DAFTAR PUSTAKA

[1] F. C. Ahmadi, E. Susanto, D. Ph, and I. P. Pangaribuan, "DESAIN DAN IMPLEMENTASI PENDULUM TERBALIK ROTASIONAL DESIGN AND IMPLEMENTATION OF ROTARY INVERTED PENDULUM USING," vol. 5, no. 3, pp. 3842–3850, 2018.

[2] A. A. Roshdy and T. Wang, "Stabilization of Real Inverted Pendulum Using Pole Separation Factor," no. Mems, pp. 711–715, 2012.

[3] G. D. Nusantoro, M. A. Muslim, and R. Indra, "Rancang Bangun Rotary Inverted Pendulum," vol. 6, no. 2, pp. 161–170, 2012.

[4] A. Faizal, K. J. Abidin, J. Hr, S. No, and S. Baru, "Analisis Sistem Kendali Hybrid Pid-Fuzzy dalam Menjaga Keseimbangan Pendulum pada Sistem Rotary Inverter Pendulum Berdasarkan Time Response," vol. 13, no. 2, pp. 242–248, 2016.

[5] M. . Khrisdyologhy, "Perancangan dan Pengontrolan pada Rotary Inverted Pendulum dengan Menggunakan Kontroler Proporsional(P)," 2018.

[6] Z. El-Azami, "ANALISA KONTROLER LINEAR QUADRATIC REGULATOR (LQR) PADA SISTEM ROTARY INVERTED PENDULUM," vol. 4, no. 1, pp. 75–84, 2019.

[7] A. Noorsal, "ANALISA KONTROLER PROPORTIONAL INTEGRAL DERIVATIVE (PID) PADA ROTARY INVERTED PENDULUM," vol. 53, no. 9, pp. 1689–1699, 2019.

[8] Muntari and H. Nurhadi, "Desain Sistem Kendali Rotary Pendulum Dengan Sliding -PID," *J. Tek. Pomits*, vol. 2, no. 2, pp. F243–F249, 2013.

[9] T. Kaliber, "Perancangan Sistem Kontrol PID Untuk Pengendali Sumbu Azimuth Turret Pada," vol. 5, no. 2, pp. 512–516, 2016.

[10] K. Ogata, *Modern Control Engineering*, 5 th. New Jersey: Pearson Education, Inc., 2010.

[11] W. P. Joko, "EVALUASI PEMANFAATAN Qt-OCTAVE DALAM MENENTUKAN LETAK KEDUDUKAN AKAR (ROOT LOCUS) BIDANG TEKNIK KONTROL," vol. 9, 2011.

[12] J. Teknik, E. Politeknik, and N. Manado, "ANALISA DAN SIMULASI SISTEM PENGENDALIAN MOTOR DC."

[13] S. Muttaqin, "Analisa Karakteristik Generator dan Motor DC," no. 21060112130034, pp. 1–11, 2013.

[14] M. Mcroberts, *Begining Arduino*, 1 st ed, New York: Apress,. 2010.

[15] M. A. Atmega, "RANCANG BANGUN PROTOTIPE ELEVATOR

MENGGUNAKAN MICROCONTROLLER ARDUINO ATMEGA 328P," vol. 4, no. 3, pp. 100–112, 2013.

[16] Q. Hidayati, "Pengaturan Kecepatan Motor DC dengan Menggunakan Mikrokontroler Atmega 8535," pp. 1–5.

LAMPIRAN A

Program Arduino

```
int encoderPin1 = 2;
int encoderPin2 = 3;

volatile int lastEncoded = 0;
volatile long encoderValue = 0;
long lastencoderValue = 0;
int lastMSB = 0;
int lastLSB = 0;

volatile long counterx = 0; // tipe data long nilai berkisat antar -2.147.483.648 hingga 2.147.483.647

#define DIR1 8 //direction motor kanan
#define DIR2 9 //direction motor kanan

unsigned char speedx;
float pwm;

float PIDvalue;
unsigned long currentTime = 0;
unsigned long previousTime = 0;
float elapsedTime = 0;

float error = 0;
float lastError = 0;
```

```
float cumError = 0;
float rateError = 0;
float error2 = 0;
float lastError2 = 0;
float cumError2 = 0;
float rateError2 = 0;

float kp = 38;
float ki = 0;
float kd = 0;
float kp2 = 18;
float ki2 = 0;
float kd2 = 0;

void setup() {
  Serial.begin (38400);

  // Pembaca Sudut THETA
  pinMode(encoderPin1, INPUT);
  pinMode(encoderPin2, INPUT);
  digitalWrite(encoderPin1, HIGH); //turn pullup resistor on
  digitalWrite(encoderPin2, HIGH); //turn pullup resistor on

  attachInterrupt(digitalPinToInterrupt(encoderPin1), updateEncoder, CHANGE);
  attachInterrupt(digitalPinToInterrupt(encoderPin2), updateEncoder, CHANGE);
```

```
  pinMode(18, INPUT_PULLUP);
  attachInterrupt(digitalPinToInterrupt(18), zsignal, HIGH);



  // Pembaca Sudut Alfa
  pinMode(21, INPUT_PULLUP); // pin untuk input
  pinMode(20, INPUT_PULLUP);
  attachInterrupt(digitalPinToInterrupt(21), ai2, RISING);//0 artinya pin intterup ke 0 (pin 21). ai0 fungsi interrupt
  attachInterrupt(digitalPinToInterrupt(20), ai3, RISING);//1 artinya pin intterup ke 0 (pin 20). ai1 fungsi interrupt

  pinMode(DIR1, OUTPUT);
  pinMode(DIR2, OUTPUT);



  pinMode(50, INPUT);

  //inisialisasi
  digitalWrite(DIR1, LOW);
  digitalWrite(DIR2, LOW);

  //  digitalWrite(PWM, HIGH); //stop motor, Sistem PWM aktif LOW
  Serial.print("Time(ms), ");
  Serial.print("PWM, ");
  Serial.print("PIDvalue, ");
  Serial.print("Alfa, ");   //1000 PPR
  Serial.println("Theta, ");  //200 PPR
  delay(2000);
```

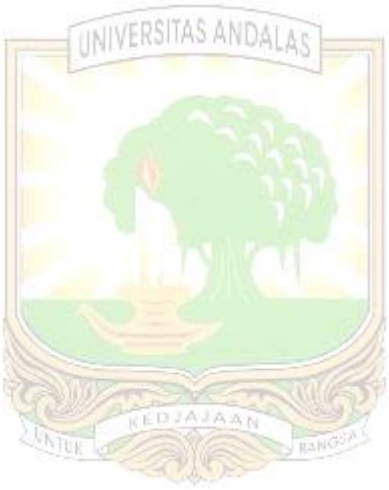
```
}


void loop() {
  PIDvalue = computePID(encoderValue, counterx);   //fungsi untuk memproleh nilai kalkulasi error dengan PID
  rotasix();
  Serial.print(currentTime);
  Serial.print(", ");
  Serial.print(pwm);
  Serial.print(", ");
  Serial.print(PIDvalue);
  Serial.print(", ");
  Serial.print(encoderValue);
  Serial.print(", ");
  Serial.println(counterx);
}


float computePID(float encoderValuey, float encoderValuex) {
  currentTime = millis();                //get current time
  elapsedTime = (float)(currentTime - previousTime) / 1000;      //compute time elapsed from previous computation
  float sudut = encoderValuey * 0.36;
  float sudut2 = encoderValuex * 0.45;
  error = 180 - sudut;                              // determine error
  error2 = 0 - sudut2;
  //cumError = (error + lastError) * elapsedTime;                // compute integral
  cumError += error ;                // compute integral
  cumError2 += error2 ;                // compute integral
  rateError = (error - lastError) / elapsedTime; // compute derivative
```

```
  rateError2 = (error2 - lastError2) / elapsedTime; // compute derivative

  float out = (kp * error + ki * cumError + kd * rateError) + (kp2 * error2 + ki2 *
cumError2 + kd2 * rateError2);        //PID output

  lastError = error;                          //remember current error
  lastError2 = error2;
  previousTime = currentTime;                      //remember current time

  return out;                                 //have function return the PID output
}

void rotasix() {
  if (error > 3) {
    if (error > 50) {
      pwm = 0;
    }
    else {
      pwm = PIDvalue;
      if (pwm > 255) {
        pwm = 255;
      }
    }
    rotasi1(pwm);
  }
  else if (error < -3) {
    if (error < -50) {
      pwm = 0 ;
    }
```

```
    else {
      pwm = (PIDvalue * (-1));
      if (pwm > 255) {
        pwm = 255;
      }
    }
    rotasi2(pwm);
  }
  else {
    pwm = 0;
    diam(pwm);
  }
}

void rotasi1(unsigned char speed_motor) {
  analogWrite(DIR2, 0);
  speedx = speed_motor; // konversi dari PWM aktif LOW ke aktif HIGH
  analogWrite(DIR1, speedx);
}
void rotasi2(unsigned char speed_motor) {
  analogWrite(DIR1, 0);
  speedx = speed_motor; // konversi dari PWM aktif LOW ke aktif HIGH
  analogWrite(DIR2, speedx);
}
void diam(unsigned char speed_motor) {
  digitalWrite(DIR1, LOW);
  digitalWrite(DIR2, LOW);
}
```

```
void updateEncoder() {
  int MSB = digitalRead(encoderPin1); //MSB = most significant bit
  int LSB = digitalRead(encoderPin2); //LSB = least significant bit

  int encoded = (MSB << 1) | LSB; //converting the 2 pin value to single number
  int sum  = (lastEncoded << 2) | encoded; //adding it to the previous encoded
value

  if (sum == 0b1101) encoderValue++;
  if (sum == 0b1110) encoderValue--;

  lastEncoded = encoded; //store this value for next time
}

void ai2() {              //pastikan jika bertambah searah jarum jam
  if (digitalRead(20) == LOW) {
    counterx++;
  } else {
    counterx--;
  }
}

void ai3() {              //pastikan jika berkurang berlawanan arah jarum jam
  if (digitalRead(21) == LOW) {
    counterx--;
  } else {
    counterx++;
  }
```

```
}

void zsignal() {
  encoderValue = 500;
}
```

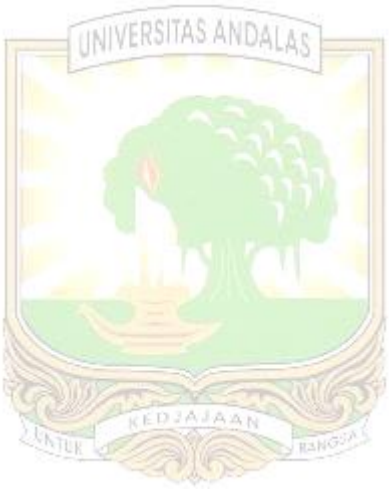


LAMPIRAN B

Program Matlab

```
clc
clear all
close all
close all hidden



% Data Rotary Inverted Pendulum dan Motor
mp=0.0499  %satuan Kg
rp=0.473     %satuan m
Ip=(1/3)*mp*rp*rp %momen inerisia rod
ml=0.0506   %Satuan Kg
rl=0.203    %Satuan m
Il=(1/3)*ml*rl*rl %momen inersia lengan
Km= 1.415   %RadPS dibagi Tegangan
Tm= 0.323      %waktu untuk 63,2% dari nilai akhir di kali nilai
arus

a=(mp*rl*rl)+Il
b=(mp*rp*rp)+Ip
c=mp*rp*rl
d=mp*9.80665*rp

A21 = ((c^2*d)-(a*b*d))/((a*b^2)-(b*c^2));
A24 = -(((a*b*c)-(c^3))/((a*b^2*Tm)-(b*c^2*Tm)));
A44 = -(1/Tm);
B12 = (Km*((a*b*c)-(b^3)))/((a*b^2*Tm)-(b*c^2*Tm));
B14 = Km/Tm;

A = [0 1 0 0; A21 0 0 A24; 0 0 0 1; 0 0 0 A44];
B = [0; B12; 0; B14];
C = [1 0 0 0;0 0 1 0];
D = [0; 0;];


states = {'alfa' 'alfadot' 'theta' 'thetadot'};
inputs = {'tegangan'};
outputs = {'alfa' 'theta'};

sys_mimo = ss(A,B,C,D,'statename',states,...
'inputname',inputs,...
'outputname',outputs);

sys_mimo


transff=tf(sys_mimo)
```

```
syspen_ol = transff(1)/transff(2)
integ_num = [1 0]
integ_den = [0 1]
integ = tf(integ_num,integ_den)
sysarm_ol = transff(2)

[C_pidarm,infoarm] = pidtune(sysarm_ol,'p')
sysarm_ol_pid = C_pidarm*sysarm_ol

sysarm_cl = feedback(sysarm_ol_pid,1)

sys_ol = sysarm_cl*syspen_ol

[C_pidpen,infopen] = pidtune(sys_ol,'p')
sys_pid=sys_ol*C_pidpen
sys_cl=feedback(sys_pid,1)

step(sys_cl)
grid on

C_pidarm
C_pidpen
```



LAMPIRAN C

Prototype *rotary inverted pendulum* dengan dua derajat kebebasan