

LAMPIRAN

Lampiran 1
Surat kesediaan Membimbing TA (Pembimbing 1)

SURAT KESEDIAAN MEMBIMBING TA

Yang bertanda tangan di bawah ini :

Nama : Miftakhul Huda, M.Kom.
NIDN : 0620127801
NIPY : 04.007.033
Jabatan Struktural : -
Jabatan fungsional : Lektor

Dengan ini Menyatakan bersedia untuk menjadi Pembimbing I pada Tugas Akhir mahasiswa berikut:

Nama : Niayu Ardelia Novella
Nim : 21041020
Program Studi : DIII Teknik Komputer

Judul TA : PENGATURAN PEMBERI PAKAN IKAN OTOMATIS
BERDASARKAN JUMLAH IKAN YANG ADA DI KOLAM

Demikian surat pernyataan saya buat agar dapat dipergunakan sebagaimana mestinya.

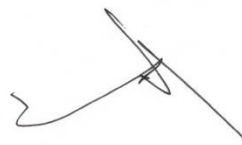
Tegal , 25 Maret 2024

Mengetahui
Ka. Prodi DIII Teknik Komputer,



Ida Afriliana, ST, M.Kom.
NIPY. 12.013.168

Dosen Pembimbing I,



Miftakhul Huda, M.Kom
NIPY. 04.007.033

Lampiran 2
Surat Kesiediaan Membimbing TA (Pembimbing 2)

SURAT KESEDIAAN MEMBIMBING TA

Yang bertanda tangan di bawah ini :

Nama : Arfan Haqiqi Sulasmoro, M.Kom.
NIDN : 0623037704
NIPY : 02.009.054
Jabatan Struktural : -
Jabatan fungsional : Asisten Ahli

Dengan ini Menyatakan bersedia untuk menjadi Pembimbing II pada Tugas Akhir mahasiswa berikut:

Nama : Niayu Ardelia Novella
Nim : 21041020
Program Studi : DIII Teknik Komputer


Judul TA : PENGATURAN PEMBERI PAKAN IKAN OTOMATIS
BERDASARKAN JUMLAH IKAN YANG ADA DI KOLAM

Demikian surat pernyataan saya buat agar dapat dipergunakan sebagaimana mestinya.

Tegal , 25 Maret 2024

Mengetahui
Ka. Prodi DIII Teknik Komputer,

Dosen Pembimbing II,


Ida Afriliana, ST, M.Kom.
NIPY. 12.013.168


Arfan Haqiqi Sulasmoro, M.Kom
NIPY. 02.009.054

Lampiran 3
Foto Dokumentasi



Lampiran 4

Kodingan Smartfeeder

```
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
#include <LiquidCrystal_I2C.h>
#include <Servo.h>

// WiFi configuration
const char* ssid = "mito";
const char* password = "saban123";
//const char* ssid = "Oppo";
//const char* password = "niayuwan0911";

// Server configuration
const char* host = "smarterfeeder.com";

// Pin configuration
#define TRIG_PIN 13 // D7 ultrasonik
#define ECHO_PIN 12 // D6 ultrasonik
#define BUZZER_PIN 15 // D8
#define ENA 0 //D3 L298N
#define SERVO_PIN 14 // D5
#define IN1 2 // D4 L298N
#define PH_PIN A0

//#define pin1 0
//#define pin2 1 //trig
//#define pin3 2
//#define pin4 3
//#define pin5 4
//#define pin6 5
//#define pin7 6
//#define pin8 7

// Initialize objects
WiFiClient NodeMCU;
LiquidCrystal_I2C lcd(0x27, 16, 2);
Servo servoMotor; // Servo object

// Global variables
bool statusPakan = false;
int sisaPakan = 0;
int jarak = 0;
int jumlahIkan = 1; // Starting with 2 fish
float nilaiPH = 0.0;
int speedMotorDc = 150;

// Calibration constants
const float PH4 = 3.226;
const float PH7 = 2.691;
const float TINGGI_WADAH = 19.0;

// Time interval
```

```

const long interval = 1000;
unsigned long previousMillis = 0;

void setup() {
  Serial.begin(115200);
  pinMode(ENA, OUTPUT);
  pinMode(IN1, OUTPUT);
  pinMode(ECHO_PIN, INPUT);
  pinMode(TRIG_PIN, OUTPUT);
  pinMode(BUZZER_PIN, OUTPUT);
  lcd.begin(16, 2);
  lcd.init();
  lcd.backlight();
  io.begin();
  servoMotor.attach(SERVO_PIN);
  setupWiFi();
}

void loop() {
  unsigned long currentMillis = millis();
  Serial.println(jumlahIkan);

  if (currentMillis - previousMillis >= interval) {
    previousMillis = currentMillis;
    getFishCountFromWeb(); //hitungjumlahikan
    tampilkanDisplay();
    updateFeedLevel(); //updatesisapakan
    beriPakanManual();
    beriPakanTerjadwal();
    sendSensorPH();
    checkFeedLevel();
    bacaPhAir();
  }
}

void setupWiFi() {
  WiFi.hostname("NodeMCU");
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print('.');
    lcd.setCursor(2, 0);
    lcd.print("Smartfeeder");
    delay(1000);
    lcd.setCursor(2, 1);
    lcd.print("Not Connected");
    delay(800);
    lcd.clear();
    delay(500);
  }
}

void tampilkanDisplay() {
  lcd.setCursor(0, 0);
  lcd.print("Sisa Pakan: " + String(sisaPakan) + "%");
  lcd.setCursor(0, 1);
  lcd.print("pH Air: " + String(nilaiPH, 2) + "%");
}

```

```

}

void bacaPhAir() {
    int nilaiAnalogPH = analogRead(PH_PIN);
    Serial.print("Nilai ADC PH: ");
    Serial.println(nilaiAnalogPH);

    double teganganPH = 3.3 / 1024.0 * nilaiAnalogPH;
    Serial.print("Tegangan PH: ");
    Serial.println(teganganPH, 3);

    float PH_step = (PH4 - PH7) / 3;
    nilaiPH = 7.00 + ((PH7 - teganganPH) / PH_step);
    Serial.print("Nilai PH Air: ");
    Serial.println(nilaiPH, 2);
}

void beriPakanManual() {
    if (checkServerStatus("/api/v1/monitoring") == "1") {
        beriPakan();
        motorHidup();
        sendSensorData();
    }
    motorMati();
    sendSensorData();
}

void beriPakanTerjadwal() {
    if (checkServerStatus("/api/v1/penjadwalanpakan/bacadata") ==
"1") {
        beriPakan();
        sendServerRequest("/api/v1/penjadwalanpakan/ubahstatus");
        sendSensorData();
    } else {
        sendSensorData();
    }
}

String checkServerStatus(const String& endpoint) {
    WiFiClient client;
    HTTPClient http;

    if (!client.connect(host, 80)) {
        Serial.println("Koneksi ke server bermasalah");
        return "";
    }

    String url = "http://" + String(host) + endpoint;
    http.begin(client, url);
    int httpCode = http.GET();
    if (httpCode == HTTP_CODE_OK) {
        return http.getString();
    }
    return "";
}

```

```

void sendServerRequest(const String& endpoint) {
    WiFiClient client;
    HTTPClient http;

    if (!client.connect(host, 80)) {
        Serial.println("Koneksi ke server bermasalah");
        return;
    }

    String url = "http://" + String(host) + endpoint;
    http.begin(client, url);
    http.GET();
}

void sendSensorData() {
    WiFiClient client;
    HTTPClient http;
    String data;

    if (jarak > 17) {
        data = "jarak=" + String(jarak) +
"&status_alarm=1&status_pakan=1&presentase_pakan=" +
String(sisaPakan);
    } else {
        data = "jarak=" + String(jarak) +
"&status_alarm=0&status_pakan=0&presentase_pakan=" +
String(sisaPakan);
    }
    // if (statusPakan) {
    //     data = "jarak=" + String(jarak) +
"&status_alarm=1&status_pakan=1&presentase_pakan=" +
String(sisaPakan);
    // } else {
    //     data = "jarak=" + String(jarak) +
"&status_alarm=0&status_pakan=0&presentase_pakan=" +
String(sisaPakan);
    // }

    http.begin(client, "http://" + String(host) +
"/api/v1/sensorjarak");
    http.addHeader("Content-Type", "application/x-www-form-
urlencoded");
    int httpCode = http.POST(data);
}

void sendSensorPH() {
    WiFiClient client;
    HTTPClient http;
    String data = "ph=" + String(nilaiPH, 2);

    http.begin(client, "http://" + String(host) +
"/api/v1/sensorair");
    http.addHeader("Content-Type", "application/x-www-form-
urlencoded");
    int httpCode = http.POST(data);
}

```



```

void beriPakan() {
  motorHidup();
  servoMotor.write(180); // Move servo to 180 degrees
  Serial.println("Servo On");
  // delay(3000); // Duration to open the feed lid
  // Tambahkan logika jumlah ikan di sini
  if (jumlahIkan < 5) {
    delay(2000); // Delay 2 detik jika jumlah ikan kurang dari 5
  } else if (jumlahIkan >= 6 && jumlahIkan <= 15) {
    delay(4000); // Delay 4 detik jika jumlah ikan antara 6 dan 15
  }
  servoMotor.write(0); // Move servo back to 0 degrees
  Serial.println("Servo Off");
  motorMati();
}

void motorHidup() {
  digitalWrite(ENA, HIGH);
  digitalWrite(IN1, HIGH);
  // digitalWrite(IN2, LOW);
  Serial.println("Motor Hidup");
}

void motorMati() {
  digitalWrite(ENA, LOW);
  digitalWrite(IN1, LOW);
  // digitalWrite(IN2, LOW);
  Serial.println("Motor Mati");
}

void updateFeedLevel() {
  // Calculate distance using ultrasonic sensor
  digitalWrite(TRIG_PIN, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIG_PIN, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);

  long duration = pulseIn(ECHO_PIN, HIGH);
  jarak = duration * 0.034 / 2;

  // Calculate remaining feed based on distance
  sisaPakan = map(jarak, 0, TINGGI_WADAH, 100, 0);
  sisaPakan = constrain(sisaPakan, 0, 100); // Ensure sisaPakan is
between 0 and 100
  Serial.println(jarak);
  Serial.println("Sisa Pakan " + String(sisaPakan));

  // Tambahkan logika jumlah ikan di sini
  if (jumlahIkan < 5) {
    delay(2000); // Delay 2 detik jika jumlah ikan kurang dari 5
  } else if (jumlahIkan >= 6 && jumlahIkan <= 15) {
    delay(4000); // Delay 4 detik jika jumlah ikan antara 6 dan 15
  }
}

```

```

}

void checkFeedLevel() {
  if (jarak > 17) {
    digitalWrite(BUZZER_PIN, HIGH); // Turn on buzzer
  } else {
    digitalWrite(BUZZER_PIN, LOW); // Turn off buzzer
  }
}

//kodingan jumlah ikan
void getFishCountFromWeb() {
  if (WiFi.status() == WL_CONNECTED) {
    HTTPClient http;
    WiFiClient client;
    // Masukkan URL API Laravel yang telah dibuat
    String url = "http://smarterfeeder.com/api/v1/jumlah-ikan"; //
    Ganti <IP_LARAVEL> dengan alamat IP Laravel

    http.begin(client, url);
    int httpCode = http.GET();

    if (httpCode > 0) {
      if (httpCode == HTTP_CODE_OK) {
        String payload = http.getString();
        Serial.println(payload); // Cetak payload yang diterima
        (jumlah ikan)
        jumlahIkan = payload.toInt(); // Simpan jumlah ikan ke
        variabel global jumlahIkan
      }
    }

    http.end();

    // Tambahkan logika jumlah ikan di sini
    if (jumlahIkan < 5) {
      delay(2000); // Delay 2 detik jika jumlah ikan kurang dari 5
    } else if (jumlahIkan >= 6 && jumlahIkan <= 15) {
      delay(4000); // Delay 4 detik jika jumlah ikan antara 6 dan 15
    }
    } else {
      Serial.println("Koneksi WiFi bermasalah");
    }
  }
}

```