**KELOMPOK3**

**Arduino**

#include <ESP8266WiFi.h>

#include <PubSubClient.h>

#include <ArduinoJson.h>

#include <OneWire.h> //Library OneWire

#include <DallasTemperature.h> // Library untuk Dallas temperature

#define buzzer D3 //pin D3 untuk buzzer

#define ONE\_WIRE\_BUS D4 // pin D4 digunakan untuk data hasil baca sensor

OneWire oneWire(ONE\_WIRE\_BUS);

DallasTemperature sensors(&oneWire);

WiFiClient espClient;

PubSubClient client(espClient);

#define WIFI\_SSID "KELOMPOK3IOT"

#define WIFI\_PASS ""

#define MQTT\_SERVER "broker.hivemq.com"

#define MQTT\_PORT 1883

void setup\_wifi() {

Serial.println();

Serial.print("Connecting to ");

Serial.println(WIFI\_SSID);

WiFi.begin(WIFI\_SSID);

while (WiFi.status() != WL\_CONNECTED) {

delay(500);

Serial.print(".");}

Serial.println("");

Serial.println("WiFi connected");

Serial.println("IP address: ");

Serial.println(WiFi.localIP());}

float suhu;

unsigned long \_waiting = millis();

unsigned long \_now;

int value = 0;

char data[50];

String buffData;

void callback(char\* topic, byte\* payload, unsigned int length) {

buffData = "";

for (int i=0; i<length; i++){

buffData += (char) payload[i];}

Serial.print(topic);

Serial.print(" ==> ");

Serial.println(buffData);

StaticJsonDocument<200> doc;

deserializeJson(doc,payload);}

void reconnect() {

while (!client.connected()) {

Serial.print("Attempting MQTT connection...");

String clientId = "Kelompok-3";

clientId += String(random(0xffff), HEX);

if (client.connect(clientId.c\_str())) {

Serial.println("connected");

client.publish("sukses\_konek", "Yess... saya terkoneksi");

client.subscribe("kelompok-3/act");

} else {

Serial.print("failed, rc=");

Serial.print(client.state());

Serial.println(" Try again In 5 Seconds");

delay(5000);}}}

void buzzerNyala(){ //Function untuk menyalakan buzzer

digitalWrite(buzzer, HIGH);

delay (10000);

digitalWrite(buzzer, LOW);

delay (500);}

void setup() {

Serial.begin(115200);

sensors.begin(); //memulai baca sensor

pinMode(buzzer, OUTPUT); //inisialisasi buzzer sebagai output

setup\_wifi();

client.setServer(MQTT\_SERVER, MQTT\_PORT);

client.setCallback(callback);}

void loop() {

if (!client.connected()) {

reconnect();}

if (suhu > 30) {

buzzerNyala(); //menyalakan buzzer}

else if (suhu < 25) {

buzzerNyala(); //menyalakan buzzer}

else {

digitalWrite(buzzer, LOW);}

client.loop();

kirimPer2menit();}

void kirimPer2menit(){

\_now = millis();

if(millis() - \_waiting > 10000){

\_waiting = \_now;

sensors.requestTemperatures(); //baca sensor

suhu = sensors.getTempCByIndex(0);

sprintf(data, "%g",

sensors.getTempCByIndex(0));

Serial.print("Publish message : ");

Serial.println(data);

client.publish("kelompok-3/sensor/temp", data);}}

**thonny**

import paho.mqtt.client as mqtt

import mysql.connector

import json

mydb = mysql.connector.connect(

host = "localhost",

user = "root",

password = "",

database = "projectiot")

mycursor = mydb.cursor()

def on\_connect(client, userdata, flags, rc):

print("Connected with result code "+str(rc))

client.subscribe("kelompok-3/sensor/temp")

def on\_message(client, userdata, msg):

arr = []

arr = str(msg.payload, 'utf-8').split('#');

suhu = arr[0];

cetak = '';

payload = '';

sql = "INSERT INTO project (suhu) VALUES (%s)"

val = (suhu,)

mycursor.execute(sql, val)

mydb.commit()

client = mqtt.Client()

client.on\_connect = on\_connect

client.on\_message = on\_message

client.connect("broker.hivemq.com", 1883, 60)

client.loop\_forever()

**KELOMPOK 1**

**Arduino**

#include <ESP8266WiFi.h>

#include <PubSubClient.h>

#include <ArduinoJson.h>

#include <Adafruit\_MLX90614.h> // Sensor Mlx90614

#define buzzer D4 // Pin D4`

Adafruit\_MLX90614 mlx = Adafruit\_MLX90614();

WiFiClient espClient;

PubSubClient client(espClient);

#define WIFI\_SSID "vivo 1610"

#define WIFI\_PASS "insankamil"

#define MQTT\_SERVER "broker.hivemq.com"

#define MQTT\_PORT 1883

float sh;

void setup\_wifi() {

Serial.println();

Serial.print("Connecting to ");

Serial.println(WIFI\_SSID);

WiFi.begin(WIFI\_SSID,WIFI\_PASS);

while (WiFi.status() != WL\_CONNECTED) {

delay(500);

Serial.print(".");

Wifi Terkonneksi }

Serial.println("");

Serial.println("WiFi connected");

Serial.println("IP address: ");

Serial.println(WiFi.localIP());

}int value = 0;

char data[10000];

String buffData;

void callback(char\* topic, byte\* payload, unsigned int length) {

buffData = "";

for (int i=0; i<length; i++){

buffData += (char) payload[i];}

Serial.print(topic);

Serial.print(" ==> ");

Serial.println(buffData);}

void reconnect() {

while (!client.connected()) {

Serial.print("Attempting MQTT connection...");

String clientId = "Kelompok-1";

clientId += String(random(0xffff), HEX);

if (client.connect(clientId.c\_str())) {

Serial.println("connected");

client.publish("Succesfully Connection....", "Terkonneksi");

} else {

Serial.print("failed, rc=");

Serial.print(client.state());

Serial.println(" Try again In 5 Seconds");

delay(5000);}}}

void setup() {

Serial.begin(9600);

setup\_wifi();

client.setServer(MQTT\_SERVER, MQTT\_PORT);

client.setCallback(callback);

pinMode(buzzer, OUTPUT);

while (!Serial);

Serial.println("Test Sensor Mlx90614");

if (!mlx.begin()) {

Serial.println("Error connecting to MLX sensor. Check wiring.");

while (1);};}

void kirimdatademam(){

sh = mlx.readObjectTempC();

sprintf(data, "\tSuhu Badan: %g C --ANDA DEMAM--",mlx.readObjectTempC());

Serial.print(data);

client.publish("kelompok-1/sensor/badan-demam", data);}

void kirimdata(){

sh = mlx.readObjectTempC();

sprintf(data, "Suhu Badan: %g C ",mlx.readObjectTempC());

Serial.print("Thermometer HEN: ");

Serial.print(data);

client.publish("kelompok-1/sensor/badan", data);}

// Eksekusi Data Program

void loop() {

if (!client.connected()) {

reconnect();

}client.loop();

kirimdata();

sh = mlx.readObjectTempC();

// jika suhu lebih dari 37 derajat maka buzzer berbunyi

if (sh > 37){

digitalWrite(buzzer, HIGH);

delay(100);

digitalWrite(buzzer, LOW);

**Thonny**

import paho.mqtt.client as mqtt

import mysql.connector

import json

mydb = mysql.connector.connect(

host = "localhost",

user = "root",

password = "",

database = "proyekiot" )

mycursor = mydb.cursor()

def on\_connect(client, userdata, flags, rc):

print("Connected with result code "+str(rc))

client.subscribe("kelompok-1/sensor/badan")

client.subscribe("kelompok-1/sensor/badan-demam")

def on\_message(client, userdata, msg):

arr = []

arr = str(msg.payload, 'utf-8').split('#');

sh = arr[0];

cetak = '';

payload = '';

sql = "INSERT INTO datasuhu(suhubadan) VALUES (%s)"

val = (sh,)

mycursor.execute(sql, val)

mydb.commit()

client = mqtt.Client()

client.on\_connect = on\_connect

client.on\_message = on\_message

client.connect("broker.hivemq.com", 1883, 60)

client.loop\_forever()

**KELOMPOK 4**

**Arduino code**

#include <ESP8266WiFi.h>

#include <PubSubClient.h>

#include <ArduinoJson.h>

WiFiClient espClient;

PubSubClient client(espClient);

int pinSensor = D1;

int pinBuzzer = D4;

int pinLED = D2;

#define WIFI\_SSID "Galaxy A30sA8D7"

#define WIFI\_PASS "hana123456"

#define MQTT\_SERVER "broker.hivemq.com"

#define MQTT\_PORT 1883

void setup\_wifi() {

Serial.println();

Serial.print("Connecting to ");

Serial.println(WIFI\_SSID);

WiFi.begin(WIFI\_SSID, WIFI\_PASS);

while (WiFi.status() != WL\_CONNECTED) {

delay(500);

Serial.print(".");}

Serial.println("");

Serial.println("WiFi connected");

Serial.println("IP address: ");

Serial.println(WiFi.localIP());}

float nilai;

unsigned long \_waiting = millis();

unsigned long \_now;

int value = 0;

char data[50];

String buffData;

void callback(char\* topic, byte\* payload, unsigned int length) {

buffData = "";

for (int i=0; i<length; i++){

buffData += (char) payload[i];}

Serial.print(topic);

Serial.print(" ==> ");

Serial.println(buffData);

StaticJsonDocument<200> doc;

deserializeJson(doc,payload);}

void reconnect() {

while (!client.connected()) {

Serial.print("Attempting MQTT connection...");

String clientId = "Kelompok-4";

clientId += String(random(0xffff), HEX);

if (client.connect(clientId.c\_str())) {

Serial.println("connected");

client.publish("sukses\_konek", "Yess... saya terkoneksi");

client.subscribe("kelompok-4/act/led");

} else {

Serial.print("failed, rc=");

Serial.print(client.state());

Serial.println(" Try again In 5 Seconds");

delay(5000);}}}

void setup() {

Serial.begin(115200);

pinMode(pinSensor, INPUT);

pinMode(pinBuzzer, OUTPUT);

pinMode(pinLED, OUTPUT);

setup\_wifi();

client.setServer(MQTT\_SERVER, MQTT\_PORT);

client.setCallback(callback);}

void loop() {

if (!client.connected()) {

reconnect();}

if (nilai < 100) {

digitalWrite(pinBuzzer, HIGH);

digitalWrite(pinLED, HIGH);

Serial.println("Bahaya Ada Api");

} else {

digitalWrite(pinBuzzer, LOW);

digitalWrite(pinLED, LOW);

Serial.println("Tenang Aman"); }

delay(1000);

client.loop();

kirimPermenit();}

void kirimPermenit(){

\_now = millis();

if(millis() - \_waiting > 2000){

\_waiting = \_now;

nilai = analogRead(pinSensor);

sprintf(data, "%g",

nilai);

Serial.print("Publish message : ");

Serial.println(data);

client.publish("kelompok-4/sensor/api", data);}}

**Thoni Code**

import paho.mqtt.client as mqtt

import mysql.connector

import json

mydb = mysql.connector.connect(

host = "localhost",

user = "root",

password = "",

database = "pendeteksiapi\_mqtt")

mycursor = mydb.cursor()

def on\_connect(client, userdata, flags, rc):

print("Connected with result code "+str(rc))

client.subscribe("kelompok-4/sensor/api")

def on\_message(client, userdata, msg):

arr = []

arr = str(msg.payload, 'utf-8').split('#');

sensorValue = arr[0];

cetak = '';

payload = '';

sql = "INSERT INTO project (nilai) VALUES (%s)"

val = (sensorValue,)

mycursor.execute(sql, val)

mydb.commit()

client = mqtt.Client()

client.on\_connect = on\_connect

client.on\_message = on\_message

client.connect("broker.hivemq.com", 1883, 60)

client.loop\_forever()

`waktu` timestamp NOT NULL DEFAULT current\_timestamp() ON UPDATE current\_timestamp(),

`nilai` varchar(1) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

**KELOMPOK 5**

**Arduino**

#include <ESP8266WiFi.h>

#include <PubSubClient.h>

#include <ArduinoJson.h>

#include <I2CSoilMoistureSensor.h> //

#define pinDigital A0

#define buzzer D1 //pin D3 untuk buzzer

WiFiClient espClient;

PubSubClient client(espClient);

#define WIFI\_SSID "METHA"

#define WIFI\_PASS "gulabatu"

#define MQTT\_SERVER "broker.hivemq.com"

#define MQTT\_PORT 1883

void setup\_wifi() {

Serial.println();

Serial.print("Connecting to ");

Serial.println(WIFI\_SSID);

WiFi.begin(WIFI\_SSID,WIFI\_PASS);

while (WiFi.status() != WL\_CONNECTED) {

delay(500);

Serial.print(".");}

Serial.println("");

Serial.println("WiFi connected");

Serial.println("IP address: ");

Serial.println(WiFi.localIP());}

float kelembapan;

unsigned long \_waiting = millis();

unsigned long \_now;

int value = 0;

char data[50];

String buffData;

void callback(char\* topic, byte\* payload, unsigned int length) {

buffData = "";

for (int i=0; i<length; i++){

buffData += (char) payload[i];}

Serial.print(topic);

Serial.print(" ==> ");

Serial.println(buffData);

StaticJsonDocument<200> doc;

deserializeJson(doc,payload);}

void reconnect() {

while (!client.connected()) {

Serial.print("Attempting MQTT connection...");

String clientId = "Kelompok-5";

clientId += String(random(0xffff), HEX);

if (client.connect(clientId.c\_str())) {

Serial.println("connected");

client.publish("sukses\_konek", "Yess... saya terkoneksi");

client.subscribe("kelompok-5/act/led");

} else {

Serial.print("failed, rc=");

Serial.print(client.state());

Serial.println(" Try again In 5 Seconds"); delay(5000);}}}

void setup() {

Serial.begin(115200);

setup\_wifi();

client.setServer(MQTT\_SERVER, MQTT\_PORT);

client.setCallback(callback);

pinMode(pinDigital, INPUT);

pinMode(buzzer, OUTPUT);}

void loop() {

if (!client.connected()) {

reconnect();}

if (kelembapan > 1000){

digitalWrite(buzzer, HIGH);

delay(1000);

digitalWrite(buzzer, LOW);

delay(1000);}

else{

digitalWrite(buzzer, LOW);

delay(1000);}

client.loop();

kirimPer2menit();}

void kirimPer2menit(){

\_now = millis();

if(millis() - \_waiting > 2000){

\_waiting = \_now;

kelembapan = analogRead (pinDigital);

sprintf(data, "%g#",

kelembapan);

Serial.print("Publish message : ");

Serial.println(data);

client.publish("kelompok-5/sensor/temp", data);}}

**Thonny**

import paho.mqtt.client as mqtt

import mysql.connector

import json

mydb = mysql.connector.connect(

host = "localhost",

user = "root",

password = "",

database = "kelompok5")

mycursor = mydb.cursor()

def on\_connect(client, userdata, flags, rc):

print("Connected with result code "+str(rc))

client.subscribe("kelompok-5/sensor/temp")

def on\_message(client, userdata, msg):

arr = []

arr = str(msg.payload, 'utf-8').split('#');

kelembapan = arr[0];

cetak = '';

payload = '';

sql = "INSERT INTO data(temperature) VALUES (%s)"

val = (kelembapan,)

mycursor.execute(sql, val)

mydb.commit()

client = mqtt.Client()

client.on\_connect = on\_connect

client.on\_message = on\_message

client.connect("broker.hivemq.com", 1883, 60)

client.loop\_forever()

**KELOMPOK6**

**Arduino**

#include <ESP8266WiFi.h>

#include <PubSubClient.h>

#include <ArduinoJson.h>

#include <MQ2.h>

#include <Wire.h>

WiFiClient espClient;

PubSubClient client(espClient);

#define WIFI\_SSID "Vnr"

#define WIFI\_PASS ""

#define MQTT\_SERVER "broker.hivemq.com"

#define MQTT\_PORT 1883

#define mqxPin A0

int sensorThres = 400;

#define redLed D3

#define whiteLed D1

int pinAout = A0;

MQ2 mq2(pinAout);

void setup\_wifi() {

Serial.println();

Serial.print("Connecting to ");

Serial.println(WIFI\_SSID);

WiFi.begin(WIFI\_SSID);

while (WiFi.status() != WL\_CONNECTED) {

delay(500);

Serial.print(".");}

Serial.println("");

Serial.println("WiFi connected");

Serial.println("IP address: ");

Serial.println(WiFi.localIP());}

float lpg\_gas, co\_gas, smoke\_gas;

float analogSensor;

unsigned long \_waiting = millis();

unsigned long \_now;

int value = 0;

char data[50];

String buffData;

void callback(char\* topic, byte\* payload, unsigned int length) {

buffData = "";

for (int i=0; i<length; i++){

buffData += (char) payload[i];}

Serial.print(topic);

Serial.print(" ==> ");

Serial.println(buffData);

StaticJsonDocument<200> doc;

deserializeJson(doc,payload);}

void reconnect() {

while (!client.connected()) {

Serial.print("Attempting MQTT connection...");

String clientId = "Kelompok-6";

clientId += String(random(0xffff), HEX);

if (client.connect(clientId.c\_str())) {

Serial.println("connected");

client.publish("sukses\_konek", "Yess... saya terkoneksi");

client.subscribe("kelompok-6/act/led");

} else {

Serial.print("failed, rc=");

Serial.print(client.state());

Serial.println(" Try again In 5 Seconds");

delay(5000);}}}

void setup() {

Serial.begin(115200);

mq2.begin();

pinMode(mqxPin, INPUT);

pinMode(redLed, OUTPUT);

pinMode(whiteLed, OUTPUT);

setup\_wifi();

client.setServer(MQTT\_SERVER, MQTT\_PORT);

client.setCallback(callback);}

void loop() {

if (!client.connected()) {

reconnect();}

if (analogSensor > sensorThres){

digitalWrite(redLed, HIGH);

digitalWrite(whiteLed, LOW);}

else{

digitalWrite(redLed, LOW);

digitalWrite(whiteLed, HIGH);}

delay(1000);

client.loop();

kirimPer2menit();}

void kirimPer2menit(){

\_now = millis();

if(millis() - \_waiting > 10000){

\_waiting = \_now;

analogSensor = analogRead(mqxPin);

lpg\_gas = mq2.readLPG();

co\_gas = mq2.readCO();

smoke\_gas = mq2.readSmoke();

sprintf(data, "%g#%i#%i#%i",

analogSensor, lpg\_gas, co\_gas, smoke\_gas);

Serial.print("Publish message : ");

Serial.println(data);

float\* values= mq2.read(true);

client.publish("kelompok-6/sensor/temp", data);}}

**Thonny**

import paho.mqtt.client as mqtt

import mysql.connector

import json

mydb = mysql.connector.connect(

host = "localhost",

user = "root",

password = "",

database = "kelompok6")

mycursor = mydb.cursor()

def on\_connect(client, userdata, flags, rc):

print("Connected with result code "+str(rc))

client.subscribe("kelompok-6/sensor/temp")

def on\_message(client, userdata, msg):

arr = []

arr = str(msg.payload, 'utf-8').split('#');

analogSensor = arr[0];

lpg\_gas = arr[1];

co\_gas = arr[2];

smoke\_gas = arr[3];

cetak = '';

payload = '';

sql = "INSERT INTO project (mq2, lpg, co, smoke) VALUES (%s, %s, %s, %s)"

val = (analogSensor, lpg\_gas, co\_gas, smoke\_gas)

mycursor.execute(sql, val)

mydb.commit()

client = mqtt.Client()

client.on\_connect = on\_connect

client.on\_message = on\_message

client.connect("broker.hivemq.com", 1883, 60)

client.loop\_forever()