

DAFTAR PUSTAKA

- Bareta, Bintara Putra Candra, Alex Harijanto, And Maryani Maryani. 2021. "Rancang Bangun Alat Ukur Sistem Monitoring Ph, Temperatur, Dan Kelembapan Akuarium Ikan Hias Berbasis Arduino Uno." *Jurnal Pembelajaran Fisika* 10(1):1. Doi: 10.19184/Jpf.V10i1.21900.
- Cahyono, Ivana Yunisa. 2018. "Pembuka Kap Dan Bagasi Mobil Menggunakan Smartphone Berbasis Bluetooth." *Pembuka Kap Dan Bagasi, Smartphone, Motor Dynamo Starter*.
- Dvali, M. F., And M. D. Belonin. 1966. "Prospects For Deep And Ultradeep Oil And Gas Deposits In U.S.S.R." *International Geology Review* 8(6):665–75. Doi: 10.1080/00206816609474324.
- Ertayan, Pretty Veronica, Porman Pangaribuan, And Agung Surya Wibowo. 2019. "Sistem Monitoring Dan Mengontrol Akuarium Dalam Pemeliharaan Ikan Hias Dari Jarak Jauh (System Monitoring And *Kontrolling* The Akuarium In The Maintenance Fish From A Distance)." 6(2):3102–8.
- Fazil, Muhammad, Saiful Adhar, And Riri Ezraneti. 2017. "Efektivitas Penggunaan Ijuk, Jerami Padi Dan Ampas Tebu Sebagai Filter Air Pada Pemeliharaan Ikan Mas Koki (*Carassius Auratus*)." *Acta Aquatica: Aquatic Sciences Journal* 4(1):37. Doi: 10.29103/Aa.V4i1.322.
- Fitrianti, Yesi Okta, Hendra Kurniawan, And Sapta Nugraha. 2016. "Sistem Monitoring Suhu Realtime Pada Kolam Pembelian Ikan Bebrasis Cloud Computing." *Jurusan Teknik Informatika, Fakultas Teknik, Universitas Maritim Raja Ali Haji* (July):1–23.
- Haidir, Muhammad. 2017. *Rancang Bangun Alat Pengendalian Kekeruhan Air Pada Akuarium Berbasis Arduino Uno*.
- Karimah, Annisa, Iwang Gumilar, And Zahidah Hasan. 2012. "Analisis Prospektif Usaha Budidaya Ikan Hias Ari Tawar Di Taman Akuarium Air Tawar (TAAT) Dan Taman Mini Indonesia (TMII) Jakarta." *Jurnal Perikanan Dan Kelautan* 3(3):145–56.
- Murti Purwantiningsih, Dwi. 2018. "C, Yang Mempengaruhi Suhu Air Akuarium Salah satunya Konsentrasi Oksigen Terlarut Pada Air. Salah Satu Contoh Pengaruh Terhadap Keadan Air Yaitu Air Akan Berbau Tidak Sedap. Konsentrasi Oksigen Terlarut Pada Air Dipengaruhi Halaman 224 Dari 451." *Prosiding* 04(1):224–31.
- N.T.Annisa. 2017. "Rancang Bangun Alat Penggantian Air Dan." *Elektro, Pendidikan Teknik Elektro, Jurusan Teknik Teknik, Fakultas* 11.
- Nugrahanto, Indrawan, Teknik Elektro, Universitas Wisnuwardhana, And Malang Email. 2017. "Pembuatan Water Level Sebagai Pengendali Water Pump

- Otomatis Berbasis Transistor.” *Jurnal Ilmu-Ilmu Teknik - Sistem* 13(1):59–70.
- Pratama, Ovan Adytyia Saputra. 2017. “Rancang Bangun Monitoring Penggunaan Air PDAM Berbasis Arduino UNO.” 1–42.
- Pratama, Rizki Priya. 2017. “Aplikasi Webservice Esp8266 Untuk Pengendali Peralatan Listrik.” 17(2). Doi: 10.31227/Osf.Io/Pjwxd.
- Prayogatama. 2020. “Aplikasi Monitoring Penjernihan Air Pada Akuarium Ikan Mas Koki Berbasis Android.” *Eprints.Uty.Ac.Id*.
- Putra Asmara, Riyan Kharisma. 2020. “Rancang Bangun Alat Monitoring Dan Penanganan Kualitas Ait Pada Akuarium Ikan Hias Berbasis *Internet Of Things* (IOT).” *Jurnal Teknik Elektro Dan Komputer TRIAC* 7(2):69–74. Doi: 10.21107/Triac.V7i2.8148.
- Ramadona, Ahmad Suci, Edy Victor Haryanto, And Muhammad Rusdi Tanjung. 2015. “Perancangan Alat Pengganti Air Akuarium Otomatis Berbasis Mikrokontroler Atmega8.” *CSRID (Computer Science Research And Its Development Journal)* 6(1):1. Doi: 10.22303/Csrid.6.1.2014.1-10.
- Sanad, Edwin Adrin Wihelmus. 2019. “Pemanfaatan Realtime Database Di Platform Firebase Pada Aplikasi E-Tourism Kabupaten Nabire.” *Jurnal Penelitian Enjiniring* 22(1):20–26. Doi: 10.25042/Jpe.052018.04.
- Sari, Maulana, Muhammmad Hatta, And Asep Permana. 2014. “Acta Aquatica.” *Acta Aquatica* 1(1):24–30.
- Sari, Meika Puspita. 2019. “Pelatihan Pembuatan Akuarium Mini Dan Teknik Pemeliharaan Ikan Hias Di Kecamatan Alang-Alang Lebar.” *Suluh Abdi: Jurnal Ilmiah Pengabdian Kepada Masyarakat* 1(2):94–97.
- Sirait, Fadli, Ilham Septian Herwiansya, And Fina Supegina. 2017. “PENINGKATAN EFISIENSI SISTEM PENDISTRIBUSIAN AIR DENGAN MENGGUNAKAN Iot (Internet Of Things).” *Jurnal Elektro* 8(3):234–39.
- Stankovic, John A. 2014. “Research Directions For The Internet Of Things.” *IEEE Internet Of Things Journal* 1(1):3–9. Doi: 10.1109/IIOT.2014.2312291.
- Suyanti, Eppy Yundra. 2019. “Rancang Bangun Deteksi Detak Jantung Manusia Dengan Metode Pulsesensorberbasis Iot (Internet Of Things).” *Jurnal Teknik Elektro* 8(1):191–98.
- Tadeus, Dista Yoel, Khasnan Azazi, And Didik Ariwibowo. 2019. “Model Sistem Monitoring PH Dan Kekeuhan Pada Akuarium Air Tawar Berbasis Internet Of Things.” *Metana* 15(2):49–56. Doi: 10.14710/Metana.V15i2.26046.
- Wicaksono, Mochamad Fajar. 2019. “Implementasi Modul Wifi Nodemcu Esp8266 Untuk Smart.” *Komputasi: Jurnal Ilmiah Ilmu Komputer Dan*

Matematika 16(2):283–88.

L

A

M

P

I

R

A

N

Lampiran 1 Source Code Program

Smart Akuarium

Kode inisialisasi library

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
// Lib Ultrasonic
#include <NewPing.h>
// Lib Suhu DS18B20
#include <OneWire.h>
#include <DallasTemperature.h>
// lib Firebase
#include <ESP8266WiFi.h> // Wifi
#include <FirebaseESP8266.h> // Firebase
#include <ArduinoJson.h> // Parsing Data
// Lib Waktu NTP
#include <NTPClient.h>
#include <WiFiUdp.h>
#include <ESP8266HTTPClient.h>
```

Kode mengkoneksikan jaringan WIFI

```
// set Wifi
#define WIFI_SSID "rahasianegara"
#define WIFI_PASSWORD "123rahasianegara"
```

Kode koneksi ke firebase database

```
#define API_KEY "ckYse9FMgpeqrPUt1JhtG1bueeMC7prJk054TYZu"
#define DATABASE_URL "sm-akuarium-default-rtdb.firebaseio.com"
// Object Firebase
FirebaseData fbdo;
```

Kode konfigurasi pin sensor dan relay

```
#define sens_temp D7
#define sens_turb A0
#define sens_us_e D6 // US Echo
#define sens_us_t D5 // US Trigger
#define rl_heater D4 // Heater
#define rl_cooler D3 // Kipas
#define rl_water_add D0 // Tambah Air
#define rl_water_red 1 // Kurangi Air
```

Kode memanggil variable lcd, suhu dan ultrasonic

```
LiquidCrystal_I2C lcd(0x27,16,2); // LCD
NewPing sonar(sens_us_t, sens_us_e, 100); // US
OneWire oneWire(sens_temp); // Suhu
DallasTemperature sensors(&oneWire); // Suhu
```

Kode menampilkan simbol derajat

```
byte degree[] = {
  B00111,
  B00101,
  B00111,
  B00000,
  B00000,
  B00000,
  B00000,
  B00000
};
```

Kode pembacaan nilai analog, realtime data dan realtime kondisi

```
int tbd_adc=0, tbd_val=0, wl=0;
// RT Data
float temp=0, rt_temp=0;
int rt_turb=0, rt_wlvl=0, val_wlvl=0;
// RT Con
String con_water_temp, con_water_turb, con_water_level;
String control_json, settings_json;
```

Kode menampilkan data agar rapi di lcd

```
char level[4];
char turbidity[4];
```

Kode request data yang diambil di internet dari aplikasi

```
String sw_manual_auto="manual";
boolean sw_water_cooler=false, sw_water_heater=false,
sw_water_addition=false, sw_water_reduction=false;
int set_min_temperature=20, set_max_temperature=32,
set_min_turbidity=0, set_max_turbidity=0, set_min_water_level=10,
set_max_water_level=90;
```

Kode interval delay realtime data yang dikirim ke firebase dan inialisasi nilai utc, standar waktu yang umum digunakan di seluruh dunia

```
unsigned long startDelay, currentDelay;
// NTP
String current_time, current_date;
WiFiUDP ntpUDP;
NTPClient timeClient(ntpUDP);
// Nama Bulan
String months[12]={"Januari", "Februari", "Maret", "April", "Mei", "Juni",
"Juli", "Augustus", "September", "Oktober", "November", "Desember"};
```

Kode koneksi ke wifi

```
void setup() {
// Koneksi Wifi
WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
//Serial.print("Connecting to Wi-Fi");
while (WiFi.status() != WL_CONNECTED)
{
delay(300);
}
```

Kode memulai koneksi ke firebase database

```
Firestore.begin(DATABASE_URL, API_KEY);
// Auto Reconnect
Firestore.reconnectWiFi(true);
```

Kode konfigurasi input dan output serta memulai koneksi ke sensor suhu

```
pinMode(sens_temp, INPUT); // Suhu
// Konfigurasi Output
pinMode(rl_heater, OUTPUT); digitalWrite(rl_heater, HIGH);
pinMode(rl_cooler, OUTPUT); digitalWrite(rl_cooler, HIGH);
pinMode(rl_water_add, OUTPUT); digitalWrite(rl_water_add, HIGH);
pinMode(rl_water_red, OUTPUT); digitalWrite(rl_water_red, HIGH);
// koneksi ke DS18B20
sensors.begin();
```

Kode memulai koneksi NTP

```
timeClient.begin();  
timeClient.setTimeOffset(28800); //GMT +8
```

Kode LCD

```
lcd.init();  
lcd.backlight(); // Lammpu latar LCD  
lcd.setCursor(0,0);  
lcd.print(" SMART AKUARIUM ");  
lcd.setCursor(0,1);  
lcd.print(" Version 1.0.0 ");  
lcd.createChar(1, degree);  
lcd.clear();  
}
```

Kode proses pemanggilan sebuah fungsi

```
void loop() {  
  request_data();  
  mode_manual:  
  read_sensors(); display_lcd(); ntp(); request_data(); set_data(); post_data();  
}
```

Kode pembacaan perintah manual dan otomatis dari Aplikasi

```
if(Firebase.getString(fbdo, "/request/auto-manual")) {  
  if(fbdo.to<String>() == "manual"){  
    // Perintah untuk menyalakan dan mematikan relay  
    if(sw_water_cooler==true) {digitalWrite(rl_cooler,LOW); }else{  
digitalWrite(rl_cooler,HIGH);}  
    if(sw_water_heater==true) {digitalWrite(rl_heater,LOW); }else{  
digitalWrite(rl_heater,HIGH);}  
    if(sw_water_addition==true) {digitalWrite(rl_water_add,LOW);}else{  
digitalWrite(rl_water_add,HIGH);}  
    if(sw_water_reduction==true){digitalWrite(rl_water_red,LOW);}else{  
digitalWrite(rl_water_red,HIGH);}  
    //Serial.println("Mode Manual");  
  }  
  if(fbdo.to<String>() == "auto"){
```



```

digitalWrite(rl_cooler,HIGH);
  digitalWrite(rl_heater,HIGH);
  digitalWrite(rl_water_add,HIGH);
  digitalWrite(rl_water_red,HIGH);
  //Serial.println("Enter to Mode Auto");
  goto mode_auto;
}
}
goto mode_manual;

mode_auto:
read_sensors(); display_lcd(); ntp(); request_data(); set_data(); post_data();
if(Firebase.getString(fbdo, "/request/auto-manual")) {
  if(fbdo.to<String>() == "auto"){
    // Cooler
    if(rt_temp >= set_max_temperature){digitalWrite(rl_cooler, LOW);}
    if(rt_temp <= set_min_temperature){digitalWrite(rl_cooler, HIGH);}
    // Heater
    if(rt_temp <= set_min_temperature){digitalWrite(rl_heater, LOW);}
    if(rt_temp >= set_max_temperature){digitalWrite(rl_heater, HIGH);}
    if((rt_temp >= set_min_temperature) && (rt_temp <=
set_max_temperature)){
      digitalWrite(rl_cooler, HIGH);
      digitalWrite(rl_heater, HIGH);
    }
    if(rt_wlvl <= set_min_water_level) {digitalWrite(rl_water_add,LOW); }
    if(rt_wlvl >= set_max_water_level) {digitalWrite(rl_water_add,HIGH);}
    if(rt_turb <= 17) {digitalWrite(rl_water_red,LOW);
digitalWrite(rl_water_add,HIGH);}
    if(rt_wlvl < set_min_water_level) {digitalWrite(rl_water_red,HIGH); }
  }
  if(fbdo.to<String>() == "manual"){
    digitalWrite(rl_cooler,HIGH);
    digitalWrite(rl_heater,HIGH);
    digitalWrite(rl_water_add,HIGH);
    digitalWrite(rl_water_red,HIGH);
    Serial.println("Enter to Mode Manual");
    goto mode_manual;
  }
}
goto mode_auto;
}

```

Data

Kode request data dengan format JSON dari firebase database

```
void request_data(){
    if(Firebase.getJSON(fbdo, "request")) {
        // Save string
        settings_json = fbdo.jsonString();
    }
}
```

Kode memarsing data dan mengecek error deserialize

```
StaticJsonDocument<512> doc;
DeserializationError error = deserializeJson(doc, settings_json);

if (error) {
    Serial.print(F("deserializeJson() failed: "));
    Serial.println(error.f_str());
    return;
}
```

Kode menyimpan data request dari aplikasi ke dalam variable tersendiri untuk bisa diambil dan dikelola

```
const char* auto_manual = doc["auto-manual"];
sw_manual_auto = auto_manual;
set_min_turbidity = doc["set-min-turbidity"];
set_max_turbidity = doc["set-max-turbidity"];
set_min_temperature = doc["set-min-temperature"];
set_max_temperature = doc["set-max-temperature"];
set_min_water_level = doc["set-min-water-level"];
set_max_water_level = doc["set-max-water-level"];
// Tombol dari aplikasi
sw_water_cooler = doc["water-cooler"];
sw_water_heater = doc["water-heater"];
sw_water_addition = doc["water-addition"];
sw_water_reduction = doc["water-reduction"];
}
}
```

Kode pengiriman data ke firebase database dengan delay 3 detik dan setingan kondisi dari data parameter pengukuran

```
void set_data(){
  currentDelay = millis();
  if(currentDelay - startDelay >= 3000){
    startDelay = currentDelay;

    // Set con tempertaure
    if(rt_temp >= set_min_temperature && rt_temp <=
set_max_temperature){
      con_water_temp="Suhu Normal";
    }
    if(rt_temp < set_min_temperature){
      con_water_temp="Suhu Kurang";
    }
    if(rt_temp > set_max_temperature){
      con_water_temp="Suhu Meningkat";
    }
    // Set con turbidity

    if(rt_turb <= 50){
      con_water_turb="Air Keruh";
    }
    if(rt_turb >= 51){
      con_water_turb="Air Jernih";
    }
    // Set con water level
    if((rt_wlvl >= set_min_water_level) && (rt_turb <=
set_max_water_level)){
      con_water_level="Air Cukup";
    }
    if(rt_wlvl < set_min_water_level){
      con_water_level="Air Kurang";
    }
    if(rt_wlvl > set_max_water_level){
      con_water_level="Air Penuh";
    }
  }
}
```

Kode pembuatan format nilai data untuk dikirim ke firebase

```
Firestore rt_json;
// RT Value
rt_json.add("water-temperature", rt_temp);
rt_json.add("water-turbidity", rt_turb);
rt_json.add("water-level", rt_wlvl);
rt_json.add("adc-turbidity", tbd_adc);
// RT Condition
rt_json.add("con-water-temperature", con_water_temp);
rt_json.add("con-water-turbidity", con_water_turb);
rt_json.add("con-water-level", con_water_level);
Firestore.setJSON(fbdo, "data", rt_json);
}
}
```

LCD

Kode penampilan data di LCD

```
void display_lcd(){
  lcd.setCursor(0, 0); lcd.print("SUHU:" + String(rt_temp)); // Menampilkan
  Nilai Suhu
  lcd.setCursor(10,0); lcd.write(1); // Menampilkan derajat
  lcd.setCursor(11,0); lcd.print("C"); // Menampilkan C

  lcd.setCursor(0,1); lcd.print("TBD :" + String(turbidity)); // Menampilkan
  Nilai Turbidity
  lcd.setCursor(9,1); lcd.print("WL:" + String(level)); // // Menampilkan Nilai
  Water Level
  lcd.setCursor(15,1);lcd.print("%");
}
```

NTP

Kode untuk mendapatkan nilai waktu realtime

```
void ntp(){
  timeClient.update();
  unsigned long epochTime = timeClient.getEpochTime();
  struct tm *ptm = gmtime ((time_t *)&epochTime);

  current_time = timeClient.getFormattedTime();
}
```

```

// Tanggal
int monthDay = ptm->tm_mday;
// Bulan
int currentMonth = ptm->tm_mon+1;
String currentMonthName = months[currentMonth-1];
// Tahun
int currentYear = ptm->tm_year+1900;

// Gabungkan format tanggal
current_date = String(monthDay) + " " + String(currentMonthName) + " " +
String(currentYear);
}

```

Post Data

Kode penjadwalan pengiriman data ke firebase dengan menyimpan data pengukuran setiap 1 menit

```

void post_data(){
currentPost = millis();
if(currentPost - startPost >= 60000){
  FirebaseJson post_json;
  // RT Value
  post_json.add("Suhu Air", rt_temp);
  post_json.add("Kekeruhan Air", rt_turb);
  post_json.add("Ketinggian Air", rt_wlvl);
  Firebase.setJSON(fbdo, "/data-kumulatif/" + String(current_date) + "/" +
String(current_time), post_json);
}
}

```

Sensors

Kode pengdeklarasian variable untuk membaca nilai sensor

```

void read_sensors(){
// Read water temp
sensors.requestTemperatures();
temp = sensors.getTempCByIndex(0);
rt_temp = roundf(temp*100.0)/100.0;
}

```

```

// Read water turbidity
tbd_adc = analogRead(sens_turb);
rt_turb = map(tbd_adc, set_max_turbidity, set_min_turbidity, 0, 100); //
Konversi ke persen
if(rt_turb<0)rt_turb=0;
if(rt_turb>100)rt_turb=100;

// Read Water level
val_wlvl= sonar.ping_cm(); // Baca sensor US
rt_wlvl = map(val_wlvl, 29, 4, 0, 100); // Konversi ketinggian air menjadi
persen
if(rt_wlvl<0)rt_wlvl=0;
if(rt_wlvl>100)rt_wlvl=100;
sprintf(level, "%3d", rt_wlvl); // Simpan ke char(tidak flicker LCD)
sprintf(turbidity, "%3d", rt_turb); // Simpan ke char(tidak flicker LCD)

delay(50);

}

```

Lampiran 2 Tampilan Aplikasi Android



Lampiran 3 Gambar Sistem Akuarium



