

# Surgeon Safe

**Sistem Pendeteksi Kualitas  
Udara Dalam Memantau  
Sterilitas Ruang Operasi  
Berbasis *Internet Of Things***

MySkill - *Internet of Things*  
Kelompok 8

The logo consists of a teal circle with the text 'MySkill' in white, centered within it. This circle is placed inside a larger white circle, which is set against a dark teal background. The white circle has a subtle drop shadow.

MySkill

# ANGGOTA KELOMPOK 8

## B36 - B40

Wanda Haniyah

B36

[Github](#)

Neysaqilla Auldey Ardyan

B37

[Github](#)

Namira Nur Rifani

B38

[Github](#)

M. Febrian Harnianza

B39

[Github](#)

Hamdan Kamal Fahrudin

B40

[Github](#)

***MENTORED BY KAK WAHYU***

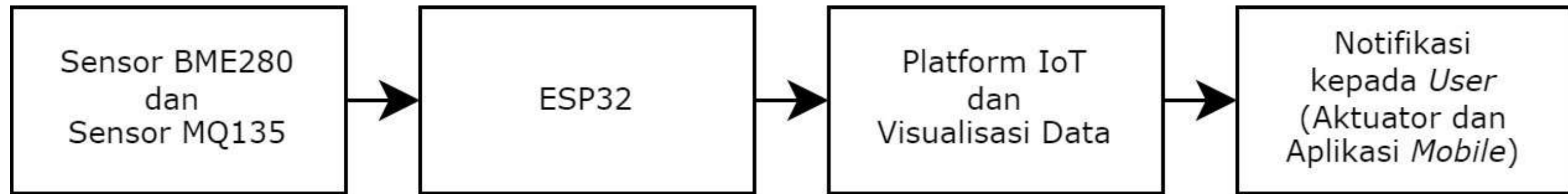
# LATAR BELAKANG

Kualitas udara dalam ruang operasi sangat penting untuk mencegah infeksi dan komplikasi pasca-operasi. Menjaga sterilitas ruang operasi seringkali sulit karena sulitnya memantau kualitas udara secara real-time. Kontaminasi bisa datang dari luar atau dalam ruang operasi. Oleh karena itu, sistem pendeteksi kualitas udara berbasis Internet of Things (IoT) dapat menjadi solusi untuk memantau dan memastikan udara tetap steril, memberikan data akurat secara langsung sehingga tindakan cepat bisa diambil.

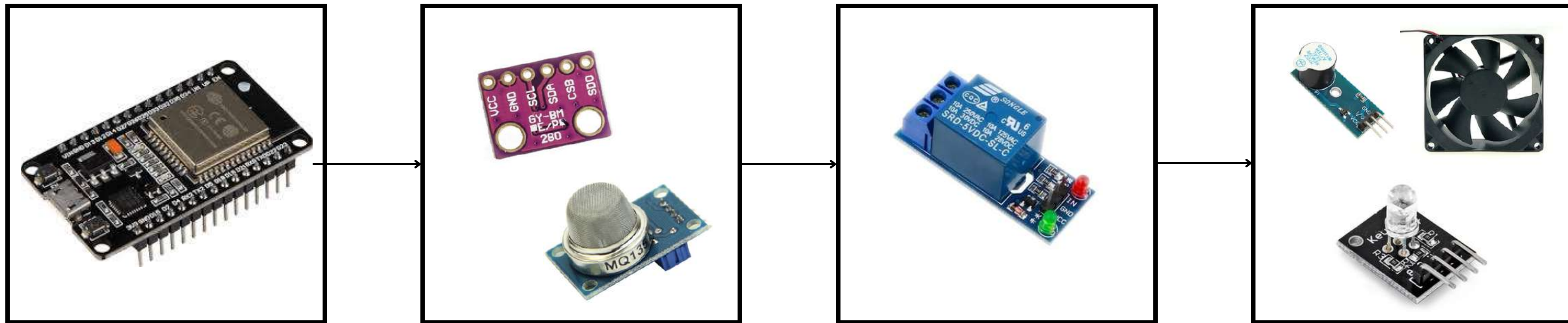
# SURGEON SAFE

Suatu sistem pemantauan *real-time* yang menggunakan sensor MQ135 dan BME280 terintegrasi dengan *board* ESP32 untuk mendeteksi dan mengukur parameter kualitas udara, suhu, dan kelembaban di ruang operasi. Data yang dikumpulkan oleh sensor ini akan dikirim secara nirkabel ke platform IoT yang memungkinkan akses mudah dan pemantauan kontinu melalui aplikasi *mobile*. Sistem ini akan memberikan notifikasi otomatis jika parameter kualitas udara melebihi batas aman yang telah ditentukan, sehingga memungkinkan staf medis untuk segera mengambil tindakan korektif guna menjaga sterilitas dan mengurangi risiko infeksi pasca operasi.

# RANCANGAN SISTEM



# RANCANGAN DESAIN



# RANCANGAN KOMUNIKASI



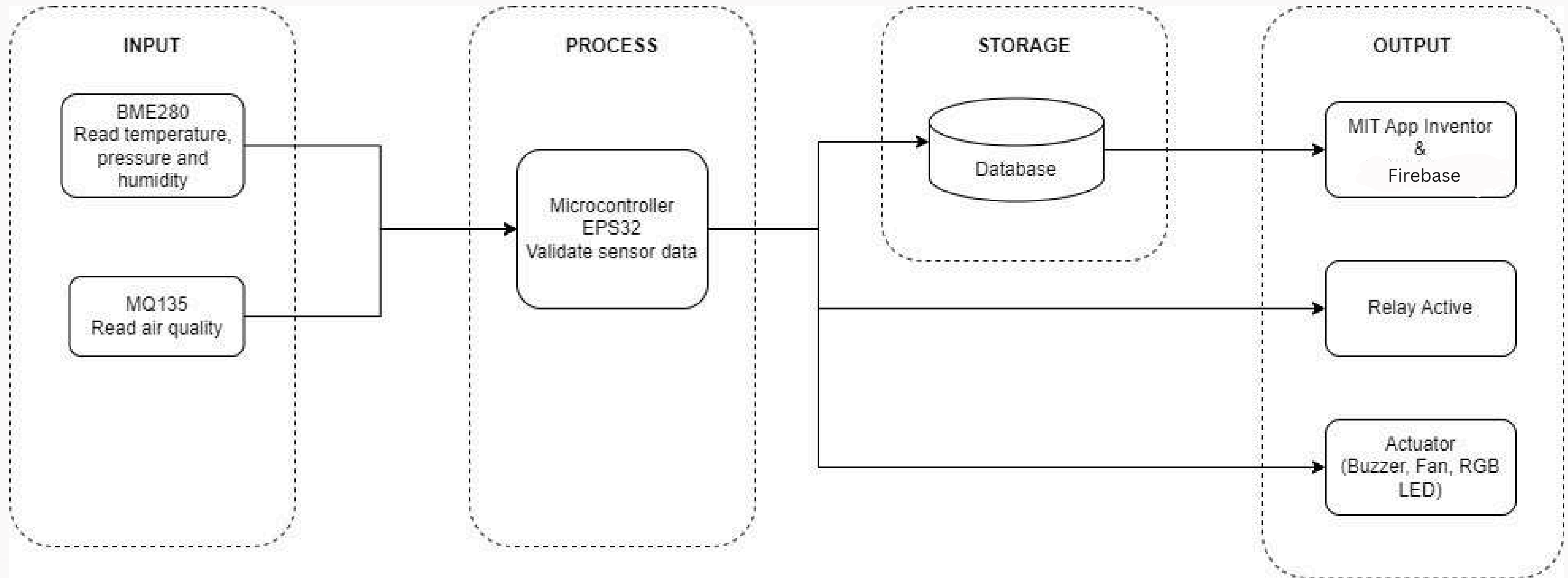
Firestore

Mikrokontroller yang digunakan disini adalah ESP32. Lalu, *platform* yang digunakan untuk perangkat IoT adalah Firebase dan aplikasi (APK) secara *mobile*. Jadi ESP32 harus terkoneksi ke internet untuk transmisi data ke *cloud* (Firestore). Lalu *cloud* tersebut terintegrasi dengan *software* yang menggunakan dari MIT App Inventor.





# ALUR DIAGRAM



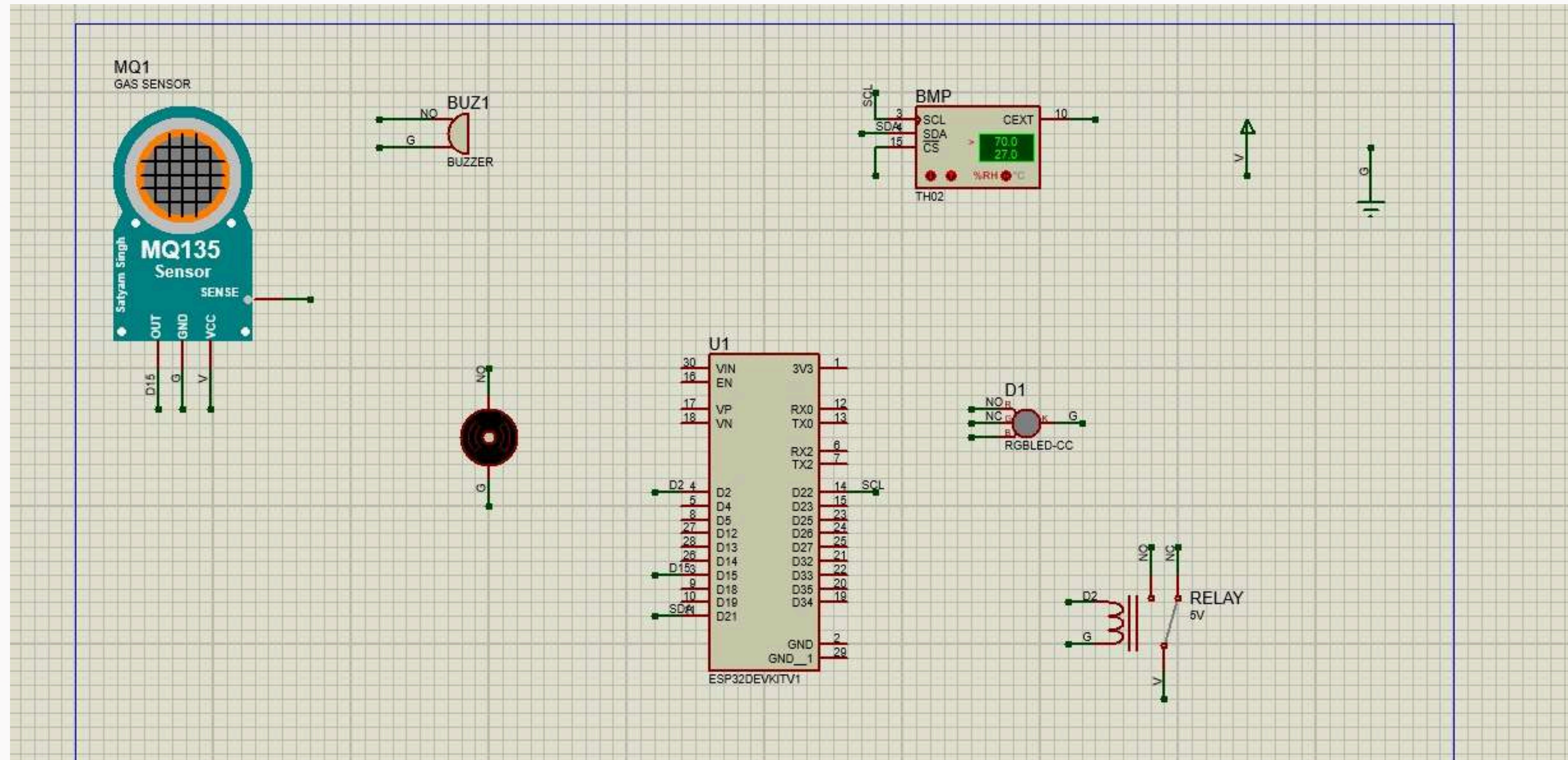


# FLOWCHART DEVICE



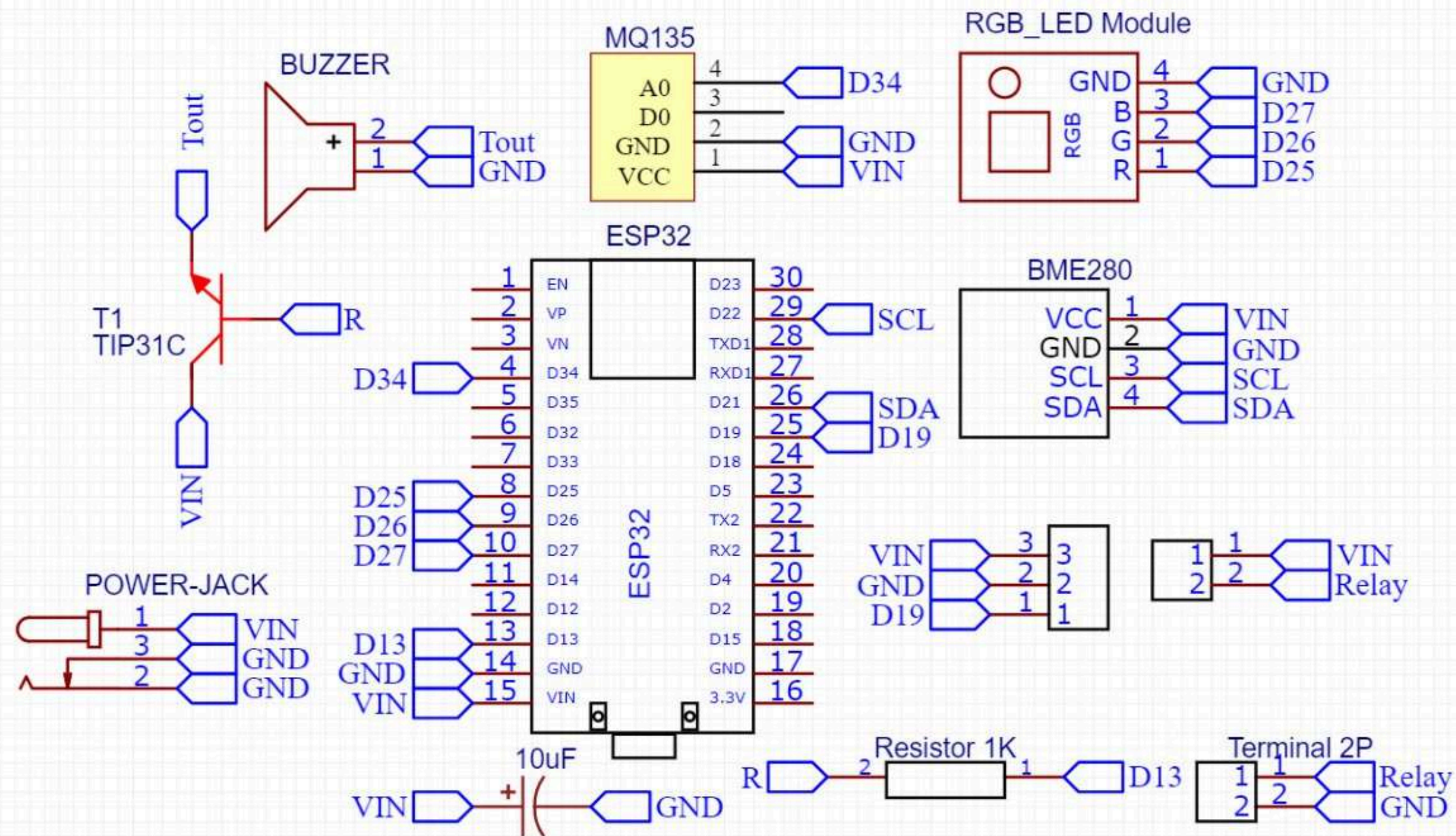


# RANGKAIAN SKEMATIK





# RANGKAIAN SKEMATIK



## KOMPONEN

1. Female Jack DC
2. ESP32
3. Transistor
4. Buzzer
5. Sensor MQ135
6. ELCO 10 uF
7. Sensor BME280 (5V)
8. RGB LED Module
9. Relay Module
10. Actuator (Output)



```

1  #include <ESP32Firebase.h>
2  #include <WiFi.h>
3  #include <WiFiClient.h>
4  #include <Wire.h>
5  #include <Adafruit_Sensor.h>
6  #include <Adafruit_BME280.h>
7  #include "MQUnifiedsensor.h"
8
9  #define Board ("ESP-32")
10 #define Pin (34)
11 #define Type ("MQ-2")
12 #define Voltage_Resolution (3.3)
13 #define ADC_Bit_Resolution (12)
14 #define RatioMQ135CleanAir (3.6)
15
16 MQUnifiedsensor MQ135(Board, Voltage_Resolution, ADC_Bit_Resolution, Pin, Type);
17
18 //define pin
19 #define RELAY_PIN 5 // Ganti dengan pin relay pada ESP32
20 #define RED_PIN 13 // Ganti dengan pin relay pada ESP32
21 #define GREEN_PIN 12 // Ganti dengan pin relay pada ESP32
22 Adafruit_BME280 bme; // I2C
23
24
25 //define wifi
26 #define WIFI_SSID "Rehat Dulu"
27 #define WIFI_PASSWORD "12345678"
28
29 #define API_KEY "AIzaSyC81uLGToGjwvb5fpu95-_5v0JKK3inj1s"
30
31 #define DATABASE_URL "https://kelompok8-f43c7-default-rtdb.firebaseio.com/"
32
33 Firebase firebase(DATABASE_URL);

```

```

34
35 void setup() {
36     //initialize serial
37     Serial.begin(9600);
38     delay(500);
39
40     //initialize bme sensor
41     if (!bme.begin(0x76)) { // Set the address of your BME280 here. Default is 0x77.
42         Serial.println("Could not find a valid BME280 sensor, check wiring!");
43         while (1);
44     }
45
46     //initialize wifi
47     WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
48     while(WiFi.status() != WL_CONNECTED) {
49         Serial.print(".");
50         delay(300);
51     }
52     Serial.println();
53     Serial.print("Connected with IP :");
54     Serial.println(WiFi.localIP());
55     Serial.println();
56
57     //pinmode
58     pinMode(RELAY_PIN, OUTPUT);
59     pinMode(RED_PIN, OUTPUT);
60     pinMode(GREEN_PIN, OUTPUT);
61
62     //initialize mq135
63     MQ135.setRegressionMethod(1);
64     MQ135.setA(110.47);
65     MQ135.setB(-2.862);
66
67     MQ135.init();
68     MQ135.setRL(10);
69

```

# PROGRAM



```

70 float calcR0 = 0;
71 for (int i = 1; i <= 10; i++) {
72     MQ135.update();
73     calcR0 += MQ135.calibrate(RatioMQ135CleanAir);
74 }
75 MQ135.setR0(calcR0 / 10);
76
77 if (isinf(calcR0)) {
78     Serial.println("Peringatan: Masalah koneksi, R0 tak terhingga (Sirkuit terbuka terdeteksi) harap periksa kabel dan suplai");
79     while (1);
80 }
81 if (calcR0 == 0) {
82     Serial.println("Peringatan: Masalah koneksi ditemukan, R0 adalah nol (Pin analog korsleting ke tanah) harap periksa kabel dan suplai");
83     while (1);
84 }
85 }
86
87 void sendToFirebase(float ppm, float temperature, float humidity) {
88     // Set data
89     firebase.setFloat("/DATA/temperature", temperature);
90     firebase.setFloat("/DATA/humidity", humidity);
91     firebase.setFloat("/DATA/ppm", ppm);
92 }
93
94 void bmp() {
95     float temperature = bme.readTemperature(); // Get temperature from BME280
96     float humidity = bme.readHumidity(); // Get humidity from BME280
97     Serial.print("Temperature: ");
98     Serial.print(temperature);
99     Serial.println(" °C, Humidity: ");
100    Serial.print(humidity);
101    Serial.println(" %");
102 }

```

# PROGRAM





```

103
104 void mq() {
105     MQ135.update();
106     float ppm = MQ135.readSensor();
107     Serial.print(ppm);
108     Serial.println(" PPM");
109
110     int value = analogRead(Pin);
111     value = map(value, 0, 4095, 0, 100);
112 }
113
114 void loop(){
115     bmp();
116     mq();
117     delay(1000);
118     float ppm = MQ135.readSensor();
119     float temperature = bme.readTemperature(); // Get temperature from BME280
120     float humidity = bme.readHumidity(); // Get humidity from BME280
121     sendToFirebase(ppm, temperature, humidity);
122
123     if(temperature < 15 || temperature > 30) {
124         // Aksi yang diinginkan jika suhu kurang dari 15 atau lebih dari 30
125         digitalWrite(RELAY_PIN, HIGH);
126         Serial.print("HIGH");
127         digitalWrite(GREEN_PIN, HIGH);
128         firebase.setString("/DATA/Indikator", "Kualitas udara tidak aman! Operasi tidak bisa dilaksanakan");
129     }
130
131     else if(humidity < 30 || humidity > 70) {
132         // Aksi yang diinginkan jika suhu kurang dari 15 atau lebih dari 30
133         digitalWrite(RELAY_PIN, HIGH);
134         Serial.print("HIGH");
135         digitalWrite(GREEN_PIN, HIGH);
136         firebase.setString("/DATA/Indikator", "Kualitas udara tidak aman! Operasi tidak bisa dilaksanakan");
137     }
138

```

```

else if(ppm > 2000) {
    // Aksi yang diinginkan jika suhu kurang dari 15 atau lebih dari 30
    digitalWrite(RELAY_PIN, HIGH);
    Serial.print("HIGH");
    digitalWrite(GREEN_PIN, HIGH);
    firebase.setString("/DATA/Indikator", "Kualitas udara tidak aman! Operasi tidak bisa dilaksanakan");
}

else {
    digitalWrite(RELAY_PIN, LOW);
    Serial.print("LOW");
    digitalWrite(GREEN_PIN, LOW);
    firebase.setString("/DATA/Indikator", "Kualitas udara aman! Operasi bisa dilaksanakan");
}
}

```


# PROGRAM



# PLATFORM IOT (FIREBASE)

 <https://surgeon-safe-default-rtdb.asia-southeast1.firebaseio.com>

<https://surgeon-safe-default-rtdb.asia-southeast1.firebaseio.com/>

DATA + 

- CO2: 205.68866
- Humidity: 63.36914
- Indikator: "Kualitas Udara Aman! Operasi Bisa Dilaksanakan!"
- Temperature: 29.49



# APLIKASI (USER INTERFACE)

Login

**Sistem Monitoring Surgeon Safe**  
*Menggunakan ESP32 dan Firebase Berbasis IoT*

**USERNAME**  
AdminRS

**PASSWORD**  
....

LOGIN

Screen2

**HOME PAGE**  
**SISTEM MONITORING SURGEON SAFE**  
*Menggunakan ESP32 dan Firebase Berbasis IoT*

**MONITORING DATA SENSOR**

BACK

Screen3

**MONITORING DATA SENSOR**  
*Secara Realtime Menggunakan Firebase*

**SENSOR SUHU**  
29.49 C

**SENSOR KELEMBAPAN**  
63.36914 %

**SENSOR CO2**  
205.68866 PPM

BACK

# SYNTAX (LOGIC CODE)

## Screen 1

```
initialize global input_username to ""
initialize global input_pw to ""
initialize global username_stored to "AdminRS"
initialize global pw_stored to "1997"

when BtnLogin . Click
do
  set global input_username to TblUsername . Text
  set global input_pw to TblPw . Text
  if
    get global input_username = get global username_stored and get global input_pw = get global pw_stored
  then
    set LblPrint . Text to "BERHASIL"
    open another screen screenName Screen2
  else
    set LblPrint . Text to "ERROR"
```

The image displays a Scratch-style logic code for a login screen. It begins with four global variables: `input_username` and `input_pw` are initialized to empty strings, while `username_stored` is set to "AdminRS" and `pw_stored` is set to "1997". The main logic is triggered by a `when BtnLogin . Click` event. Inside a `do` loop, the code sets `input_username` to the text of `TblUsername` and `input_pw` to the text of `TblPw`. An `if` statement then checks if `input_username` equals `username_stored` and `input_pw` equals `pw_stored`. If both conditions are met, it sets `LblPrint` to "BERHASIL" and opens `Screen2`. Otherwise, it sets `LblPrint` to "ERROR".

# SYNTAX (LOGIC CODE)

## Screen 3

## Screen 2

```
when BtnSenDat .Click
do open another screen screenName Screen3

when BtnBack .Click
do open another screen screenName Screen1
```

```
when FirebaseDB1 .DataChanged
tag value
do
  if get tag = "Temperature"
  then set LblSuhu .Text to join get value " C "

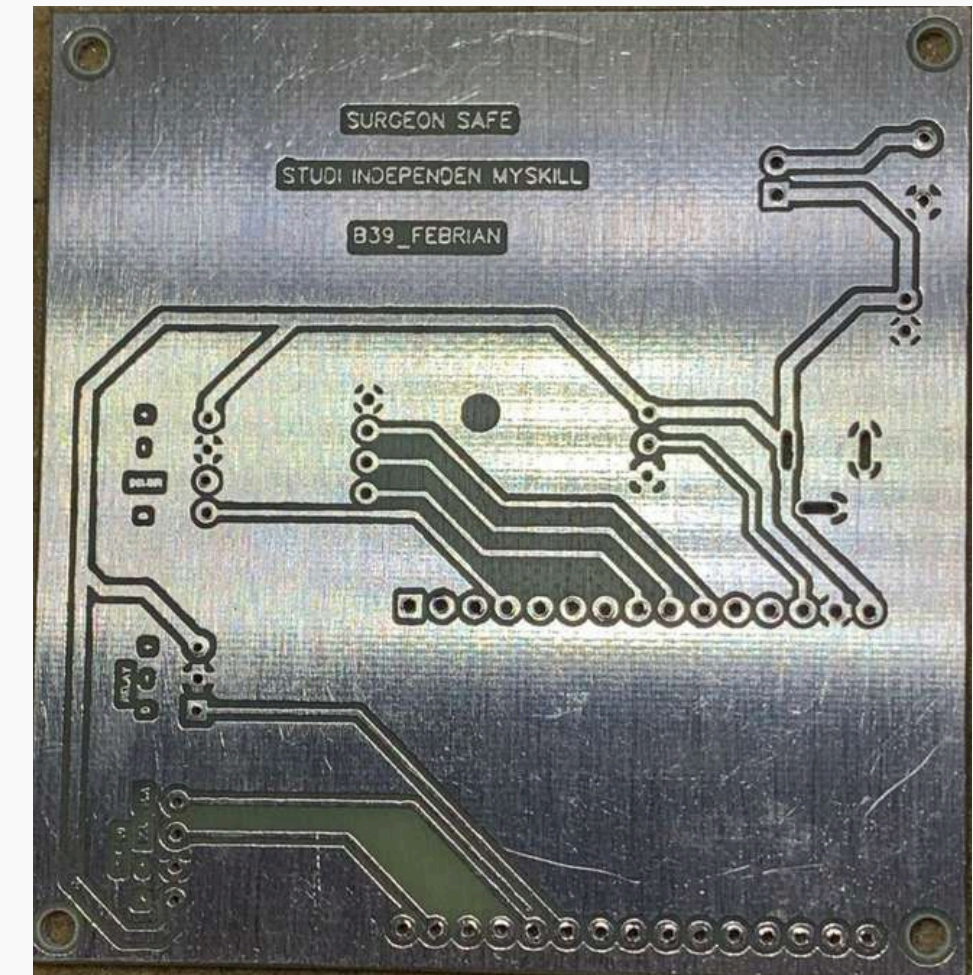
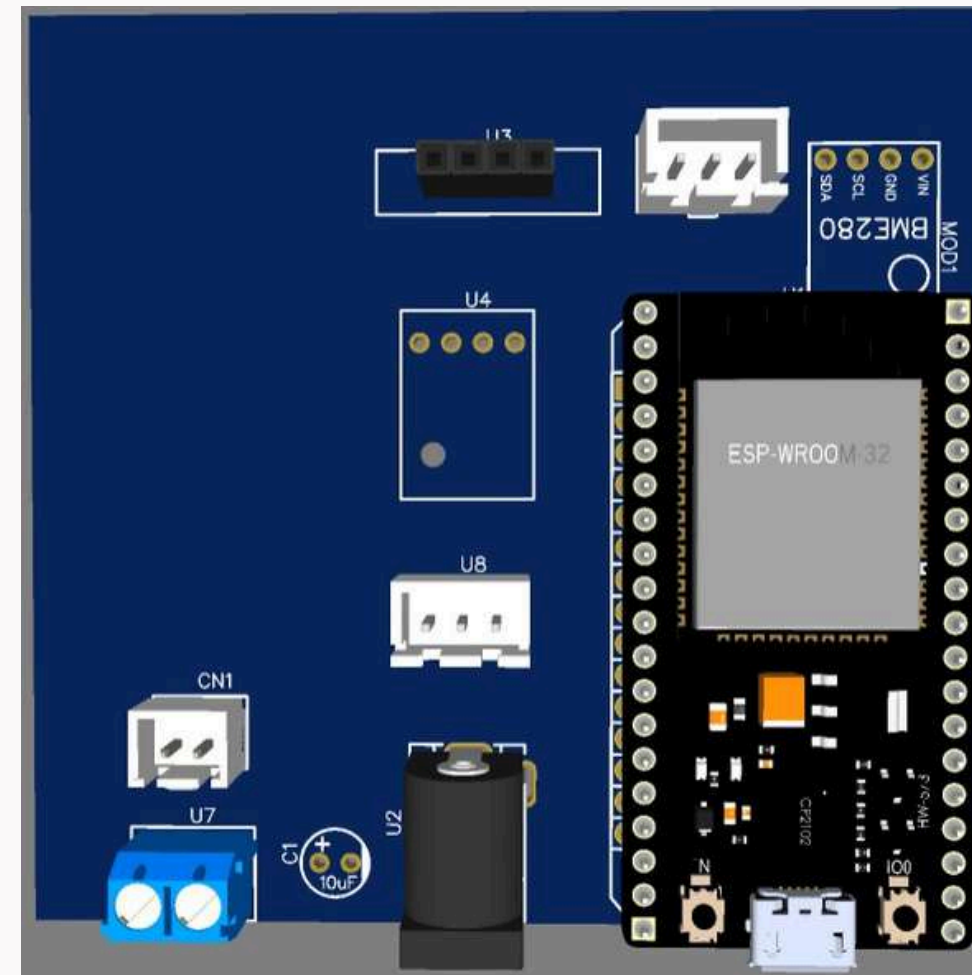
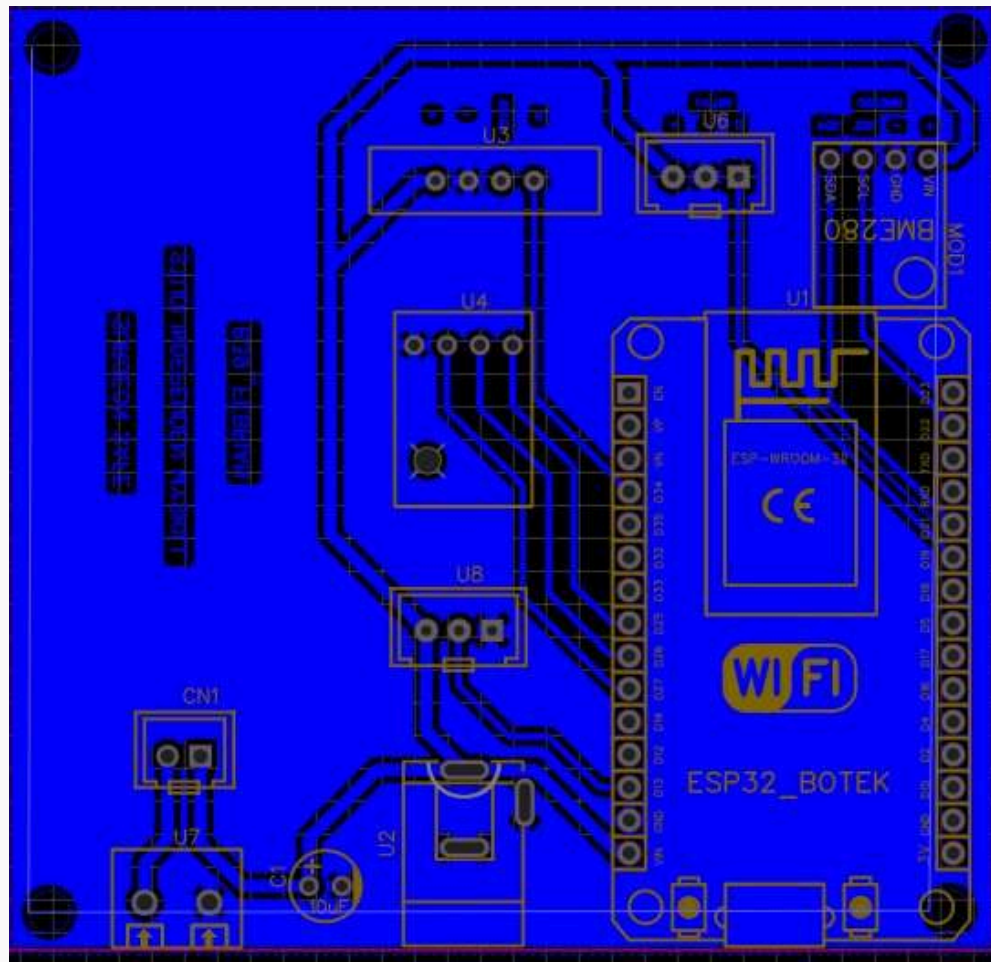
  if get tag = "Humidity"
  then set LblHumi .Text to join get value "% "

  if get tag = "CO2"
  then set LblCO2 .Text to join get value " PPM "

when BtnBack .Click
do open another screen screenName Screen2
```



# BOX ENCLOSURE





# FOTO DOKUMENTASI





**VIDEO UJI COBA:**

**[https://drive.google.com/drive/u/1/folders/1M5n\\_xjaK7rd6IK3RrSce6Y0D5PZzRNGY](https://drive.google.com/drive/u/1/folders/1M5n_xjaK7rd6IK3RrSce6Y0D5PZzRNGY)**



# HASIL UJI FUNGSI

Hasil uji fungsi alat menunjukkan bahwa sistem bekerja dengan baik dan sesuai dengan spesifikasi yang diharapkan. Pemantauan real-time melalui aplikasi berhasil dilakukan dengan respon cepat dan data akurat. Sensor MQ135 mendeteksi kadar CO2 dengan presisi tinggi, sementara sensor BME280 memberikan pengukuran suhu dan kelembaban yang konsisten. Indikator RGB LED berfungsi sesuai kondisi ruangan: merah untuk kualitas udara buruk dan hijau untuk kondisi steril. Buzzer menyala konstan dan fan aktif saat LED merah menyala, menandakan ruangan tidak steril, dan buzzer mati ketika kondisi ruangan baik. Secara keseluruhan, alat ini memberikan indikasi yang jelas dan dapat diandalkan mengenai kualitas udara dan kondisi lingkungan ruangan.





THANK  
YOU