

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



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 sistem kendali berbasis PID dengan error yang dihubungkan dengan konstanta proporsional dan derivatif

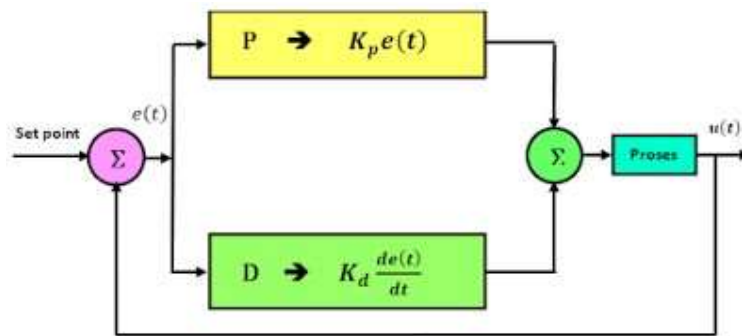
## B. Alat dan Bahan

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.



Persamaan hubungan antara keluaran sistem dengan sinyal error pada kombinasi aksi kendali proporsional-derivative adalah sebagai berikut.

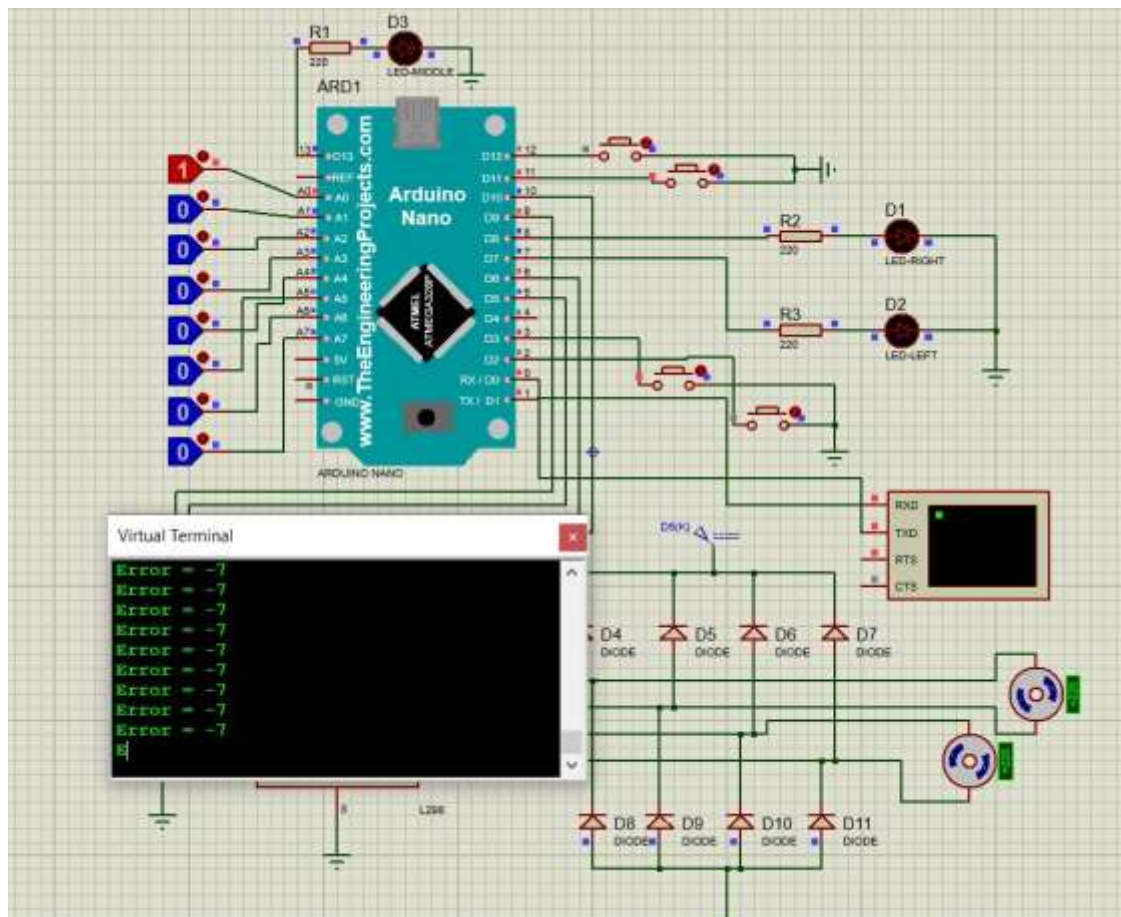
$$u(t) = K_p e(t) + K_d \frac{d}{dt} e(t)$$

$$u(t) = K_p e(t) + K_p T_d \frac{de(t)}{dt}$$

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  $D = K_d/T_s \times \text{rate}$ , dimana  $T_s$  adalah time sampling atau waktu cuplik dan  $\text{rate} = e(n) - e(n-1)$ . Dalam program, nilai error ( $SP - PV$ ) saat itu menjadi nilai `last_error`, sehingga `rate` didapat dari `error - last_error`

#### D. Hasil Percobaan



pd

```
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 Kp = 15;
int Kd = 5;
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];

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;
    }
}

void loop(){
    int kondisi0 = digitalRead(p1);
    int kondisi1 = digitalRead(p2);
    int kondisi2 = digitalRead(p3);
    int kondisi3 = digitalRead(p4);
    int kondisi4 = digitalRead(p5);
    int kondisi5 = digitalRead(p6);
    int kondisi6 = digitalRead(p7);
    int kondisi7 = digitalRead(p8);

    for(int i=0; i<8; i++){
        sensor[i]==kondisi[i];
        if(sensor[i] > NilaiMinSensor[i]){
            NilaiMinSensor[i] =sensor[i];
        }
        if else(sensor[i] < NilaiMaxSensor[i]){
            NilaiMaxSensor[i] =sensor[i];
        }
        NilaiTengahSensor[i]= NilaiMinSensor[i]+NilaiMaxSensor[i])/2;
        // blink led
    }
}
```

```

string kondisi = String(kondisi0+kondisi1+kondisi2+kondisi3+kondisi4+kondisi5+kondisi6+kondisi7);

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"){
    limabelas();
}
else if(kondisi = "00000000"){

```



```

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 LastError = -6;
    moveControl = Kp*LastError;
    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 LastError = -5;
    moveControl = Kp*LastError;
    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 LastError = -4;
    moveControl = Kp*LastError;
    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 lima(){
    int LastError = -3;
    moveControl = Kp*LastError;
    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 = -3");
}

void enam(){
    int LastError = -2;
    moveControl = Kp*LastError;
    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 = -2");
}

```

```

void tujuh() {
    int LastError = -1;
    moveControl = Kp*LastError;
    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 = -1");
}

void delapan() {
    int LastError = 0;
    moveControl = Kp*LastError;
    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 = 0");
}

void sembilan() {
    int LastError = 1;
    moveControl = Kp*LastError;
    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 = 1");
}
}

```

```

void sepuluh() {
    int LastError = 2;
    moveControl = Kp*LastError;
    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 = 2");
}

void sebelas() {
    int LastError = 3;
    moveControl = Kp*LastError;
    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 = 3");
}

void duabelas() {
    int LastError = 4;
    moveControl = Kp*LastError;
    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 tigabelas(){
    int LastError = 5;
    moveControl = Kp*LastError;
    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 empatbelas(){
    int LastError = 6;
    moveControl = Kp*LastError;
    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 LastError = 7;
    moveControl = Kp*LastError;
    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");
}

```



Sensor								Error	Nilai Setpoint	Analog Value	
0	1	2	3	4	5	6	7			Motor Kiri	Motor Kanan
1	0	0	0	0	0	0	0	-7	150	115	185
1	1	0	0	0	0	0	0	-6	150	120	180
0	1	0	0	0	0	0	0	-5	150	125	175
0	1	1	0	0	0	0	0	-4	150	130	170
0	0	1	0	0	0	0	0	-3	150	135	165
0	0	1	1	0	0	0	0	-2	150	140	160
0	0	0	1	0	0	0	0	-1	150	145	155
0	0	0	1	1	0	0	0	0	150	150	150
0	0	0	0	1	0	0	0	1	150	155	145
0	0	0	0	1	1	0	0	2	150	160	140
0	0	0	0	0	1	0	0	3	150	165	135
0	0	0	0	0	1	1	0	4	150	170	130
0	0	0	0	0	0	1	0	5	150	175	125
0	0	0	0	0	0	1	1	6	150	180	120
0	0	0	0	0	0	0	1	7	150	185	115

### E. Kesimpulan

kesimpulan pada praktikum kali ini adalah nilai eror dari system kendali P disertai analog value menunjukkan bahwa data eror mempengaruhi pada kecepatan motor saat berjalan

### F. Link Video Kegiatan praktikum

<https://www.youtube.com/watch?v=VSLuMSGJA0c>