Lampiran-lampiran

```
// untuk  kondisi dekat
  if (sensorUS <= 10)
  { jarak [0] = 1;}
  else if (sensorUS > 10 && sensorUS <= 20)
  {  jarak [0] = (20 - sensorUS)/(20 - 10); }
  else
  { jarak [0] = 0;}

  // untuk kondisi sedang
  if (sensorUS <= 10)
  { jarak [1] = 0;}
  else if (sensorUS > 10 && sensorUS <= 20)
  { jarak [1] = (sensorUS -10)/(20-10);}
  else if (sensorUS > 20 && sensorUS <= 40)
  { jarak [1] = (40-sensorUS)/(40 - 20);}
  else
  { jarak [1] = 0;}

  // untuk kondisi jauh
  if (sensorSuhu <= 20)
  { jarak [2] = 0;}
  else if (sensorUS > 20 && sensorUS <= 30)
  { jarak [2] = (sensorUS-20)/(30-20);}
  else if (sensorUS > 30)
  { jarak [2] = 1;}
}
```

```
/*
  PROGRAM FOR IR SENSOR & FAN MOTOR
*/

// initialization flame IR sensor
#define flame A5

// initialization fan motor
#define in_A 22
#define in_B 23

// initialization driver motor
#define en_A 2
#define en_B 3
#define right_1 4
#define right_2 5
#define left_1 7
#define left_2 6
int speed = 150;

void setup()  {
  Serial.begin(9600);
  pinMode(flame, INPUT); // set as input
  pinMode(in_A, OUTPUT); // set fan motor as
output
  pinMode(in_B, OUTPUT);
}
```

```
void loop() {
  // read flame value from IR sensor
  int value = digitalRead(flame);

  // display value to serial
  Serial.println(value);

  // if flame exist, fan motor ON
  if(value == 0){
    // motor off
    analogWrite(en_A, 0);
    analogWrite(en_B, 0);
    digitalWrite(right_1, LOW);
    digitalWrite(right_2, LOW);
    digitalWrite(left_1, LOW);
    digitalWrite(left_2, LOW);

    // add delay
    delay(2000);

    // fan on
    digitalWrite(in_A, HIGH);
    digitalWrite(in_B, LOW);
  }
  // if doesn't exist, fan motor OFF
  else{
    // motor on
    analogWrite(en_A, 150);
    analogWrite(en_B, 150);
    digitalWrite(right_1, HIGH);
    digitalWrite(right_2, LOW);
```

```
    digitalWrite(left_1, HIGH);
    digitalWrite(left_2, LOW);

    // fan off
    digitalWrite(in_A, LOW);
    digitalWrite(in_B, LOW);
  }
}
```

```
#include <string.h>
#define IBUS_BUFFSIZE 32
#define IBUS_MAXCHANNELS 10 // I am using only
10 channels because my TX (FlySky i6) supports max 10 channels
#include <Servo.h>
static uint8_t ibusIndex = 0;
static uint8_t ibus[IBUS_BUFFSIZE] = {0};
static uint16_t rcValue[IBUS_MAXCHANNELS];

static boolean rxFrameDone;

int ch_width_3;
int ch_width_4;
```

```
Servo ch3;
Servo ch4;

void setup()
{
  Serial.begin(115200);
  ch3.attach(4);
  ch4.attach(6);
}

void loop()
{
  readRx();

 }

void readRx()
{
  rxFrameDone = false;

  if (Serial.available())
  {
    uint8_t val = Serial.read();
    // Look for 0x2040 as start of packet
    if (ibusIndex == 0 && val != 0x20)
    {
      ibusIndex = 0;
      return;
    }
    if (ibusIndex == 1 && val != 0x40)
    {
      ibusIndex = 0;
      return;
    }

    if (ibusIndex == IBUS_BUFFSIZE)
    {
      ibusIndex = 0;
      int high=3;
      int low=2;
      for(int i=0;i<IBUS_MAXCHANNELS; i++)
```

```
                    {
                      rcValue[i] = (ibus[high] << 8) +
ibus[low];
                      high += 2;
                      low += 2;
                    }

                    ch_width_3 = map(rcValue[2], 1000, 2000,
1000, 2000);
                    ch3.writeMicroseconds(ch_width_3);

                    Serial.print(ch_width_3);
                    Serial.print("     ");

                    ch_width_4 = map(rcValue[3], 1000, 2000,
1000, 2000);
                    ch4.writeMicroseconds(ch_width_4);

                    Serial.print(ch_width_4);
                    Serial.print("     ");

                    rxFrameDone = true;
                    return;
                  }
                  else
                  {
                    ibus[ibusIndex] = val;
                    ibusIndex++;
                  }
                }
              }
```