**MODUL ARDUINO**

**BY. YENI FITRIA**

#include <Wire.h>    // Library Wire untuk komunikasi I2C

#include <LiquidCrystal\_I2C.h> // Library LiquidCrystal\_I2C

LiquidCrystal\_I2C lcd(0x27, 16, 2); // Alamat I2C umum untuk modul LCD 16x2

void setup() // Melakukan penulisan perintah proses inisialisasi program (1 kali)

{ // Bagian yang digunakan untuk mengawali void setup

lcd.init(); // Berfungsi untuk inisialisasi LCD

lcd.backlight();  // Nyalakan backlight LCD

lcd.setCursor(0,0); // Posisi kursor di kolom 0, baris 0

lcd.print("Hi, Semuanya!");   // Untuk mengetik text & menampilkan di LCD

lcd.setCursor(0,1); // Penempatan kursor di kolom 0, baris 1

lcd.print("Tugas TIK-Syifa");   // Untuk mengetik text & menampilkan di LCD

delay(3000);  // Menampilkan LCD dan text selama 3 detik

lcd.clear();  // Menghapus layar LCD dan posisi kursor

} // Bagian yang digunakan mengakhiri void setup

void loop() // Bagian IDE dilakukan secara berulang, berurutan

{ // Bagian yang digunakan untuk mengawali void loop

lcd.setCursor(0,0); // Posisi kursor di kolom 0, baris 0

lcd.print("Kali ini, aku"); // Untuk mengetik text & menampilkan di LCD

lcd.setCursor(4,1); // Posisi kursor di kolom 4, baris 1

lcd.print("ada pantun!"); // Untuk mengetik text & menampilkan di LCD

delay(3000); // Menampilkan LCD dan text selama 3 detik

lcd.clear(); // Menghapus layar LCD dan posisi kursor

lcd.setCursor(0,0); // Posisi kursor di kolom 0, baris 0

lcd.print("Yuk, bilang"); // Untuk mengetik text & menampilkan di LCD

lcd.setCursor(6,1); // Posisi kursor di kolom 6, baris 1

lcd.print("cakep, ya!"); // Untuk mengetik text & menampilkan di LCD

delay(3000); // Menampilkan LCD & text selama 3 detik

lcd.clear(); // Menghapus layar LCD dan posisi kursor

lcd.setCursor(0,0); // Posisi kursor di kolom 0, baris 0

lcd.print("Beli baju"); // Untuk mengetik text & menampilkan di LCD

lcd.setCursor(9, 1); // Posisi kursor di kolom 9, baris 1

lcd.print("batik"); // Untuk mengetik text & menampilkan di LCD

delay(3000); // Menampilkan LCD dan text selama 3 detik

lcd.clear(); // Menghapus layar LCD dan posisi kursor

lcd.setCursor(0,0); // Posisi kursor di kolom 0, baris 0

lcd.print ("Ternyata"); // Untuk mengetik text & menampilkan di LCD

lcd.setCursor(2,1); // Posisi kursor di kolom 2, baris 1

lcd.print("bahannya halus"); // Untuk mengetik text & menampilkan di LCD

delay(3000); // Menampilkan LCD dan text selama 3 detik

lcd.clear(); // Menghapus layar LCD dan posisi kursor

lcd.setCursor(0,0); // Posisi kursor di kolom 0, baris 0

lcd.print("Hai Ms. Aina "); // Untuk mengetik text & menampilkan di LCD

lcd.setCursor(0,1); // Posisi kursor di kolom 0, baris 1

lcd.print("yang cantik :D"); // Untuk mengetik text & menampilkan di LCD

delay(3000); // Menampilkan LCD dan text selama 3 detik

lcd.clear(); // Menghapus layar LCD dan posisi kursor

lcd.setCursor(0,0); // Posisi kursor di kolom 0, baris 0

lcd.print("Boleh dong kasih"); // Untuk mengetik text & menampilkan di LCD

lcd.setCursor(0,1); // Posisi kursor di kolom 0, baris 1

lcd.print("nilai bagus <3"); // Untuk mengetik text & menampilkan di LCD

delay(3000); // Menampilkan LCD dan text selama 3 detik

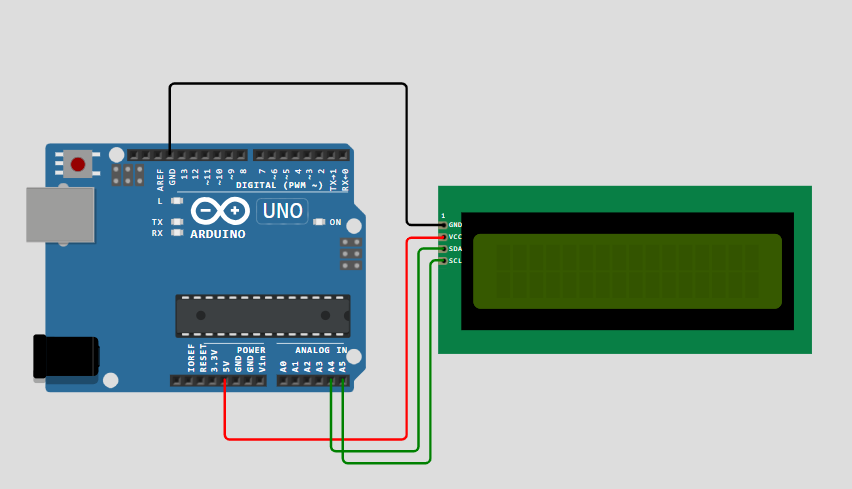
lcd.clear(); // Menghapus layar LCD dan posisi kursor

} // Bagian yang digunakan untuk mengakhiri void loop

// Kesimpulan : projek LCD 16x2 12C yang menggunakan Arduino Uno ini

// -> dapat menampilkan text yang sudah diketik, pada layar LCD

// -> sesuai dengan sketsa, posisi kursor yang sudah teratur dan benar



Void loop()

{

Int i;

For (i=0;i<16;i++)

{

Lcd.ScrollDisplayRight();

Delay(200);

}

}

int tombol = 2;

int led\_hijau = 3;

int led\_kuning = 4;

int led\_merah = 5;

int jumlahTekanan;

void setup(){

  pinMode(tombol, INPUT);

  pinMode(led\_hijau, OUTPUT);

  pinMode(led\_kuning, OUTPUT);

  pinMode(led\_merah, OUTPUT);

}

void loop(){

  int statusTombol = digitalRead(tombol);

  if (statusTombol == HIGH){

    jumlahTekanan ++;

    delay(200);

    if(jumlahTekanan==1){

      digitalWrite(led\_hijau, HIGH);

    }

    else if(jumlahTekanan==2){

      digitalWrite(led\_kuning, HIGH);

    }

    else if(jumlahTekanan==3){

      digitalWrite(led\_merah, HIGH);

    }

    else if(jumlahTekanan==4){

      digitalWrite(led\_hijau, LOW);

      digitalWrite(led\_kuning, LOW);

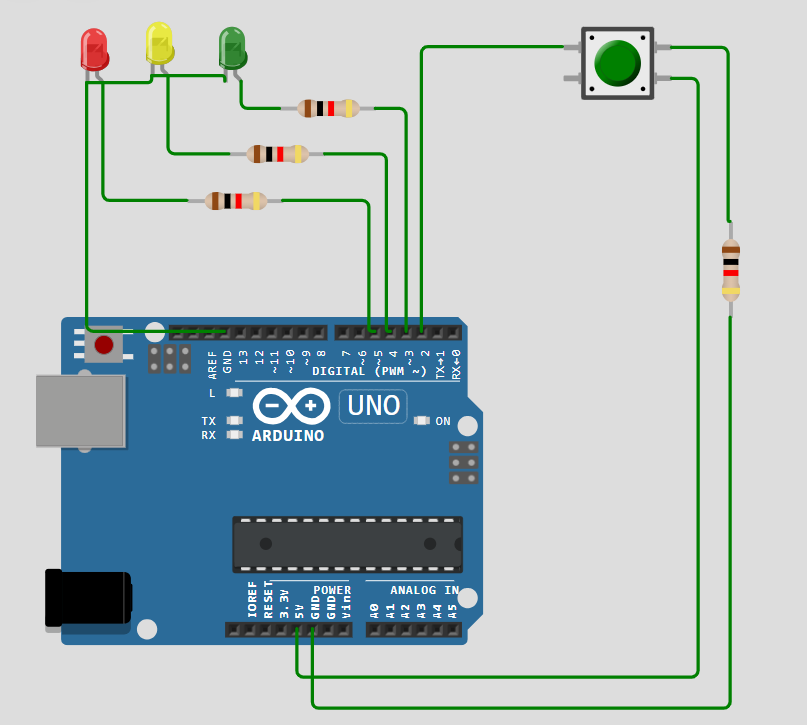
      digitalWrite(led\_merah, LOW);

      jumlahTekanan=0;

    }

  }

}



#include <Wire.h>  // Library Wire digunakan untuk komunikasi I2C

#include <LiquidCrystal\_I2C.h>  // Library LiquidCrystal\_I2C

LiquidCrystal\_I2C lcd(0x27, 16, 2); // Alamat I2C umum untuk modul LCD 16x2

byte Char1[] = {

  B00001,

  B00001,

  B00001,

  B11001,

  B11001,

  B11001,

  B11111,

  B11111

};

byte Char2[] = {

  B11110,

  B11110,

  B10000,

  B10000,

  B10000,

  B10000,

  B10000,

  B11111

};

byte Char3[] = {

  B11111,

  B00001,

  B00001,

  B00001,

  B00001,

  B00001,

  B01111,

  B01111

};

byte Char4[] = {

  B11111,

  B10011,

  B10011,

  B10011,

  B10000,

  B10000,

  B10000,

  B10000

};

void setup() {

  lcd.init();                      // Inisialisasi LCD

  lcd.backlight();

  lcd.createChar(0,Char1);

   lcd.createChar(1,Char2);

    lcd.createChar(2,Char3);

     lcd.createChar(3,Char4);

}

void loop() {

  lcd.setCursor(0,0);

  lcd.write(byte(0));

  lcd.write(byte(1));

  lcd.setCursor(0,1);

  lcd.write(byte(2));

  lcd.write(byte(3));

  lcd.setCursor(2,0); // menulis karakter pada kolom 0 baris 0

  lcd.print("count:"); // Tampilkan karakter pada LCD

  for(int i=0;i<=10;i++){

    lcd.setCursor(8,0); // menulis karakter pada kolom 0 baris 0

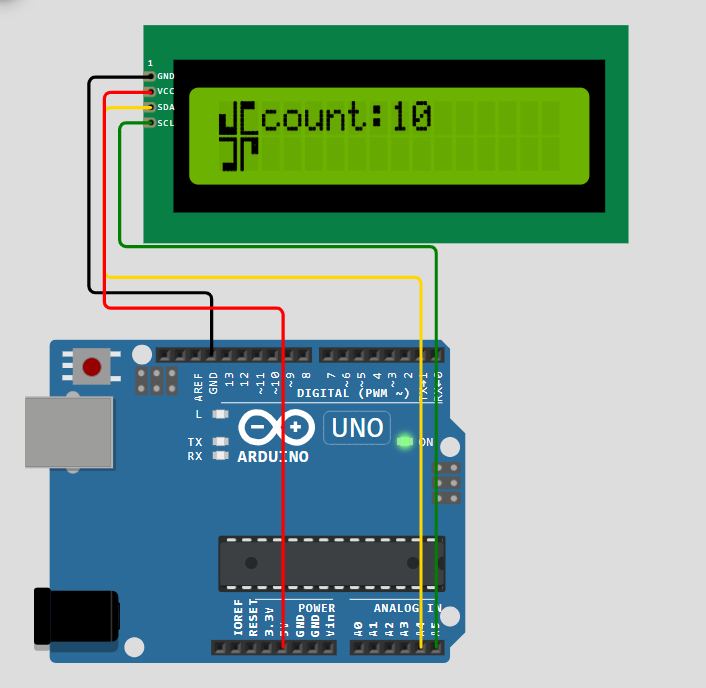
  lcd.print(i);

  delay(1000);

  }

  // Tampilkan karakter pada LCD

}



#define buzzer 12

void setup()

{

pinMode(buzzer, OUTPUT);

}

void loop()

{

// first phase

  //tone(buzzer, 330); delay(155); //fa

  //tone(buzzer, 373); delay(158); //sol

  //tone(buzzer, 438); delay(158); //la

  //tone(buzzer, 373); delay(150); //sol

  //tone(buzzer, 568); delay(290); //re

  //noTone(buzzer);

  //delay(70);

  //tone(buzzer, 560); delay(390); //re

  //tone(buzzer, 500); delay(780); //do

  //noTone(buzzer);

  //delay(10);

// second phase

  //tone(buzzer, 330); delay(140); //fa

  //tone(buzzer, 373); delay(140); //sol

  //tone(buzzer, 425); delay(140); //la

  //tone(buzzer, 319); delay(140); //fa

  //tone(buzzer, 498); delay(220); //do

  //noTone(buzzer);

  //delay(100);

  //tone(buzzer, 489); delay(390); //do

  //tone(buzzer, 445); delay(395); //la

  //tone(buzzer, 430); delay(120); //la

  //tone(buzzer, 380); delay(250); //sol

//third phase

  //tone(buzzer, 330); delay(140); //fa

  //tone(buzzer, 373); delay(140); //sol

  //tone(buzzer, 425); delay(140); //la

  //tone(buzzer, 340); delay(140); //fa

  //tone(buzzer, 450); delay(530); //la

  //tone(buzzer, 510); delay(290); //do

  //tone(buzzer, 426); delay(590); //la

  //tone(buzzer, 340); delay(390); //sol

  //noTone(buzzer);

  //delay(70);

  //tone(buzzer, 350); delay(300); //fa

  //tone(buzzer, 515); delay(500); //fa

  //tone(buzzer, 456); delay(1000); //la

// fourth phase

  //tone(buzzer, 330); delay(155); //fa

  //tone(buzzer, 373); delay(158); //sol

  //tone(buzzer, 438); delay(158); //la

  //tone(buzzer, 373); delay(150); //sol

  //tone(buzzer, 568); delay(390); //re

  //noTone(buzzer);

  //delay(70);

  //tone(buzzer, 560); delay(390); //re

  //tone(buzzer, 500); delay(780); //do

// fifth phase

  //tone(buzzer, 330); delay(155); //fa

  //tone(buzzer, 363); delay(158); //sol

  //tone(buzzer, 448); delay(158); //la

  //tone(buzzer, 383); delay(150); //fa

  //tone(buzzer, 678); delay(700); //fa

  //tone(buzzer, 429); delay(340); //la

  //tone(buzzer, 455); delay(450); //la

  tone(buzzer, 434); delay(150); //la

  tone(buzzer, 380); delay(350); //sol

// sixth phase

  tone(buzzer, 386); delay(140); //fa

  tone(buzzer, 433); delay(140); //sol

  tone(buzzer, 488); delay(140); //la

  tone(buzzer, 399); delay(140); //fa

  tone(buzzer, 535); delay(530); //la

  tone(buzzer, 590); delay(290); //do

  tone(buzzer, 486); delay(590); //la

  tone(buzzer, 390); delay(390); //sol

  noTone(buzzer);

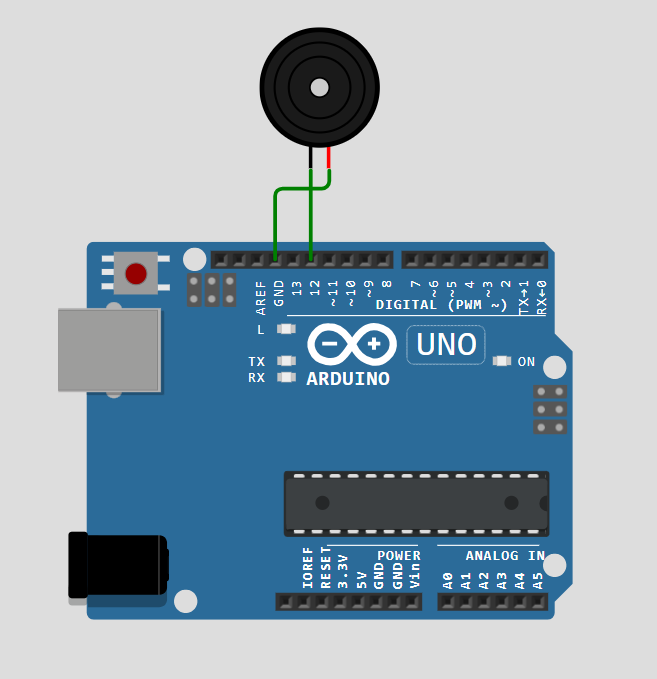
  delay(70);

  tone(buzzer, 385); delay(300); //fa

  tone(buzzer, 578); delay(500); //fa

  tone(buzzer, 516); delay(1000); //la

}



#define ECHO\_PIN 5

#define TRIG\_PIN 4

void setup() {

**Serial**.begin(115200);

  pinMode(LED\_BUILTIN, OUTPUT);

  pinMode(TRIG\_PIN, OUTPUT);

  pinMode(ECHO\_PIN, INPUT);

  pinMode(6, OUTPUT);

  pinMode(7, OUTPUT);

  pinMode(8, OUTPUT);

}

float readDistanceCM() {

  digitalWrite(TRIG\_PIN, LOW);

  delayMicroseconds(2);

  digitalWrite(TRIG\_PIN, HIGH);

  delayMicroseconds(10);

  digitalWrite(ECHO\_PIN, LOW);

  int duration = pulseIn(ECHO\_PIN,HIGH);

  return duration \* 0.034 / 2;

}

void loop() {

  digitalWrite(TRIG\_PIN, LOW);

  delayMicroseconds(2);

  digitalWrite(TRIG\_PIN, HIGH);

  delayMicroseconds(10);

  digitalWrite(TRIG\_PIN, LOW);

  int duration = pulseIn(ECHO\_PIN, HIGH);

  float distance = duration \* 0.034 / 2;

**Serial**.print("Jarak: ");

**Serial**.print(distance) ;

if(distance >100){

  digitalWrite(6, HIGH);

  digitalWrite(7, LOW);

  digitalWrite(8, LOW);

**Serial**.println("Aman");

} else if (distance <100 && distance >=50){

  digitalWrite(6, LOW);

  digitalWrite(7, HIGH);

  digitalWrite(8, LOW);

**Serial**.println("Siaga");

} else if (distance <50) {

  digitalWrite(6, LOW);

  digitalWrite(7, LOW);

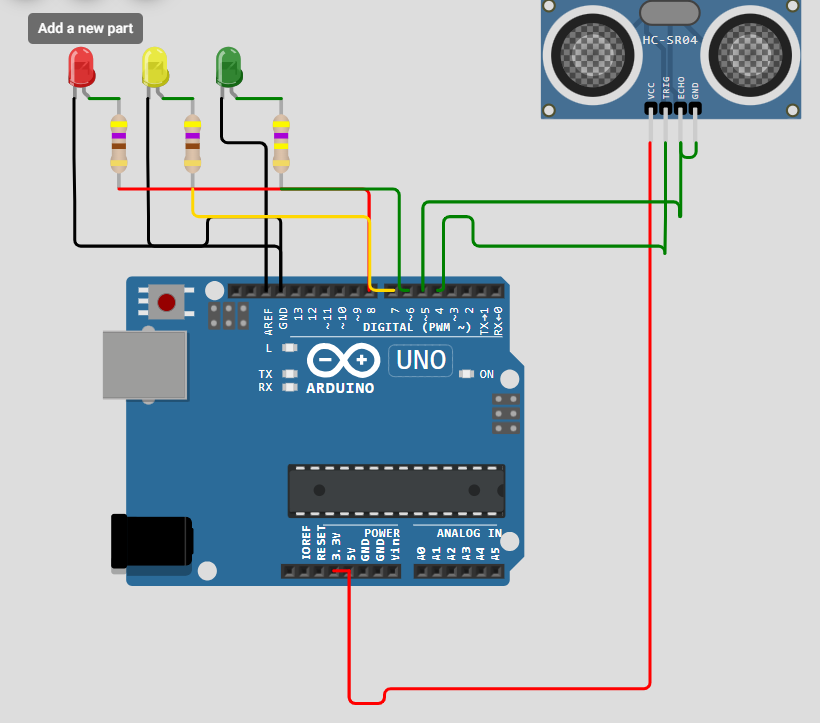
  digitalWrite(8, HIGH);

**Serial**.println("Banjir");

}

delay(100);

}



int BUTTON = 12;

int BUZZER = 6;

#define NOTE\_C4 262 //DO

#define NOTE\_D4 294 //RE

#define NOTE\_E4 330 //MI

#define NOTE\_F4 349 //FA

#define NOTE\_G4 392 //SO

#define NOTE\_A4 440 //LA

#define NOTE\_B4 494 //SI

#define NOTE\_C5 523 //DO (higher octave)

// Speaker is on pin 6

const int pinSpeaker = 6;

void setup(){

  pinMode(pinSpeaker, OUTPUT);

  pinMode(BUTTON, INPUT\_PULLUP); // Pull-up resistor enabled for button

  pinMode(BUZZER, OUTPUT);

}

// Function to play "Pelangi Pelangi"

void playPelangiMelody() {

  // First line: "Pelangi pelangi"

  tone(pinSpeaker, NOTE\_G4, 500);  // Pe

  delay(500);

  tone(pinSpeaker, NOTE\_A4, 500);  // lan

  delay(500);

  tone(pinSpeaker, NOTE\_G4, 500);  // gi

  delay(500);

  tone(pinSpeaker, NOTE\_E4, 750);  // pe

  delay(750);

  tone(pinSpeaker, NOTE\_G4, 500);  // lan

  delay(500);

  tone(pinSpeaker, NOTE\_F4, 750);  // gi

  delay(750);

  // Second line: "Alangkah indahmu"

  tone(pinSpeaker, NOTE\_E4, 500);  // A

  delay(500);

  tone(pinSpeaker, NOTE\_F4, 500);  // lan

  delay(500);

  tone(pinSpeaker, NOTE\_G4, 500);  // kah

  delay(500);

  tone(pinSpeaker, NOTE\_A4, 750);  // in

  delay(750);

  tone(pinSpeaker, NOTE\_C5, 500);  // dah

  delay(500);

  tone(pinSpeaker, NOTE\_A4, 750);  // mu

  delay(750);

  // Third line: "Pelangi pelangi"

  tone(pinSpeaker, NOTE\_G4, 500);  // Pe

  delay(500);

  tone(pinSpeaker, NOTE\_A4, 500);  // lan

  delay(500);

  tone(pinSpeaker, NOTE\_G4, 500);  // gi

  delay(500);

  tone(pinSpeaker, NOTE\_E4, 750);  // pe

  delay(750);

  tone(pinSpeaker, NOTE\_G4, 500);  // lan

  delay(500);

  tone(pinSpeaker, NOTE\_F4, 750);  // gi

  delay(750);

  // Final line: "Ciptaan Tuhan"

  tone(pinSpeaker, NOTE\_D4, 500);  // Cip

  delay(500);

  tone(pinSpeaker, NOTE\_E4, 500);  // ta

  delay(500);

  tone(pinSpeaker, NOTE\_F4, 500);  // an

  delay(500);

  tone(pinSpeaker, NOTE\_G4, 750);  // Tu

  delay(750);

  tone(pinSpeaker, NOTE\_C5, 1000); // han

  delay(1000);

}

void loop(){

  if (digitalRead(BUTTON) == LOW) {

    // Play the Pelangi Pelangi melody when button is pressed

    playPelangiMelody();

    // Turn on buzzer

    digitalWrite(BUZZER, HIGH);

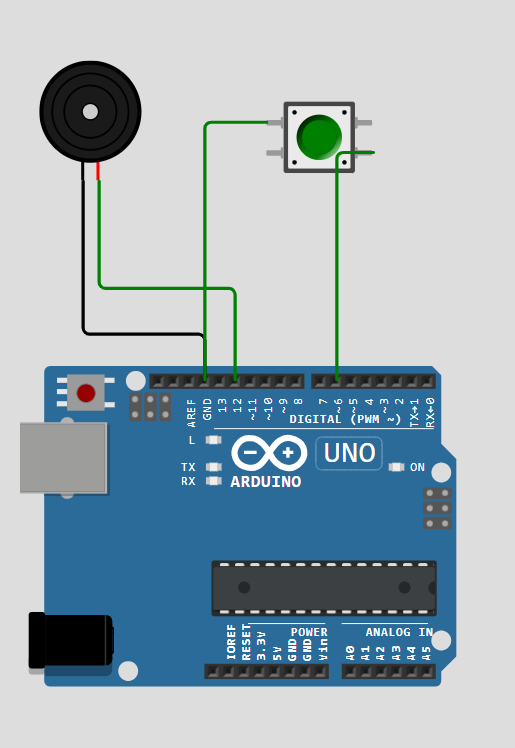
  } else {

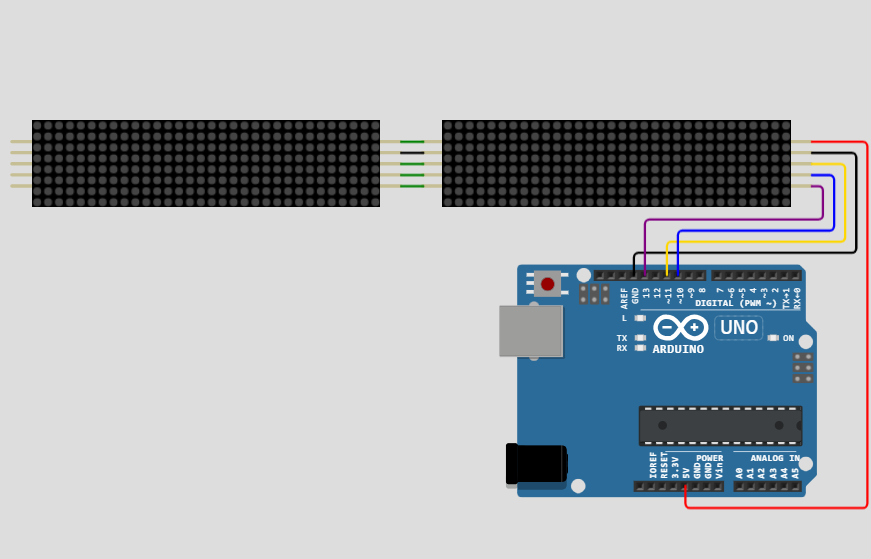
    // Turn off buzzer when button is not pressed

    digitalWrite(BUZZER, LOW);

  }

}





#include <MD\_Parola.h>

#include <MD\_MAX72xx.h>

#include <SPI.h>

// Define hardware type, size, and output pins:

#define HARDWARE\_TYPE MD\_MAX72XX::PAROLA\_HW

#define MAX\_DEVICES 8

#define CS\_PIN 10

#define DATA\_PIN 11

#define CLK\_PIN 13

// Create a new instance of the MD\_Parola class with hardware SPI connection:

//MD\_Parola myDisplay = MD\_Parola(HARDWARE\_TYPE, CS\_PIN, MAX\_DEVICES);

// Setup for software SPI:

MD\_Parola myDisplay = MD\_Parola(HARDWARE\_TYPE, DATA\_PIN, CLK\_PIN, CS\_PIN, MAX\_DEVICES);

void setup() {

  // Intialize the object:

  myDisplay.begin();

  // Set the intensity (brightness) of the display (0-15):

  myDisplay.setIntensity(0);

  // Clear the display:

  myDisplay.displayClear();

  myDisplay.displayText("HELLO", PA\_CENTER, 100, 0, PA\_SCROLL\_LEFT, PA\_SCROLL\_LEFT); //(Text, Letak, speed)

}

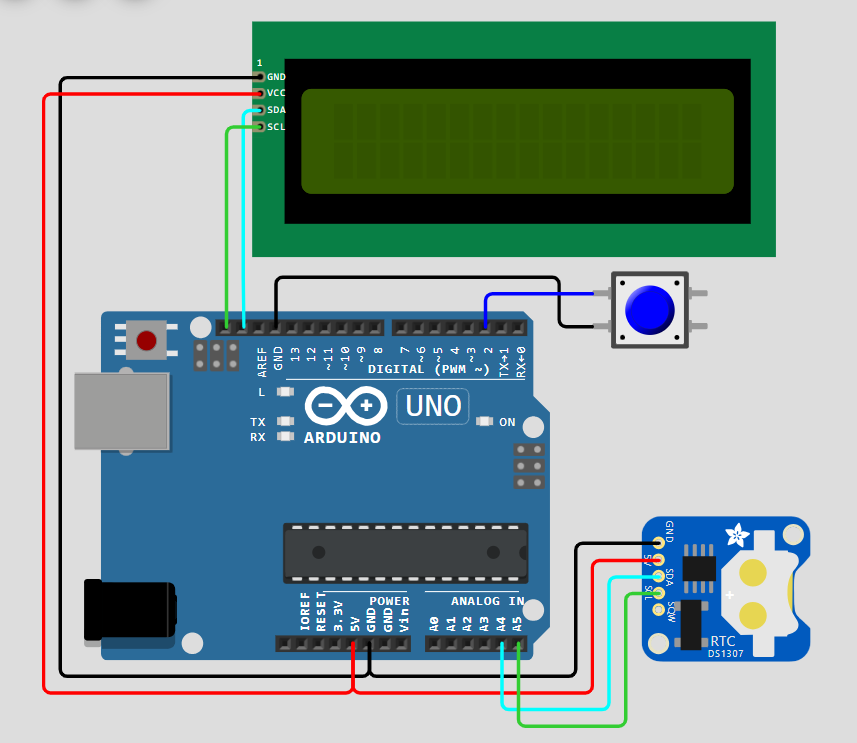
void loop() {

  if (myDisplay.displayAnimate()) {

    myDisplay.displayReset();

  }

}



#include <RTClib.h>

#include <Wire.h>

#include <LiquidCrystal\_I2C.h>

const int buttonPin = 2;  // Pin pushbutton

int buttonState = 0;      // Variabel untuk menyimpan status tombol

bool lcdOn = true;        // Status LCD (hidup atau mati)

RTC\_DS1307 RTC;

LiquidCrystal\_I2C lcd(0x27, 16, 2);

void setup() {

  // Inisialisasi Serial, RTC, dan LCD

**Serial**.begin(9600);

  RTC.begin();

  lcd.init();

  lcd.backlight();

  pinMode(buttonPin, INPUT\_PULLUP);  // Mengatur pin pushbutton sebagai input dengan pull-up resistor

  // Cek ketersediaan RTC

  if (!RTC.begin()) {

**Serial**.println("Couldn't find RTC");

    while (1);

  }

  // Atur waktu RTC secara manual jika tidak berjalan

  if (!RTC.isrunning()) {

**Serial**.println("RTC is NOT running!");

    RTC.adjust(DateTime(F(\_\_DATE\_\_), F(\_\_TIME\_\_)));

  }

}

void loop() {

  buttonState = digitalRead(buttonPin);

  if (buttonState == LOW) {

    // Tombol ditekan, ubah status LCD

    lcdOn = !lcdOn;

    delay(500);  // Memberikan waktu debouncing

    if (lcdOn) {

      lcd.backlight();  // Hidupkan LCD jika status LCD On

    } else {

      lcd.noBacklight();  // Matikan LCD jika status LCD Off

      lcd.clear();        // Bersihkan tampilan LCD

    }

  }

  if (lcdOn) {

    // Menampilkan tanggal dan waktu pada LCD

    DateTime now = RTC.now();

    // LCD

    lcd.setCursor(0, 0);

    lcd.print("Date: ");

    lcd.print(now.day());

    lcd.print("/");

    lcd.print(now.month());

    lcd.print("/");

    lcd.print(now.year());

    lcd.setCursor(0, 1);

    lcd.print("Time: ");

    lcd.print(now.hour());

    lcd.print(":");

    if (now.minute() < 10) {  // Mengatur angka tetap 2 digit ketika angka kurang dari 10

      lcd.print("0"); }

    lcd.print(now.minute());

    lcd.print(":");

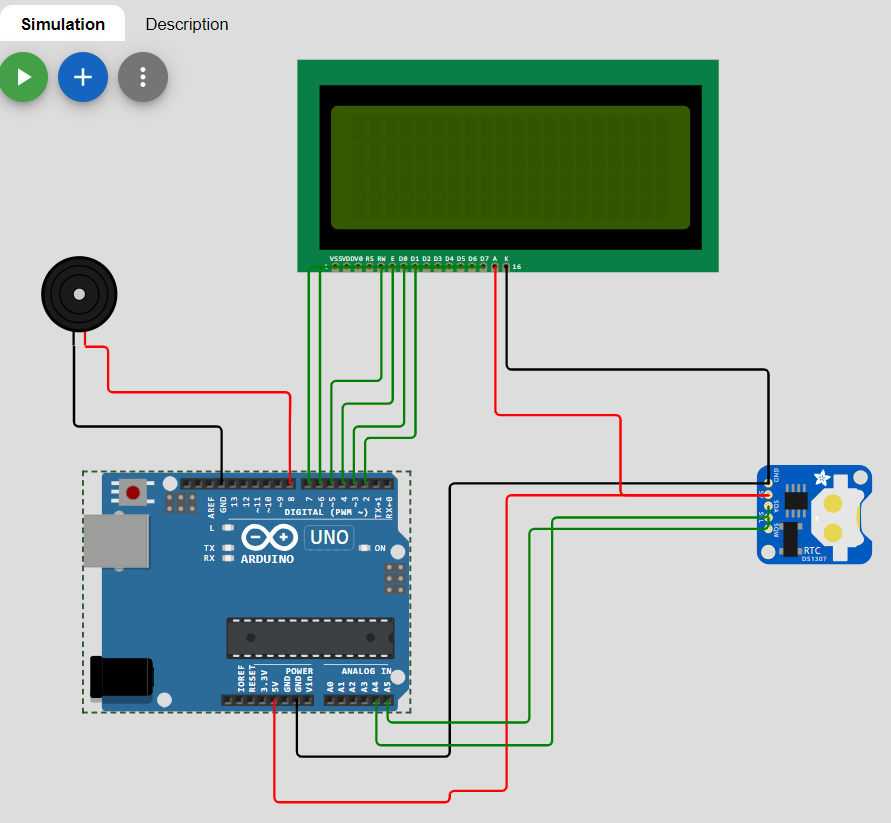
    if (now.second() < 10) {  // Mengatur angka tetap 2 digit ketika angka kurang dari 10

      lcd.print("0"); }

    lcd.print(now.second());

  }

}



#include <Wire.h>

#include "RTClib.h"

#include <LiquidCrystal.h>

LiquidCrystal lcd(7, 6, 5, 4, 3, 2);

RTC\_DS3231 rtc;

char nameDay[7][12] = {"Ahad", "Senin", "Selasa", "Rabu", "Kamis", "Jumat", "Sabtu"};

void setup()

{

 pinMode(10, OUTPUT);

 lcd.begin(20, 4);

 rtc.begin();

 if (rtc.lostPower()) {

 rtc.adjust(DateTime(F(\_\_DATE\_\_), F(\_\_TIME\_\_)));

 }

lcd.setCursor(0,0);

lcd.print("   JAM DIGITAL   ");

lcd.setCursor(0,1);

lcd.print("   LAB LISTRIK   ");

delay(1000);

lcd.clear();

}

 void loop()

 {

 DateTime now = rtc.now();

 lcd.setCursor(2,0);

 lcd.print (nameDay[now.dayOfTheWeek()]);

 lcd.print (",");

 printAngka (now.day());

 lcd.print (",");

 printAngka (now.month());

 lcd.print (",");

 printAngka (now.year());

 lcd.setCursor(5,1);

 printAngka (now.hour());

 lcd.print (",");

 printAngka (now.minute());

 lcd.print (",");

 printAngka (now.second());tone(8,1000,50);

 lcd.setCursor (2,3);

 lcd.print (" SMKN 2 SURAKARTA ");

 delay(250);

 }

 void printAngka (int digits){

  if (digits < 10){

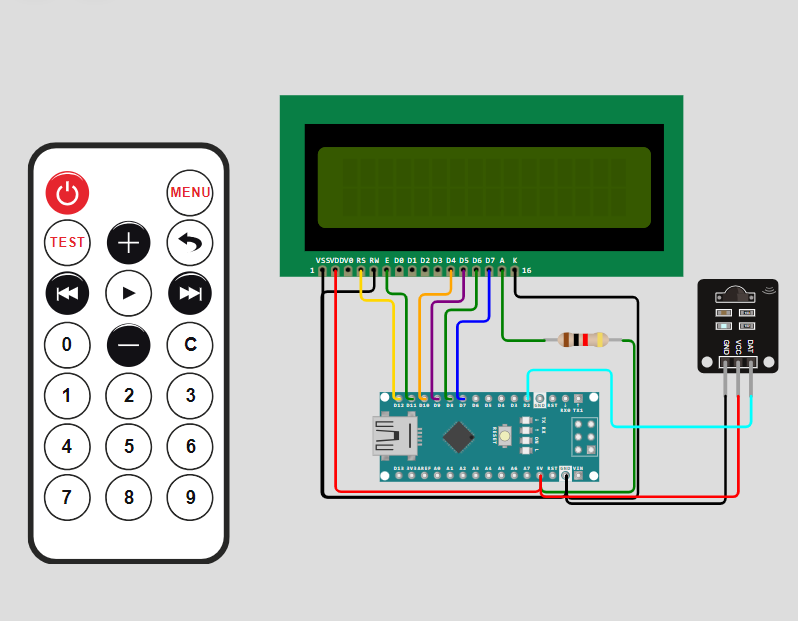
    lcd.print("0");

    lcd.print(digits);

 }

 else lcd.print(digits);

 }



#include <IRremote.h> // memanggil library IRremote

#include <LiquidCrystal.h> // memanggil library LiquidCrystal

// Melakukan pendeklarasian variabel

int Receiver = 2;

int A = 0;

int B = 0;

int Result;

int nilai;

IRrecv irrecv(Receiver);

decode\_results results;

LiquidCrystal lcd(12, 11, 10, 9, 8, 7);

void setup() {

**Serial**.begin(9600); // berkomunikasi dengan PC atau komputer

  irrecv.enableIRIn();

}

void loop() {

  if (irrecv.decode()) {

    lcd.clear();

    button();

    if (A == 0) {

      A = nilai;

    } else {

      B = nilai;

    }

    lcd.setCursor(0, 0);

    lcd.print(A);

    lcd.print("+");

    lcd.print(B);

    lcd.print("=");

    lcd.setCursor(5, 1);

    if (A > 0 && B > 0) {

      Result = A + B;

    }

**Serial**.print("Hasil A:"); // mencetak pada serial monitor

**Serial**.println(A);

**Serial**.print("Hasil B:");

**Serial**.println(B);

**Serial**.print("Hasil :");

**Serial**.println(Result);

    lcd.print(Result);

    irrecv.resume();

  }

}

// fungsi untuk inputan remote

void button() {

  int inputan = irrecv.decodedIRData.command;

  if (inputan == 48) {

    nilai = 1;

  }

  if (inputan == 24) {

    nilai = 2;

  }

  if (inputan == 122) {

    nilai = 3;

  }

  if (inputan == 16) {

    nilai = 4;

  }

  if (inputan == 56) {

    nilai = 5;

  }

  if (inputan == 90) {

    nilai = 6;

  }

  if (inputan == 66) {

    nilai = 7;

  }

  if (inputan == 74) {

    nilai = 8;

  }

  if (inputan == 82) {

    nilai = 9;

  }

  if (inputan == 82) {

    nilai = 0;

  }

  if (inputan == 34) {

    Result = 0;

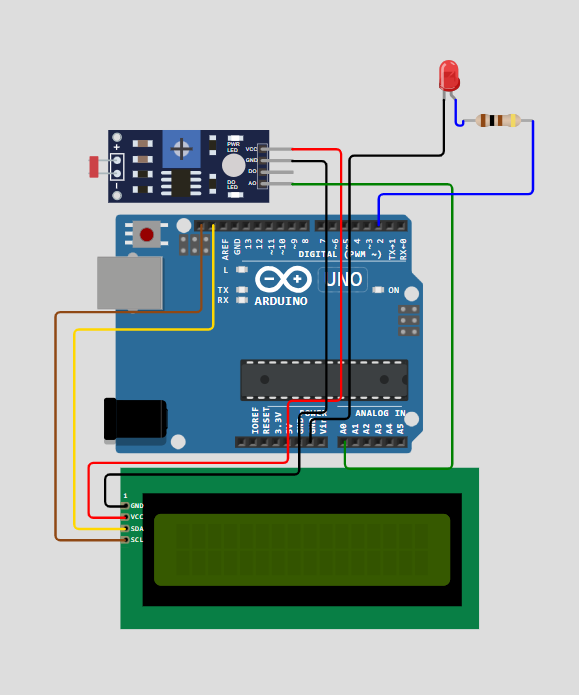
    A = 0;

    B = 0;

    nilai = 0;

  }

}



#include <LiquidCrystal\_I2C.h> //memanggir libabry LiquidCrystal\_I2C ke dalam program

int ldrPin = A0; //deklarasi pin ldr dengan nama variabel ldrPin yang dihubungkan ke pin A0

int ledPin = 2; //deklarasi pin led dengan nama ledPin yang dihubungkan ke pin D2

const float GAMMA = 0.7; //nilai ketetapan gama untuk mencari tingkat kecerahan(lux)

const float RL10 = 50; //nilai ketetapan gama untuk mencari tingkat kecerahan(lux)

LiquidCrystal\_I2C lcd(0x27, 16, 2); //deklarasi alamat, dan jenis lcd yang digunakan

void setup() {

  // put your setup code here, to run once:

  pinMode(ledPin, OUTPUT); //inisialisai ledPin sebagai pin output

  lcd.init(); //perintah menyalakan lcd

  lcd.backlight(); //perintah menyalakan backlight dari lcd

}

void loop() {

  // put your main code here, to run repeatedly:

  int ldrValue = analogRead(ldrPin); //membaca nilai analog pada pin A0 dari sensor ldr

  float voltase = ldrValue / 1024. \* 5; //rumus mengubah nilai sinyal analog dari ldr menjadi nilai voltase

  float resistansi = 2000 \* voltase / (1-voltase / 5); //rumus mencari nilai resistansi dari nilai voltase sebelumnya

  float kecerahan = pow(RL10 \* 1e3 \* pow(10, GAMMA) / resistansi, (1/GAMMA)); //rumus mencari nilai tingkat kecerahan (lux)

  lcd.setCursor(0, 0); //set posisi karakter di lcd pada kolom 0, baris 0

  lcd.print("Cahaya: "); //menampilkan tulisan Cahaya: ke lcd

  if(kecerahan > 50){ //mengecek nilai dari kecerahan, bila lebih dari 50 skrip di bawah if akan dieksekusi

    lcd.print("Terang"); //menampilkan keterangan terang ke dalam lcd

    digitalWrite(ledPin, LOW); //memamtikan led ketika cahaya terang

  }

  else{ //jika kondisi kecerahan kurang dari 50 skrip akan dieksekusi

    lcd.print("Gelap "); //menampilkan tulisan gelap ke lcd

    digitalWrite(ledPin, HIGH); //menyalakan led karena suasana gelap

  }

  lcd.setCursor(0, 1); //set posisi karakter di lcd pada kolom 0, baris 1

  lcd.print("Lux: ");

  lcd.print(kecerahan);

  lcd.print("     ");

  delay(100);

}

**// Prototype Tempat Otomatis Berbasis Arduino**

**// Sensor Ultrasonic HC-SR04 dan Servo**

**// Dibuat ole : M.Nofgi Y.P.U.**

**// www.nofgipiston.wordpress.com**

**// Tokopedia/Bukalapak : Nofgi Piston Shop**

**// Mohon dipelajari program yang diberikan agar mendapat ilmu yg bermanfaat :)**

**// menambahkan lubrary Servo**

**#include <Servo.h>**

**// inialisasi variabel untuk servo**

**Servo servo1;**

**// inialisasi pin ultrasonic**

**const int pinTRIGGER = 8;**

**const int pinECHO = 9;**

**// inialisasi variabel pembacaan sensor ultrasonic**

**// gunakan "float" untuk angka dengan koma**

**// atau gunakan "int" untuk angka tanpa koma**

**float durasi, jarak;**

**// ======================= program pengaturan awal ====================== //**

**void setup()**

**{**

**// inialisasi baud rate serial monitor**

**Serial.begin(9600);**

**// inialisasi pin servo**

**servo1.attach(10);**

**// menulis derajat awal servo (menutup)**

**servo1.write(0);**

**// inialisasi stauts I/O pin**

**pinMode(pinTRIGGER, OUTPUT);**

**pinMode(pinECHO, INPUT);**

**delay(1000);**

**}**

**// ============================== program utama ============================= //**

**void loop()**

**{**

**//pembacaan sesnor ulrasonic**

**//Mengaktifak pin Trigger selama 10mikrodetik**

**//Lalu dia non aktif**

**digitalWrite(pinTRIGGER, HIGH);**

**delayMicroseconds(10);**

**digitalWrite(pinTRIGGER, LOW);**

**//Mengaktifkan pin Echo yang menerima pantulan gelombang ultrasonik**

**//pulseIn = Lama durasi pin Echo menerima pantulan gelombang**

**//Menghitung rumus jarak(Lama durasi dibagi dua)**

**durasi = pulseIn(pinECHO, HIGH);**

**jarak = ((durasi \* 0.034) / 2);**

**// menulis data pembacaan ultrasonic pada serial monitor**

**// jadi kita tahu jarak obejak yang terbaca**

**Serial.print("Jarak: ");**

**Serial.println(jarak);**

**// program buka tutup servo otomatis**

**// jika jarak objek terbaca <= 7 cm**

**// ubah dan sesuaikan jarak objek saat buka di sini, satuan cm**

**if (jarak <= 7)**

**{**

**// servo membuka, mnuju derajat 90**

**// ubah dan asesuaikan derajat buka di sini**

**servo1.write(90);**

**// delay membuka servo**

**// di sini disetting 3 detik membukanya**

**// ubah durasi buka di sini**

**delay(3000);**

**// servo menutup kembali, menuju derajat 0**

**// ubah dan sesuaikan derajat tutup di sini**

**servo1.write(0);**

**}**

**// jika jarak objek terbaca lebih dari pernyataan di atas**

**else**

**{**

**// servo menutup, menuju derajat 0**

**// ubah dan sesuaikan derajat tutup di sini**

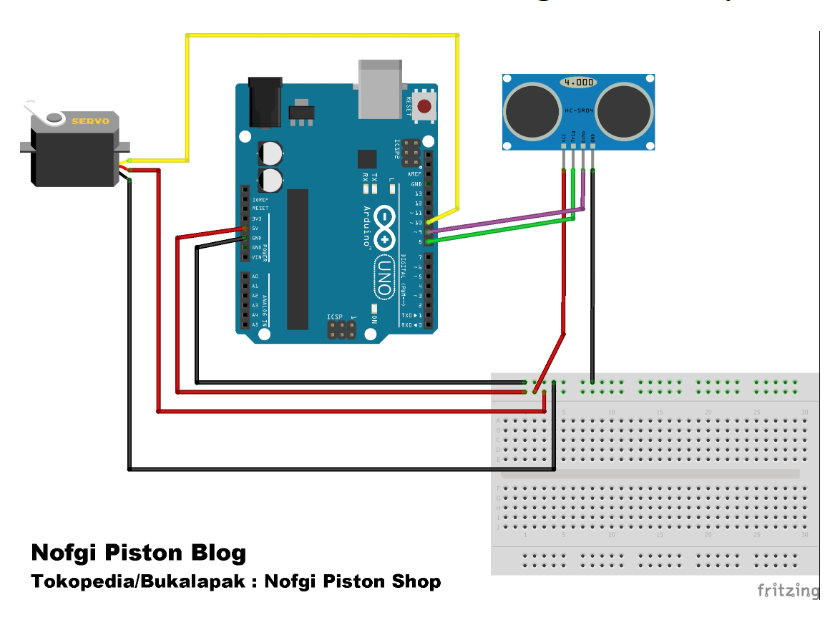
**servo1.write(0);**

**}**

**// delay program**

**delay(100);**

**}**



**Keterangan Rangkaian :**

* **Bisa Menggunakan Segala Jenis Arduino (UNO, NANO, MEGA, MINI, dll), karena penomoran pinnya sama. Tinggal cocokkan Saja nomor Pin-Pin nya.**
* **Kabel Data Untuk Servo (Biasanya Berwarna Orange/kuning) ===> pin 10 Arduino**
* **Pin Trigger Sensor Ultrasonic HC-SR04 ===> pin 8 Arduino**
* **Pin Echo Sensor Ultrasonic HC-SR04 ===> pin 9 Arduino**
* **Kabel VCC Untuk Servo (Biasanya Berwarna Merah) ===> 5V**
* **Kaki vcc pada Sensor hubungkan ke sumber power positif 5v (bisa dari 5v Arduino)**
* **Kaki gnd/ground atau kabel hitam pada masing-masing komponen hubungkan ke sumber negatif power/ground (dihubungkan ke pin gnd Arduino)**

//Bel Sederhana menggunakan arduino

void setup() {

  pinMode (8, INPUT);

  pinMode(5, OUTPUT);

}

void loop() {

  int Switch\_State=digitalRead(8);

  if(Switch\_State==HIGH){

    digitalWrite(5, HIGH);

    tone(5,1000);

  }

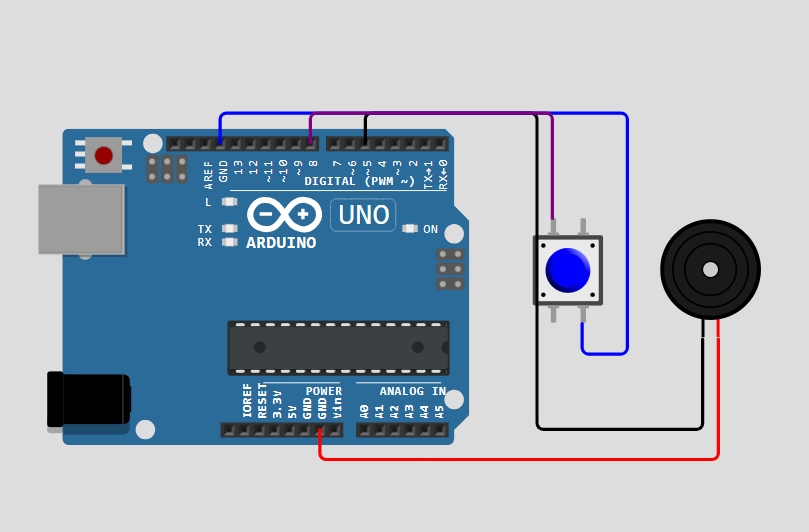
  else{

    digitalWrite(5,LOW);

    noTone(5);

  }

}



Bahan yang harus disediakan:

1x Arduino Uno : http://jogjarobotika.com/arduino-uno-r3-compatible-cable-data

2x TCRT5000 : http://jogjarobotika.com/sensor-garis-tcrt5000-infrared-module

1x L298N Driver Motor : http://jogjarobotika.com/l298n-driver-modul

1x Sasis Kit 2WD : http://jogjarobotika.com/2wd-sasis-kit

2xSpacer 3cm :http://jogjarobotika.com/spacer-kuningan-30mm6mm

1x Kabel Jumper Male-Female http://jogjarobotika.com/kabel-jumper-male-female-40pc

Mur baut 3mm, Jack DC Male

Setelah bahan ready maka tahap selanjutnya adalah merakitnya, namun sebelum itu solderlah motor dc, saklar, dan jack dc terlebih dahulu.

Penyolderan Saklar dilakukan dengan cara membagi jumper male-female menjadi dua. bagian female disolder ke bagian 0 (mati) dan bagian male disolder ke bagian - (Hidup). Sedangkan Jack DC disolder dengan cara membuang bagian female dari kabel jumper, perlu dicatat warna dan polaritasnya! Begitupun dengan penyolderan Motor DC..

Sudah?? Lanjut rakit sasis kitnyaa...

Saya rasa perakitan sasis nggk perlu dijelaskan :P

Oke lanjut ke pengkabelan..

