**MAIN PROGRAM**

#include <Servo.h>

// pins

#define echoKiri 7

#define trigKiri 8

#define echoKanan 6

#define trigKanan 5

#define rudderPin 3

#define servoPin 4

#define rudderNormal 80

#define rudderMin 30

#define rudderMax 135

#define propeller A1

#define fanPin A0

// variable

int kp = 20;

int ki = 2;

int kd = 4;

int ts = 10;

int setPoint = 20;

int fanpos = 110;

Servo fan,rudder;

void setup(){

rudder.attach(rudderPin);

fan.attach(servoPin);

rudder.write(rudderNormal);

pinMode(propeller, OUTPUT);

pinMode(fanPin, OUTPUT);

digitalWrite(propeller, LOW);

digitalWrite(fanPin, LOW);

Serial.begin(9600);

}

void loop(){

int jarakKiri = get\_data(trigKiri, echoKiri);

int jarakKanan = get\_data(trigKanan, echoKanan);

int besarSudut = pid(jarakKanan, 200, kp, ki, kd, ts, setPoint);

besarSudut = rudderNormal + besarSudut;

if(besarSudut < rudderMin){

besarSudut = rudderMin;

}

else if(besarSudut > rudderMax){

besarSudut = rudderMax;

}

if(jarakKiri<55 && jarakKanan<30){

digitalWrite(fanPin, HIGH);

digitalWrite(propeller, LOW);

rudder.write(rudderMax);

for(fanpos = 85; fanpos <= 165; fanpos += 1){

fan.write(fanpos);

delay(30);

}

for(fanpos = 165; fanpos >= 85; fanpos -= 1){

fan.write(fanpos);

delay(30);

}

}

else{

digitalWrite(fanPin, LOW);

digitalWrite(propeller, HIGH);

}

rudder.write(besarSudut);

fan.write(fanpos);

Serial.println("Kiri: "+String(jarakKiri)+" Kanan: "+String(jarakKanan)+" Sudut: "+String(besarSudut)+"degree");

}

**TAB PID**

int pid(int jarak1, int jarak2, int Kp, int Ki, int Kd, int Ts, int Sp){

double Pv, Error, Sp\_sensor, P, Last\_error, Pid, Pd;

double I1, I2, I3, I, D1,D2, D3, D;

Error = Sp - jarak1;

Error = Error \* 1.9;

if(jarak2 < 25){

Error=30;

}

P = Kp \* Error;

D1 = Kd \* Ts;

D2 = D1 / Ts;

D3 = Error - Last\_error;

D = D2 \* D3;

//=======KONSTANTA INTEGRATIF=======

I1 = Ki / Ts;

I2 = Error + Last\_error;

I3 = I1 \* I2;

I = I3 \* Ts;

//=======ERROR LAMPAU=======

Last\_error = Error;

//====proportional-INTEGRATIF-derivative control====

Pd = P + D;

Pid = Pd + I;

Pid = Pid/10;

return Pid;

}

**TAB SENSOR ULTRASONIC**

int get\_data(int trigPin, int echoPin){

long duration, distance;

pinMode(trigPin, OUTPUT);

pinMode(echoPin, INPUT);

digitalWrite(trigPin, LOW);

delayMicroseconds(2);

digitalWrite(trigPin, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin, LOW);

duration = pulseIn(echoPin, HIGH);

distance = (duration/2)/29.1;

delayMicroseconds(200);

return distance;

}