

DAFTAR PUSTAKA

- Hasiholan, C., Primananda, R., & Amron, K. (2018). Implementasi Konsep *Internet of Things* pada Sistem Monitoring Banjir menggunakan Protokol *MQTT*. *Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 2(12), 6128–6135.
- Cahyono, I. Y. (2018). Pembuka Kap Dan Bagasi Mobil Menggunakan *Smartphone* Berbasis *Bluetooth*.
- Husdi. (2018). Monitoring Kelembaban Tanah Pertanian Menggunakan *Soil Moisture Sensor YL-69* Dan *Arduino UNO*. *ILKOM Jurnal Ilmiah*, 10(2), 237–243.
- Islam, H. I., Nabilah, N., Atsaurry, S. S., Saputra, D. H., Pradipta, G. M., Kurniawan, A., Syafutra, H., Irmansyah, & Irzaman. (2016). Sistem Kendali Suhu dan Pemantauan Kelembaban Udara Ruangan Berbasis *Arduino UNO* Dengan Menggunakan Sensor *DHT22* Dan *Passive Infrared (PIR)*. *SNF2016,V(Lcd)*, 119–124.
- Kaya, E. (2012). Pengaruh Kompos Jerami dan Pupuk *NPK* Terhadap *N-Tersedia* Tanah, Serapan-*N*, Pertumbuhan, Dan Hasil Padi Sawah (*Oryza sativa L*). *Ilmu Budidaya Tanaman*, 1(2), 91–169.
- Masyudi, M., Sotyohadi, & Limpraptono, F. Y. (2020). Sistem Kontrol dan *Monitoring* Air Pada Tanaman Bawang Merah Berbasis *Web* Dengan Menggunakan *Wireless Sensor Network (WSN)*. Institut Teknologi Nasional, Malang, Indonesia, 1–11.
- Pratama, O. A. S. (2017). Rancang Bangun Monitoring Penggunaan Air PDAM Berbasis *Arduino UNO*.
- Pratama, R. P. (2017). Aplikasi *Websserver ESP8266* Untuk Pengendali Peralatan Listrik. *INVOTEK*, 17(2), 40–44.
- Rahmat Saputra. (2021). Sistem Monitoring Kelembaban Tanah dan Suhu

- Greenhouse* Tanaman Bawang Merah Berbasis *IoT*. Jurnal Perencanaan, Sains, Teknologi, Dan Komputer, 4(1), 981–990.
- Rahmatullah, W. (2019). Rancang Bangun Data *Logger* Berbasis Sensor *DHT22* Untuk Mengukur Suhu dan Kelembaban Habitat Satwa *Herpetofauna* Secara *Real-Time*.
- Sinuraya, M. C., Satoto, K. I., & Isnanto, R. R. (2013). Perancangan Sistem Informasi *Geografis* Perpajakan Pada Perangkat Bergerak Menggunakan Sistem Operasi *Android*. TRANSIENT, 2(1), 80–86.
- Suhaeb, S., Djawad, Y. A., Jaya, H., Ridwansyah, Sabran, & Risal, A. (2017). *Mikrokontroler dan Interface*.
- Sumarni, N., & Hidayat, A. (2005). Budidaya Bawang Merah.
- Suyanti, & Yundra, E. (2019). Rancang Bangun Deteksi Detak Jantung Manusia dengan Metode *Pulse* Sensor Berbasis *IoT (Internet of Things)*. Jurnal Teknik Elektro, 8(1), 191–198.
- Syaputra, D. (2021). Rancang Bangun Ruang Pengering Solar *Dryer* Kulit Kayu Manis PT. CASSIA CO-OP.
- Tabuni, A. (2017). Budidaya Tanaman Bawang Merah. 87(1,2), 149–200.
- Utari, T. L., Ms, A. U., & Alfita, R. (2019). Rancang Bangun Sistem Irigasi Otomatis Pada Tanaman Bawang Merah Berbasis *Short Message Service (SMS)*. Seminar Nasional Fortei7, VII, 243–247.
- Wihelmus, E. A., Achmad, A., & Dewiani. (2019). Pemanfaatan *Realtime Database* di *Platform Firebase* Pada Aplikasi *E-Tourism* Kabupaten Nabire. Jurnal Penelitian Enjiniring, 22(1), 20–26.

**L
A
M
P
I
R
A
N**

Lampiran 1 Source Code Program

GH_Onion_Master

Kode inisialisasi library

```
#include <NewPing.h>

#include "DHT.h"

#include <ESP8266WiFi.h>

#include <FirebaseESP8266.h>

#include <NTPClient.h>

#include <WiFiUdp.h>

#include <ESP8266HTTPClient.h>
```

Kode inisialisasi wi-fi

```
#define WIFI_SSID "Syarifah"

#define WIFI_PASSWORD "12345679"
```

Kode inisialisasi firebase

```
#define API_KEY
"o1FbG0xq5yVGCHfoc4PXsqeO11EskFXW0C3xxyOB"

#define DATABASE_URL "gh-onion-default-rtdb.firebaseio.com"

FirebaseData fbdo;
```

Kode inisialisasi pin sensor

```
#define sens_dht D1

#define sens_soil_moisture A0

#define sens_us_echo D5

#define sens_us_trig D6
```

Kode inisialisasi variabel

```
NewPing sonar(sens_us_trig, sens_us_echo, 100);

DHT dht(sens_dht, DHT11);

float rt_temperature, rt_humidity;

int sm_adc, rt_sm, rt_wl, val_wl;

unsigned long startDelay, currentDelay;

unsigned long startPost, currentPost;

unsigned long startRead, currentRead;

String current_time, current_date;

WiFiUDP ntpUDP;

NTPClient timeClient(ntpUDP);

String months[12]={"Januari", "Februari", "April", "April", "Mei", "Juni",
"Juli", "Augustus", "September", "Oktober", "November", "Desember"};
```

Kode mengatur mode pin dht

```
void setup() {

  Serial.begin(115200); // Komunikasi Serial

  dht.begin();

  // Sensor DHT konfigurasi Input

  pinMode(sens_dht, INPUT_PULLUP);
```

Kode mengoneksikan wi-fi dan firebase

```
WiFi.begin(WIFI_SSID, WIFI_PASSWORD);

Serial.print("Connecting to Wi-Fi");

while (WiFi.status() != WL_CONNECTED)
```

```

{
  Serial.print(".");
  delay(300);
}
Serial.println();
Serial.print("Connected with IP: ");
Serial.println(WiFi.localIP());
Serial.println();
Serial.printf("Firebase          Client          v%s\n\n",
FIREBASE_CLIENT_VERSION);

  Firebase.begin(DATABASE_URL, API_KEY);
  Firebase.reconnectWiFi(true);
  timeClient.begin();
  timeClient.setTimeOffset(28800); //GMT +8
}

```

Kode memanggil method `read_sensors`, `set_data`, `ntp`, dan `post_data`

```

void loop() {
  read_sensors();
  set_data();
  ntp();
  post_data();
}

```

Data

Kode perintah mengambil data sensor dan mengirim data ke firebase

```
void set_data(){  
  
    currentDelay = millis();  
  
    if(currentDelay - startDelay >= 3000){  
  
        startDelay = currentDelay;  
  
        FirebaseJson rt_json;  
  
        rt_json.add("rt-temperature", rt_temperature);  
  
        rt_json.add("rt-humidity", rt_humidity);  
  
        rt_json.add("rt-soil-moisture", rt_sm);  
  
        rt_json.add("rt-water-level", rt_wl);  
  
        Firebase.setJSON(fbdo, "data", rt_json);  
  
    }  
  
}
```

NTP

Kode perintah menampilkan waktu, tanggal, bulan, dan tahun data di firebase

```
void ntp(){
    timeClient.update();
    unsigned long epochTime = timeClient.getEpochTime();
    struct tm *ptm = gmtime ((time_t *)&epochTime);
    current_time = timeClient.getFormattedTime();
    int monthDay = ptm->tm_mday;
    int currentMonth = ptm->tm_mon+1;
    String currentMonthName = months[currentMonth-1];
    int currentYear = ptm->tm_year+1900;
    current_date = String(monthDay) + " " + String(currentMonthName) + " "
+ String(currentYear);
}
```


Post_Data

Kode tampilkan data kumulatif ke firebase

```
void post_data(){
    currentPost = millis();
    if(currentPost - startPost >= 60000){

        FirebaseJson post_json;

        post_json.add("Suhu Udara", rt_temperature);
        post_json.add("Kelembaban Udara", rt_humidity);
        post_json.add("Kelembaban Tanah", rt_sm);
        post_json.add("Ketinggian Air", rt_wl);

        Firebase.setJSON(fbdo, "/data-kumulatif/" + String(current_date) + "/" +
String(current_time), post_json);

        WiFiClient client;

        HTTPClient http;

        int httpResponseCode = http.POST(httpRequestData);

        if(httpResponseCode>0) {
            //Serial.print("HTTP Response code: ");

            //Serial.println(httpResponseCode);
        }else{
            //Serial.print("Error code: ");
```

```
//Serial.println(httpResponseCode);  
  
}  
  
http.end();  
  
startPost = currentPost;  
  
}  
  
}
```

Sensors

Kode mendeklarasikan nilai sensor kelembaban tanah

```
void read_sensors(){  
    sm_adc = analogRead(sens_soil_moisture);  
    rt_sm = map(sm_adc, 0, 1023, 0, 100); // Nilai 1023 di ganti dengan nilai  
    adc air  
    if(rt_sm>100)rt_sm=100;  
    if(rt_sm<0)rt_sm=0;
```

Kode mendeklarasikan nilai sensor dht

```
currentRead = millis();  
if(currentRead - startRead >= 2000){  
    rt_temperature = dht.readTemperature();  
    rt_humidity = dht.readHumidity();  
    if (isnan(rt_temperature) || isnan(rt_humidity)){  
        rt_temperature = 0;  
        rt_humidity = 0;  
    }  
    if(rt_temperature>100)rt_temperature=100;  
    if(rt_temperature<0)rt_temperature=0;  
    if(rt_humidity>100)rt_humidity=100;  
    if(rt_humidity<0)rt_humidity=0;  
    startRead = currentRead;  
}
```

Kode mendeklarasikan nilai sensor ultrasonik

```
val_wl = sonar.ping_cm();  
  
rt_wl = map(val_wl, 36, 4, 0, 100);  
  
if(rt_wl<0)rt_wl=0;  
  
if(rt_wl>100)rt_wl=100;  
  
  
Serial.println("ADC : " + String(sm_adc));  
  
Serial.println();  
  
}
```

GH_Onion_Slave

Kode inisialisasi library

```
#include <ESP8266WiFi.h>

#include <FirebaseESP8266.h>

#include <Wire.h>

#include <LiquidCrystal_I2C.h>

#include <ArduinoJson.h>
```

Kode inisialisasi wi-fi

```
#define WIFI_SSID "Syarifah" //"Syarifah"

#define WIFI_PASSWORD "12345679" //"12345679"
```

Kode inisialisasi firebase

```
#define API_KEY
"o1FbG0xq5yVGCHfoc4PXsqeOI1EskFXW0C3xxyOB"

#define DATABASE_URL "gh-onion-default-rtdb.firebaseio.com"

FirebaseData fbdo;
```

Kode inisialisasi pin relay

```
#define rl_cooler D3

#define rl_heater D4

#define rl_watering D5

#define rl_fill_water D6
```

Kode menampilkan simbol derajat pada LCD

```
byte degree[] = {

  B00111,
```

```
B00101,  
  
B00111,  
  
B00000,  
  
B00000,  
  
B00000,  
  
B00000,  
  
B00000  
};
```

Kode inisialisasi variabel

```
String request_json, data_json;  
  
boolean gh_cooler, gh_heater, gh_watering, gh_fill_water;  
  
int set_min_temperature, set_max_temperature,  
    set_humidity, set_min_soil_moisture, set_max_soil_moisture,  
    set_min_water_level, set_max_water_level;  
  
float rt_temperature;  
  
int rt_humidity, rt_soil_moisture, rt_water_level;  
  
char humd[3], sm[3], wl[3];  
  
LiquidCrystal_I2C lcd(0x27,16,2);
```

Kode pengoneksian wi-fi dan firebase

```
void setup() {  
  
    Serial.begin(115200);
```

```

WiFi.begin(WIFI_SSID, WIFI_PASSWORD);

Serial.print("Connecting to Wi-Fi");

while (WiFi.status() != WL_CONNECTED)

{

  Serial.print(".");

  delay(300);

}

Serial.println();

Serial.print("Connected with IP: ");

Serial.println(WiFi.localIP());

Serial.println();

Serial.printf("Firebase          Client          v%s\n",
FIREBASE_CLIENT_VERSION);

Firebase.begin(DATABASE_URL, API_KEY);

Firebase.reconnectWiFi(true);

```

Kode mengatur pinmode relay

```

pinMode(rl_cooler, OUTPUT);  digitalWrite(rl_cooler, HIGH);

pinMode(rl_heater, OUTPUT);  digitalWrite(rl_heater, HIGH);

pinMode(rl_watering, OUTPUT); digitalWrite(rl_watering, HIGH);

pinMode(rl_fill_water, OUTPUT); digitalWrite(rl_fill_water, HIGH);

```

Kode menampilkan data ke LCD

```
// LCD

lcd.init();

lcd.backlight();

lcd.setCursor(0,0);

lcd.print(" RUMAH BAWANG ");

lcd.setCursor(0,1);

lcd.print(" Version 1.0.0 ");

delay(2000);

lcd.createChar(1, degree);

request_data();

lcd.clear();

}
```

Kode pengaturan manual/auto pada akuator

```
void loop() {

  mode_manual:

  display_lcd(); request_data();

  if(Firebase.getString(fbdo, "/request/auto-manual")) {

    if(fbdo.to<String>() == "manual"){

      // Manual Cooler

      if(gh_cooler==true)                {digitalWrite(rl_cooler,LOW);

    }else{digitalWrite(rl_cooler,HIGH);}

      if(gh_heater==true)                {digitalWrite(rl_heater,LOW);

    }else{digitalWrite(rl_heater,HIGH);}

    }
```



```

        if(gh_watering==true)                {digitalWrite(rl_watering,LOW);
    }else{digitalWrite(rl_watering,HIGH);}

    if(gh_fill_water==true){digitalWrite(rl_fill_water,LOW);}else{digitalWrite
    (rl_fill_water,HIGH);}

        //Serial.println("Mode Manual");

    }

    if(fbdo.to<String>() == "auto"){

        digitalWrite(rl_cooler,HIGH);

        digitalWrite(rl_heater,HIGH);

        digitalWrite(rl_watering,HIGH);

        digitalWrite(rl_fill_water,HIGH);

        Serial.println("Enter to Mode Auto");

        goto mode_auto;

    }

}

goto mode_manual;

mode_auto:

display_lcd(); request_data();

if(Firebase.getString(fbdo, "/request/auto-manual")) {

    if(fbdo.to<String>() == "auto"){

        // Cooler

        if(rt_temperature >= set_max_temperature){digitalWrite(rl_cooler,

```

```

LOW);}

    if(rt_temperature <= set_min_temperature){digitalWrite(rl_cooler,
HIGH);}

    // Heater

    if(rt_temperature <= set_min_temperature){digitalWrite(rl_heater,
LOW);}

    if(rt_temperature >= set_max_temperature){digitalWrite(rl_heater,
HIGH);}

    // Watering

    if(rt_soil_moisture <=
set_min_soil_moisture){digitalWrite(rl_watering, LOW);}

    if(rt_soil_moisture >=
set_max_soil_moisture){digitalWrite(rl_watering, HIGH);}

    // Water Level

    if(rt_water_level <= set_min_water_level){digitalWrite(rl_fill_water,
LOW);}

    if(rt_water_level >= set_max_water_level){digitalWrite(rl_fill_water,
HIGH);}

    //Serial.println("Mode Auto");

}

if(fbdo.to<String>() == "manual"){

    digitalWrite(rl_cooler,HIGH);

    digitalWrite(rl_heater,HIGH);

    digitalWrite(rl_watering,HIGH);

    digitalWrite(rl_fill_water,HIGH);

```

```
Serial.println("Enter to Mode Manual");
```

```
goto mode_manual;
```

```
}
```

```
}
```

```
goto mode_auto;
```

```
}
```

LCD

Kode menampilkan data sensor ke LCD 16x2

```
void display_lcd(){

    lcd.setCursor(0,0); lcd.print("T:" + String(rt_temperature));

    lcd.setCursor(7,0); lcd.write(1);   lcd.setCursor(8,0); lcd.print("C");

    lcd.setCursor(10,0); lcd.print("H:"); sprintf(humd, "%3d", rt_humidity);

    lcd.setCursor(12,0); lcd.print(humd); lcd.setCursor(15,0); lcd.print("%");

    lcd.setCursor(0,1);          lcd.print("SM:");   sprintf(sm,      "%3d",
rt_soil_moisture);

    lcd.setCursor(3,1); lcd.print(sm);   lcd.setCursor(6,1); lcd.print("%");

    lcd.setCursor(9,1); lcd.print("WL:"); sprintf(wl, "%3d", rt_water_level);

    lcd.setCursor(12,1); lcd.print(wl);   lcd.setCursor(15,1); lcd.print("%");

}
```

Request Data

Kode request data dari firebase

```
void request_data(){  
  
    if(Firebase.getJSON(fbdo, "request")) {  
  
        request_json = fbdo.jsonString();  
  
        //Serial.println(data_json);  
  
        StaticJsonDocument<512> doc;  
  
        DeserializationError error = deserializeJson(doc, request_json);  
  
        if (error) {  
  
            Serial.print(F("deserializeJson() failed: "));  
  
            Serial.println(error.f_str());  
  
            return;  
  
        }  
  
    }
```

Kode menampilkan status akuator pada firebase

```
gh_cooler = doc["gh-cooler"];  
  
gh_fill_water = doc["gh-fill-water"];  
  
gh_heater = doc["gh-heater"];  
  
gh_watering = doc["gh-watering"];  
  
set_humidity = doc["set-humidity"];
```

Kode menampilkan nilai min dan max pada firebase

```

set_max_water_level = doc["set-max-water-level"];

set_min_water_level = doc["set-min-water-level"];

set_min_soil_moisture = doc["set-min-soil-moisture"];

set_max_soil_moisture = doc["set-max-soil-moisture"];

set_min_temperature = doc["set-min-temperature"];

set_max_temperature = doc["set-max-temperature"];

}

```

Kode menampilkan realtime data sensor pada firebase

```

//RT Data

if(Firebase.getJSON(fbdo, "data")) {

    data_json = fbdo.jsonString();

    StaticJsonDocument<256> doc;

    DeserializationError error = deserializeJson(doc, data_json);

    if (error) {

        Serial.print(F("deserializeJson() failed: "));

        Serial.println(error.f_str());

        return;

    }

    rt_humidity = doc["rt-humidity"];

    rt_soil_moisture = doc["rt-soil-moisture"];

    rt_temperature = doc["rt-temperature"];

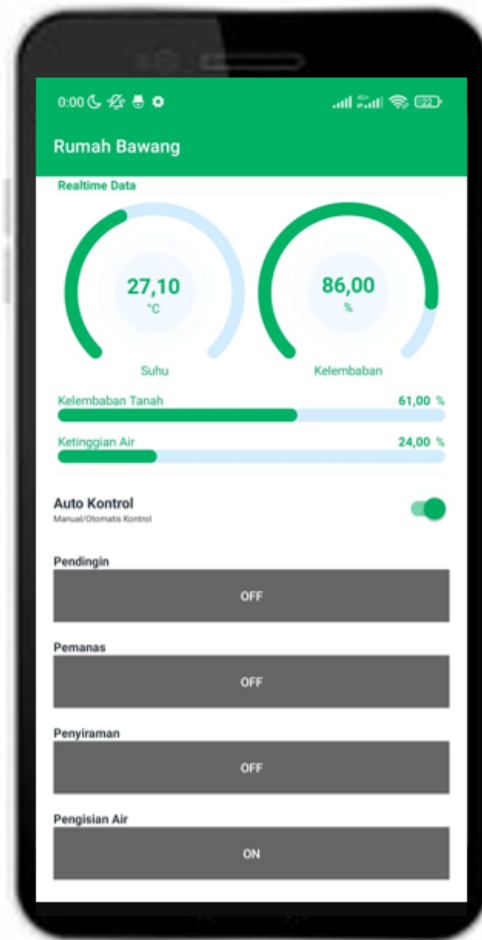
```

```
rt_water_level = doc["rt-water-level"];
```

```
}
```

```
}
```

Lampiran 2 Tampilan Aplikasi Android



Lampiran 3 Gambar Sistem Tanaman Bawang Merah



