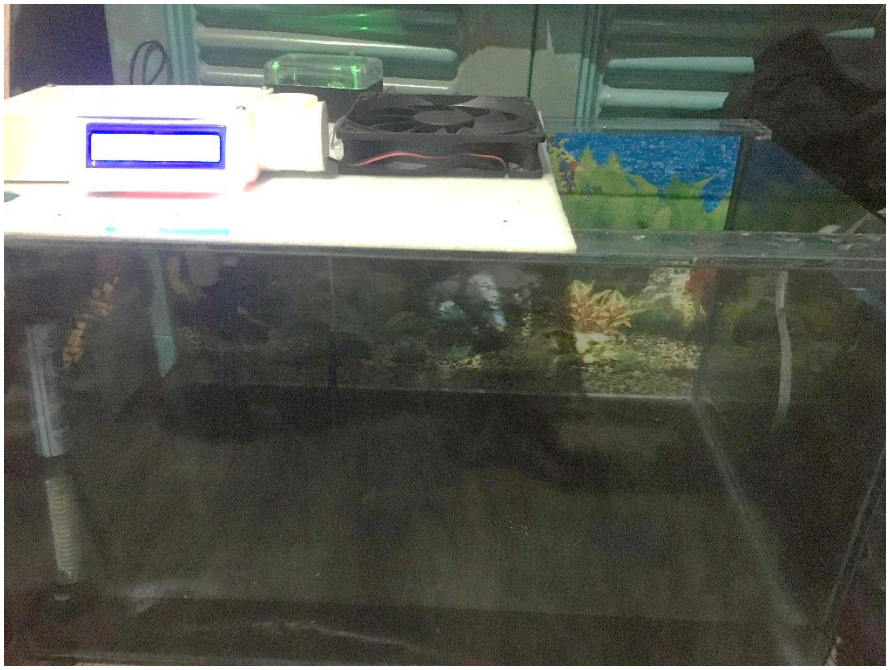
LAMPIRAN

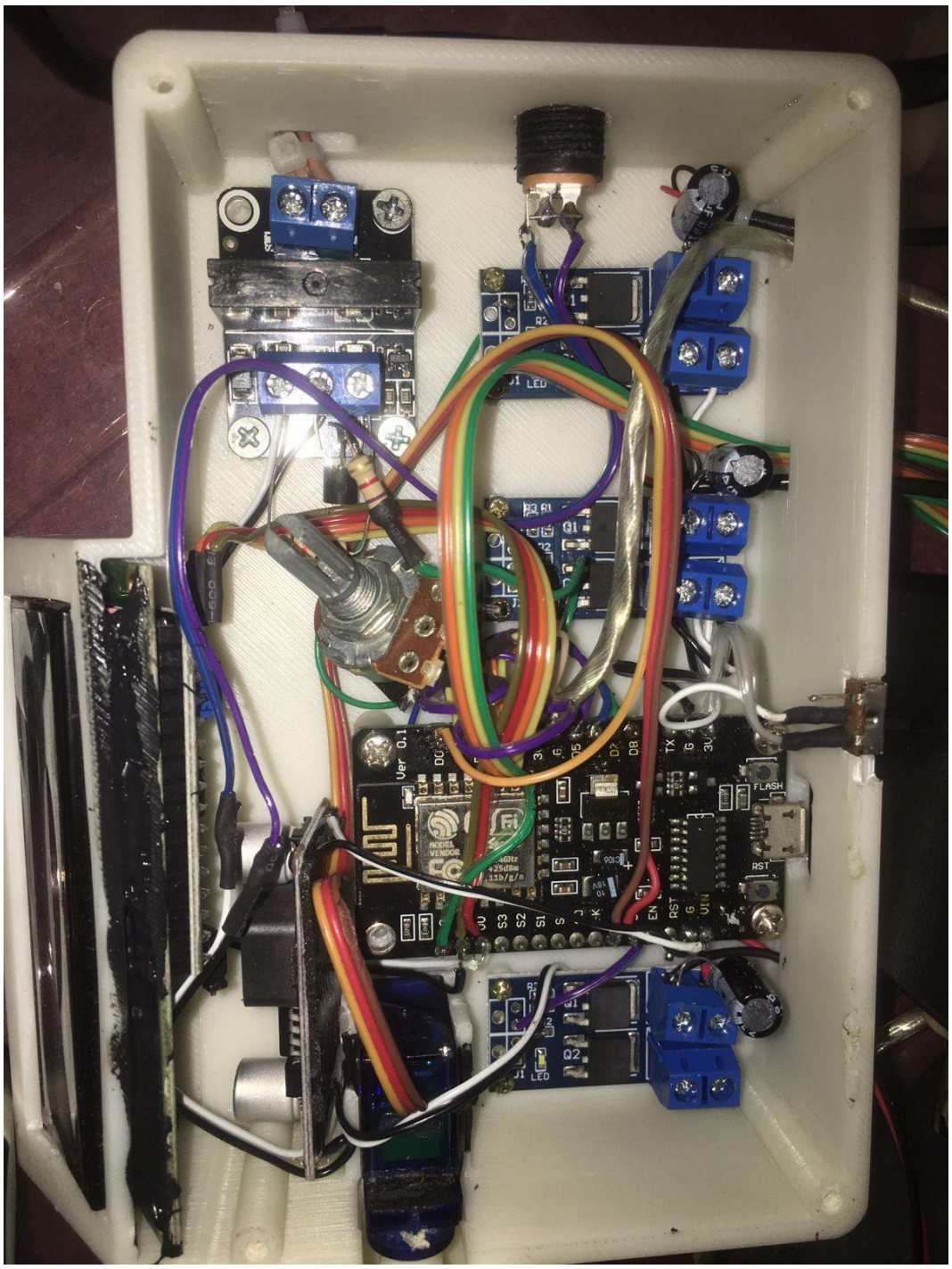
Lampiran 1. Sistem Kontrol Pada Aquarium Rancang Bangun Sistem Akuarium Ikan Discus Dengan Kontrol PH, Suhu dan Pemberian Pakan Menggunakan Aplikasi *Telegram.*



Lampiran 2. Alat Kontrol Rancang Bangun Sistem Akuarium Ikan Discus Dengan Kontrol PH, Suhu dan Pemberian Pakan Menggunakan Aplikasi *Telegram.*



Lampiran 3. Mikrokontroller Alat Rancang Bangun Sistem Akuarium Ikan Discus Dengan Kontrol PH, Suhu dan Pemberian Pakan Menggunakan Aplikasi *Telegram.*



```
#include <ESP8266WiFi.h>
#include <WiFiClientSecure.h>
#include <UniversalTelegramBot.h>
#include <OneWire.h>
#include <DallasTemperature.h>
#include <Servo.h> // tambahkan library Servo
#include <LCDI2C_Multilingual.h>

// Wifi network station credentials
#define WIFI_SSID "Andryan_805"
#define WIFI_PASSWORD "baba9807"
// Telegram BOT Token (Get from Botfather)
#define BOT_TOKEN "6566818844:AAG1xEyr59ur4_CCQ6NmQMca1hfb8EvpIss"

const unsigned long BOT_MTBS = 1000; // mean time between scan messages

X509List cert(TELEGRAM_CERTIFICATE_ROOT);
WiFiClientSecure secured_client;
UniversalTelegramBot bot(BOT_TOKEN, secured_client);
unsigned long bot_lasttime; // last time messages' scan has been done

LCDI2C_Generic lcd(0x27, 16, 2);
```

```
// Setup oneWire instance untuk komunikasi dengan perangkat OneWire
OneWire oneWire(2); // GPIO4 pada ESP8266 untuk DS18B20
DallasTemperature sensors(&oneWire);

#define ssr 0
#define fan 14
#define motor 12
#define motor2 13

bool state = false;

Servo servo; // Inisialisasi objek Servo

const int pH_Pin = A0;
float Po = 0;
float PH_Step;
int nilai_analog_PH;
double TeganganPh;

float PH4 = 3.010;
float PH7 = 2.910;
```

```
unsigned long startMillis; //some global variables available anywhere in the program

unsigned long currentMillis;

const unsigned long period = 10000;


void handleNewMessages(int numNewMessages) {

Serial.print("handleNewMessages ");

Serial.println(numNewMessages);


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

String chat_id = bot.messages[i].chat_id;

String text = bot.messages[i].text;


String from_name = bot.messages[i].from_name;

if (from_name == "")

from_name = "Guest";


if (text == "1") {

nilai_analog_PH = analogRead(pH_Pin);

TeganganPh = 3.3 / 1024.0 * nilai_analog_PH;

PH_Step = (PH4 - PH7) / 3;

Po = 7.00 + ((PH7 - TeganganPh) / PH_Step);
```

```
String message = "menampilkan kadar Ph air: " + String(Po, 1);
bot.sendMessage(chat_id, message, "");
}

if (text == "2") {
bot.sendMessage(chat_id, "Ph air sudah ditambahkan", ""); // Pesan saat pemberian pakan sedang dilakukan
analogWrite(motor, 255);
delay(3000);
analogWrite(motor, 0);
}

if (text == "3") {
bot.sendMessage(chat_id, "Ph air sudah dikurangi", ""); // Pesan saat pemberian pakan sedang dilakukan
analogWrite(motor2, 255);
delay(3000);
analogWrite(motor2, 0);
}

if (text == "4") {
sensors.requestTemperatures(); // Minta suhu dari sensor DS18B20
```

```
float temperatureC = sensors.getTempCByIndex(0); // Baca suhu dalam derajat Celsius

String message = "Suhu air saat ini: " + String(temperatureC, 1) + "°C";

bot.sendMessage(chat_id, message, "");

}


if (text == "5") {

bot.sendMessage(chat_id, "Pemberian pakan sedang dilakukan", ""); // Pesan saat pemberian pakan sedang dilakukan

servo.write(180); // Posisi tengah servo

delay(2000); // Tunggu 6 detik (6000 ms)

bot.sendMessage(chat_id, "Pakan telah diberikan", ""); // Pesan saat selesai memberikan pakan

servo.write(0); // Matikan servo

}


if (text == "/start") {

String welcome = "Hallo, Selamat datang dikontrolling ikan discus by, " + from_name + ".\n";

welcome += "Silahkan kirim perintah dengan mengirim angka.\n\n";

welcome += "1 : Cek pH air\n";

welcome += "2 : Tambah pH air\n";

welcome += "3 : Kurangi pH air\n";

welcome += "4 : Cek suhu air\n";
```

```
welcome += "5 : Pemberian pakan ikan\n";
bot.sendMessage(chat_id, welcome, "Markdown");
}
}
}
void setup() {
Serial.begin(115200);
Serial.println();
analogWrite(ssr, 0);
lcd.init();
lcd.backlight();
lcd.setCursor(0, 0);
lcd.print("Waiting for Network");
pinMode(pH_Pin, INPUT);
// attempt to connect to Wifi network:
configTime(0, 0, "pool.ntp.org"); // get UTC time via NTP
secured_client.setTrustAnchors(&cert); // Add root certificate for api.telegram.org
Serial.print("Connecting to Wifi SSID ");
Serial.print(WIFI_SSID);
WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
while (WiFi.status() != WL_CONNECTED) {
Serial.print(".");
```

```
delay(500);
}

Serial.print("\nWiFi connected. IP address: ");
Serial.println(WiFi.localIP());

// Check NTP/Time, usually it is instantaneous and you can delete the code below.
Serial.print("Retrieving time: ");
time_t now = time(nullptr);
while (now < 24 * 3600) {
Serial.print(".");
delay(100);
now = time(nullptr);
}

Serial.println(now);
// Menginisialisasi sensor suhu DS18B20
sensors.begin();
// Inisialisasi servo
servo.attach(16); // Attach servo ke pin GPIO2
servo.write(0); // Matikan servo saat booting
pinMode(ssr, OUTPUT); // put your setup code here, to run once:
```

```
pinMode(fan, OUTPUT);
pinMode(motor, OUTPUT);
pinMode(motor2, OUTPUT);
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Connected to");
lcd.print(WIFI_SSID);
delay(5000);
lcd.clear();
startMillis = millis();
}

void loop() {
currentMillis = millis();
if (millis() - bot_lasttime > BOT_MTBS) {
int numNewMessages = bot.getUpdates(bot.last_message_received + 1);
while (numNewMessages) {
Serial.println("got response");
handleNewMessages(numNewMessages);
numNewMessages = bot.getUpdates(bot.last_message_received + 1);
}
bot_lasttime = millis();
```

```
}
// Membaca suhu dari sensor DS18B20
sensors.requestTemperatures();
float temperatureC = sensors.getTempCByIndex(0);
// Logika kontrol suhu
if (temperatureC > 33) {
analogWrite(fan, 255);
} else {
analogWrite(fan, 0);
}
if (temperatureC < 32) {
analogWrite(ssr, 255);
} else {
analogWrite(ssr, 0);
}
delay(1000); // Delay sebelum pembacaan suhu berikutnya
// Membaca ph
nilai_analog_PH = analogRead(pH_Pin);
TeganganPh = 3.3 / 1024.0 * nilai_analog_PH;
PH_Step = (PH4 - PH7) / 3;
Po = 7.00 + ((PH7 - TeganganPh) / PH_Step);
lcd.setCursor(0, 0);
```

```
lcd.print("Suhu :");
lcd.setCursor(7, 0);
lcd.print(temperatureC);
lcd.setCursor(0, 1);
lcd.print("Ph :");
lcd.setCursor(5, 1);
lcd.print(Po);
}
```