, LOW);          // mematikan lampu LED (off)
    }
}
```

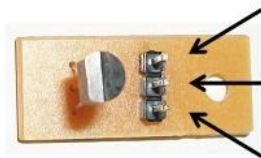
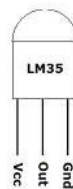
Catatan: Pin signal modul LDR (juga keluarga resistor yg lain, misalnya Potensiometer, Thermistor) dihubungkan dengan port Analog Arduino, bukan port digital

Project 10. TERMOMETER DIGITAL

Untuk mengukur suhu, caranya cukup mudah. IC jenis LM 35 ini cukup praktis, bentuknya kecil dan akurasinya tinggi. Cukup dengan menghubungkan kakinya ke kutub + , A0 dan – seperti pada gambar, nilai tegangan listrik yang didapat akan dikonversi menjadi nilai suhu dalam satuan derajat Celsius/Rheamur/Fahrenheit.



SENSOR SUHU



Vcc (+) Arduino

Signal

Gnd (-) Arduino

Sketch:

a. Nilai temperatur ditampilkan di layar komputer

```
int potPin = 0;           // select the input pin for the LM35
float temperature = 0;     // type float -> 2 angka di blknng koma
long val = 0;             // tyle long = int, hanya range nilainya lebih besar

void setup()
{
  Serial.begin(9600);
}
```

```

void loop()
{
  if (Serial.available())           // if monitor screen opened
  {
    val = analogRead(potPin);       // read the value from the sensor
    temperature = (5.0 * val * 100.0)/1024.0; // conversion formula to Celcius
    Serial.println(temperature);    // write temperature to notebook monitor
    // Serial.println((long)temperature); // jika nilainya ingin dibulatkan
  }
  delay(1000);
}

```

b. Nilai temperatur ditampilkan di layar LCD

```

#include <LiquidCrystal.h>           // include library for LCD

LiquidCrystal lcd(12, 11, 5, 4, 3, 2); // pin to LCD

int potPin = 0;                     // select the input pin for the LM35
float temperature = 0;
long val = 0;

void setup()
{
  lcd.begin(16, 2);                 // set up the LCD's number of rows and columns
}

void loop()
{
  val = analogRead(potPin);         // read the value from the sensor
  temperature = (5.0 * val * 100.0)/1024.0; // convert to Celcius

  lcd.clear();                      // clear LCD screen
  lcd.setCursor(0,0);               // set text to LCD row 1
  lcd.print("current temp. ");      // some text to add meaning to the numbers
  lcd.setCursor(0,1);               // set text to LCD row 2
  lcd.print((long)temperature);      // writing temperature value
  lcd.print(" deg.C");

  delay(1000);
}

```


c. Thermometer digital dengan fitur lampu indikator

Tambahkan pengukur suhu ini dengan fitur lampu indikator (rangkaian traffict light) yang menyala bergantian pada suhu tertentu. Gunakan logika IF-ELSE atau SWITCH-CASE.

```
int potPin = 0;           // select the input pin for the LM35
float temperature = 0;
int suhu;
long val = 0;
int redLight = 7;
int yellowLight = 6;
int greenLight = 4;

void setup()
{
  Serial.begin(9600);

  pinMode(redLight, OUTPUT);
  pinMode(yellowLight, OUTPUT);
  pinMode(greenLight, OUTPUT);
}

void loop()
{
  if (Serial.available())      // if monitor screen opened
  {
    val = analogRead(potPin); // read the value from the sensor
    temperature = (5.0 * val * 100.0)/1024.0;
    suhu = temperature;
    Serial.println("Suhu sekarang adalah : ");
    Serial.println((long)temperature);

    if (suhu > 30) { // Jika lebih besar dari 30 derajat
      digitalWrite(redLight,HIGH);
      digitalWrite(yellowLight,LOW);
      digitalWrite(greenLight,LOW);
    }

    if (suhu >=27 || suhu <= 30) { // Jika suhunya antara 27-30 derajat
      digitalWrite(yellowLight,HIGH);
      digitalWrite(greenLight,LOW);
      digitalWrite(redLight,LOW);
    }
  }
}
```

```

if (suhu < 27) { // Jika suhunya kurang dari 27 derajat
    digitalWrite(greenLight,HIGH);
    digitalWrite(redLight,LOW);
    digitalWrite(yellowLight,LOW);
}

}
delay(5000);
}

```

Catatan:

----- **Jika menggunakan IF - ELSE**

```

if (suhu > 24) //
{
}
if (suhu == 26) // 'sama dengan' ditulis dengan simbol == . Tidak sama dengan, simbolnya !=
{
}
if (suhu != 24 || suhu !=26) // simbol || artinya OR, sedangkan simbol && artinya AND
{
}

```

----- **Jika menggunakan SWICH - CASE**

```

switch (suhu)
{
    case 24 : // Jika suhunya 24 derajat
        .....
        break;

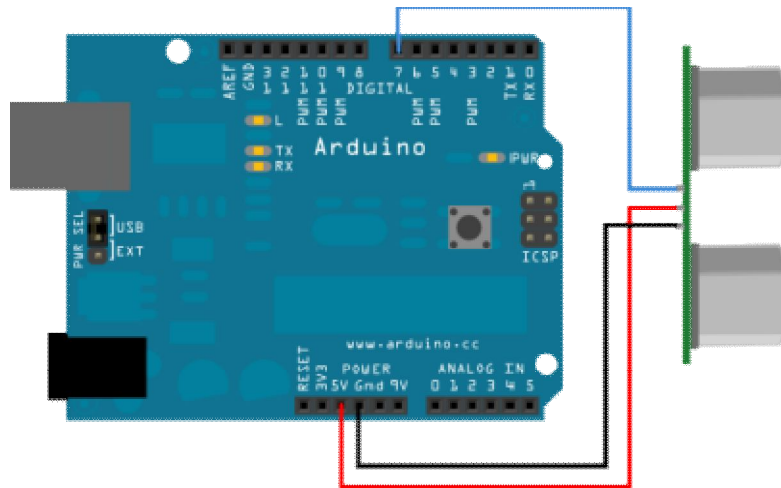
    case 26 : // Jika suhunya 23 derajat
        .....
        break;

    default : // jika suhunya selain 23 dan 24 derajat
        .....
}

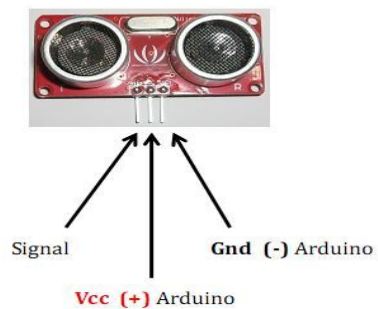
```

Project 11. PENGUKUR JARAK

Untuk mengukur jarak digunakan Sensor Ultrasonic yang cara kerjanya adalah memancarkan gelombang ultrasonic dan menangkap pantulannya jika mengenai benda di depannya. Waktu pantul itulah yang akan di konversi kedalam satuan jarak.



SENSOR JARAK ULTRASONIC



Sketch:

```
const int pingPin = 7;

void setup() {
  Serial.begin(9600);
}
```

```

void loop() {

    long duration, cm;

    pinMode(pingPin, OUTPUT);
    digitalWrite(pingPin, LOW);
    delayMicroseconds(2);
    digitalWrite(pingPin, HIGH);
    delayMicroseconds(5);
    digitalWrite(pingPin, LOW);

    pinMode(pingPin, INPUT);
    duration = pulseIn(pingPin, HIGH);

    cm = microsecondsToCentimeters(duration);

    Serial.print(cm);
    Serial.print("cm");
    Serial.println();

    delay(1000);
}

long microsecondsToCentimeters(long microseconds)
{
    return microseconds / 29 / 2;
}

```

```

#include <LiquidCrystal.h>                // include library for LCD

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);    // pin to LCD
int pingPin = 7;

void setup()
{
    lcd.begin(16, 2);                    // set up the LCD's number of rows and columns
}

void loop() {

    long duration, cm;

```

```

pinMode(pingPin, OUTPUT);
digitalWrite(pingPin, LOW);
delayMicroseconds(2);
digitalWrite(pingPin, HIGH);
delayMicroseconds(5);
digitalWrite(pingPin, LOW);

pinMode(pingPin, INPUT);
duration = pulseIn(pingPin, HIGH);

cm = microsecondsToCentimeters(duration);

lcd.clear();
lcd.setCursor(0, 0);
lcd.print(cm);
lcd.print("cm");

delay(1000);
}

long microsecondsToCentimeters(long microseconds)
{
    return microseconds / 29 / 2;
}

```

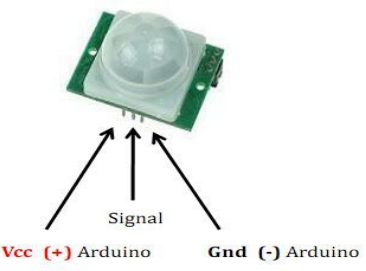
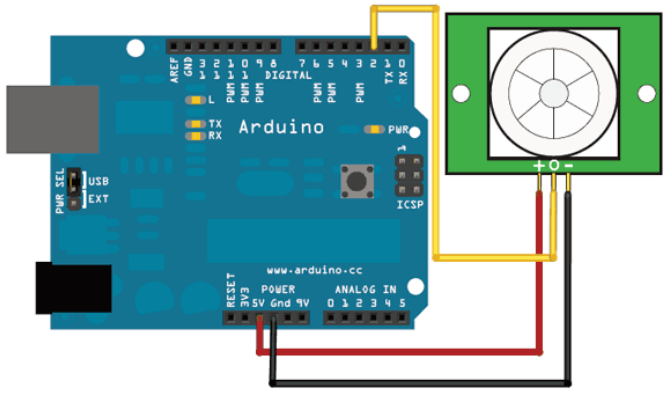
Catatan:

Sensor jarak ini banyak diimplementasikan untuk berbagai macam proyek robotik, misalnya:

- Penaksir jarak
- Robot obstacle avoider (Penghindar halangan)
- Radar pendeteksi benda asing
- Pengukur tinggi badan
- Pengukur ketinggian air

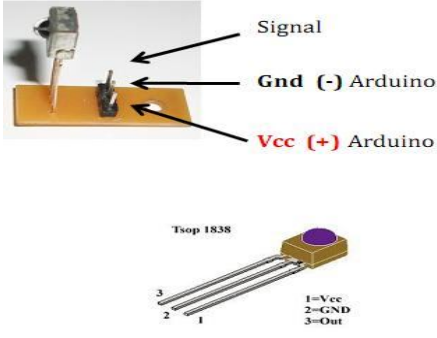
Project 12. PENDETEKSI GERAK

Untuk mendeteksi gerakan, diperlukan sensor gerak yang biasa disebut Passive Infra Red (PIR), yang cara kerjanya adalah mendeteksi adanya perbedaan/perubahan suhu sekarang dan sebelumnya.

<p>SENSOR GERAK (PIR)</p>  <p>Vcc (+) Arduino Signal Gnd (-) Arduino</p>	
<pre>/* Motion detector */ const byte ledPin = 13; // internal LED pada pin 13 const byte motionPin = 2; // pin Signal sensor gerak dihubungkan dengan port 2 byte senseMotion = 0; // variable deteksi gerak void setup() { pinMode(ledPin, OUTPUT); pinMode(motionPin, INPUT); } void loop() { senseMotion = digitalRead(motionPin); // membaca nilai sensor gerak if (senseMotion == HIGH) { // jika terdeteksi ada gerakan digitalWrite(ledPin, HIGH); delay(2000); } else { // jika tidak ada gerakan digitalWrite(ledPin, LOW); } }</pre>	

Project 13. PENGENDALI PERALATAN LISTRIK DENGAN REMOTE CONTROL TV

Sensor yang dapat menangkap signal dari remote control TV disebut Infra Red Decoder (IR Decoder). Setiap tombol remote control memiliki nilai yang berbeda. Kita bisa menangkap nilai tombol yang diinginkan dengan menampilkannya pada layar monitor dan memberikan perintah tertentu untuk tombol tersebut.

<p>SENSOR REMOTE CONTROL TV</p> 	<p>Catatan:</p> <ul style="list-style-type: none">- Gunakan rangkaian lampu traffict light sebagai output- Sinyal remote control yg ditangkap IR Decoder dan nilainya ditampilkan dilayar monitor.- Selanjutnya, tombol yang diinginkan (misal: 1,2,3) dapat diberi perintah untuk menyalakan LED 1, 2 atau 3
<p>// Apabila menggunakan Remote control merek Sonny</p> <pre>#include <IRremote.h> int RECV_PIN = 11; IRrecv irrecv(RECV_PIN); // Output signal sensor remote dihubungkan ke pin 11 decode_results results; int key; int led1 = 5; // lampu1 pada pin 5 int led2 = 6; // lampu2 pada pin 6 int led3 = 7; // lampu3 pada pin 7 void setup() { pinMode(led1, OUTPUT); pinMode(led2, OUTPUT); pinMode(led3, OUTPUT); Serial.begin(9600); irrecv.enableIRIn(); // Start the receiver }</pre>	

```

unsigned long last = millis();

void loop() {
  if (irrecv.decode(&results)) {
    // If it's been at least 1/4 second since the last
    if (millis() - last > 250) {
      key = results.value, HEX;
      Serial.println(key);          // Mengetahui nilai tombol remote

      if(key == 16) {
        Serial.println("tombol 1"); // jika ditekan tombol 1
        digitalWrite(led1, HIGH);
        digitalWrite(led2, LOW);
        digitalWrite(led3, LOW);
      }
      if(key == 2064) {
        Serial.println("tombol 2"); // jika ditekan tombol 2
        digitalWrite(led1, LOW);
        digitalWrite(led2, HIGH);
        digitalWrite(led3, LOW);
      }
      if(key == 1040) {
        Serial.println("tombol 3"); // jika ditekan tombol 3
        digitalWrite(led1, LOW);
        digitalWrite(led2, LOW);
        digitalWrite(led3, HIGH);
      }
    }
    last = millis();
    irrecv.resume(); // Receive the next value
  }
}

```

Catatan: Gunakan statement **SWITCH-CASE** sebagai pengganti **IF – ELSE**, agar proses berjalan lebih cepat !

Project 14. SENSOR GARIS / DETEKTOR WARNA

Sepasang sensor garis biasanya digunakan pada Robot Line tracer untuk mendeteksi warna dasar putih dan garis hitam pada papan lintasan. Sensor ini menggunakan pancaran sinar infrared yang akan dipantulkan dan ditangkap oleh fotodiode. Nilai yang terbaca selanjutnya akan diolah oleh mikrokontroler untuk menjalankan aksi tertentu.

SENSOR GARIS / WARNA INFRARED



Sketch:

```
/*
 * Infrared and Arduino
 * Reads an input from IR, sends it to serial monitor and
 * turns a LED off or on.
 */

// variables
int LDR_pin = 0; // analog pin 0 (connect IR here)
int LDR_val = 0; // variable use to read input data
int LEDpin = 13; //This is the LED

// a threshold to decide when the LED turns on
// batas bawah = 50 (hitam) , atas = 400 (kertas putih)

int threshold = 50;

void setup(){
  pinMode(LDR_pin, INPUT);
  pinMode(LEDpin, OUTPUT);
  Serial.begin(9600);
}

void loop(){

  LDR_val = analogRead(LDR_pin);

  if (LDR_val > threshold)
    digitalWrite(LEDpin, HIGH);
  else
    digitalWrite(LEDpin, LOW);
}
```

	<pre>// output 'LDR_val' value into the console Serial.print("LDR = "); Serial.print(LDR_val); Serial.println(); delay (1000); }</pre>
--	--

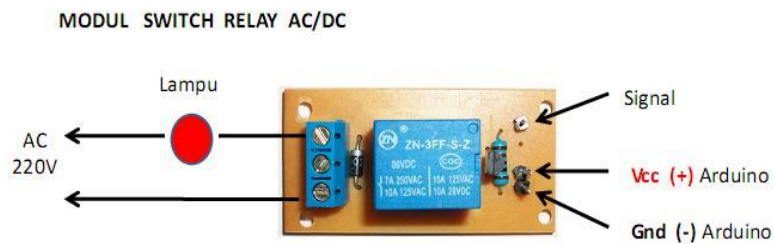
Catatan:

Cobalah dengan mendeteksi berbagai macam warna kertas yang permukaannya tidak mengkilat.

Tangkap nilainya, dan berikan statement IF untuk menterjemahkannya menjadi nama WARNA tertentu.

Project 15. SWITCH ON/OFF PERALATAN LISTRIK DENGAN RELAY

Output dari pembacaan sensor (misalnya sensor garis, suhu, cahaya, warna, remote control, gerak, jarak), dapat dihubungkan dengan relay yang berfungsi sebagai switch / tombol ON/OFF berbagai peralatan listrik. Skemanya adalah sebagai berikut:



Sketch:

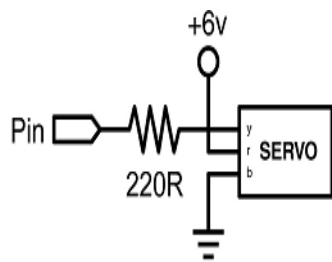
```
int relay1 = 3;          // pin 3 Arduino dihubungkan dengan pin SIGNAL modul Relay

void setup()
{
    pinMode(relay1, OUTPUT);    // pin 13 sebagai output
}

void loop()
{
    digitalWrite(relay1, HIGH);    // led On
    delay(1000);                  // tunggu 1 detik
    digitalWrite(relay1, LOW);     // led Off
    delay(1000);                  // tunggu 1 detik
}
```

Project 16. MENGERAKKAN LENGAN ROBOT DENGAN SERVO

Servo adalah motor yang sudut putarnya antara 0 – 180 derajat. Berputarnya Servo didasarkan atas perintah mikrokontroler untuk berputar dengan sudut tertentu dan kecepatan tertentu.



Catatan:

- Gunakan sumber tegangan sesuai spesifikasi Servo
- Biasanya digunakan untuk menggerakkan lengan / kaki Robot
- Program berikut ini adalah untuk memutar Servo dengan sudut putar 10 – 170 derajat ke kanan dan kekiri.

```
int servoPin = 2; // servo connected to digital pin 2
int myAngle; // angle of the servo roughly 0-180
int pulseWidth; // servoPulse function variable

void setup()
{
    pinMode(servoPin, OUTPUT); // sets pin 2 as output
}
void servoPulse(int servoPin, int myAngle)
{
    pulseWidth = (myAngle * 10) + 600; // determines delay
    digitalWrite(servoPin, HIGH); // set servo high
    delayMicroseconds(pulseWidth); // microsecond pause
    digitalWrite(servoPin, LOW); // set servo low
}
void loop()
{
    // servo starts at 10 deg and rotates to 170 deg
    for (myAngle=10; myAngle<=170; myAngle++)
    {
        servoPulse(servoPin, myAngle); // send pin and angle
        delay(20); // refresh cycle
    }
    // servo starts at 170 deg and rotates to 10 deg
    for (myAngle=170; myAngle>=10; myAngle--)
    {
        servoPulse(servoPin, myAngle); // send pin and angle
        delay(20); // refresh cycle
    }
}
```

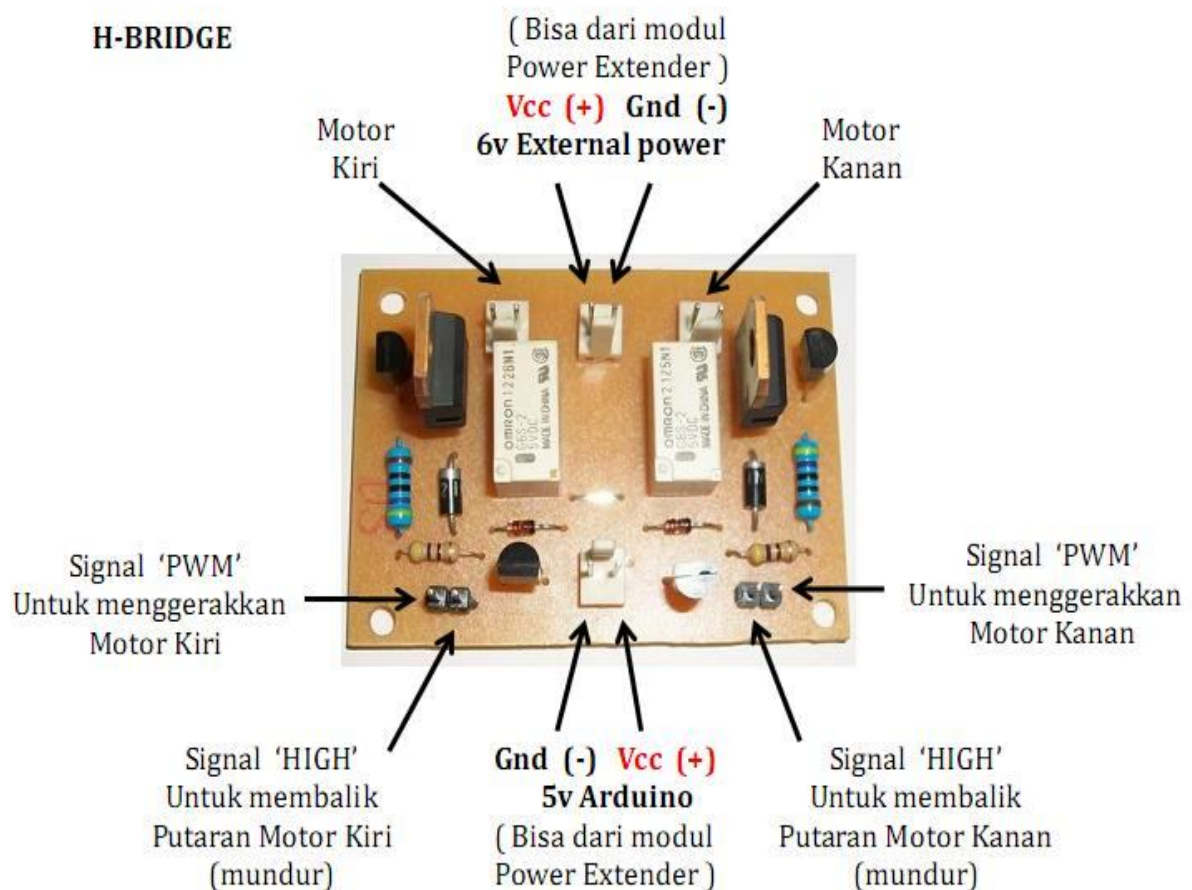
```
/*  
PROGRAM UNTUK MEMUTAR SERVO DENGAN POTENTIOMETER  
*/  
  
#include <Servo.h>  
  
Servo myservo;    // create servo object to control a servo  
int sliderpin = 0; // analog pin potentiometer (CENTER PIN)  
int sensorValue;  // variable to read the value from the analog pin  
  
void setup()  
{  
  myservo.attach(0); // attaches the servo on pin 0 to the servo object  
}  
  
void loop()  
{  
  sensorValue = analogRead(sliderpin); // reads the value of the slider (value between 0 -1023)  
  sensorValue = sensorValue/5.68;      // scale it to use it with the servo (value between 0 -180)  
  myservo.write(sensorValue);           // sets the servo position according to the scaled value  
  delay(15);                           // waits for the servo to get there  
}
```

Project 17. H-BRIDGE MOTOR DRIVER

Motor DC, misalnya motor Tamiya, tidak dapat langsung dihubungkan dengan output Mikrokontroler, karena arusnya terlalu kecil. Untuk itu diperlukan rangkaian H-bridge sebagai penguat arus sekaligus pengendali kecepatan motor dan arah putaran motor DC.

Catatan:

- Rangkai modul H-Bridge, bodi Robot mobil (dengan 2 DC motor Tamiya) dan sumber tenaganya.
- Dengan bantuan sensor, misalnya sensor jarak, kita dapat membuat Robot Line tracer, Obstacle Avoider, light sensing, dll.
- Program berikut ini adalah untuk menggerakkan Robot maju full speed dan mundur half speed.



Sketch:

```
int transistorPin = 9;    // dari digital pin 9 ke pin signal PWM kiri (menggerakkan motor)
int transistorPin2 = 11; // dari digital pin 11 ke pin signal PWM kanan (menggerakkan motor)
int relayPin = 7;        // dari digital pin 7 ke pin signal relay kiri (membalik putaran motor)
int relayPin2 = 5;       // dari digital pin 5 ke pin signal relay kanan (membalik putaran motor)

void setup() {
// set the transistor pin as an output
pinMode(transistorPin, OUTPUT);
pinMode(transistorPin2, OUTPUT);
// set the relay pin as an output
pinMode(relayPin, OUTPUT);
pinMode(relayPin2, OUTPUT);
}

void loop()
{
digitalWrite(7, LOW); // relay1 off - MAJU
digitalWrite(5, LOW); // relay2 off - MAJU
analogWrite(9, 255); // motor 1 on, full speed - (nilai 0-255)
analogWrite(11, 255); // motor 2 on, full speed - (nilai 0-255)

delay(3000);

analogWrite(9, 0);    // motor 1 off
analogWrite(11, 0);   // motor 2 off

delay(3000);

digitalWrite(7, HIGH); // relay1 on - MUNDUR
digitalWrite(5, HIGH); // relay2 on - MUNDUR
analogWrite(9, 125);   // motor 1 on, half speed - (nilai 0-255)
analogWrite(11, 125);  // motor 2 on, half speed - (nilai 0-255)
delay(3000);

analogWrite(9, 0);    // motor 1 off
analogWrite(11, 0);   // motor 2 off
delay(3000);
}
```