



**MANUAL BOOK**

# **MONITORING WATER QUALITY WITH IOT**

## **TIM PENYUSUN:**

- **RIANTO, S.T., M.T.**
- **DR. ARADEA, S.T., M.T.**
- **HUSNI MUBAROK, STP., M.T.**
- **DR. IR. NUR WIDIYASONO, M.KOM**
- **MAULANA DECKY RAKHMAN**



# DAFTAR ISI

<b>DAFTAR ISI.....</b>	<b>i</b>
<b>1. PENJELASAN UMUM.....</b>	<b>1</b>
a. Mikrokontroler.....	2
b. NodeMCU ESP8266.....	2
c. Sensor pH.....	2
d. Sensor Suhu DS18B20.....	3
e. Firebase.....	3
f. Mit App Inventor.....	4
<b>2. Desain Sistem.....</b>	<b>4</b>
<b>3. SOURCE CODE .....</b>	<b>5</b>
<b>4. INSTALASI APLIKASI .....</b>	<b>12</b>
a. Kebutuhan sistem .....	12
b. Tahap Instalasi .....	12
<b>5. SIMULASI .....</b>	<b>15</b>

## **PEMEGANG HAK CIPTA**

1. Rianto, S.T., M.T.
2. Dr. Aradea, S.T., M.T.
3. Husni Mubarak, STP., M.T.
4. Dr. Ir. Nur Widiyasono, M.Kom
5. Maulana Decky Rakhman

Tasikmalaya, November 2023

Pemegang Hak Cipta

## 1. PENJELASAN UMUM

### a. Mikrokontroler

Mikrokontroler adalah perangkat elektronik digital yang memiliki kemampuan untuk menerima input, menghasilkan output, dan dikendalikan melalui program yang dapat ditulis dan dihapus. Cara kerja mikrokontroler sebenarnya melibatkan proses membaca dan menulis data. Mikrokontroler terdiri dari berbagai komponen yang terintegrasi dalam satu chip, termasuk prosesor, memori, dan antarmuka input/output (I/O), yang membuatnya berfungsi sebagai sebuah komputer kecil yang dapat digunakan secara inovatif sesuai dengan kebutuhan sistem (Santoso, 2015).

### b. NodeMCU ESP8266

NodeMCU adalah sebuah platform Internet of Things (IoT) yang bersifat open source. Ini terdiri dari perangkat keras yang menggunakan System On Chip ESP8266 yang dibuat oleh Espressif System, serta firmware yang menggunakan bahasa pemrograman skrip Lua. Istilah NodeMCU pada dasarnya merujuk pada firmware yang digunakan daripada kit pengembangan perangkat keras. NodeMCU bisa dianggap sebagai versi papan pengembangan Arduino untuk ESP8266. NodeMCU telah mengemas ESP8266 ke dalam papan yang lebih kompak dengan berbagai fitur mirip mikrokontroler dan kemampuan untuk mengakses Wi-Fi, serta chip komunikasi USB ke serial. Oleh karena itu, untuk memprogramnya, hanya perlu menggunakan kabel data USB ekstensi yang sama seperti yang digunakan untuk mengisi daya ponsel Android (Mariza Wijayanti, 2022).

### c. Sensor pH

Sensor pH meter adalah alat yang digunakan untuk mengukur tingkat keasaman atau kebasaan dalam cairan atau larutan. Prinsip kerja sensor pH utamanya terletak pada probe sensor yang terbuat dari elektroda kaca. Elektroda kaca ini memiliki larutan HCL pada ujungnya, dan sensor probe ini bekerja dengan mengukur konsentrasi ion  $H_3O^+$  dalam larutan, sehingga dapat menentukan nilai pH dari larutan atau cairan tersebut (Denanta Bayuguna Perteka, Piarsa and Wibawa, 2020). Elektroda yang digunakan dalam sensor pH air terbuat dari lapisan kaca yang sensitif dan memiliki impedansi rendah, sehingga dapat memberikan hasil pembacaan dan penilaian yang stabil dan cepat, baik

pada suhu tinggi maupun rendah dalam cairan atau larutan. Nilai pH yang diukur oleh sensor pH dapat diakses oleh mikrokontroler melalui antarmuka PH 2.0 yang tersedia pada modul sensor pH air. Sensor pH air ini sangat cocok untuk melakukan pengukuran kadar pH dalam cairan dengan interval waktu yang lama (Syafiqoh, Sunardi and Yudhana, 2018).

d. Sensor Suhu DS18B20

Sensor suhu DS18B20 adalah sebuah perangkat pengukur suhu yang menggunakan antarmuka satu kabel (one wire), sehingga hanya memerlukan sedikit kabel dalam proses pemasangannya. Keistimewaan dari sensor ini adalah kemampuannya untuk dihubungkan secara paralel dengan satu input. Sensor ini dapat digunakan lebih dari satu sensor DS18B20, semua keluaran sensor tersebut dapat dihubungkan ke satu pin pada papan Arduino. Fitur ini membuat sensor ini sangat populer, terutama karena sensor ini juga tahan air (waterproof), sehingga sensor ini dapat digunakan untuk mengukur dan mengontrol suhu pemanas air.

e. Firebase

Firebase Realtime Database adalah sebuah database realtime yang disimpan di cloud dan mendukung berbagai platform seperti Android, iOS, dan Web. Data dalam Firebase disimpan dalam format struktur JSON (JavaScript Object Notation). Firebase Database secara otomatis melakukan sinkronisasi dengan aplikasi klien yang terhubung, sehingga aplikasi multiplatform yang menggunakan SDK Android, iOS, dan JavaScript akan secara otomatis menerima pembaruan data terbaru saat terhubung ke server Firebase (Sudiarta, Indrayana and Suasnawa, 2018). Firebase Realtime Database adalah platform database yang digunakan untuk aplikasi yang berjalan secara realtime. Ketika terjadi perubahan data, aplikasi yang terhubung dengan Firebase akan secara otomatis diperbarui di setiap perangkat, termasuk website dan perangkat mobile. Firebase juga dapat diintegrasikan dengan berbagai kerangka kerja lain seperti node, java, javascript, dan sebagainya (Sanad, 2019).

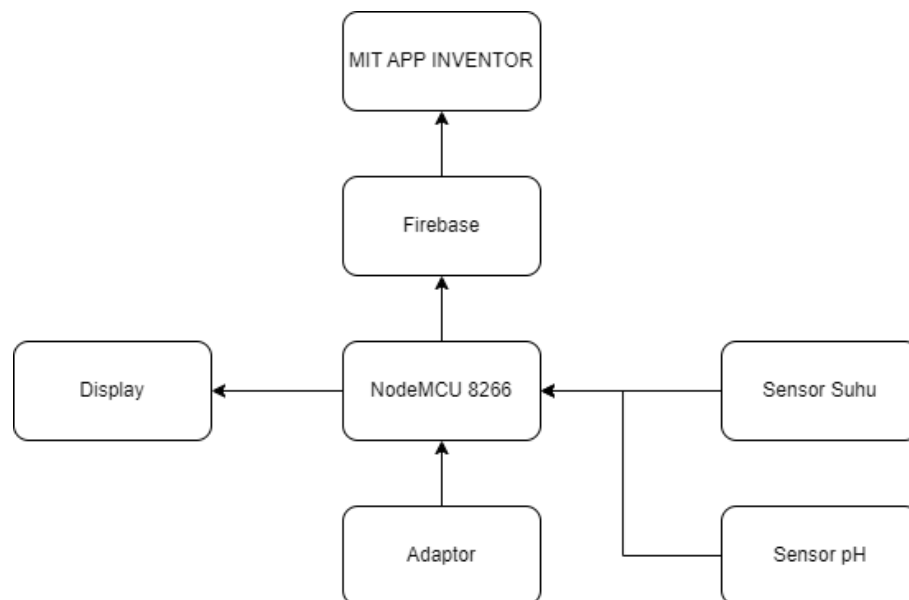
f. IoT

Internet of Things, atau yang sering disingkat sebagai IoT, adalah konsep di mana semua objek fisik di dunia nyata dapat saling berkomunikasi sebagai bagian dari sebuah sistem terintegrasi menggunakan jaringan internet sebagai perantara. Contohnya adalah CCTV yang terpasang di berbagai lokasi dihubungkan melalui koneksi internet dan dikendalikan dari satu pusat kontrol yang mungkin berjarak beberapa puluh kilometer. Atau, sebuah rumah pintar yang dapat dikelola melalui smartphone dengan bantuan konektivitas internet. (Efendi, 2018).

g. Mit App Inventor

MIT App Inventor merupakan sebuah platform pemrograman visual yang memungkinkan pengguna untuk merancang dan membuat aplikasi seluler yang dapat dijalankan pada sistem Android dengan cara menggunakan, melihat, menyusun, serta menarik dan menjatuhkan blok-blok perintah yang merupakan simbol-simbol instruksi (Sintia br Ginting *et al.*, 2022).

## 2. Desain Sistem



Gambar 1. Arsitektur Water Monitoring IoT

Arsitektur pada gambar 1 menunjukkan arsitektur dari keseluruhan Water Monitoring IoT. Di dalam arsitektur tersebut terdapat beberapa komponen yaitu:

1. NodeMCU ESP8266, berupa mikrokontroler yang menerima input dari sensor pH dan suhu. Data dari kedua sensor ini diproses sesuai dengan program yang telah diunggah. Selanjutnya, data dari NodeMCU dikirim ke Firebase dan ditampilkan di layar LCD.
2. Sensor suhu DS18B20, sensor tersebut dipasang ke NodeMCU kemudian dari sensor tersebut akan mengirimkan nilai yang sudah terdeteksi secara realtime.
3. Sensor pH, sensor yang digunakan adalah sensor *pH meter module pH-4502C with probe electrode*. Sensor tersebut dihubungkan ke NodeMCU dan dari sensor pH tersebut dapat mengirimkan nilai pH air secara realtime ke NodeMCU.
4. Display LCD, display LCD yang dipakai berukuran 16x2. Display LCD ini bertujuan untuk menampilkan data yang telah diterima NodeMCU dari tiap sensor.
5. Firebase, firebase dapat menerima data yang dikirimkan oleh NodeMCU, yaitu sebuah data sensor pH dan suhu yang telah diterima oleh NodeMCU dari tiap sensor. Data tersebut tersimpan di Firebase sehingga bisa dikirimkan lagi.
6. Mit App Inventor, Mit App Inventor berfungsi untuk membuat aplikasi secara praktis. Dalam Mit app inventor, untuk mendapatkan data dari firebase harus mengambil dulu API key nya kemudian baru bisa mendapatkan data tersebut.
7. Adaptor, ini diperlukan untuk sumber daya ke NodeMCU agar bisa berfungsi dengan baik.

### 3. SOURCE CODE

```
#include <OneWire.h>
#include <DallasTemperature.h>

const int oneWireBus = 2; // Misalnya, gunakan pin digital 2
OneWire oneWire(oneWireBus);
DallasTemperature sensors(&oneWire);

const int sensorPin = A0;
float Kadar_pH = 0;

#include <Wire.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 16, 2); // set the LCD address to 0x27 for a 16 chars and 2 line display

#if defined(ESP32)
  #include <WiFi.h>
#elif defined(ESP8266)
  #include <ESP8266WiFi.h>
#endif
#include <Firebase_ESP_Client.h>

#include <addons/TokenHelper.h>
#include <addons/RTDBHelper.h>
```

```

#define WIFI_SSID "ssid_wifi"
#define WIFI_PASSWORD "password_wifi"

#define API_KEY "AlzaSyB331SCnBRhHhONWYklEgtHNFeC07O-2NI"
#define DATABASE_URL "https://water-monitoring-project-25b0d-default-rtdb.firebaseio.com/"

FirebaseData fbdo;
FirebaseAuth auth;
FirebaseConfig config;

unsigned long previousMillis_DS18B20 = 0;
unsigned long interval_DS18B20 = 1000;

unsigned long previousMillis_pH_meter = 0;
unsigned long interval_pH_meter = 1000;

unsigned long previousMillis_displayTemp = 0;
unsigned long interval_displayTemp = 1000;

unsigned long previousMillis_displaypH = 0;
unsigned long interval_displaypH = 1000;

unsigned long previousMillis_display = 0;
unsigned long interval_display = 200;

int count = 0;

bool signupOK = false;

byte Temp[8] = //icon for termometer
{
  B00100,
  B01010,
  B01010,
  B01110,
  B01110,
  B11111,
  B11111,
  B01110
};

byte pH[8] = //icon for water droplet
{
  B00100,
  B00100,
  B01110,
  B01110,
  B11111,
  B11111,
  B11111,
  B01110,
};

void setup() {

```



```

Serial.begin(115200);

pinMode (sensorPin, INPUT);

lcd.init();    // initialize the lcd
lcd.backlight(); // Turn on the LCD screen backlight

lcd.createChar(1, Temp); // create a new icon for thermometer
lcd.createChar(2, pH); // create a new icon for water droplet

WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
while (WiFi.status() != WL_CONNECTED) {

    modeOffline();
}
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Connected:");
lcd.setCursor(0, 1);
lcd.print(WiFi.localIP());
Serial.println();
Serial.print("connected: ");
Serial.println(WiFi.localIP());
Serial.println();

config.api_key = API_KEY;
config.database_url = DATABASE_URL;

if (Firebase.signUp(&config, &auth, "", "")){
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Sign up OK");
    Serial.println("ok");
    signupOK = true;
    delay(1500);
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Online");
    delay(1000);
}
else {
    lcd.clear();
    lcd.print("Sign up FAILED");
    delay(1500);

    while (signupOK != true) {
        modeOffline();
    }
}

config.token_status_callback = tokenStatusCallback;

Firebase.begin(&config, &auth);
Firebase.reconnectWiFi(true);
}

```

```

void loop() {
    unsigned long currentMillis = millis();

    if (Firebase.ready() && signupOK && (currentMillis - previousMillis_DS18B20 >=
interval_DS18B20 || previousMillis_DS18B20 == 0 )) {

        sensors.requestTemperatures();

        float temperatureCelsius = sensors.getTempCByIndex(0);
        float temperatureFahrenheit = sensors.getTempFByIndex(0);

        char temperatureCelsiusStr[10];
        dtostrf(temperatureCelsius, 4, 1, temperatureCelsiusStr); // 4 karakter total, 1 angka desimal

        if (Firebase.RTDB.setFloat(&fbdo, "Water_Monitoring_Project_v1/Celsius",
atof(temperatureCelsiusStr))){
            Serial.print("Suhu dalam Celsius: ");
            Serial.print(temperatureCelsiusStr);
            Serial.println(" °C");
        }
        else {
            Serial.println("FAILED");
            Serial.println("REASON: " + fbdo.errorReason());
        }

        char temperatureFahrenheitStr[10];
        dtostrf(temperatureFahrenheit, 4, 1, temperatureFahrenheitStr); // 4 karakter total, 1 angka
desimal

        if (Firebase.RTDB.setFloat(&fbdo, "Water_Monitoring_Project_v1/Fahrenheit",
atof(temperatureFahrenheitStr)) {
            Serial.print("Suhu dalam Fahrenheit: ");
            Serial.print(temperatureFahrenheitStr);
            Serial.println(" °F");
        } else {
            Serial.println("FAILED");
            Serial.println("REASON: " + fbdo.errorReason());
        }

        previousMillis_DS18B20 = currentMillis;
    }

    if (Firebase.ready() && signupOK && (currentMillis - previousMillis_pH_meter >=
interval_pH_meter || previousMillis_pH_meter == 0 )) {

        int ADC_pH = analogRead(sensorPin);
        double Tegangan_pH = 5 / 1024.0 * ADC_pH;

        Kadar_pH = 7.00 + ((5 - Tegangan_pH) / 0.17);

        char ADC_pHStr[10];
        dtostrf(ADC_pH, 4, 1, ADC_pHStr); // 4 karakter total, 1 angka desimal

        if (Firebase.RTDB.setFloat(&fbdo, "Water_Monitoring_Project_v1/ADC_pH", atof(ADC_pHStr))) {

```

```

    Serial.print("Nilai ADC Ph: ");
    Serial.println(ADC_pHStr);
} else {
    Serial.println("FAILED");
    Serial.println("REASON: " + fbdo.errorReason());
}

char Tegangan_pHStr[15];
dtostrf(Tegangan_pH, 4, 1, Tegangan_pHStr);

if (Firebase.RTDB.setFloat(&fbdo, "Water_Monitoring_Project_v1/Tegangan_pH",
atof(Tegangan_pHStr))) {
    Serial.print("Tegangan pH : ");
    Serial.println(Tegangan_pHStr);
} else {
    Serial.println("FAILED");
    Serial.println("REASON: " + fbdo.errorReason());
}

char Kadar_pHStr[15];
dtostrf(Kadar_pH, 4, 1, Kadar_pHStr);

if (Firebase.RTDB.setFloat(&fbdo, "Water_Monitoring_Project_v1/Kadar_pH",
atof(Kadar_pHStr))) {
    Serial.print("Kadar pH : ");
    Serial.println(Kadar_pHStr);
} else {
    Serial.println("FAILED");
    Serial.println("REASON: " + fbdo.errorReason());
}

previousMillis_pH_meter = currentMillis;
}

if (Firebase.ready() && signupOK && (currentMillis - previousMillis_displayTemp >=
interval_displayTemp || previousMillis_displayTemp == 0 )) {

    sensors.requestTemperatures();

    float temperatureCelsius = sensors.getTempCByIndex(0);
    float temperatureFahrenheit = sensors.getTempFByIndex(0);

    char temperatureCelsiusStr[10];
    dtostrf(temperatureCelsius, 4, 1, temperatureCelsiusStr); // 4 karakter total, 1 angka desimal

    lcd.setCursor(0, 1);
    lcd.write(1); // write the termometer icon
    lcd.setCursor(2, 1);
    lcd.print(temperatureCelsiusStr);
    lcd.print((char)223);
    lcd.print("C");

    previousMillis_displayTemp = currentMillis;
}

```

```

    if (Firebase.ready() && signupOK && (currentMillis - previousMillis_displaypH >=
interval_displaypH || previousMillis_displaypH == 0 )) {

        int ADC_pH = analogRead(sensorPin);
        double Tegangan_pH = 5 / 1024.0 * ADC_pH;

        Kadar_pH = 7.00 + ((5 - Tegangan_pH) / 0.17);

        char Kadar_pHStr[15];
        dtostrf(Kadar_pH, 4, 1, Kadar_pHStr);

        lcd.setCursor(9, 1);
        lcd.write(2); // write the water droplet icon
        lcd.setCursor(11, 1);
        lcd.print(Kadar_pHStr);

        previousMillis_displaypH = currentMillis;
    }

    if (Firebase.ready() && signupOK && (currentMillis - previousMillis_display >= interval_display ||
previousMillis_display == 0 )) {

        for (int kolom = 1; kolom <= 19; kolom++) {
            lcd.setCursor(kolom - 1, 0); // Set kursor ke kolom yang sesuai di baris pertama
            lcd.print("~"); // Tampilkan gelombang (~) di kolom yang ditunjuk
            delay(150); // Jeda antara setiap langkah
            lcd.setCursor(kolom - 1, 0); // Set kursor ke kolom yang ditunjuk kembali
            lcd.print(" "); // Hapus gelombang (~) dengan mengganti dengan spasi
        }

        previousMillis_display = currentMillis;
    }
}

void modeOffline() {

    unsigned long currentMillis = millis();

    if ((currentMillis - previousMillis_displayTemp >= interval_displayTemp ||
previousMillis_displayTemp == 0 )) {

        sensors.requestTemperatures();

        float temperatureCelsius = sensors.getTempCByIndex(0);
        float temperatureFahrenheit = sensors.getTempFByIndex(0);

        char temperatureCelsiusStr[10];
        dtostrf(temperatureCelsius, 4, 1, temperatureCelsiusStr); // 4 karakter total, 1 angka desimal

        lcd.setCursor(0, 1);
        lcd.write(1); // write the termometer icon
        lcd.setCursor(2, 1);
        lcd.print(temperatureCelsiusStr);
        lcd.print((char)223);
        lcd.print("C");
    }
}

```

```

    previousMillis_displayTemp = currentMillis;
}

if ((currentMillis - previousMillis_displaypH >= interval_displaypH || previousMillis_displaypH == 0 )) {

    int ADC_pH = analogRead(sensorPin);
    double Tegangan_pH = 5 / 1024.0 * ADC_pH;

    Kadar_pH = 7.00 + ((5 - Tegangan_pH) / 0.17);

    char Kadar_pHStr[15];
    dtostrf(Kadar_pH, 4, 1, Kadar_pHStr);

    lcd.setCursor(9, 1);
    lcd.write(2); // write the water droplet icon
    lcd.setCursor(11, 1);
    lcd.print(Kadar_pHStr);

    previousMillis_displaypH = currentMillis;
}

if ((currentMillis - previousMillis_display >= interval_display || previousMillis_display == 0 )) {

    // Gelombang air dari kolom 1 ke kolom 8
    for (int kolom = 1; kolom <= 19; kolom++) {
        lcd.setCursor(kolom - 1, 0); // Set kursor ke kolom yang sesuai di baris pertama
        lcd.print("~");           // Tampilkan gelombang (~) di kolom yang ditunjuk
        delay(150);                // Jeda antara setiap langkah
        lcd.setCursor(kolom - 1, 0); // Set kursor ke kolom yang ditunjuk kembali
        lcd.print(" ");           // Hapus gelombang (~) dengan mengganti dengan spasi
    }

    previousMillis_display = currentMillis;
}
}

```

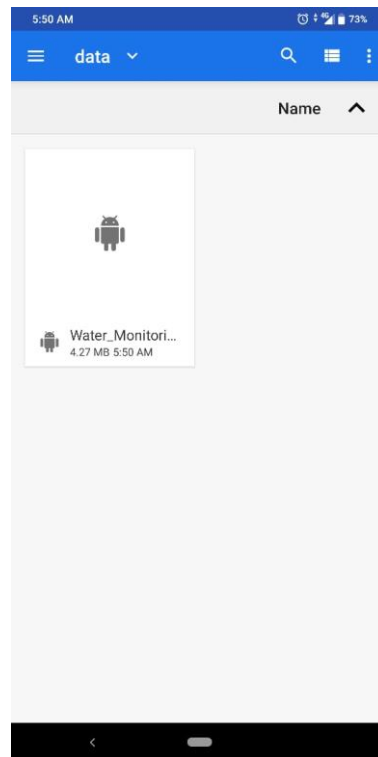
#### 4. INSTALASI APLIKASI

##### a. Kebutuhan sistem

1. iOS 9.0 atau lebih baru di iPhone, iPad, atau iPod Touch
2. macOS 11 atau lebih tinggi di komputer dengan Apple M1 Silicon
3. Sistem Operasi Android 2.1 (“Eclair”) atau lebih tinggi untuk perangkat Android

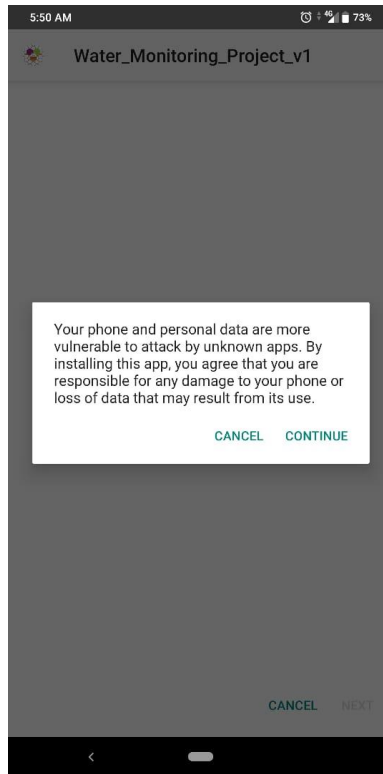
##### b. Tahap Instalasi

1. Sebelum melakukan instalasi pastikan sudah mendownload APK file nya dari Mit App Inventor yang sudah di buat sebelumnya.



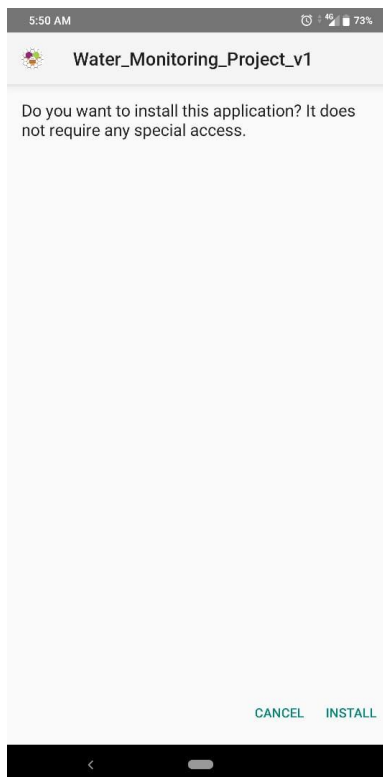
*Gambar 2. APK file*

2. Buka file tersebut, jika ada peringatan tekan CONTINUE.



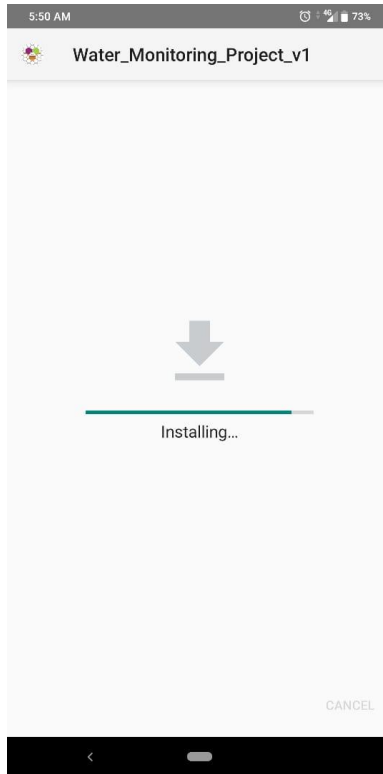
*Gambar 3. warning message*

3. Selanjutnya tekan INSTALL untuk melakukan penginstalan aplikasinya.



*Gambar 4. pre-installing*

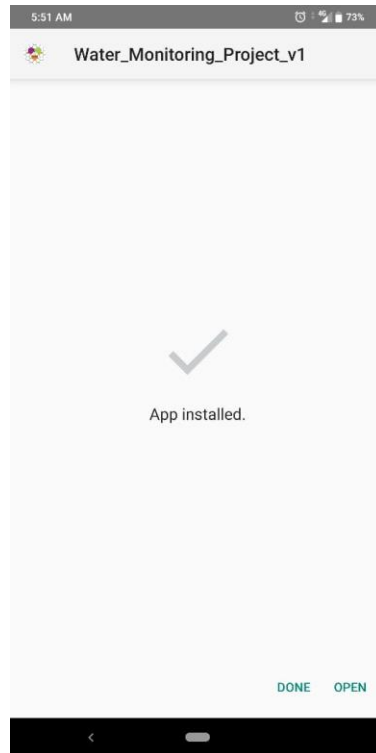
4. Setelah tekan INSTALL, maka proses installing akan berjalan.



*Gambar 5. proses install*

5. Setelah proses INSTALL berhasil, maka akan menerima informasi bahwa proses INSTALL sudah berhasil dan bisa langsung dibuka.





*Gambar 6. App installed*

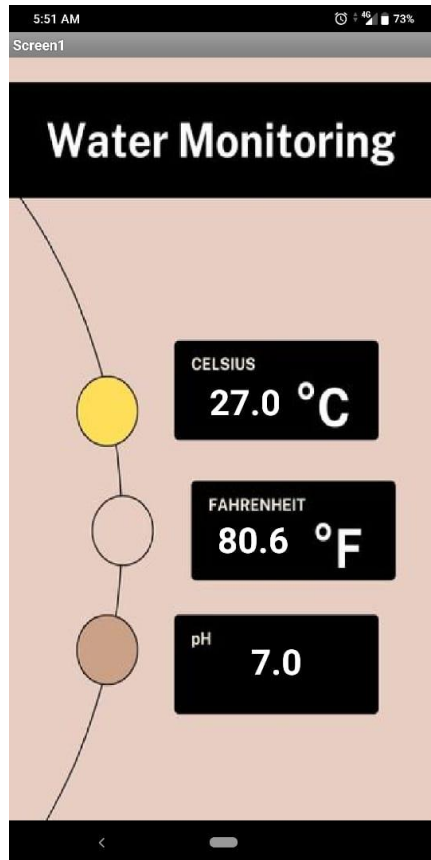
## **5. SIMULASI**

- a. Buka aplikasi yang sudah terinstall sebelumnya.



*Gambar 7. APK on android*

- b. Setelah aplikasi dibuka, maka aplikasi sudah memonitoring suhu dan pH air secara realtime. Aplikasi akan berada dalam mode offline ketika tidak ada internet masuk dan hanya akan menampilkan nilai defaultnya saja.



*Gambar 8. APK interface*