

DAFTAR PUSTAKA

- Charlie, K., Primananda, R., & Data, M. (2018). Analisis Performa Load Balancing Pada Broker MQTT Menggunakan Algoritma Round Robin. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 2(9), 2902–2908.
- Dewi, N. H. L., Rohmah, M. F., & Mahasiswa, S. Z. (2019). Prototype Smart Home Dengan Modul Nodemcu Esp8266 Berbasis Internet of Things (Iot). *Teknologi Informasi*, 3–3.
- ETSI. (1999). Telecommunications and Internet Protocol Harmonization Over Networks (TIPHON); General aspects of Quality of Service (QoS). France: European Telecommunications Standards Institute.
- Fernando, R. & S. (2023). In-Car Air Quality Notification Using Internet of Things Platform. *Engineering, Mathematics and Computer Science (EMACS) Journal*, 5(2), 53–61. <https://doi.org/10.21512/emacsjournal.v5i2.9950>
- Gessal, C. I. Y., Lumenta, A. S. M., & Sugiarto, B. A. (2019). Kolaborasi Aplikasi Android Dengan Sensor Mq-135 Melahirkan Detektor Polutan Udara. *Jurnal Teknik Informatika*, 14(1), 109–120.
- Goh, C. C., Kamarudin, L. M., Zakaria, A., Nishizaki, H., Ramli, N., Mao, X., Zakaria, S. M. M. S., Kanagaraj, E., Sukor, A. S. A., & Elham, M. F. (2021). Real-time in-vehicle air quality monitoring system using machine learning prediction algorithm. *Sensors*, 21(15), 1–16. <https://doi.org/10.3390/s21154956>
- Kurniawan, D. E., Iqbal, M., Friadi, J., Borman, R. I., & Rinaldi, R. (2019, November). Smart monitoring temperature and humidity of the room server using raspberry pi and whatsapp notifications. In *Journal of Physics: Conference Series* (Vol. 1351, No. 1, p. 012006). IOP Publishing.



, D. setia & E. F. (2020). Rancang Bangun Prototype Robot Penghisap u Menggunakan Optical Dust Sensor Gp2Y1010Au0F. *Rancang Bangun*
at, C. A. (2021). *RANCANG BANGUN SISTIM KEAMANAN*

PERANGKAT IoT DENGAN METODE AUTENTIKASI MENGGUNAKAN JSON WEB TOKEN PADA PROTOKOL MQTT.

- Nizam, M., Yuana, H., & Wulansari, Z. (2022). MIKROKONTROLLER ESP 32 SEBAGAI ALAT MONITORING PINTU BERBASIS WEB. *Jurnal Teknik Informatika*, 6(2), 767–772.
- Pengaruh Polusi Karbon Monoksida bagi Kesehatan. (2023). *Kemendes Ditjen Yankes*. https://yankes.kemkes.go.id/view_artikel/2575/pengaruh-polusi-karbon-monoksida-bagi-kesehatan
- Pratama, A. P., Maulana, F. S., & Kusumadiarti, R. S. (2021). Sistem Pemantauan Ketebalan Debu & Suhu Pada Ruangan Menggunakan Aplikasi Telegram berbasis IoT. *JATISI (Jurnal Teknik Informatika Dan Sistem Informasi)*, 8(4), 2198–2211. <https://doi.org/10.35957/jatisi.v8i4.1191>
- Prayoga, I. D. T. S. (2020). *Sistem Monitoring Kualitas Udara Secara Realtime Dengan Peringatan Bahaya Kualitas Udara Tidak Sehat Menggunakan Push Notification*. 7(2), 9–19.
- Rangan, A. Y., Amelia Yusnita, & Muhammad Awaludin. (2020). Sistem Monitoring berbasis Internet of things pada Suhu dan Kelembaban Udara di Laboratorium Kimia XYZ. *Jurnal E-Komtek (Elektro-Komputer-Teknik)*, 4(2), 168–183. <https://doi.org/10.37339/e-komtek.v4i2.404>
- Rizon, F. M., & Sarmidi. (2018). Alat Pendeteksi Udara Di Dalam Mobil Menggunakan Arduino Uno. *Jumantaka*, 02(01), 31–40.
- Rostanti, Q. (2023). Ayah dan Anak Ditemukan tak Bernyawa di Mobil, Diduga Keracunan Gas CO dari AC. *Republika*. <https://news.republika.co.id/berita/rxd5z3425/ayah-dan-anak-ditemukan-tak-bernyawa-di-mobil-diduga-keracunan-gas-co-dari-ac>

Saputra, H. A., Pohny, & Saputra, G. M. (2020). Analisis QOS Jaringan 4G Dengan menggunakan Aplikasi Wireshark (Studi Kasus : Tepian Samarinda, Taman arinda, dan Taman Cerdas). Prosiding Seminar Nasional Ilmu Komputer Teknologi Informasi, 13-18



Saputra, R. A. D. (2017). *Prototipe Pengendali Kualitas Udara Indoor Berbasis Mikrokontroller ATMEGA 328P*. 2.

Satryawan, M. A., & Susanti, E. (2023). PERANCANGAN ALAT PENDETEKSI KUALITAS UDARA DENGAN IoT (Internet of Things) MENGGUNAKAN WEMOS ESP32 D1 R32. *Sigma Teknika*, 6(2), 410–419. <https://doi.org/10.33373/sigmateknika.v6i2.5646>

Sreenivasan, V. (2022). What are PM2.5 and PM10? How They Affect You and Solutions. In *Devic Earth*. <https://www.devic-earth.com/blog/what-are-pm2-5-and-pm10-how-they-affect-you-and-solutions>

Sutarti, Siswanto, & Subandi, A. (2018). Implementasi Dan Analisis QoS (Quality of Service) Pada VoIP (Voice Over Internet Protocol) Berbasis Linux. *Jurnal*

Wahyudi, A. (2020). *Alat Pendeteksi Co Dengan Sensor Mq7* (pp. 1–82).

Waworundeng, J. M. S., & Lengkong, O. (2018). Sistem Monitoring dan Notifikasi Kualitas Udara dalam Ruangan dengan Platform IoT. *CogITo Smart Journal*, 4(1), 94–103. <https://doi.org/10.31154/cogito.v4i1.105.94-103>

Wulandari, R. (2016). ANALISIS QoS (QUALITY OF SERVICE) PADA JARINGAN INTERNET (STUDI KASUS : UPT LOKA UJI TEKNIK PENAMBANGAN JAMPANG KULON-LIPI). *Jurnal Teknik Informatika dan Sistem Informasi*, 162-172

Yulianto, S. V., Setia, L. D., & Atmaja, A. P. (2021, March). The use of *whatsapp* gateway for automatic notification system. In *Journal of Physics: Conference Series* (Vol. 1845, No. 1, p. 012014). IOP Publishing.



LAMPIRAN

Lampiran 1 dokumentasi kalibrasi sensor MQ135 dan MQ7 dan AQI



Lampiran 2 dokumentasi kalibrasi sensor DHT11 dan HTC



Lampiran 3 dokumentasi monitoring LCD dan *Blynk*



Lampiran 4 dokumentasi pengambilan data



Lampiran 5 program pengirim pada Arduino IDE

```
#define BLYNK_TEMPLATE_ID "TMPL6Pu-o7v_0"
#define BLYNK_TEMPLATE_NAME "Tugas Akhir Iffah"
#define BLYNK_AUTH_TOKEN "H3ts9V4ybHTWuwhJfugo9LtoYS0hg9di"

#include <TinyGPS++.h> // Library untuk membaca data GPS
#include <HardwareSerial.h> // Library untuk komunikasi
serial dengan GPS
#include <DHT.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <BlynkSimpleEsp32.h> // Library Blynk untuk ESP32
#include <HTTPClient.h> // Library untuk koneksi HTTP

#define DHTPIN 4 // Pin tempat sensor DHT11 terhubung
#define DHTTYPE DHT11 // Tipe sensor DHT (DHT11, DHT22,
DHT21)
DHT dht(DHTPIN, DHTTYPE);

#define MQ7PIN 35 // Pin tempat sensor MQ7 terhubung
#define MQ135PIN 34 // Pin tempat sensor MQ135
terhubung
#define GPS_TX 16 //
#define GPS_RX 17

HardwareSerial gpsSerial(1); // Inisialisasi komunikasi
serial dengan GPS menggunakan Serial1 pada ESP32
TinyGPSPlus gps; // Objek untuk membaca data GPS
```



```
return nilai awal
Value = 0;
```

```

int MQ135Value = 0;
float CO = 0;
float AQI = 0;

LiquidCrystal_I2C lcd(0x27, 20, 4); // Alamat I2C dan
ukuran LCD (20x4)
WidgetLCD lcd_widget(V5); // Widget LCD pada Blynk

char ssid[] = "Wanda"; // Nama WiFi Anda
char pass[] = "12345678"; // Password WiFi Anda

char auth[] = BLYNK_AUTH_TOKEN; // Token otentikasi Blynk

BlynkTimer timer;

bool isWifiConnected = false; // Indikator koneksi WiFi
bool isBlynkConnected = false; // Indikator koneksi ke Blynk

int buzzerPin = 32; // Pin untuk buzzer

String getGoogleMapsLink();

void kirimPesanWhatsapp() {
  HTTPClient http;
  String url =
    "https://api.callmebot.com/whatsapp.php?phone=6287865104219&
    text=Perhatian!!!+Kualitas+udara+di+dalam+mobil+Anda+sedang
    +buruk.+Pastikan+untuk+membuka+kaca+secara+parsial++CO:";
    url += String(CO) + "+ppm,+C02:";
    url += String(AQI) + "ppm++";
    url += "Lokasi:";
    url += getGoogleMapsLink() + "&apikey=7788288";
  http.begin(url);

```



```

int httpCode = http.GET();

if (httpCode > 0) {
    if (httpCode == HTTP_CODE_OK || httpCode ==
HTTP_CODE_CREATED) {
        Serial.println("Pesan Whatsapp berhasil dikirim");
    } else {
        Serial.println("Gagal mengirim pesan Whatsapp, kode
respons: " + String(httpCode));
    }
    String payload = http.getString();
    Serial.println("Respons server Whatsapp: " + payload);
} else {
    Serial.println("Gagal terhubung ke server Whatsapp");
}

http.end();
}

void kirimDataSensor() {
    float suhu = dht.readTemperature();
    float kelembaban = dht.readHumidity();

    if (isnan(suhu) || isnan(kelembaban)) {
        Serial.println("Gagal membaca sensor DHT11");
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("Error: DHT11");
        return;
    }
}

```



```

MQ7Value = analogRead(MQ7PIN);
CO = (float)MQ7Value / 1023.0 * 10.0;
MQ135Value = analogRead(MQ135PIN);
AQI = map(MQ135Value, 0, 1023, 0, 2046);

```

```

lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Tem :");
lcd.setCursor(6, 0);
lcd.print(suhu);
lcd.print(" C");

lcd.setCursor(0, 1);
lcd.print("Hum : ");
lcd.setCursor(6, 1);
lcd.print(kelembaban);
lcd.print(" %");

```

```

lcd.setCursor(0, 2);
lcd.print("CO  : ");
lcd.setCursor(6, 2);
lcd.print(CO);
lcd.print(" ppm");

```

```

lcd.setCursor(0, 3);
lcd.print("C02 : ");
lcd.setCursor(6, 3);
lcd.print(AQI);
lcd.print(" ppm");

```

```

l1.print("tem: ");
l1.print(suhu);

```



```

Serial.println(" C");

Serial.print("Hum: ");
Serial.print(kelembaban);
Serial.println(" %");

Serial.print("C0: ");
Serial.print(AQI);
Serial.println(" ppm");

Serial.print("CO2: ");
Serial.print(CO);
Serial.println(" ppm");

if (isBlynkConnected) {
    Blynk.virtualWrite(V1, suhu);
    Blynk.virtualWrite(V2, kelembaban);
    Blynk.virtualWrite(V3, AQI);
    Blynk.virtualWrite(V4, CO);
}

if (CO >= 35 || AQI >= 1000) {
    digitalWrite(buzzerPin, HIGH);
    kirimPesanWhatsapp();
} else {
    digitalWrite(buzzerPin, LOW);
}

```



```

CONNECTED() {
    isBlynkConnected = true;
}

```

```

    Serial.println("Terhubung ke server Blynk.");
}

BLYNK_DISCONNECTED() {
    isBlynkConnected = false;
    Serial.println("Terputus dari server Blynk.");
}

void setup() {
    Serial.begin(115200);
    dht.begin();
    pinMode(MQ7PIN, INPUT);
    pinMode(MQ135PIN, INPUT);
    pinMode(buzzerPin, OUTPUT);

    lcd.init();
    lcd.backlight();
    lcd.setCursor(1, 1);
    lcd.print("Menghubungkan WiFi.");
    delay(1000);

    Blynk.begin(auth, ssid, pass);

    gpsSerial.begin(9600, SERIAL_8N1, GPS_RX, GPS_TX); //
    Inisialisasi komunikasi serial dengan GPS
    gpsSerial.setTimeout(2000); // Set timeout serial GPS

    timer.setInterval(5000L, kirimDataSensor);

```



```

    op() {
        .run();
    }

```

```

    timer.run();

    while (gpsSerial.available() > 0) {
        if (gps.encode(gpsSerial.read())) {
            Serial.print("Latitude: ");
            Serial.println(gps.location.lat(), 6);
            Serial.print("Longitude: ");
            Serial.println(gps.location.lng(), 6);
            // Do nothing, just read GPS data
        }
    }
    //}

    if (WiFi.status() == WL_CONNECTED && !isWifiConnected) {
        isWifiConnected = true;
        lcd.clear();
        lcd.setCursor(3, 1);
        lcd.print("WiFi Terhubung");
        delay(1000);
        lcd.setCursor(0, 1);
        lcd.print(" Terhubung ke Blynk ");

    } else if (WiFi.status() != WL_CONNECTED) {
        isWifiConnected = false;
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("Menghubungkan WiFi...");
    }
}

String getGoogleMapsLink() {
    String latitude = String(gps.location.lat(), 6);
    String longitude = String(gps.location.lng(), 6);

```



```
String link = "www.google.com/maps?q=" + latitude + "," +  
longitude;  
    return link;  
}
```

