

## BAB II

### METODE PENELITIAN

#### 2.1. Tempat dan Waktu Penelitian

Penelitian ini akan dilakukan pada kantor UPTD. Pusat Layanan Disabilitas dan Pendidikan Inklusi Kota Samarinda, dengan rentang waktu pengerjaannya yaitu selama 2 bulan.

#### 2.2. Objek Penelitian

Objek dalam penelitian ini adalah pengembangan sistem kontrol aerator otomatis yang menggunakan sensor *waterflow* sebagai sensor untuk mendeteksi aliran air, dengan mikrokontroler sebagai pusat pengendalian *relay* yang akan menonaktifkan dan mengaktifkan aerator lalu kemudian akan memberikan notifikasi pemberitahuan ke aplikasi *Blynk*. Penelitian ini mencakup perancangan perangkat keras, pengembangan perangkat lunak, dan uji coba sistem pada lingkungan akuarium. Objek penelitian ini melibatkan interaksi antara mikrokontroler ESP8266, sensor *waterflow*, *relay*, aktuator aerator, pompa sirkulasi air dan lingkungan akuarium.

#### 2.3. Alat dan Bahan

##### A. Mikrokontroler NodeMCU ESP8266

NodeMCU ESP8266 menggabungkan NodeMCU dan mikrokontroler ESP8266 menjadi satu, sehingga tidak perlu membuatnya terlebih dahulu. NodeMCU ESP8266 membantu pembuatan *prototype* IoT dengan menggunakan *sketch* di arduino IDE (Damayanti et al., 2023). NodeMCU ESP8266 merupakan mikrokontroler tambahan untuk dapat terhubung pada jaringan internet WiFi. Dalam NodeMCU ESP8266 sudah dilengkapi dengan prosesor, memori, dan GPIO (Ilhami & Sokibi, 2019).



**Gambar 2. 1** Node MCU ESP 8266

##### B. Sensor *Waterflow*

Sensor *waterflow* terdiri dari roda berputar yang dipasang didalam saluran air dan berfungsi untuk mendeteksi aliran air yang melewatinya. Sensor ini sering digunakan untuk mengukur aliran air di banyak sistem, termasuk sistem aliran air limbah, pendinginan, dan irigasi. Sensor *waterflow* ini mengukur laju aliran air yang masuk atau keluar dari aerator di akuarium. Ini membantu memantau seberapa baik aerator memenuhi kebutuhan oksigenasi dan sirkulasi air (Julius H et al., 2023).



**Gambar 2. 2** Sensor *Waterflow*

### C. Modul *Relay*

komponen saklar elektronik yang disebut *Relay* dapat digerakkan oleh arus listrik (Arianto et al., 2024). Ini biasanya digunakan untuk menggerakkan tegangan yang besar dengan memakai tegangan yang kecil (Sulistiyorini et al., 2022). *Relay* ini digunakan untuk mengaktifkan dan menonaktifkan aerator berdasarkan data yang terbaca dari sensor *waterflow* dikirimkan melalui mikrokontroler ESP8266.



**Gambar 2. 3** Modul *Relay*

### D. Aktuator Aerator

Aerator membantu menyalurkan oksigen ke dalam akuarium, karena biota air seperti ikan hias membutuhkannya untuk dapat bertahan hidup. Aerator dalam akuarium terdiri dari motor yang menghasilkan gelembung udara, yang kemudian disemprotkan kedalam air. Aerator ini juga berfungsi untuk mengurangi jumlah karbondioksida yang mengendap dalam air (Widodo et al., 2020).



**Gambar 2. 4** Aerator Akuarium

#### E. Akuarium

Tempat yang sering digunakan untuk memelihara ikan hias adalah Akuarium. Akuarium tidak hanya berfungsi sebagai tempat hidup dan berkembang biak ikan, seiring perkembangan zaman, akuarium juga berfungsi sebagai interior memperindah ruangan (Basri & Surbakti, 2023). Akuarium juga menjadi hal yang sangat diperhatikan bagi kalangan pecinta ikan hias (Masykuroh et al., 2023).



**Gambar 2. 5** Akuarium Ikan Hias

#### F. Pompa Sirkulasi Air

Pompa sirkulasi air merupakan alat yang berfungsi mengalirkan dan mengedarkan air dan juga berfungsi menciptakan arus dalam air sehingga memberikan oksigen pada akuarium (Amatullah et al., 2024).



**Gambar 2. 6** Pompa Sirkulasi Air

### 2.4. Prosedur Penelitian

Adapun prosedur penelitian yang digunakan pada penelitian ini adalah sebagai berikut:

#### A. Studi Literatur

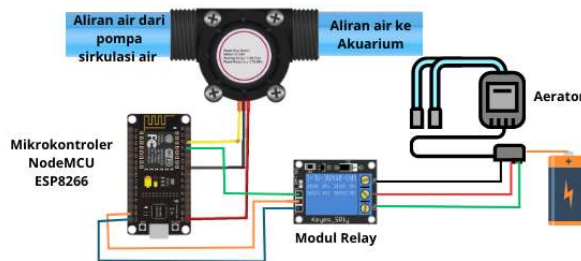
Proses mencari referensi dan pemahaman dari berbagai sumber seperti artikel – artikel, jurnal ilmiah, majalah dan buku sebelumnya terkait dengan penelitian yang akan dilakukan. Studi literatur digunakan sebagai menambah pengetahuan dan informasi yang dapat membantu proses dari penelitian “Kontrol Aerator Otomatis pada Akuarium Ikan Hias Menggunakan Mikrokontroler ESP8266”.

#### B. Analisis Kebutuhan Sistem

Proses analisis kebutuhan sistem digunakan untuk menentukan kebutuhan sistem kontrol aerator otomatis untuk penelitian ini. Hal ini memastikan bahwa setiap kebutuhan yang digunakan sesuai dengan hasil yang akan dihasilkan.

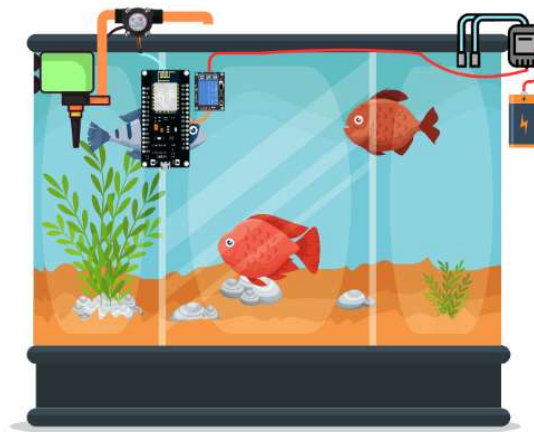
C. Perancangan dan Perakitan Perangkat Keras

Perancangan adalah proses penting yang harus dilakukan sebelum pembuatan suatu alat. Hal ini memungkinkan untuk mengurangi jumlah komponen yang tidak terpakai dan mengurangi biaya. Untuk menghindari kerusakan komponen, maka perlu untuk memahami karakteristiknya. Dibawah ini merupakan hasil dari perancangan rangkaian dimana pada gambar tersebut sensor *waterflow* akan membaca laju aliran air akuarium yang mengalir dari sumber air PLN melalui pompa air, lalu data dari sensor *waterflow* akan dikirim pada mikrokontroler ESP8266 selanjutnya diproses untuk menentukan apakah aerator perlu diaktifkan atau tidak dengan mengirimkan sinyal pada *relay*. Jika aerator perlu diaktifkan maka mikrokontroler akan menutup kontak *relay* untuk mengaktifkan aerator. Jika tidak, mikrokontroler membuka kontak *relay* untuk mematikan aerator. Berikut keseluruhan perancangan perangkat keras dapat dilihat pada gambar 2.7. Skematik



Gambar 2. 7 Skematik

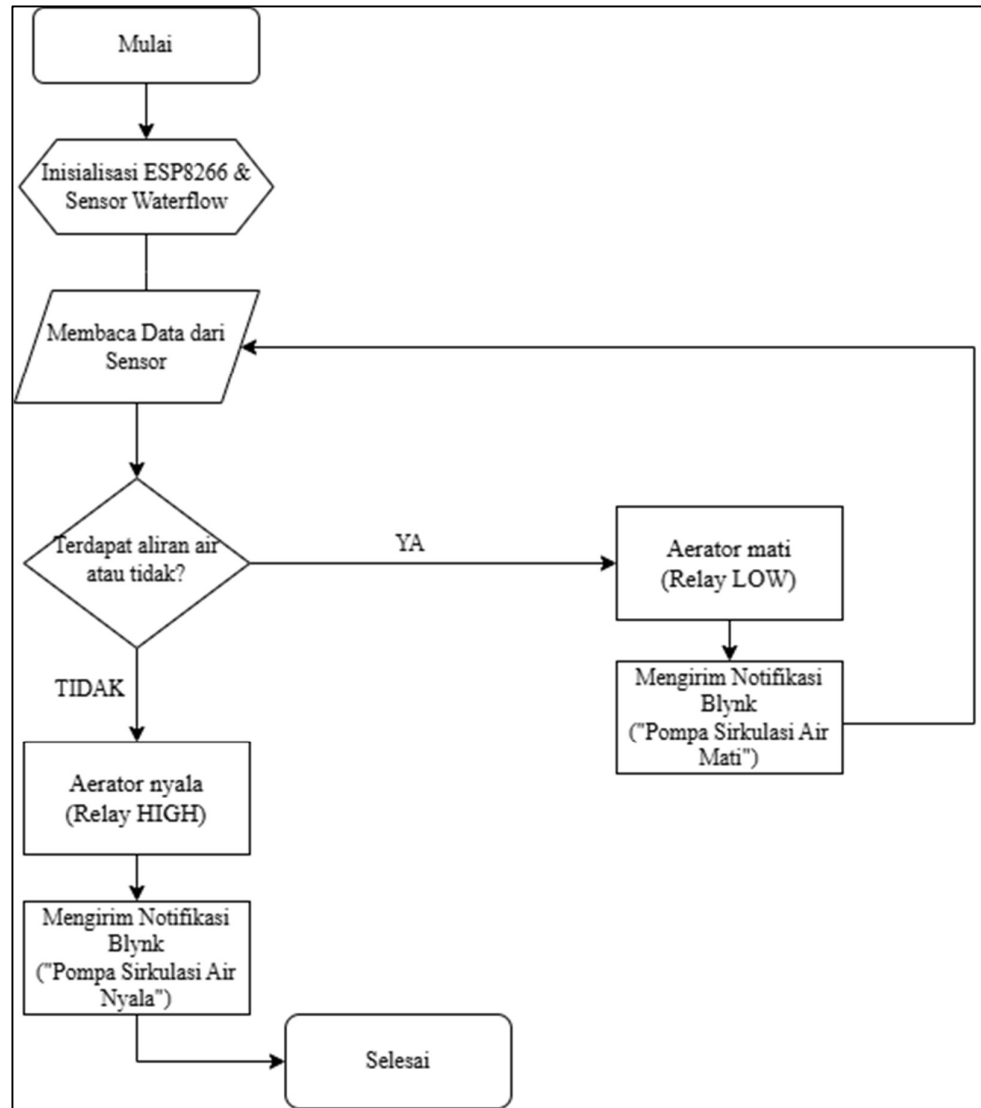
Pada Gambar 2.8 dibawah ini merupakan desain visual dari keseluruhan rancangan yang akan dibuat. Dimana terdapat akuarium sebagai wadah dari ikan hias dan juga sebagai tempat implementasi dari rangkaian skematik diatas. Terdapat juga pompa sirkulasi air yang membantu dalam memberikan suplai oksigen pada akuarium dimana sistem pompa sirkulasi air ini bergantung pada daya listrik (PLN) sebagai sumber daya.



Gambar 2. 8 Desain Visual

#### D. Perancangan Perangkat Lunak

Perancangan perangkat lunak dibuat dengan membuat gambar *flowchart* kebutuhan sistem untuk dijalankan pada perangkat keras. Pada gambar 2.9 dibawah merupakan *flowchart* alur kerja sistem dari program yang akan dibuat pada penelitian ini.



**Gambar 2.9** *FlowChart*

Dari flowchart pada gambar 2.9. Peneliti dapat mengetahui bagaimana gambaran kerja keseluruhan sistem sehingga dapat dibuat program sesuai dengan rangkaian kerja sistem yang sudah dibuat. Dengan begitu dibawah ini merupakan keseluruhan dari program yang digunakan dalam menjalankan rangkaian perangkat keras yang ditunjukkan pada gambar 2.10 sampai dengan 2.13 dibawah ini.

```

#define BLYNK_TEMPLATE_ID "TMPL61fCZA6Db"
#define BLYNK_TEMPLATE_NAME "Aerator"
#define BLYNK_AUTH_TOKEN "uKLHGnj9GYwtcIkMSzfSk8_v7uzMzG-7"

#include <Arduino.h>
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>

// Pin dan variabel untuk sensor aliran air dan relay
const int flowSensorPin = D3; // Pin untuk sensor aliran air
const int relayPin = D1;      // Pin untuk relay
volatile int pulseCount = 0;  // Jumlah pulsa yang diterima dari sensor
unsigned long lastTime = 0;
unsigned long currentTime;
float flowRate;
float flowRateML;
float calibrationFactor = 4.5; // Kalibrasi sensor aliran air (bisa disesuaikan)

```

**Gambar 2. 10** Program 1

```

// Informasi jaringan WiFi
char ssid[] = "AERATOR";
char pass[] = "11111111";

void IRAM_ATTR pulseCounter() {
  pulseCount++;
}

void setup() {
  Serial.begin(115200);
  pinMode(flowSensorPin, INPUT_PULLUP);
  pinMode(relayPin, OUTPUT);
  digitalWrite(relayPin, LOW); // Matikan pompa di awal

  Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);

  // Attach interrupt untuk sensor aliran air
  attachInterrupt(digitalPinToInterrupt(flowSensorPin), pulseCounter, FALLING);
}

```

**Gambar 2. 11** Program 2

```

void loop() {
  Blynk.run();

  currentTime = millis();
  // Hitung laju aliran setiap 1 detik
  if (currentTime - lastTime >= 1000) {
    detachInterrupt(digitalPinToInterrupt(flowSensorPin));

    // Hitung laju aliran air dalam Liter per menit
    flowRate = ((1000.0 / (currentTime - lastTime)) * pulseCount) / calibrationFactor;
    // Hitung laju aliran air dalam mL/detik
    flowRateML = (flowRate * 1000) / 60;

    // Cetak nilai currentTime - lastTime dan pulseCount ke serial monitor
    Serial.print("currentTime - lastTime: ");
    Serial.println(currentTime - lastTime);
    Serial.print("pulseCount: ");
    Serial.println(pulseCount);

    lastTime = currentTime;
    pulseCount = 0;

    Serial.print("Flow Rate: ");
    Serial.print(flowRate);
    Serial.println(" L/min");

    Serial.print("Flow Rate ML: ");
    Serial.print(flowRateML);
    Serial.println(" mL/detik");
  }
}

```

**Gambar 2. 12** Program 3

```

// Kirim data laju aliran ke Blynk Virtual Pin V1
Blynk.virtualWrite(V1, flowRate);

// Nyalakan/matikan pompa berdasarkan laju aliran air
if (flowRate > 1) {
  if (digitalRead(relayPin) == LOW) {
    Blynk.logEvent("waterflow_meter", "Terdeteksi aliran air pada sensor!. Pompa ON.");
  }
  digitalWrite(relayPin, HIGH); // Nyalakan pompa
} else {
  if (digitalRead(relayPin) == HIGH) {
    Blynk.logEvent("waterflow_meter", "Tidak Terdeteksi aliran air pada sensor! Pompa OFF.");
  }
  digitalWrite(relayPin, LOW); // Matikan pompa
}

attachInterrupt(digitalPinToInterrupt(flowSensorPin), pulseCounter, FALLING);
}

```

**Gambar 2. 13** Program 4

#### E. Pengujian Sistem

Setelah perancangan perangkat keras dan perangkat lunak sudah dilakukan maka selanjutnya adalah pengujian pada tiap-tiap rangkaian memastikan jika rangkaian sudah sesuai dan tidak ada error didalamnya.

#### F. Implementasi

Selanjutnya rancangan yang sudah dibuat akan di implementasi secara sungguhan. Proses implemetasi sistem ini merupakan tahapan akhir dari perancangan sistem yang sudah dilakukan sebelumnya. Pada tahap ini, seluruh komponen akan dipasang sesuai dengan sistem yang telah dibuat sebelumnya.

#### G. Evaluasi Sistem

Mengevaluasi sistem berjalan sesuai rancangan atau tidak, jika tidak maka akan dicari solusi untuk menanganinya.

#### H. Analisa Kerja

Untuk analisa kerja, dilakukan bersamaan dengan uji coba alat yang bertujuan dapat mengetahui kerja alat tersebut. Selain itu juga dilakukan analisa terhadap respon sensor *waterflow* dan juga aerator terhadap aliran air yang terdeteksi, juga respon pada aplikasi *Blynk* yang berbasis Mikrokontroler. Berdasarkan hasil pengujian dan analisis guna memastikan sistem yang telah dibuat sesuai dengan harapan.