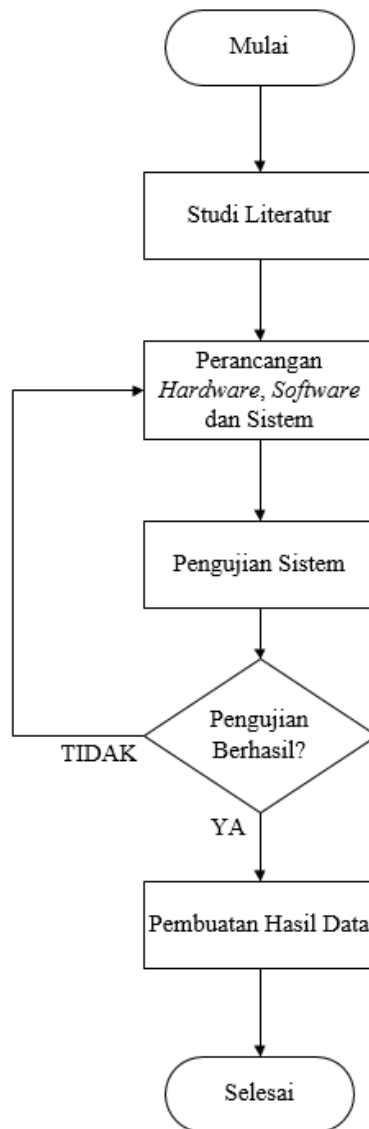


## BAB III METODE PENELITIAN

### 3.1 ALUR PENELITIAN

Dalam perancangan penelitian dibutuhkan tahapan alur penelitian yang dapat membantu proses perancangan berjalan sesuai dengan yang apa penulis susun secara teratur. Penelitian dilakukan dalam beberapa tahap yaitu tahap mempersiapkan alur penelitian, dalam alur penelitian dibentuk dalam bentuk *flowchart* yang dapat menjelaskan proses-proses yang dilakukan saat penelitian yang akan dibuat, seperti pada gambar 3.1.



Gambar 3.1 *Flowchart* Alur Penelitian

Sesuai dengan gambar 3.1 *flowchart* alur penelitian dimulai dari pencarian studi literatur, dalam proses ini bertujuan untuk membandingkan kajian-kajian teori dari penelitian perancangan yang sebelumnya, selain itu juga studi literatur dapat dilakukan dengan membaca jurnal ilmiah, artikel dari sumber daring, dan juga buku-buku yang berkaitan dengan perancangan yang akan dibuat. Pada blok diagram perancangan *hardware*, *software* dan sistem merupakan tahap proses pengumpulan alat dan bahan yang digunakan terdiri dari mikrokontroler dan juga sensor *turbidity* yang akan digunakan dalam penelitian. Setelah perancangan *hardware* proses selanjutnya yaitu perancangan *software*, proses perancangan *software* merupakan proses pembuatan Blynk yang digunakan untuk memantau atau *memonitoring* kekeruhan air irigasi secara *realtime*. Dalam *memonitoring* menggunakan aplikasi Blynk yang sudah tersedia di *marketplace Play Store* pada *smartphone* OS Android. Nilai kekeruhan air dapat ditampilkan di aplikasi Blynk secara *realtime* dikarenakan aplikasi Blynk menggunakan *cloud* yang biasa dikenal *Blynk cloud*. Setelah perancangan *hardware* dan *software* tahap selanjutnya yaitu pengujian sistem, jika pada pengujian tersebut terdapat kendala dan hasil tidak sesuai dengan yang diinginkan maka dilakukan kembali perancangan *hardware* dan *software* kembali hingga pengujian tersebut berhasil dan hasil data yang diinginkan sesuai maka melanjutkan ke tahap selanjutnya pembuatan hasil data.

### 3.2 PERANCANGAN HARDWARE

Dalam perancangan rancang bangun sistem *monitoring* kekeruhan air irigasi sawah berbasis *Internet of Things* (IoT) ada beberapa komponen dalam proses perancangan *hardware*. Alat dan bahan yang diperlukan dalam perancangan dapat dilihat dalam tabel 3.1.

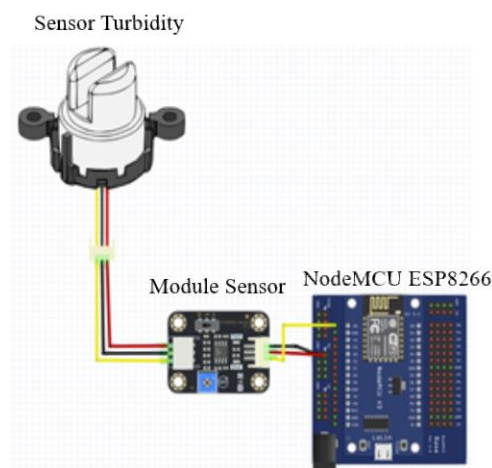
Tabel 3.1 Alat dan Bahan

No	Alat dan Bahan	Jumlah (Buah)
1	NodeMCU ESP8266	1
2	<i>Baseboard</i> NodeMCU	1
3	<i>Relay</i>	1
4	Pompa air DC	1

No	Alat dan Bahan	Jumlah (Buah)
5	Sensor <i>Turbidity</i>	1
6	Motor Servo	1
7	Adaptor 2A 5V	1
8	Step Down LM2596	1
9	Kabel Mikro USB	1
10	Kabel <i>Jumper</i>	Secukupnya
11	Laptop	1
12	<i>Smartphone</i> Android	1

### 3.2.1 Perancangan Sensor *Turbidity* dengan NodeMCU ESP8266

Penulis menggunakan sensor *turbidity* dengan *module* TSW-30. Perangkat ini menggunakan cahaya untuk mendeteksi partikel yang tertahan didalam air dengan cara mengukur transmisi cahaya dan tingkat penghamburan cahaya yang berubah sesuai dengan jumlah *Total Suspended Solids*. Rangkaian sensor *turbidity* TSW-30 dengan NodeMCU ESP8266 dapat dilihat pada gambar 3.2 dan konfigurasi pin dapat dilihat pada tabel 3.2.



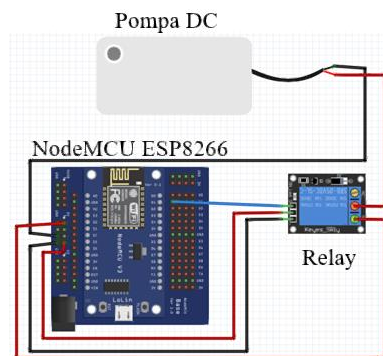
Gambar 3.2 Rangkaian Sensor *Turbidity* dengan NodeMCU ESP8266

Tabel 3.2 Konfigurasi Pin Sensor *Turbidity* dengan NodeMCU ESP8266

<i>Turbidity</i> TSW-30	NodeMCU ESP8266
VCC	5V
GND	GND
OUT	A0

### 3.2.2 Perancangan *Module Relay* dengan NodeMCU ESP8266

Dalam perancangan *module relay*, penulis menggunakan *module relay* 1 *channel* untuk mengaktifkan dan mematikan pompa air DC. Rangkaian *module relay* dengan NodeMCU ESP8266 dapat dilihat pada gambar 3.3 dan konfigurasi pin dapat dilihat pada tabel 3.3.



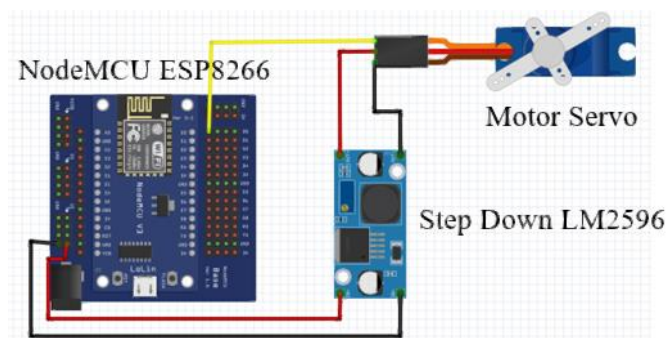
Gambar 3.3 Rangkaian *Module Relay* dengan NodeMCU ESP8266

Tabel 3.3 Konfigurasi Pin *Module Relay* dengan NodeMCU ESP8266

<i>Module Relay</i>	NodeMCU ESP8266
Pin 1	D1
Pin 2	5V
Pin 3	GND

### 3.2.3 Perancangan Motor Servo dengan NodeMCU ESP8266

Dalam perancangan motor servo, penulis menggunakan motor servo untuk membuka dan menutup suatu objek. Rangkaian motor servo dengan NodeMCU ESP8266 dapat dilihat pada gambar 3.4 dan konfigurasi pin dapat dilihat pada tabel 3.4.



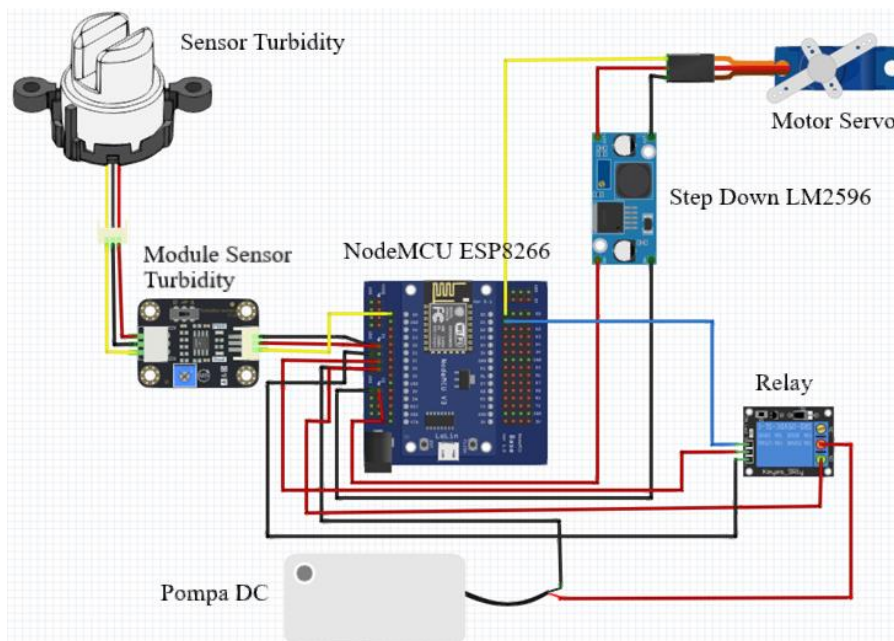
Gambar 3.4 Rangkaian Motor Servo dengan NodeMCU ESP8266

Tabel 3.4 Konfigurasi Pin Motor Servo dengan NodeMCU ESP8266

Motor Servo	LM2596 <i>Out</i>	LM2596 <i>In</i>	NodeMCU ESP 8266
<i>Signal</i>	-	-	D0
5V	<i>Out +</i>	<i>In +</i>	V1
GND	<i>Out -</i>	<i>In -</i>	GND

### 3.2.4 Skema Rangkaian Elektronika

Gambaran sebuah rangkaian elektronika yang telah penulis buat, perangkat input dan output dirangkai menjadi satu rangkaian dapat dilihat pada gambar 3.5.



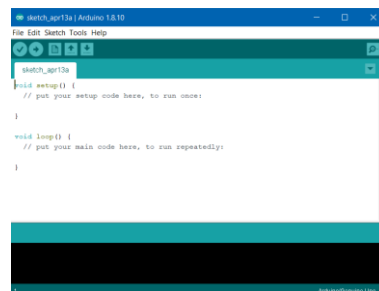
Gambar 3.5 Skema Rangkaian Elektronika

## 3.3 PERANCANGAN SOFTWARE

### 3.3.1 Software Arduino IDE

Penggunaan aplikasi Arduino IDE pada perancangan *software* sebagai wadah penulis memberikan perintah-perintah pada komponen elektronik dan diunggah ke mikrokontroler NodeMCU ESP8266 untuk pengaplikasiannya agar dapat dioperasikan. Dalam penulisan kode program dilakukan untuk memberikan instruksi perintah yang ingin diberikan dengan menggunakan bahasa pemrograman C. Penulis juga membutuhkan beberapa *library* pendukung, *library* pendukung dibutuhkan agar semuanya dapat saling berkomunikasi diantaranya *library* dari

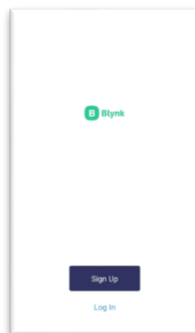
*Blynk* yang berfungsi untuk penghubung antara perangkat elektronik dengan *cloud* IoT yang telah disediakan oleh *Blynk*. *Library* motor servo yang berfungsi untuk menghubungkan motor servo dengan NodeMCU ESP8266, *library* sensor *turbidity* yang berfungsi untuk menghubungkan sensor *turbidity* dengan NodeMCU ESP8266, setelah *library* sensor *turbidity* adapun *library* ArduinoOTA yang berfungsi untuk jembatan penghubung antara *module* NodeMCU ESP8266 dengan Arduino IDE secara *wireless* sehingga pada *module* NodeMCU ESP8266 dapat diprogram dan diperbarui melalui jaringan WiFi. Tampilan *software* Arduino IDE dapat dilihat pada gambar 3.6.



Gambar 3.6 Tampilan Awal Arduino IDE

### 3.3.2 Software Blynk

Perancangan antarmuka Blynk dapat mendukung suatu proyek IoT dengan banyak *widget* yang sudah disediakan guna mendukung proyek IoT. Blynk diciptakan dengan salah satu tujuannya yakni *monitoring* kekeruhan air irigasi secara jarak jauh (*remote*) menggunakan komunikasi data internet ataupun intranet. Pada aplikasi Blynk menampilkan data parameter kekeruhan air secara *realtime*, sehingga dapat mengetahui kualitas air kapanpun. Pada gambar 3.7 dapat dilihat merupakan tampilan awal aplikasi Blynk.



Gambar 3.7 Tampilan Awal Aplikasi Blynk

### 3.4 PERANCANGAN SISTEM

#### 3.4.1 Konsep Dasar Sistem

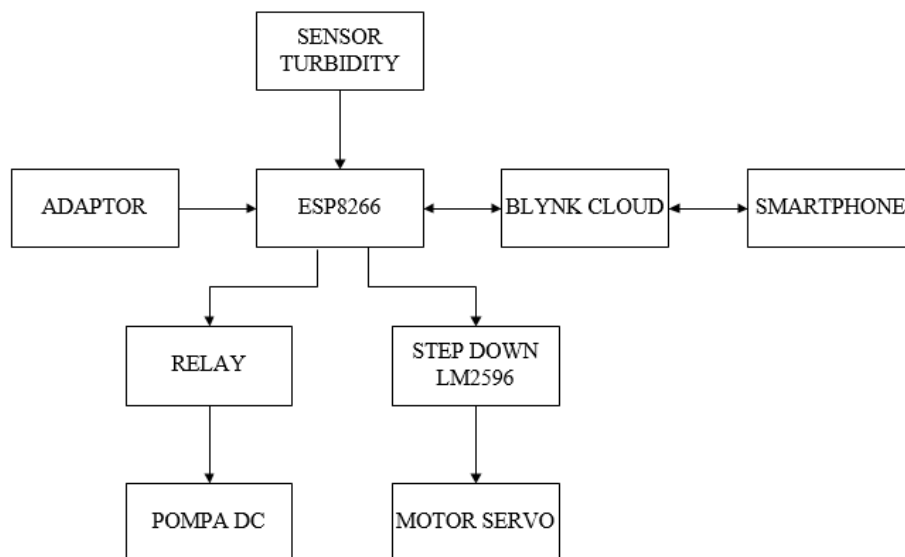
Perancangan sistem dirancang untuk melakukan *monitoring* kekeruhan air irigasi melalui *smartphone*, *output* yang didapatkan berupa kualitas air tersebut tergolong bersih, keruh dan kotor. Sensor *turbidity* telah diprogram pada NodeMCU ESP8266 untuk mendeteksi kekeruhan air pada saluran irigasi, apabila air dalam kondisi kotor maka motor servo akan bekerja untuk membuka pintu pembuangan air. Hasil data diolah kemudian dikirimkan ke *Blynk cloud*, setelah dikirimkan ke *Blynk cloud* data ditampilkan pada *smartphone*. Adapun gambaran alur konsep dasar sistem dapat dilihat pada gambar 3.8.



Gambar 3.8 Alur Konsep Dasar Sistem

#### 3.4.2 Diagram Blok Sistem

Diagram blok menggambarkan dasar perancangan sistem, komponen-komponen pada diagram blok memiliki fungsi kerjanya masing-masing.



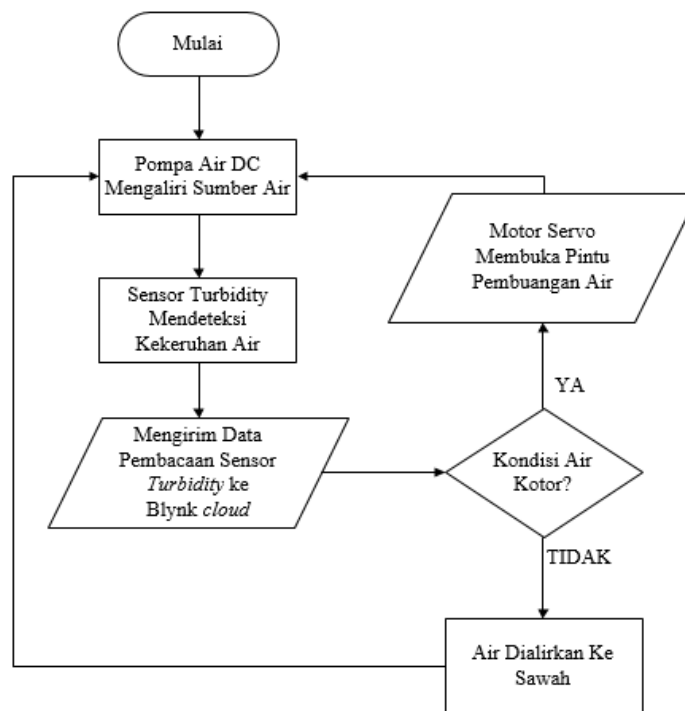
Gambar 3.9 Diagram Blok Sistem

Pada gambar 3.9 dapat dijelaskan proses dari perancangan sistem yang dibuat. Adaptor yang akan digunakan menggunakan daya sebesar 5 Volt dan 12 Ampere. Sensor *turbidity* sebagai perangkat *input* yang berfungsi untuk mendeteksi

kekeruhan air irigasi. ESP8266 berfungsi sebagai mikrokontroler yang mengolah *input*, *output* dan mengatur koneksi internet, setelah data diolah diteruskan ke perangkat *output relay* dan *Blynk cloud*. *Relay* yang digunakan yaitu jenis 1 *channel* yang berfungsi untuk saklar pompa air DC, jika air sudah tertampung dalam suatu tempat irigasi maka sensor *turbidity* dapat dijalankan untuk mendeteksi kekeruhan air irigasi dan dapat dipantau melalui *smartphone*, sensor *turbidity* akan membaca tingkat kekeruhan air apabila air dalam kondisi kotor maka motor servo akan bekerja namun jika air dalam kondisi bersih maka motor servo tidak akan bekerja. Data dikirim dari ESP 8266 ke *Blynk cloud* tersebut akan disimpan secara *realtime*.

### 3.4.3 Flowchart Sistem

Alur jalannya sistem dari awal mulai proses hingga ke proses selanjutnya, dapat dilihat pada gambar 3.10 *flowchart* sistem.



Gambar 3.10 *Flowchart* Sistem

Proses pada gambar flowchart 3.10 merupakan tahapan dalam sebuah sistem perancangan penulis. Pompa air DC sebagai sumber pengaliran air ke dalam irigasi, selanjutnya sensor *turbidity* akan mendeteksi kekeruhan air pada irigasi, hasil pendeteksian tersebut dikirimkan ke *Blynk cloud*, *Blynk cloud* akan



mengirimkan pemberitahuan melalui aplikasi pada *smartphone*. Pada aplikasi *Blynk smartphone* akan ditampilkan kualitas air, apabila kondisi air tersebut kotor maka dapat mengaktifkan motor servo agar air dalam irigasi tersebut dapat dialirkan ke saluran pembuangan air kotor.

### **3.5 PENGUJIAN SISTEM**

#### **3.5.1 Pengujian Sistem Sensor *Turbidity***

Pengujian sensor *turbidity* dilakukan dengan melihat tingkat kekeruhan air yang telah dideteksi dan ditampilkan pada aplikasi *Blynk*. Pengujian dilakukan 3 sampel air yang berbeda dengan pengujian 5 kali untuk *memonitoring* kualitas air dari 3 sampel air tersebut.

#### **3.5.2 Pengujian Sistem Pompa DC, Motor Servo dan Blynk**

Pengujian pompa DC dilakukan dengan menekan tombol pada aplikasi *Blynk*. Pengujian pompa DC dilakukan sebanyak 8 kali sebagai sumber air yang akan masuk ke irigasi. Pengujian motor servo dilakukan melalui aplikasi *Blynk*, pengujian motor servo dilakukan 8 kali dengan tingkat kekeruhan yang berbeda untuk memastikan dengan tingkat kekeruhan yang berbeda dapatkah pintu pembuangan dapat terbuka atau tidak.

#### **3.5.3 Pengujian Parameter QoS**

Pengujian parameter QoS dilakukan untuk mengetahui waktu *delay* dalam mengirim sebuah paket dari penerima ke pengirim, pengujian *throughput* dilakukan untuk mengetahui berapa besar paket data yang telah berhasil dikirimkan pada jaringan dan pengujian *packet loss* dilakukan untuk mengetahui paket yang hilang selama pengiriman data.