

# SISTEM MONITORING SUHU DAN PENDETEKSI KEBAKARAN KELOMPOK 7 TT4A

DANDI KUSUMA

YELI HERLINA

ZULEFIKAR BELLA ALI

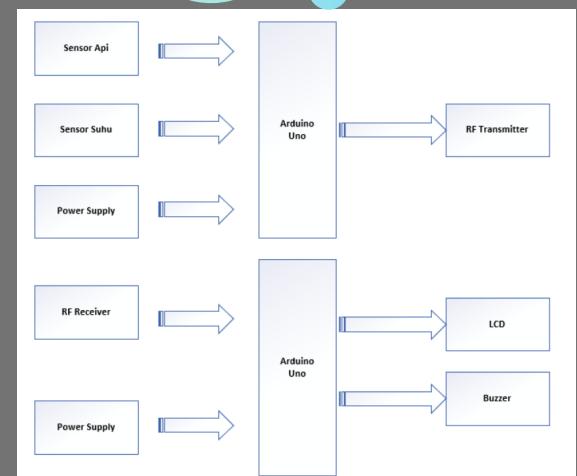
## Desc

Dalam rancangan proyek 1 kami ingin merancang komunikasi nirkabel antara dua papan Arduino menggunakan modul transceiver NRF24L01, yaitu Monitoring Suhu dan Pendeksi kebakaran berbasis nRF24L01& Arduino dengan DHT11.

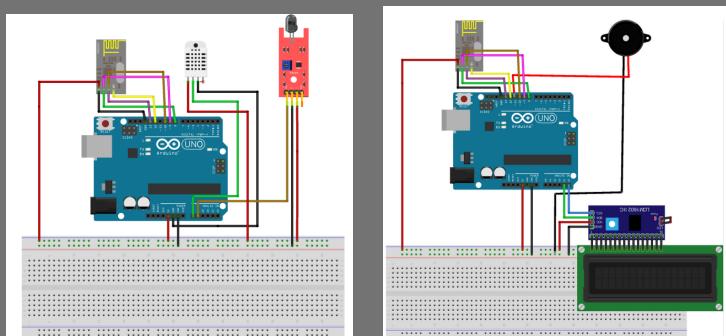
Dengan cara kerja jika suhu yang terdeteksi melebihi yang ditentukan, buzzer akan berbunyi menandakan bahwa suhu sudah melewati ambang batas.

RF pengirim akan mengirimkan data menuju RF penerima berupa data suhu terkini.

## Diagram Block



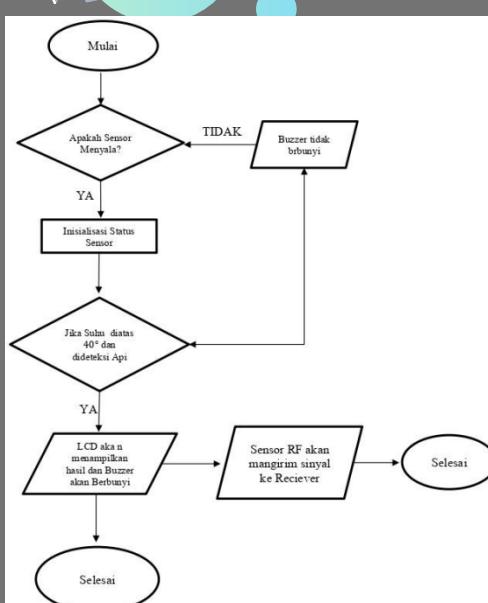
## Wiring Diagram



## Bill of Materials

Bill Of Materials				
No	Description	Num. Units	Cost/Unit	Total Cost
1	Arduino Uno	2	Rp.80.000	Rp.160.000
2	NRF24L01 Transmitter	1	Rp.11.500	Rp.11.500
3	NRF24L01 Receiver	1	Rp.11.500	Rp.11.500
4	Sensor Api (Flame Module)	1	Rp.6.500	Rp.6.500
5	Sensor DHT11	1	Rp.17.500	Rp.17.500
6	Active Piezo Buzzer 5Volt	1	Rp.2.000	Rp.2.000
7	Protoboard	2	Rp.14.500	Rp.29.000
8	Jumper	30	Rp.450	Rp.13.500
9	LCD 16x2	1	Rp.34.000	Rp.34.000
10	Resistor 220 ohm	1	Rp.500	Rp.500
Total Cost				Rp.286.000

## Flowchart



## Picture



# SKETCH

## Tranceiver

```
#include <DHT.h>
#include <DHT_U.h>
#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>
#include <RF24_config.h>
#include <dht.h>

const uint64_t pipeOut = 0xE8E8F0F0E1LL;
#define DHTPIN A0
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);
#define BUZZER 3
RF24 radio(9, 10); // CN and CSN pins of nrf
struct MyData {
    byte h;
    byte t;
};
MyData data;
void setup()
{
    Serial.begin(9600);
    dht.begin();
    radio.begin();
    radio.setAutoAck(false);
    radio.setDataRate(RF24_250KBPS);
    radio.openWritingPipe(pipeOut);
    pinMode(BUZZER, OUTPUT);
}
void loop()
{
    data.h = dht.readHumidity();
    data.t = dht.readTemperature();
    if (isnan(data.h) || isnan(data.t)){
        Serial.println(F("Failed to read from DHT sensor!"));
        return;
    }
    {
        Serial.print("Humidity: ");
        Serial.print(data.h);
        Serial.print("Temperature: ");
        Serial.print(data.t);
        radio.write(&data, sizeof(MyData));
    }
}
```

## Receiver

```
#include <LiquidCrystal_I2C.h>
#include <nRF24L01.h>
#include <RF24.h>
#include <RF24_config.h>
#include <Wire.h>
#include <SPI.h>
#define BUZZERPIN 3
int pesan [1];
LiquidCrystal_I2C lcd(0x27, 16, 2);
const uint64_t pipeIn = 0xE8E8F0F0E1LL;
RF24 radio(9, 10);
struct MyData {
    byte h;
    byte t;
};
MyData data;
void setup()
{
    Serial.begin(9600);
    radio.begin();
    lcd.begin();
    lcd.home();
    lcd.backlight();
    lcd.clear();
    radio.setAutoAck(false);
    radio.setDataRate(RF24_250KBPS);
    radio.openReadingPipe(1, pipeIn);
    radio.startListening();
    //lcd.println("Receiver ");
}
void recvData()
{
    if ( radio.available() ) {
        radio.read(&data, sizeof(MyData));
    }
}
void loop()
{
    recvData();
    Serial.print("Humidity: ");
    Serial.print(data.h);
    lcd.setCursor(0,0);
    lcd.print("Humidity:");
    lcd.print(data.h);
    lcd.print("%");
    lcd.setCursor(0,1);
    Serial.print("Temperature: ");
    Serial.print(data.t);
    lcd.print("Temperature:");
    lcd.print(data.t);
    lcd.print(" C");
    //Serial.print("\n");
    if ((data.t > 0) && (data.t <= 32))
    {
        delay(10);
        digitalWrite(BUZZERPIN, LOW);
    }
    else if ((data.t > 34) && (data.t <= 45))
    {
        delay(10);
        digitalWrite(BUZZERPIN, HIGH);
        tone(3, 500, 1000);
        delay(10);
        digitalWrite(BUZZERPIN, LOW);
    }
}
```