LAMPIRAN A
Program

```
#include <PS2X_lib.h>  //for v1.6
#define pwm1 2 // belakang kiri niup
#define pwm2 3 // belakang kiri nyedot
#define pwm5 5 // depan kiri niup
#define pwm6 7 // depan kiri nyedot
#define pwm3 8 // belakang kanan niup
#define pwm4 10 // belakang kanan nyedot
#define pwm7 12 // kanan depan niup
#define pwm8 13 // kanan depan nyedot
const int relay1 = 48; // Kamera
const int relay2 = 49; // Senter
#define B 47 // BUZZER
float Tegangan;
float BacaTegangan;
/*
  right now, the library does NOT support hot pluggable controllers, meaning
  you must always either restart your Arduino after you conect the controller,
  or call config_gamepad(pins) again after connecting the controller.
*/
int relayON = LOW;
int relayOFF = HIGH;
// create PS2 Controller Class
PS2X ps2x;
int error = 0;
byte type = 0;
byte vibrate = 0;

void setup()
{
  Serial.begin(57600);
  // CHANGES for v1.6 HERE!!! *****PAY ATTENTION****
  // setup pins and settings:  GamePad(sck52, miso,ss, mosi, Pressures?, Rumble?) check for error SS53 CLK52 MISO51 MOSI50
  error = ps2x.config_gamepad(52,51,53,50, true, true);
```

```
pinMode(pwm1, OUTPUT);
pinMode(pwm2, OUTPUT);
pinMode(pwm3, OUTPUT);
pinMode(pwm4, OUTPUT);
pinMode(pwm5, OUTPUT);
pinMode(pwm6, OUTPUT);
pinMode(pwm7, OUTPUT);
pinMode(pwm8, OUTPUT);
pinMode (relay1,OUTPUT);
pinMode (relay2,OUTPUT);
digitalWrite(relay1, relayOFF);
digitalWrite(relay2, relayOFF);
pinMode (B,OUTPUT);
  if(error == 0)
  {
    Serial.println("Found Controller, configured successful");
    Serial.println("Try out all the buttons, X will vibrate the controller,
faster as you press harder;");
    Serial.println("holding L1 or R1 will print out the analog stick
values.");
    Serial.println("Go to www.billporter.info for updates and to report
bugs.");
  }
  else if(error == 1)
    Serial.println("No controller found, check wiring, see readme.txt to
enable debug. visit www.billporter.info for troubleshooting tips");
  else if(error == 2)
    Serial.println("Controller found but not accepting commands. see
readme.txt to enable debug. Visit www.billporter.info for
troubleshooting tips");
  else if(error == 3)
    Serial.println("Controller refusing to enter Pressures mode, may not
support it. ");

  // Serial.print(ps2x.Analog(1), HEX);
  type = ps2x.readType();
  switch(type)
  {
    case 0:
```

```
      Serial.println("Unknown Controller type");
      break;
    case 1:
      Serial.println("DualShock Controller Found");
      break;
    case 2:
      Serial.println("GuitarHero Controller Found");
      break;
  }
}

void loop()
{
  /*
    You must Read Gamepad to get new values
    Read GamePad and set vibration values
    ps2x.read_gamepad(small motor on/off, larger motor strenght from 0-
255)
    if you don't enable the rumble, use ps2x.read_gamepad(); with no
values
    you should call this at least once a second
  */
  BacaTegangan=analogRead(0);
Tegangan=((BacaTegangan*0.00489)*5);
Serial.print(Tegangan);
Serial.println("V");
if (Tegangan > 11){
  digitalWrite(B,LOW);
  }
if (Tegangan < 11){
  digitalWrite(B,HIGH);
  }
  // skip loop if no controller found
  if(error == 1)
    return;

  if(type == 2)
  {
    // Guitar Hero Controller
```

```
  // read controller
  ps2x.read_gamepad();

  if(ps2x.ButtonPressed(GREEN_FRET))
    Serial.println("Green Fret Pressed");
  if(ps2x.ButtonPressed(RED_FRET))
    Serial.println("Red Fret Pressed");
  if(ps2x.ButtonPressed(YELLOW_FRET))
    Serial.println("Yellow Fret Pressed");
  if(ps2x.ButtonPressed(BLUE_FRET))
    Serial.println("Blue Fret Pressed");
  if(ps2x.ButtonPressed(ORANGE_FRET))
    Serial.println("Orange Fret Pressed");

  if(ps2x.ButtonPressed(STAR_POWER))
    Serial.println("Star Power Command");

  // will be TRUE as long as button is pressed
  if(ps2x.Button(UP_STRUM))
    Serial.println("Up Strum");
  if(ps2x.Button(DOWN_STRUM))
    Serial.println("DOWN Strum");

  // will be TRUE as long as button is pressed
  if(ps2x.Button(PSB_START))
    Serial.println("Start is being held");
  if(ps2x.Button(PSB_SELECT))
    Serial.println("Select is being held");

  // print stick value IF TRUE
  if(ps2x.Button(ORANGE_FRET))
  {
    Serial.print("Wammy Bar Position:");
    Serial.println(ps2x.Analog(WHAMMY_BAR), DEC);
  }
}
else
{
  // DualShock Controller
```

```
  // read controller and set large motor to spin at 'vibrate' speed
  ps2x.read_gamepad(false, vibrate);

  // will be TRUE as long as button is pressed
  if(ps2x.Button(PSB_START))
    Serial.println("Start is being held");
  if(ps2x.Button(PSB_SELECT))
    Serial.println("Select is being held");

  // will be TRUE as long as button is pressed
  if(ps2x.Button(PSB_PAD_UP))
  {
    Serial.print("Up held this hard: ");
    Serial.println(ps2x.Analog(PSAB_PAD_UP), DEC);
 }
  }
  if(ps2x.Button(PSB_PAD_RIGHT))
  {
    Serial.print("Right held this hard: ");
    Serial.println(ps2x.Analog(PSAB_PAD_RIGHT), DEC);
  }
  if(ps2x.Button(PSB_PAD_LEFT))
  {
    Serial.print("LEFT held this hard: ");
    Serial.println(ps2x.Analog(PSAB_PAD_LEFT), DEC);
  }
  if(ps2x.Button(PSB_PAD_DOWN))
  {
    Serial.print("DOWN held this hard: ");
    Serial.println(ps2x.Analog(PSAB_PAD_DOWN), DEC);
  }

  // this will set the large motor vibrate speed based on how hard you
press the blue (X) button
  vibrate = ps2x.Analog(PSAB_BLUE);

  // will be TRUE if any button changes state (on to off, or off to on)
  if (ps2x.NewButtonState())
  {
```

```
    if(ps2x.Button(PSB_L3))
      Serial.println("L3 pressed");
    if(ps2x.Button(PSB_R3))
      Serial.println("R3 pressed");
    if(ps2x.Button(PSB_L2))
      Serial.println("L2 pressed");

    if(ps2x.Button(PSB_R2))
      Serial.println("R2 pressed");

    if(ps2x.Button(PSB_GREEN))
      Serial.println("Triangle pressed");
   }

   // will be TRUE if button was JUST pressed
   if(ps2x.NewButtonState(PSB_RED)){
   Serial.println("Circle just pressed");
}
   // will be TRUE if button was JUST released
   if(ps2x.NewButtonState(PSB_PINK)){
     Serial.println("Square just pressed");
    ;
  }

   // will be TRUE if button was JUST pressed OR released
   if(ps2x.NewButtonState(PSB_BLUE)){
     Serial.println("X just changed");

}
   // print stick values if either is TRUE
   if(ps2x.Analog(PSS_LY) || ps2x.Analog(PSS_LX))
   {
     Serial.print("Stick Values:");
     // Left stick, Y axis. Other options: LX, RY, RX
     Serial.print(ps2x.Analog(PSS_LY), DEC);
     Serial.print(",");
     Serial.print(ps2x.Analog(PSS_LX), DEC);
     Serial.print(",");
     Serial.print(ps2x.Analog(PSS_RY), DEC);
```

```
      Serial.print(",");
      Serial.println(ps2x.Analog(PSS_RX), DEC);
   }
   if (ps2x.Analog(PSS_LY)>129)// mundur pelan
   {
    analogWrite(pwm1, 0);
    analogWrite(pwm2, 150);
    analogWrite(pwm3, 0);
    analogWrite(pwm4, 150);}
 if (ps2x.Analog(PSS_LY)==255)// mundur cepat
 {
      analogWrite(pwm1, 0);
    analogWrite(pwm2, 200);
    analogWrite(pwm3, 0);
    analogWrite(pwm4, 200);
     }
      if ((ps2x.Analog(PSS_LY)>122) &&
 (ps2x.Analog(PSS_LY)<129))// stop
      {
      analogWrite(pwm1, 0);
    analogWrite(pwm2, 0);
    analogWrite(pwm3, 0);
    analogWrite(pwm4, 0);
      }
       if (ps2x.Analog(PSS_LY)<121) // maju pelan
       {
      analogWrite(pwm1, 150);
    analogWrite(pwm2, 0);
    analogWrite(pwm3, 150);
    analogWrite(pwm4, 0);}
 if (ps2x.Analog(PSS_LY)==0)// maju cepat
 {
     analogWrite(pwm1, 200);
    analogWrite(pwm2, 0);
    analogWrite(pwm3, 200);
    analogWrite(pwm4, 0);
    }
    if (ps2x.Analog(PSS_RY)>130)// turun pelan
    {
```

```
      analogWrite(pwm5, 0);
   analogWrite(pwm6, 150);
   analogWrite(pwm7, 150);
   analogWrite(pwm8, 0);}
if (ps2x.Analog(PSS_RY)==255)// turun cepat
{
    analogWrite(pwm5, 0);
   analogWrite(pwm6, 200);
   analogWrite(pwm7, 200);
   analogWrite(pwm8, 0);
     }
      if
((ps2x.Analog(PSS_RY)<129)&&(ps2x.Analog(PSS_RY)>122))// stop
      {
   analogWrite(pwm5, 0);
   analogWrite(pwm6, 0);
   analogWrite(pwm7, 0);
   analogWrite(pwm8, 0);
      }
       if (ps2x.Analog(PSS_RY)<121) // naik pelan
       {
      analogWrite(pwm5, 150);
   analogWrite(pwm6, 0);
   analogWrite(pwm7, 0);
   analogWrite(pwm8, 150);}
if (ps2x.Analog(PSS_RY)==0)// naik cepat
{
   analogWrite(pwm5, 200);
   analogWrite(pwm6, 0);
   analogWrite(pwm7, 0);
   analogWrite(pwm8, 200);
    }
      if (ps2x.Analog(PSS_LX)==255)// belok kanan
      {
        analogWrite(pwm1, 200);
   analogWrite(pwm2, 0);
      }
       if (ps2x.Analog(PSS_LX)==0)// belok kiri
       {
```

```
      analogWrite(pwm3, 200);
 analogWrite(pwm4, 0);
    }
    if (ps2x.Button(PSB_GREEN)){
      digitalWrite(relay1, relayON);

    }
    if (ps2x.Button(PSB_PINK)){
      digitalWrite(relay1, relayOFF);
    }

    if (ps2x.Button(PSB_RED)){

    digitalWrite(relay2, relayON);
    }
    if (ps2x.Button(PSB_BLUE)){

    digitalWrite(relay2, relayOFF);}
delay(50);
```

BIODATA PENULIS

Nama	:	Aditya Putra Pambayun
Tempat, Tanggal Lahir	:	Cilacap, 18 Mei 2001
Agama	:	Islam
Alamat	:	Jl. Pakis RT 04 RW 09 Karangtalun
E-mail	:	adithoppus182@gmail.com
Telepon/Hp	:	081575917964
Hobby	:	Bermusik , Menonton Anime, Bermain Game
Motto	:	Hidup seperti bass, panjang berat tapi kalau tau kuncinya bakal asik.

Riwayat Pendidikan

- SD Negeri 01 Karangtalun — Tahun 2007 - 2013
- SMP Negeri 05 Cilacap — Tahun 2013 – 2016
- SMK Negeri 2 Cilacap — Tahun 2016 – 2019
  Jurusan Teknik Instalasi Pemanfaatan Tenaga Listrik
- Politeknik Negeri Cilacap — Tahun 2020 – 2023
  Prodi D3 Teknik Elektronika

Penulis telah mengikuti seminar hasil Tugas Akhir pada tanggal 15 Agustus 2023 sebagai salah satu persyaratan untuk memperoleh gelar Ahli Madya (A.Md)