

BAB V

SIMPULAN DAN SARAN

5.1 Simpulan

Berdasarkan dari hasil dan pembahasan Sistem *Prototype* Sistem Optimalisasi Pemeliharaan Ikan dengan Teknologi IoT pada Akuarium Menggunakan Arduino. dapat disimpulkan hal-hal sebagai berikut:

1. Menghasilkan sistem pemantauan dan pemeliharaan ikan berbasis IoT yang dapat mendeteksi kondisi air akuarium, sistem juga telah mengintegrasikan ke sensor TDS, Suhu, Turbidity, pH.
2. Memiliki indikasi penggunaan sistem IoT berpengaruh positif terhadap kualitas air yang memiliki pH 7,77 dan tds 73,80 ppm memiliki pengaruh terhadap kesehatan ikan lohan.
3. Sistem ini dapat memberikan efisiensi waktu dan tenaga terhadap pemantauan dan pemeliharaan ikan lohan, dan efektivitas dalam pemberian pakan.

5.2 Saran

Berdasarkan hasil penelitian dan analisis SWOT, beberapa saran yang dapat dipertimbangkan:

1. Sistem ini diimplementasikan untuk mengetahui kondisi air, penghobi ikan hias diharapkan jika sensor sudah menunjukkan indikasi tidak normal maka disarankan untuk segera mengganti air akuarium.
2. Sistem ini juga memiliki sistem pemberian pakan otomatis, disarankan kepada penghobi untuk memberi pakan ikan 3 kali sehari agar kualitas air tetap terjaga dan ikan mendapatkan nutrisi yang cukup.

3. Untuk penelitian selanjutnya sistem ini dapat dikembangkan untuk menampilkan histori pemantauan kualitas air di aplikasi, dan juga kontrol suhu air secara otomatis.

Dengan mempertimbangkan saran-saran ini, penelitian dapat lebih efektif dalam mencapai tujuannya dan memberikan manfaat yang lebih besar bagi pemilik akuarium ikan hias.



DAFTAR PUSTAKA

- Agronet. (2018). *Louhan si "jenong."* Portal Pertanian Dan Perikanan. <https://www.agronet.co.id/detail/budi-daya/perikanan/2917-Louhan-si-Jenong>
- Agus Wantoro, Slamet Samsugi, & Muhammad Joko Suharyanto. (2021). Sistem Monitoring Perawatan dan Perbaikan Fasilitas PT PLN (Studi Kasus : Kota Metro Lampung). *Jurnal Tekno Kompak*, 15(1), 116–130.
- Atthaariq Maulana, M., Harianto, Musayyanah, & Indah Kusumawati, W. (2022). Monitoring Dan Controlling Pemberi Pakan Ikan Otomatis Pada Akuarium Menggunakan MQTT. *Journal of Advances in Information and Industrial Technology*, 4(2), 69–80. <https://doi.org/10.52435/jaiit.v4i2.248>
- Chela Ramita, Indri Ariyanti, & Novianti, L. (2020). Aplikasi Monitoring dan Pengaduan Inventaris Barang Pada Jurusan Manajemen Informatika Berbasis Website. *JASISFO (Jurnal Sistem Informasi)*, 1(2), 79–89.
- Danar Virdaus. (2023). *Ikan Louhan*. www.Ikanpedia.Com.
- Deriota. (2023). *Pengertian Smart Aquarium Dan Hubungannya Dengan Smart Home*. Deriota. <https://deriota.com/news/read/1097/pengertian-smart-aquarium-dan-hubungannya-dengan-smart-home.html>
- Effendy, E., Siregar, E. A., Fitri, P. C., & Damanik, I. A. S. (2023). Mengenal Sistem Informasi Manajemen Dakwah (Pengertian Sistem, Karakteristik Sistem). *Jurnal Pendidikan Dan Konseling*, 5(2), 4343–4349.
- Fahrullah. (2021). Implementasi Pengujian Black Box Pada Sistem Informasi Monitoring Akademik Dengan Pendekatan Teknik Equivalence Partitions. *Jurnal Teknosains Kodepena*, 1(2), 94–100.
- Fernanda, R., & Wellem, T. (2022). Perancangan Dan Implementasi Sistem Pemberi Pakan. *Jurnal Teknik Informatika Dan Sistem Informasi*, 9(2), 1261–1274.
- Fonna, M. Z., Husaini, H., & Indrawati, I. (2020). Penerapan Iot (Internet Of Things) Untuk Pemberian Pakan Ikan Pada Aquarium. *Jurnal Teknologi Rekayasa Informasi Dan Komputer*, 3(2), 20–26.
- Garden, E. (2023). *Pengertian Akuarium, Sejarah, Jenis dan Manfaatnya*. Emir Garden. <https://www.emirgarden.com/pengertian-akuarium-sejarah-jenis-dan-manfaatnya/>
- Hayati, A. (2020). *Biologi Reproduksi Ikan*. Airlangga University Press. <https://books.google.co.id/books?id=YhrIDwAAQBAJ>
- Huda, I. A. (2020). Perkembangan Teknologi Informasi Dan Komunikasi (Tik) Terhadap

- Kualitas Pembelajaran Di Sekolah Dasar. *Jurnal Pendidikan Dan Konseling (JPDK)*, 2(1), 121–125. <https://doi.org/10.31004/jpdk.v1i2.622>
- Junaedi, & Ki, H. (2022). Smart Aquarium with IoT based as Monitoring in Fish Farming. *Bit-Tech*, 4(3), 116–122. <https://doi.org/10.32877/bt.v4i3.441>
- Koromari, B. I., & David, F. (2023). Perancangan Dan Implementasi Sistem Pakan Otomatis Dan Monitoring Tds Pada Aquarium Ikan Hias Berbasis Iot. *Jurnal Penerapan Teknologi Informasi Dan Komunikasi*, 02, 154–169.
- Kusumawijaya, W., & Aula, A. (2024). Water Quality Monitoring And Control System In Fish Cultivation Pond Using Arduino Cloud. *Computers, and Electricals Engineering Journal (TELECTRICAL)*, 1(3), 333–340. <https://doi.org/10.26418/telectrical.v1i3.73567>
- Laksmiana, I., Jingga, T. Z., Febrina, W., Khomarudin, A. N., Putri, E. E., Nazli, R., & Novita, R. (2022). *TEKNOLOGI INTERNET OF THINGS (IoT) DAN HIDROPONIK*. Goresan Pena. <https://books.google.co.id/books?id=dJWdEAAAQBAJ>
- Latuconsina, H., & Press, U. G. M. (2021). *Ekologi Ikan Perairan Tropis: Biodiversitas Adaptasi Ancaman dan Pengelolaannya*. Gadjah Mada University Press. <https://books.google.co.id/books?id=dvAUEAAAQBAJ>
- Lestari, D., Yuniarti, E., & Dinda Sari, Y. (2024). Sistem Monitoring Kualitas Air dan Pakan Otomatis Pada Aquarium Ikan Mas Koki Terintegrasi IoT. *BEES: Bulletin of Electrical and Electronics Engineering*, 4(3), 103–111. <https://doi.org/10.47065/bees.v4i3.4624>
- Masril, M. A., Aritonang, M. A. S., Saputra, M. H. K., Mulia, S. B., Simatupang, F., Hasan, H., Nurjannah, D. R., Iskandar, R., Andaria, A. C., & Caniago, D. P. (2024). *Mikrokontroler dan Arduino*. Yayasan Tri Edukasi Ilmiah. <https://books.google.co.id/books?id=3f0bEQAAQBAJ>
- Muhamad, F. N., Yulianto, D. T., & Aulia Fathurohman, M. A. (2023). Aquarium Monitoring and Automatic Feeding System Based on Internet of Things. *International Journal of Research and Applied Technology*, 3(1), 123–130. <https://doi.org/10.34010/injuratech.v3i1.10012>
- Muhammad, H., Ahfas, A., & Ayuni, S. D. (2023). SISTEM MONITORING KUALITAS AIR DAN PAKAN IKAN OTOMATIS BERBASIS IOT DENGAN SISTEM KENDALI APLIKASI BLYNK. *JURNAL TEKNIK MESIN, ELEKTRO DAN ILMU KOMPUTER*, 3(1), 41–46. https://scholar.google.com/scholar?hl=en&as_sdt=0%2C5&q=SISTEM+MONITORI NG+KUALITAS+AIR+DAN+PAKAN+IKAN+OTOMATIS+BERBASIS+IOT+DE

- Muthi, R. F. (2023). *PENGERTIAN PEMELIHARAAN MENURUT PARA AHLI*. SCRIBD. <https://www.scribd.com/document/348108220/Pengertian-Pemeliharaan>
- Nugraha, M. I., & Cahyono, B. D. (2023). *Arduino-based Automatic Fish Feeding Device Prototype. 1*(October), 1–11.
- Nur Hartawati, E. (2021). Identifying The Working Process Of Automatic Fish Feed Tool Based On Arduino Uno. *International Journal of Science, Technology & Management*, 2(5), 1830–1835. <https://doi.org/10.46729/ijstm.v2i5.356>
- Nurhadi, A. M., Midyanti, D. M., & ... (2023). Otomatisasi Pengontrolan Kualitas Air Pada Aquarium Ikan Arwana Berbasis IOT Menggunakan Logika Fuzzy Tsukamoto. *Jurnal Sistem ...*, 5(2), 476–488. <https://doi.org/10.30865/json.v5i2.6947>
- Octaviani, R. D., Subariyanti, H., Widiyanto, H., Arisanti, I., Hartono, M., Nurmayanti, S., Simamora, R. B., Sinaga, T. M., Simanjuntak, V. C., & Yulianto, A. R. (2023). *SUMBER DAYA MANUSIA DALAM ORGANISASI*. CV. Intelektual Manifes Media. <https://books.google.co.id/books?id=tsziEAAAQBAJ>
- Permatasari, D. I. (2020). Pengujian Aplikasi menggunakan metode Load Testing dengan Apache JMeter pada Sistem Informasi Pertanian. *Jurnal Sistem Dan Teknologi Informasi (JUSTIN)*, 8(1), 135. <https://doi.org/10.26418/justin.v8i1.34452>
- Prof. Dr. Ir. Amos Neolaka, M. P., & Grace Amialia A. Neolaka, S. P. M. P. (2015). *Landasan Pendidikan Dasar Pengenalan Diri Sendiri Menuju Perubahan Hidup: Edisi Pertama*. Prenada Media Group. <https://books.google.co.id/books?id=7BVNDwAAQBAJ>
- Prof. Dr. Yuni Pantiwati, M. M. M. P., Fendy Hardian Permana, M. P., Aminudin, S. K. M. C., & Tasya Novian Indah Sari, M. P. (2024). *PROTOTYPE E-MODUL MODEL PEMBELAJARAN LI-PRO-GP*. UMMPress. <https://books.google.co.id/books?id=o-USEQAAQBAJ>
- Ramdani, S. K., & Zakaria, H. (2023). Penerapan Framework Laravel Dalam Rancangan Aplikasi Data Warehouse Untuk Optimalisasi Pencarian Barang Dengan Metode Lifo (Studi Kasus : Kickoff Sports). *JURIHUM: Jurnal Inovasi Dan Humaniora*, 1(4), 486–498.
- Rina Noviana. (2022). Pembuatan Aplikasi Penjualan Berbasis Web Monja Store Menggunakan Php Dan Mysql. *Jurnal Teknik Dan Science*, 1(2), 112–124. <https://doi.org/10.56127/jts.v1i2.128>
- Sitorus, B. B. (2021). Perancangan dan Pembuatan Purwarupa Internet of Things (IoT)

- Pemantauan Kualitas Air Aquarium Multi Parameter. *Jurnal Informatika Dan Bisnis*. <http://jurnal.kwikkiangie.ac.id/index.php/JIB/article/view/745%0Ahttps://jurnal.kwikkiangie.ac.id/index.php/JIB/article/download/745/493>
- Sitorus Pane, U. F. S., & Andriyani, I. A. (2024). Sistem Pendeteksi Kualitas Air Pada Budidaya Ikan Air Tawar Berbasis Internet Of Things (IoT). *J-SISKO TECH (Jurnal Teknologi Sistem Informasi Dan Sistem Komputer TGD)*, 7(1), 84. <https://doi.org/10.53513/jsk.v7i1.9562>
- Somantri, N. T., Darwin, N., Nurjaman, D. F., Hidayat, M. R., & Winanti, N. (2022). *Sistem Monitoring Kualitas Air pada Akuarium Budidaya Ternak Ikan Guppy Menggunakan Mikrokontroler Berbasis IoT*. 21(02), 144–157. <http://creativecommons.org/licenses/by/4.0/>
- Subandri, M. A., & Danuri, D. (2023). Early Warning System Of Water Quality Changes In Fishponds. *Jurnal Teknologi Dan Open Source*, 6(1), 55–62. <https://doi.org/10.36378/jtos.v6i1.3109>
- Susila, A. (2023). Aplikasi Point Of Sales (POS) Berbasis Website Dengan Menggunakan Laravel (Studi Kasus: Bakmi Djowo). *Jurnal Ilmu Komputer Dan Pendidikan*, 2(1), 160–167.
- Syarief, A. N. L., Yanfika, H., & Mutolib, A. (2021). Pemahaman dan Persepsi Penyuluh Perikanan terhadap Penggunaan Teknologi Informasi dan Komunikasi dalam Penyuluhan di Kota Bandar Lampung. *ETHOS: Jurnal Penelitian Dan Pengabdian Kepada Masyarakat*, 9(1). <https://doi.org/10.29313/ethos.v9i1.6433>
- Syarifah, Chairullah Naury, & Wahyuni Nurindah Sulistiyowati. (2022). Perancangan Prototype Sistem Informasi Repository Skripsi Berbasis Web Di UNA'IM Yapis Wamena Papua. *SATESI: Jurnal Sains Teknologi Dan Sistem Informasi*, 2(1), 25–31. <https://doi.org/10.54259/satesi.v2i1.682>
- Tahir, A., Nurdjan, N., & Israkwaty. (2023). Konstruksi Mesin Pakan Ikan Otomatis. *Jurnal Vokasi Teknik Mesin Dan Fabrikasi Logam*, 2(1), 54–63.
- Yazid, A., Kusumawati, W. I., & Febriliana, R. (2023a). *Prototipe Sistem Monitoring Suhu , Ketinggian Air , dan Kontrol Otomatis pada Budidaya Ikan dalam Ember Berbasis IoT*. 5(2), 37–48. <https://journal.ittelkom-sby.ac.id/jaiit/article/view/371>
- Yazid, A., Kusumawati, W. I., & Febriliana, R. (2023b). Prototipe Sistem Monitoring Suhu , Ketinggian Air , dan Kontrol Otomatis pada Budidaya Ikan dalam Ember Berbasis IoT. *Journal of Advances in Information and Industrial Technology*, 5(2), 37–48. <https://doi.org/10.52435/jaiit.v5i2.371>

- Zalukhu, A., Purba, S., Darma, D., Zalukhu¹, A., Purba², S., Darma³, D., Teknik Informatika, M., & Industri, F. T. (2023). Perangkat Lunak Aplikasi Pembelajaran Flowchart. *Jurnal Teknologi, Informasi Dan Industri*, 4(1), 61–70.
<https://ejurnal.istp.ac.id/index.php/jtii/article/view/351>
- Zulfa, I., & Wanda, R. (2023). Klik: kajian ilmiah informatika dan komputer rancangan sistem informasi akademik berbasis website menggunakan php dan mysql. *Klik: Kajian Ilmiah Informatika Dan Komputer*, 3(4), 393–399.
<https://djournals.com/klik/article/view/617>



LAMPIRAN

Lampiran 1 Kode Arduino

```
#include <Wire.h> // Library untuk komunikasi I2C
#include <Adafruit_GFX.h> // Library grafik Adafruit
#include <Adafruit_SSD1306.h> // Library untuk layar OLED SSD1306
#include <OneWire.h> // Library OneWire untuk sensor suhu DS18B20
#include <DallasTemperature.h> // Library untuk sensor DS18B20
#include <WiFi.h> // Library Wi-Fi untuk ESP32
#include <ESP32Servo.h> // Library Servo untuk ESP32
#include <HTTPClient.h> // Library untuk HTTP client
#include <ArduinoJson.h> // Library untuk memproses JSON

#define SCREEN_WIDTH 128 // Lebar layar OLED
#define SCREEN_HEIGHT 64 // Tinggi layar OLED
#define SCREEN_ADDRESS 0x3C // Alamat I2C layar OLED

Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1); // Objek layar OLED

const char* ssid = "Galaxy A7"; // SSID Wi-Fi
const char* password = "11111111"; // Password Wi-Fi

// Mendefinisikan pin sensor
#define TdsSensorPin 35 // Pin sensor TDS
#define VREF 3.3 // Tegangan referensi ADC
#define SCOUNT 30 // Jumlah sampel untuk TDS
const int sensorkekeruhan = 39; // Pin sensor kekeruhan
const int ds18b20Pin = 32; // Pin sensor suhu DS18B20
#define PH_PIN 34 // Pin sensor pH

// Mendefinisikan pin tombol
const int buttonUpPin = 5;
const int buttonDownPin = 15;
const int buttonRightPin = 4;
const int buttonLeftPin = 18;
const int buttonOkPin = 19;
const int servoPin = 33; // Pin servo

Servo myServo; // Membuat objek Servo
int servoPosition = 0; // Posisi awal servo
```

```

OneWire oneWire(ds18b20Pin);
DallasTemperature sensors(&oneWire); // Objek sensor DS18B20

int analogBuffer[SCOUNT]; // Buffer bacaan TDS
int analogBufferTemp[SCOUNT];
int analogBufferIndex = 0; // Indeks buffer
float averageVoltage = 0; // Tegangan rata-rata
float tdsValue = 0; // Nilai TDS
float temperature = 25; // Suhu untuk kompensasi

float latestPH = 0.0; // Nilai pH terbaru
float latestTempC = 0.0; // Suhu terbaru dalam Celsius
float latestTDS = 0.0; // Nilai TDS terbaru
int latestTurbidity = 0; // Nilai kekeruhan terbaru

unsigned long lastButtonPressTime = 0; // Waktu terakhir tombol ditekan
const unsigned long debounceDelay = 200; // Delay untuk menghindari bounce pada tombol

enum DisplayMode { DISPLAY_PH, DISPLAY_TEMP, DISPLAY_TDS, DISPLAY_TURBIDITY }; // Mode
tampilan
DisplayMode currentDisplayMode = DISPLAY_PH; // Mode tampilan saat ini

void setup() {
    Serial.begin(115200); // Memulai komunikasi Serial

    if (!display.begin(SSD1306_SWITCHCAPVCC, SCREEN_ADDRESS)) {
        Serial.println(F("SSD1306 allocation failed")); // Menampilkan pesan jika alokasi gagal
        for (;;) // Loop tanpa akhir jika gagal
        }
    display.clearDisplay(); // Menghapus tampilan awal

    pinMode(TdsSensorPin, INPUT); // Mengatur pin sensor TDS sebagai input
    pinMode(sensorkekeruhan, INPUT); // Mengatur pin sensor kekeruhan sebagai input
    pinMode(PH_PIN, INPUT); // Mengatur pin sensor pH sebagai input

    pinMode(buttonUpPin, INPUT_PULLUP); // Mengatur pin tombol Up sebagai input dengan pull-up
    pinMode(buttonDownPin, INPUT_PULLUP); // Mengatur pin tombol Down sebagai input dengan pull-up
    pinMode(buttonRightPin, INPUT_PULLUP); // Mengatur pin tombol Right sebagai input dengan pull-up
    pinMode(buttonLeftPin, INPUT_PULLUP); // Mengatur pin tombol Left sebagai input dengan pull-up

```

```

pinMode(buttonOkPin, INPUT_PULLUP); // Mengatur pin tombol OK sebagai input dengan pull-up

myServo.attach(servoPin); // Menghubungkan objek servo ke pin
myServo.write(servoPosition); // Mulai di 0 derajat

Serial.print("Connecting to WiFi"); // Menampilkan pesan menghubungkan ke Wi-Fi
WiFi.begin(ssid, password); // Memulai koneksi Wi-Fi
while (WiFi.status() != WL_CONNECTED) { // Menunggu hingga terhubung
    delay(500);
    Serial.print("."); // Menampilkan titik selama proses koneksi
}
Serial.println(" connected!"); // Menampilkan pesan jika terhubung
Serial.print("IP address: "); // Menampilkan alamat IP
Serial.println(WiFi.localIP()); // Menampilkan alamat IP
}

// Fungsi untuk mendapatkan nilai median dari array
int getMedianNum(int bArray[], int iFilterLen) {
    int bTab[iFilterLen]; // Membuat salinan bacaan
    for (byte i = 0; i < iFilterLen; i++)
        bTab[i] = bArray[i];

    // Mengurutkan array
    for (int j = 0; j < iFilterLen - 1; j++) {
        for (int i = 0; i < iFilterLen - j - 1; i++) {
            if (bTab[i] > bTab[i + 1]) {
                int bTemp = bTab[i];
                bTab[i] = bTab[i + 1];
                bTab[i + 1] = bTemp;
            }
        }
    }
}

// Menghitung dan mengembalikan median
return (iFilterLen & 1) ? bTab[(iFilterLen - 1) / 2] : (bTab[iFilterLen / 2] + bTab[iFilterLen / 2 - 1]) / 2;
}

// Fungsi untuk memproses sensor
void processSensors() {
    static unsigned long analogSampleTimepoint = millis(); // Inisialisasi timer analog

```

```

if (millis() - analogSampleTimepoint > 400) { // Mengambil sampel setiap 40 ms
    analogSampleTimepoint = millis();
    analogBuffer[analogBufferIndex] = analogRead(TdsSensorPin); // Membaca sensor TDS
    analogBufferIndex++;
    if (analogBufferIndex == SCOUNT) {
        analogBufferIndex = 0; // Reset indeks jika buffer penuh
    }
}

// Memperbarui nilai TDS
for (int copyIndex = 0; copyIndex < SCOUNT; copyIndex++) {
    analogBufferTemp[copyIndex] = analogBuffer[copyIndex];
}
averageVoltage = getMedianNum(analogBufferTemp, SCOUNT) * VREF / 4096.0; // Menghitung tegangan
rata-rata

float compensationCoefficient = 1.0 + 0.02 * (temperature - 25.0); // Koefisien kompensasi suhu
float compensationVoltage = averageVoltage / compensationCoefficient; // Tegangan kompensasi
tdsValue = (133.42 * compensationVoltage * compensationVoltage * compensationVoltage - 255.86 *
compensationVoltage * compensationVoltage + 857.39 * compensationVoltage) * 0.5; // Menghitung nilai TDS

float sensorKekeruhanValue = analogRead(sensorkekeruhan); // Membaca nilai sensor kekeruhan
latestTurbidity = map(sensorKekeruhanValue, 112, 4095, 100, 0); // Memetakan nilai kekeruhan

sensors.requestTemperatures(); // Meminta pembacaan suhu
latestTempC = sensors.getTempCByIndex(0); // Mendapatkan suhu dalam Celsius

int sensorPHValue = analogRead(PH_PIN); // Membaca nilai sensor pH
float voltage = sensorPHValue * (3.3 / 4095.0); // Menghitung tegangan dari nilai pH
latestPH = 7 + ((2.5 - voltage) / 0.215); // Menghitung nilai pH
}

// Fungsi untuk mengirim data ke server
void sendDataToServer() {
    if (WiFi.status() == WL_CONNECTED) { // Memeriksa koneksi Wi-Fi
        HTTPClient http; // Membuat objek HTTPClient
        http.begin("https://157.245.52.6/sensor"); // Mengatur URL untuk permintaan
        http.addHeader("Content-Type", "application/x-www-form-urlencoded"); // Menambahkan header
    }
}

```

```

// Menyusun data untuk dikirim
String postData = "ph=" + String(latestPH) + "&suhu=" + String(latestTempC) + "&tds=" +
String(tdsValue) + "&turbidity=" + String(latestTurbidity);

int httpResponseCode = http.POST(postData); // Mengirim permintaan POST
if (httpResponseCode > 0) { // Memeriksa apakah permintaan berhasil
    String response = http.getString(); // Mendapatkan respon dari server
    Serial.println(httpResponseCode); // Menampilkan kode status HTTP
    Serial.println(response); // Menampilkan respon server
    // Memproses respon JSON
    DynamicJsonDocument doc(1024); // Membuat dokumen JSON
    DeserializationError error = deserializeJson(doc, response); // Mengurai respon

    if (!error) { // Memeriksa kesalahan saat mengurai
        // Memeriksa kunci "waktu_kasih_makan" dan nilainya
        if (doc["waktu_kasih_makan"]) {
            Serial.println("Triggering servo!"); // Menampilkan pesan pemicu servo
            myServo.write(360); // Memutar servo
            delay(1000); // Menunggu 1 detik
            myServo.write(0); // Kembali ke posisi awal
        }
        } else {
            Serial.print("Failed to parse JSON: "); // Menampilkan pesan kesalahan saat mengurai JSON
            Serial.println(error.c_str());
        }
    } else {
        Serial.print("Error on sending POST: "); // Menampilkan pesan kesalahan saat mengirim POST
        Serial.println(httpResponseCode);
    }
}

http.end(); // Mengakhiri koneksi
} else {
    Serial.println("WiFi Disconnected"); // Menampilkan pesan jika Wi-Fi terputus
}
}

// Fungsi untuk menampilkan data di layar OLED
void displayData() {
    display.clearDisplay(); // Menghapus layar untuk data baru
    display.setTextSize(2); // Mengatur ukuran teks untuk keterbacaan

```

```

display.setTextColor(SSD1306_WHITE); // Mengatur warna teks
display.setCursor(0, 0); // Mengatur posisi kursor

// Menampilkan berdasarkan mode saat ini
switch (currentDisplayMode) {
    case DISPLAY_PH:
        display.println("pH: "); // Menampilkan label pH
        display.print(latestPH); // Menampilkan nilai pH
        display.println(" pH"); // Menampilkan satuan pH
        break;
    case DISPLAY_TEMP:
        display.println("Suhu: "); // Menampilkan label suhu
        display.print(latestTempC); // Menampilkan nilai suhu
        display.println(" C"); // Menampilkan satuan Celsius
        break;
    case DISPLAY_TDS:
        display.println("TDS: "); // Menampilkan label TDS
        display.print(tdsValue, 0); // Menampilkan nilai TDS
        display.println(" ppm"); // Menampilkan satuan ppm
        break;
    case DISPLAY_TURBIDITY:
        display.println("Kekeruhan: "); // Menampilkan label kekeruhan
        display.print(latestTurbidity); // Menampilkan nilai kekeruhan
        display.println(" NTU"); // Menampilkan satuan NTU
        break;
}

// Menambahkan status konektivitas Wi-Fi ke layar
display.setTextSize(1); // Ukuran teks lebih kecil
display.setCursor(0, 50); // Mengatur posisi kursor
if (WiFi.status() == WL_CONNECTED) {
    display.println("WiFi: Terhubung"); // Menampilkan status terhubung
} else {
    display.println("WiFi: Tidak Terhubung"); // Menampilkan status tidak terhubung
}

display.display(); // Memperbarui tampilan dengan data baru
}

void loop() {

```

```

processSensors(); // Membaca nilai dari sensor
sendDataToServer(); // Mengirim data ke server

// Memeriksa status tombol tanpa delay
unsigned long currentMillis = millis(); // Mendapatkan waktu saat ini

// Status tombol
int buttonUpState = digitalRead(buttonUpPin);
int buttonDownState = digitalRead(buttonDownPin);
int buttonRightState = digitalRead(buttonRightPin);
int buttonLeftState = digitalRead(buttonLeftPin);
int buttonOkState = digitalRead(buttonOkPin);

// Tombol Up (Tampilkan pH terus-menerus)
if (buttonUpState == LOW && (currentMillis - lastButtonPressTime) > debounceDelay) {
    lastButtonPressTime = currentMillis; // Memperbarui waktu tekan tombol terakhir
    currentDisplayMode = DISPLAY_PH; // Mengubah mode tampilan ke pH
}

// Tombol Down (Tampilkan TDS terus-menerus)
if (buttonDownState == LOW && (currentMillis - lastButtonPressTime) > debounceDelay) {
    lastButtonPressTime = currentMillis;
    currentDisplayMode = DISPLAY_TDS; // Mengubah mode tampilan ke TDS
}

// Tombol Left (Tampilkan Suhu terus-menerus)
if (buttonLeftState == LOW && (currentMillis - lastButtonPressTime) > debounceDelay) {
    lastButtonPressTime = currentMillis;
    currentDisplayMode = DISPLAY_TEMP; // Mengubah mode tampilan ke suhu
}

// Tombol Right (Tampilkan Kekeruhan terus-menerus)
if (buttonRightState == LOW && (currentMillis - lastButtonPressTime) > debounceDelay) {
    lastButtonPressTime = currentMillis;
    currentDisplayMode = DISPLAY_TURBIDITY; // Mengubah mode tampilan ke kekeruhan
}

// Tombol OK (Putar Servo dua kali)
if (buttonOkState == LOW && (currentMillis - lastButtonPressTime) > debounceDelay) {
    lastButtonPressTime = currentMillis;

```

```

    for (int i = 0; i < 2; i++) { // Rotasi dua kali
        myServo.write(360); // Putar ke 360 derajat
        delay(1000);      // Delay untuk memberi waktu pada servo bergerak
        myServo.write(0);  // Kembali ke posisi awal
    }
}

displayData(); // Menampilkan data terbaru di OLED

// Mencetak semua bacaan ke Serial Monitor terus-menerus
Serial.print("pH: ");
Serial.println(latestPH); // Menampilkan nilai pH
Serial.print("Suhu: ");
Serial.print(latestTempC); // Menampilkan nilai suhu
Serial.println(" C"); // Menampilkan satuan Celsius
Serial.print("TDS: ");
Serial.print(tdsValue, 0); // Menampilkan nilai TDS
Serial.println(" ppm"); // Menampilkan satuan ppm
Serial.print("Kekeruhan: ");
Serial.print(latestTurbidity); // Menampilkan nilai kekeruhan
Serial.println(" NTU"); // Menampilkan satuan NTU
if (WiFi.status() == WL_CONNECTED) {
    Serial.println("WiFi: Terhubung"); // Menampilkan status terhubung
} else {
    Serial.println("WiFi: Tidak Terhubung"); // Menampilkan status tidak terhubung
}

Serial.println("-----");

delay(100); // Delay singkat untuk mencegah loop terlalu padat
}

```

Lampiran 2 Kode Laravel

1. Tab Home

```
<?php

namespace App\Livewire;
use App\Models\jadwal_pelet;
use Livewire\Component;
class Home extends Component
{

    public ?string $jam = "";
    public function save() {

        if(is_null($this->jam) || empty($this->jam)) {
            session()->flash("error", "Jam tidak boleh kosong");
            return redirect()->route('home');
        }

        jadwal_pelet::create([
            "jam" => $this->jam,
        ]);

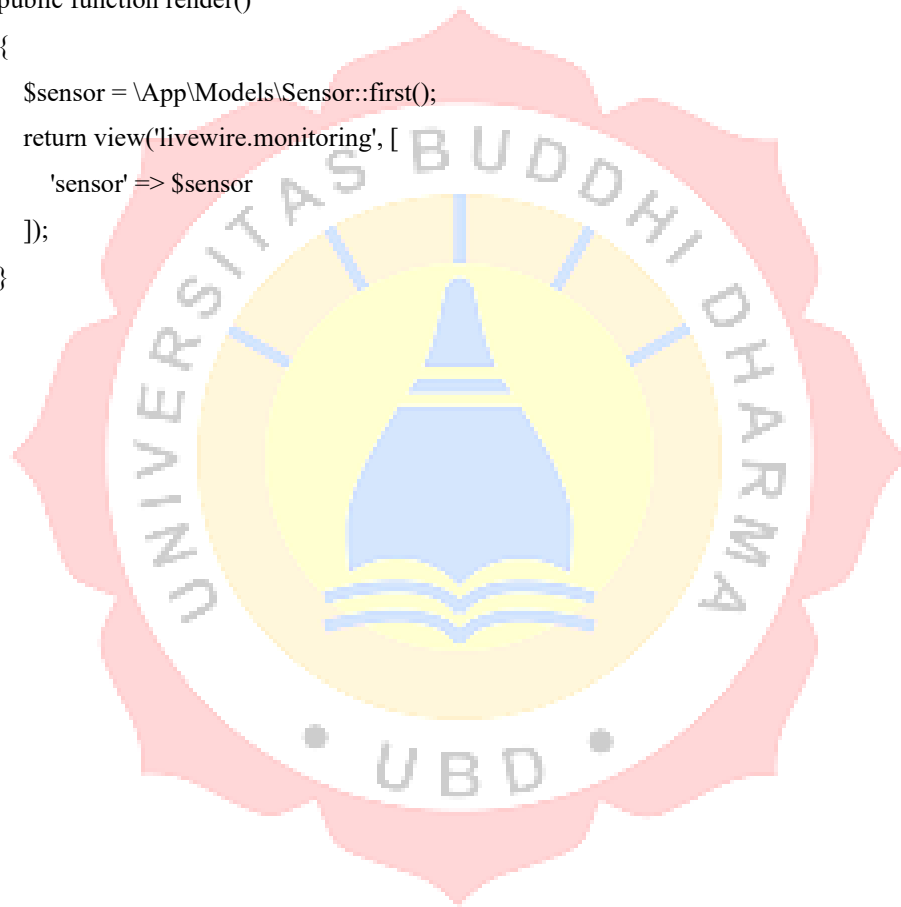
        session()->flash("message", "Berhasil tambah jadwal");
        return redirect()->route('home');
    }

    public function remove(string $id) {
        jadwal_pelet::findOrFail($id)->delete();

        session()->flash("message", "Berhasil hapus jadwal");
        return redirect()->route('home');
    }

    public function render()
    {
        $jadwals = jadwal_pelet::all();
        return view('livewire.home', [
            "jadwals" => $jadwals
        ]);
    }
}
```

```
}  
2. Tab Monitoring  
<?php  
  
namespace App\Livewire;  
  
use Livewire\Component;  
  
class Monitoring extends Component  
{  
    public function render()  
    {  
        $sensor = \App\Models\Sensor::first();  
        return view('livewire.monitoring', [  
            'sensor' => $sensor  
        ]);  
    }  
}
```



Lampiran 3 Requirement Elicitation 1

Requirement Elicitation

No	Analisa Kebutuhan Sistem	Keterangan
	Saya ingin sistem dapat:	
1	Monitoring kualitas air	✓
2	Kontrol suhu otomatis	X
3	Pengelolaan pemberian pakan	✓
4	Notifikasi dan alarm	X
5	Akses remote melalui aplikasi/web	✓
6	Penyimpanan data historis	X
7	Antarmuka yang <i>user friendly</i>	✓
8	Kompabilitas dengan perangkat iot yang lain	✓

Tangerang, 05 Februari 2025

Pembimbing,



Dr. Yakub, M.Kom., M.M.

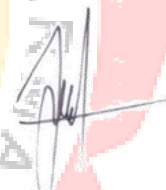
NIDN: 0304056901

Responden,



Hervin

Mahasiswa,



Andi Kusuma

NIM: 20201000037

Lampiran 4 Requirement Elicitation 1

Requirement Elicitation

No	Analisa Kebutuhan Sistem	Keterangan
	Saya ingin sistem dapat:	
1	Monitoring kualitas air	✓
2	Kontrol suhu otomatis	X
3	Pengelolaan pemberian pakan	✓
4	Notifikasi dan alarm	X
5	Akses remote melalui aplikasi/web	✓
6	Penyimpanan data historis	X
7	Antarmuka yang <i>user friendly</i>	✓
8	Kompabilitas dengan perangkat iot yang lain	✓

Tangerang, 05 Februari 2025

Pembimbing,



Dr. Yakub, M.Kom., M.M.

NIDN: 0304056901

Responden,



Ryan Salim

Mahasiswa,



Andi Kusuma

NIM: 20201000037



KARTU BIMBINGAN TA/SKRIPSI

NIM : 20201000037
Nama Mahasiswa : ANDI KUSUMA
Fakultas : Sains dan Teknologi
Program Studi : Teknik Informatika
Jenjang : Strata Satu
Tahun Akademik/Semester : 2024/2025 Genap
Dosen Pembimbing : Yakub, MM, M.Kom
Judul Skripsi : PROTOTYPE SISTEM MONITORING PEMELIHARAAN IKAN
DENGAN TEKNOLOGI IOT PADA AKUARIUM MENGGUNAKAN
ARDUINO

Tanggal	Catatan	Paraf
2024-09-25	Konsultasi judul skripsi	A
2024-10-02	Bab 1: Bagian latar belakang	A
2024-10-16	Bab 1 (latar belakang, identifikasi masalah, ruang lingkup, tujuan dan manfaat, dan sistematika penulisan)	A
2024-10-30	Bab 2: Tata cara penulisan kutipan dan penguraian teori yang berkaitan dengan topik penelitian	A
2024-11-12	Bab 3: Gambaran umum, analisis masalah, identifikasi kebutuhan, RE, Flowchart penggunaan metode, jadwal penelitian	A
2024-11-20	Bab 3: Revisi analisis masalah, identifikasi kebutuhan, RE, Flowchart penggunaan metode	A
2024-11-27	Bab 4: Pembahasan hasil penelitian, Flowchart Penggunaan Alat, Hasil Pembuatan Alat.	A
2024-12-04	Bab 4: Revisi Pembahasan hasil penelitian, Flowchart Penggunaan Alat, Hasil Pembuatan Alat dan testing sistem	A
2024-12-18	Hasil Angket dan Bab 5: Kesimpulan dan Saran	A

Mengetahui
Ketua Program Studi



Tangerang, 12 February 2025

Pembimbing




DAFTAR RIWAYAT HIDUP



BIODATA DIRI

Nama Lengkap	Andi Kusuma
Tempat, Tanggal Lahir	Tangerang, 04 Oktober 2002
Jenis Kelamin	Laki-laki
Agama	Buddha
Kewarganegaraan	Indonesia
Alamat	Jl Pinus 4 blok E-5/34, Kel. Kuta Jaya, Kec. Pasar Kemis, Kab Tangerang, Banten
Email	Kandi5372@gmail.com
No. Telepon	081514345317
IPK Terakhir	3.29

RIWAYAT PENDIDIKAN

- 2006 s/d 2007 TK ARIYA METTA
- 2007 s/s 2014 SDN PASAR BARU 1
- 2014 s/d 2017 SMP ARIYA METTA
- 2017 s/d 2020 SMK ARIYA METTA
- 2020 s/d 2025 UNIVERSITAS BUDDHI DHARMA

Tangerang, 04 Februari 2025

Hormat, Saya,

Andi Kusuma

NIM : 20201000037