

# **SISTEM KENDALI PID KASUS P DAN D DENGAN EPROM**



Mata Kuliah : Sistem Kendali

Kode Dosen : AJR

Kelas : D3TK-43-02

Anggota Kelompok :

1. Wahyu Esysa Nasution (6702194052)
2. Farhan Ulil Fajri (6702190077)

**PROGRAM STUDI D3 TEKNOLOGI KOMPUTER  
FAKULTAS ILMU TERAPAN  
UNIVERSITAS TELKOM  
BANDUNG  
2021**

## **A. Tujuan**

Maksud dan tujuan dari praktikum ini adalah :

1. Mahasiswa dapat memahami fungsi dan cara kerja PID pada motor DC
2. Mahasiswa dapat membuat program untuk menggunakan EEPROM untuk penyimpanan data sensor yang telah dikalibrasi.
3. Mahasiswa dapat menggunakan peripheral berupa push button untuk menambah konstanta  $K_p$  dan  $K_d$ .

## **B. Alat dan Bahan**

Peralatan yang dibutuhkan dalam praktikum ini adalah :

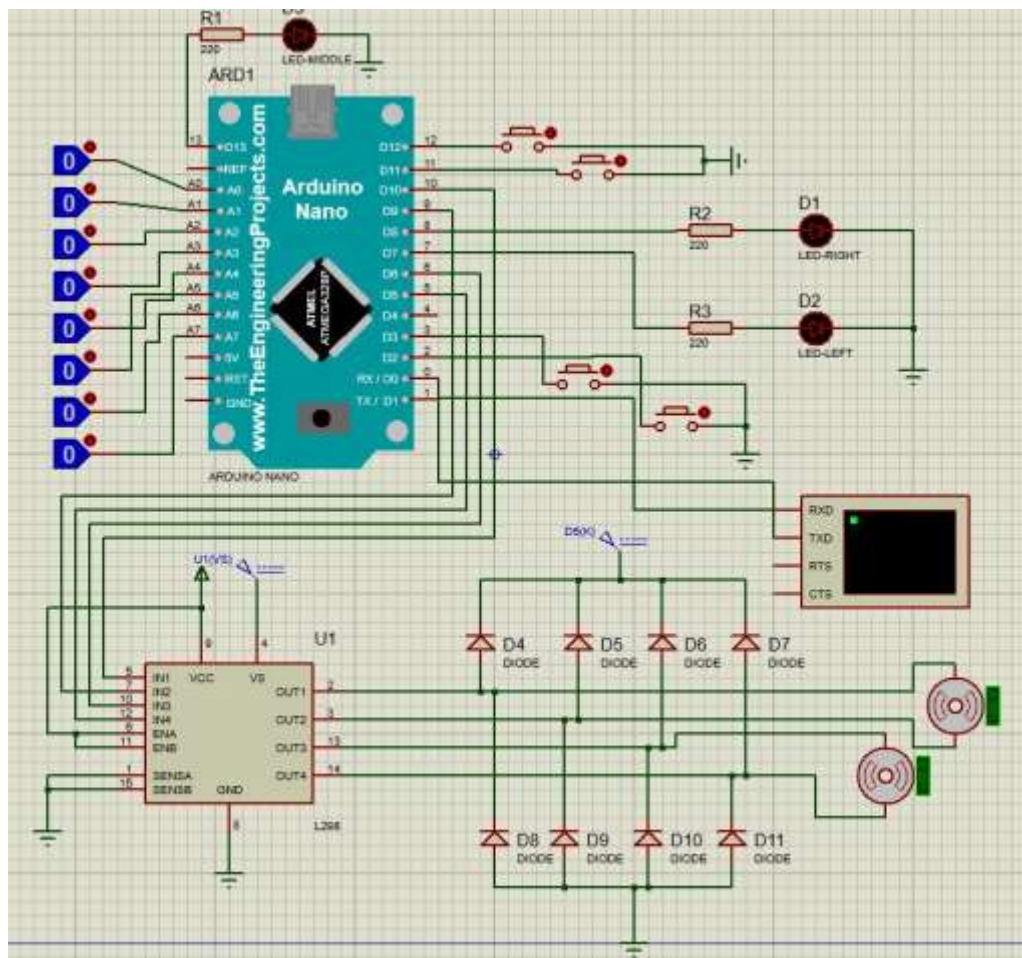
1. Proteus (apk)
2. Arduino IDE

## **C. Teori dasar**

Teknik kendali proporsional-derivatif (PD) adalah pengendali yang merupakan gabungan antara teknik kendali proporsional (P) dengan teknik kendali derivatif (D). Gambar 1 merupakan gambar diagram blok sistem kendali PD. Program Studi D3 Teknologi Komputer Fakultas Ilmu Terapan 2019 Hal 12 Dalam penerapannya di software, kondisi ideal pada robot adalah bergerak maju lurus mengikuti garis, dengan kata lain  $error = 0$ . Dari sini dapat diasumsikan bahwa Set Point (SP) / kondisi ideal adalah saat  $SP = 0$ . Nilai sensor yang dibaca oleh sensor disebut Process Variable (PV) / nilai aktual pembacaan. Menyimpangnya posisi robot dari garis disebut sebagai error (e), yang didapat dari  $e = SP - PV$ . Dengan mengetahui besar error, mikrokontroler dapat memberikan nilai PWM motor kiri dan kanan yang sesuai agar dapat menuju ke posisi ideal ( $SP = 0$ ). Besarnya nilai PWM ini dapat diperoleh dengan menggunakan kontrol Proporsional (P), dimana  $P = e \times K_p$  ( $K_p$  adalah konstanta proporsional yang nilainya diset sendiri dari hasil tuning/trial and error). Jika pergerakan robot masih terlihat bergelombang, dapat ditambahkan parameter kontrol Derivatif (D). Kontrol D digunakan untuk mengukur seberapa cepat robot bergerak dari kiri ke kanan atau dari kanan ke kiri. Semakin cepat bergerak dari satu sisi ke sisi lainnya, maka semakin besar nilai D. Konstanta D ( $K_d$ ) digunakan untuk menambah atau mengurangi imbas dari derivatif. Dengan mendapatkan nilai  $K_d$  yang tepat pergerakan sisi ke sisi yang bergelombang akibat dari kontrol proporsional dapat diminimalisasi. Dengan mendapatkan nilai  $K_d$  yang tepat pergerakan sisi ke sisi yang bergelombang akibat dari kontrol proporsional bisa diminimalisasi. Nilai D didapat dari  $D = K_d / T_s \times rate$ , dimana  $T_s$  adalah time sampling atau waktu cuplik dan  $rate = e(n) - e(n-1)$ . Dalam program, nilai error ( $SP - PV$ ) saat itu menjadi nilai `last_error`, sehingga `rate` didapat dari `error - last_error` Agar konfigurasi atau hasil

kalibrasi sensor tidak hilang ketika robot dimatikan atau kehilangan daya, EEPROM pada Arduino Nano dimanfaatkan untuk menyimpan data tersebut. Arduino Nano dengan mikrokontroler ATmega328 memiliki EEPROM dengan kapasitas 1024 byte. Kemudian untuk mempermudah user dalam memanfaatkan EEPROM untuk menyimpan dan menggunakan data, 4 buah push button yang disediakan pada robot digunakan.

#### D. Hasil Percobaan



```

#include <EEPROM.h>
int mKanan1=9;
int mKanan2=10;
int mKiri1=6;
int mKiri2=5;
int EnA= 11;
int EnB= 3;

char p1=A0;
char p2=A1;
char p3=A2;
char p4=A3;
char p5=A4;
char p6=A5;
char p7=7;
char p8=8;

int rate = 0;
int moveControl = 0;
int kecepatanMotorKanan = 0;
int kecepatanMotorKiri = 0;
int kecepatanSetPoint = 150;

int sensor[8];
int NilaiMaxSensor[8];
int NilaiMinSensor[8];
int NilaiTengahSensor[8];

int state =0;
int setting =0;
int peka[8];
int Kp=20;
int Kd=5;
int kondisi[8];

//button
int kirib = 3;
int kirid = 2;
int kananb = 12;
int kanand = 11;

```

```

//led
int ledkanan = 8;
int ledkiri = 7;
void setup(){
  Serial.begin(9600);
  pinMode(mKanan1,OUTPUT);
  pinMode(mKanan2,OUTPUT);
  pinMode(mKiri1,OUTPUT);
  pinMode(mKiri2,OUTPUT);
  pinMode(EnA,OUTPUT);
  pinMode(EnB,OUTPUT);

  pinMode(p1,INPUT);
  pinMode(p2,INPUT);
  pinMode(p3,INPUT);
  pinMode(p4,INPUT);
  pinMode(p5,INPUT);
  pinMode(p6,INPUT);
  pinMode(p7,INPUT);
  pinMode(p8,INPUT);

  for(int i=0; i<8; i++){
    NilaiMaxSensor[i]=1023;
    NilaiMinSensor[i]=0;
    peka[i]=500;
  }

  kondisi[0] = digitalRead(p1);
  kondisi[1] = digitalRead(p2);
  kondisi[2] = digitalRead(p3);
  kondisi[3] = digitalRead(p4);
  kondisi[4] = digitalRead(p5);
  kondisi[5] = digitalRead(p6);
  kondisi[6] = digitalRead(p7);
  kondisi[7] = digitalRead(p8);

  for(int i=0; i<8; i++){
    sensor[i]==kondisi[i];
    if(sensor[i] > NilaiMinSensor[i]){
      NilaiMinSensor[i] =sensor[i];
    }
    if(sensor[i] < NilaiMaxSensor[i]){
      NilaiMaxSensor[i] =sensor[i];
    }
    NilaiTengahSensor[i]= (NilaiMinSensor[i]+NilaiMaxSensor[i])/2;
  }
  for(int i=0; i<8; i++){
    EEPROM.write(i, NilaiTengahSensor[i]);
    peka[i]=EEPROM.read(i)*4;
    EEPROM.write(8, Kp);
    Kp=EEPROM.read(8);
    EEPROM.write(9, Kd);
    Kd=EEPROM.read(9);
    Serial.print("peka(");
    Serial.print(i);
    Serial.print(") : ");
    Serial.print(peka[i]);
    Serial.print("Kp :");
    Serial.print(Kp);
    Serial.print("Kd :");
    Serial.print(Kd);
    delay(1000);
  }
}

```

```

void loop() {

    if(digitalRead(kirib)==LOW){
        state==1;
    }if(digitalRead(kirid)==LOW){
        state==2;
    }if(digitalRead(kirib)==LOW && setting==1){
        state==3;
    }if(digitalRead(kirid)==LOW && setting==1){
        state==4;
    }if(digitalRead(kananb)==LOW && setting==1){
        state==5;
    }if(digitalRead(kanand)==LOW && setting==1){
        state==6;
    }

    //state
    if(state==1){
        //auto calibration
    }if(state==2){
        for(int i=0; i<8; i++){
            EEPROM.write(i,peka[i]/4);
        }
        EEPROM.write(8,Kd);
        EEPROM.write(9,Kp);
    }if(state==3){
        digitalWrite(ledkiri,HIGH);
        delay(1000);
        digitalWrite(ledkiri,LOW);
        Kp=Kp-1;
        digitalWrite(8,Kp);
    }if(state==4){
        digitalWrite(ledkiri,HIGH);
        delay(1000);
        digitalWrite(ledkiri,LOW);
        Kp=Kp+1;
        digitalWrite(8,Kp);
    }if(state==5){
        digitalWrite(ledkiri,HIGH);
        delay(1000);
        digitalWrite(ledkiri,LOW);
        Kd=Kd-1;
        digitalWrite(9,Kd);
    }if(state==6){
        digitalWrite(ledkiri,HIGH);
        delay(1000);
        digitalWrite(ledkiri,LOW);
        Kd=Kd+1;
        digitalWrite(9,Kd);
    }
}

```

```

String kondisi = String(kondisi[0]+kondisi[1]+kondisi[2]+kondisi[3]+kondisi[4]+kondisi[5]+kondisi[6]+kondisi[7]);

if(kondisi = "10000000"){
    satu();
}
else if(kondisi = "11000000"){
    dua();
}
else if(kondisi = "01000000"){
    tiga();
}
else if(kondisi = "01100000"){
    empat();
}
else if(kondisi = "00100000"){
    lima();
}
else if(kondisi = "00110000"){
    enam();
}
else if(kondisi = "00010000"){
    tujuh();
}
else if(kondisi = "00011000"){
    delapan();
}
else if(kondisi = "00001000"){
    sembilan();
}
else if(kondisi = "00001100"){
    sepuluh();
}
else if(kondisi = "00000100"){
    sebelas();
}
else if(kondisi = "00000110"){
    duabelas();
}
else if(kondisi = "00000010"){
    tigabelas();
}
else if(kondisi = "00000011"){
    empatbelas();
}
else if(kondisi = "00000001"){
}

void satu(){
    int Error = -7;
    int LastError = Error;
    int rate = (Error-LastError);
    moveControl = (Kp*Error)+(Kd+rate);
    kecepatanMotorKanan = kecepatanSetPoint - moveControl;
    kecepatanMotorKiri = kecepatanSetPoint + moveControl;
    digitalWrite(mKanan1,HIGH);
    digitalWrite(mKanan2,LOW);
    digitalWrite(mKiri1,HIGH);
    digitalWrite(mKiri2,LOW);
    analogWrite(EnA,kecepatanMotorKanan);
    analogWrite(EnB,kecepatanMotorKiri);
    Serial.println("Error = -7");
}

void dua(){
    int Error = -6;
    int LastError = Error;
    int rate = (Error-LastError);
    moveControl = (Kp*Error)+(Kd+rate);
    kecepatanMotorKanan = kecepatanSetPoint - moveControl;
    kecepatanMotorKiri = kecepatanSetPoint + moveControl;
    digitalWrite(mKanan1,HIGH);
    digitalWrite(mKanan2,LOW);
    digitalWrite(mKiri1,HIGH);
    digitalWrite(mKiri2,LOW);
    analogWrite(EnA,kecepatanMotorKanan);
    analogWrite(EnB,kecepatanMotorKiri);
    Serial.println("Error = -6");
}

void tiga(){
    int Error = -5;
    int LastError = Error;
    int rate = (Error-LastError);
    moveControl = (Kp*Error)+(Kd+rate);
    kecepatanMotorKanan = kecepatanSetPoint - moveControl;
    kecepatanMotorKiri = kecepatanSetPoint + moveControl;
    digitalWrite(mKanan1,HIGH);
    digitalWrite(mKanan2,LOW);
    digitalWrite(mKiri1,HIGH);
    digitalWrite(mKiri2,LOW);
    analogWrite(EnA,kecepatanMotorKanan);
    analogWrite(EnB,kecepatanMotorKiri);
    Serial.println("Error = -5");
}

void empat(){
    int Error = -4;
    int LastError = Error;
    int rate = (Error-LastError);
    moveControl = (Kp*Error)+(Kd+rate);
    kecepatanMotorKanan = kecepatanSetPoint - moveControl;
    kecepatanMotorKiri = kecepatanSetPoint + moveControl;
    digitalWrite(mKanan1,HIGH);
    digitalWrite(mKanan2,LOW);
    digitalWrite(mKiri1,HIGH);
    digitalWrite(mKiri2,LOW);
    analogWrite(EnA,kecepatanMotorKanan);
    analogWrite(EnB,kecepatanMotorKiri);
    Serial.println("Error = -4");
}

void empatbelas(){
    int Error = 6;
    int LastError = Error;
    int rate = (Error-LastError);
    moveControl = (Kp*Error)+(Kd+rate);
    kecepatanMotorKanan = kecepatanSetPoint - moveControl;
    kecepatanMotorKiri = kecepatanSetPoint + moveControl;
    digitalWrite(mKanan1,HIGH);
    digitalWrite(mKanan2,LOW);
    digitalWrite(mKiri1,HIGH);
    digitalWrite(mKiri2,LOW);
    analogWrite(EnA,kecepatanMotorKanan);
    analogWrite(EnB,kecepatanMotorKiri);
    Serial.println("Error = 6");
}

void limabelas(){
    int Error = 7;
    int LastError = Error;
    int rate = (Error-LastError);
    moveControl = (Kp*Error)+(Kd+rate);
    kecepatanMotorKanan = kecepatanSetPoint - moveControl;
    kecepatanMotorKiri = kecepatanSetPoint + moveControl;
    digitalWrite(mKanan1,HIGH);
    digitalWrite(mKanan2,LOW);
    digitalWrite(mKiri1,HIGH);
    digitalWrite(mKiri2,LOW);
    analogWrite(EnA,kecepatanMotorKanan);
    analogWrite(EnB,kecepatanMotorKiri);
    Serial.println("Error = 7");
}

void enambelas(){
    digitalWrite(mKanan1,LOW);
    digitalWrite(mKanan2,LOW);
    digitalWrite(mKiri1,LOW);
    digitalWrite(mKiri2,LOW);
    analogWrite(EnA,0);
    analogWrite(EnB,0);
    Serial.println("STOP");
}

```



### **E. Kesimpulan**

Kesimpulan praktikum kali ini tidak berbeda jauh dari praktikum sebelumnya hanya mengubah nilai EPROM

### **F. Link Video Kegiatan praktikum**

<https://www.youtube.com/watch?v=yxKX1CCRx6k>

<https://github.com/foolaisx/kelompok-wahyu-farhanulil>