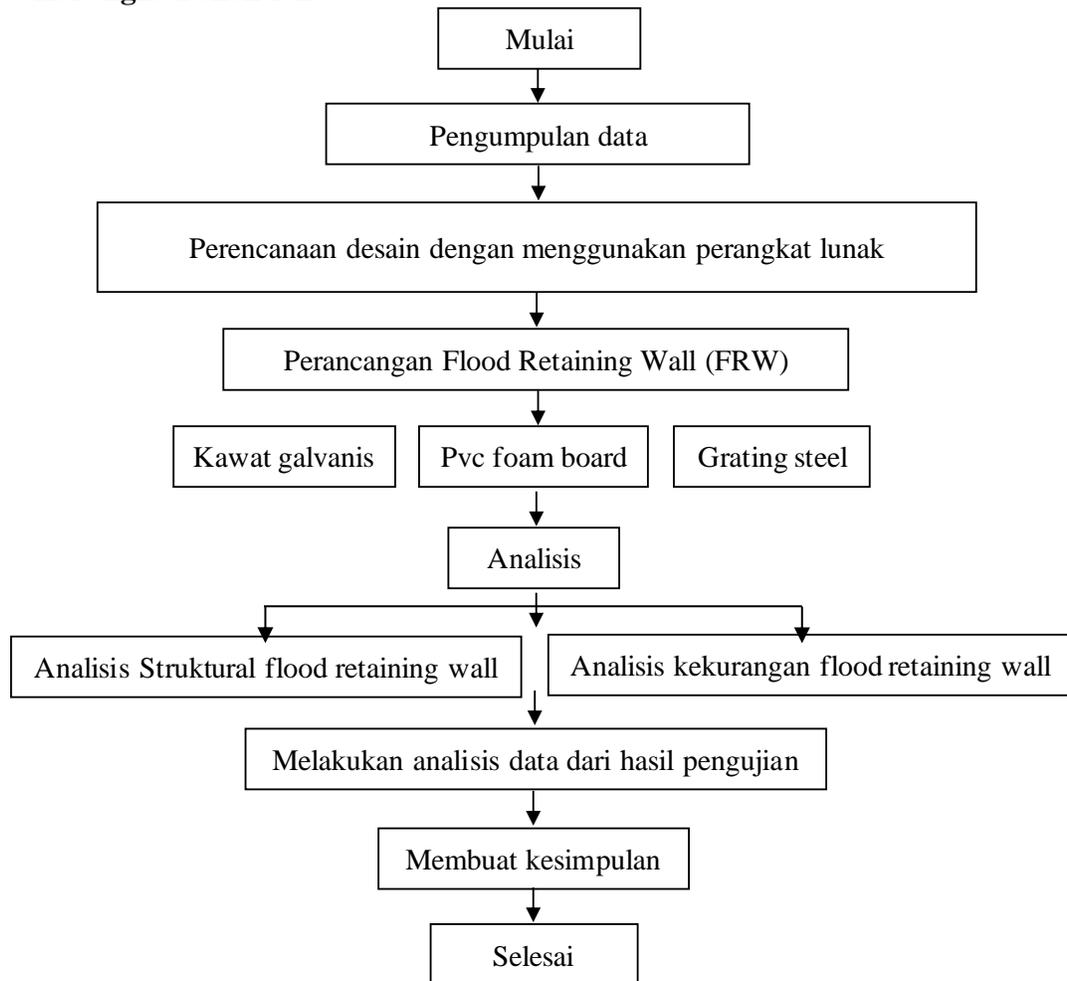


BAB III

METODE PENELITIAN

Metode penelitian yang digunakan dalam penelitian ini adalah metode analisis perancangan. Analisis untuk mendapatkan gambaran umum mengenai kejadian banjir, dan sistem yang akan di jalankan berdasarkan permasalahan yang ada. Perancangan yang dilakukan adalah dengan mendesain sistem dan menemukan bahan yang cocok untuk di aplikasikan pada sistem agar bisa mencapai tujuan.

3.1 Kerangka Penelitian



Gambar 3. 1 Flow Chart

3.2 Pengumpulan Data

Tahap awal adalah pendataan yang cukup mengenai intensitas banjir tahun-tahun sebelumnya dari Badan Penanggulangan Bencana Daerah (BPBD). Dan juga penting untuk mengumpulkan informasi yang diperlukan mengenai setiap outlet dalam bentuk aliran alami atau sungai & data curah hujan yang akurat yang akan memberikan bantuan besar dalam desain penghalang.

3.3 Bahan

Pada penelitian kali ini penulis akan menggunakan bahan *PVC Foam Board* yang merupakan busa polistiren yang diekstrusi yang terbungkus di antara lapisan melamin dan veneer serat kayu. *PVC Foam Board* ini memiliki permukaan yang sangat kaku yang tahan terhadap kelembaban dan tidak akan mudah pecah atau melengkung. *Veneer* serat kayu dilapisi dengan resin yang mencegah kerusakan, noda dan air. *PVC Foam Board* akan mempertahankan kekuatannya terlepas dari lamanya waktu. *PVC Foam Board* adalah pilihan yang sangat baik untuk menggantikan kayu lapis dalam aplikasi. Ini juga merupakan alternatif ekonomis untuk inti lain yang digunakan dalam produksi struktur komposit *sandwich* karena menawarkan kekuatan yang sangat baik dan stabilitas dimensi pada suhu tinggi. *PVC Foam Board* ini dapat mempercepat proses manufaktur, menghemat biaya tenaga kerja, dan hampir menghilangkan limbah material.

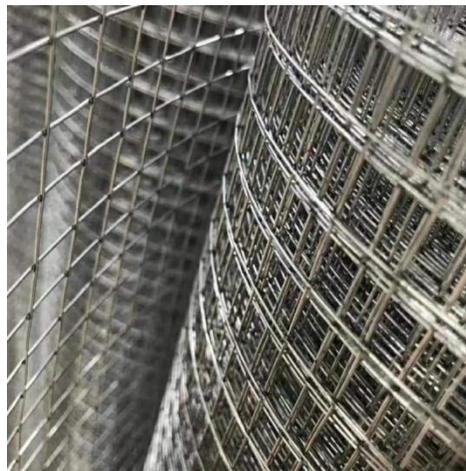
Bahan selanjutnya yang di perlukan adalah grill besi. Grill besi adalah sejenis baja yang biasanya di pasang di atas selokan atau drainase jalan untuk menyaring partikel besar tidak ikut masuk ke dalam. grill besi ini dapat berbentuk bujur sangkar atau silang yang saling menyatu pada antar bagian dengan melalui proses pengelasan (Indah dkk, 2020). Grill besi ini di pasang di atas sistem FRW dan pintu pertama yang akan di lewati air banjir sebelum masuk ke dalam wilayah sistem FRW.



Gambar 3. 2 Grill besi/ Grating Steel

(Sumber : tokopedia.com)

Bahan terakhir yang dipakai dalam penelitian ini adalah kawat galvanis dengan lubang sebesar 12 mm x 12 mm dan tebal kawat 1 mm. Kawat ini berfungsi untuk menyaring kembali puing puing yang bisa jadi terbawa arus dari banjir sebelum air memasuki bak tertutup yang akan mengangkat dinding penghalang.



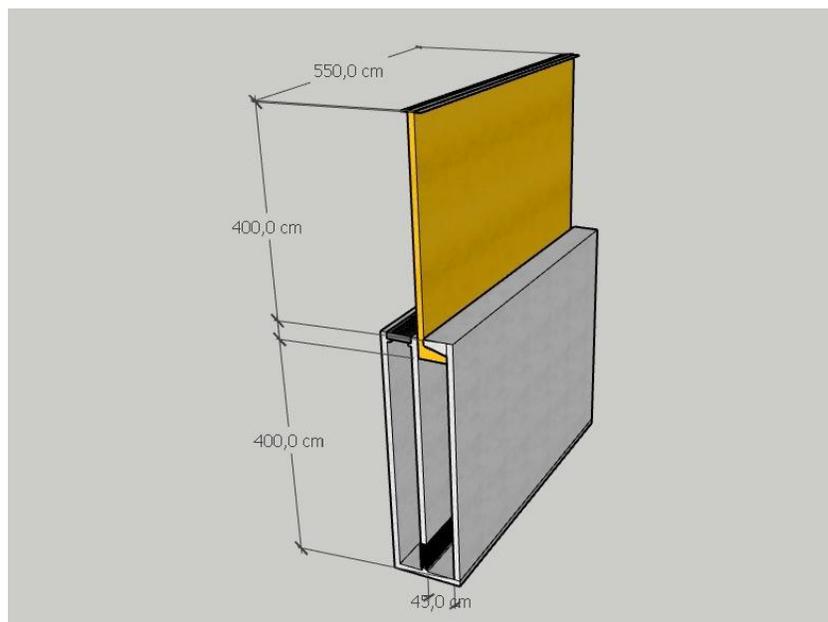
Gambar 3. 3 Kawat Galvanis

(Sumber : pinterest.com)

3.4 Perancangan

Desain *Flood Retaining Wall* (FRW) dilakukan dengan menggunakan perangkat lunak SketchUp, Blynk dan Arduino IDE. Desainnya menggunakan air banjir yang mendekat untuk secara otomatis menaikkan penghalang. Ukuran yang kami coba adalah dengan panjang hingga 5 m dan tinggi 4 m. Beberapa unit dapat dihubungkan bersama untuk membuat jangka panjang jika diperlukan, dengan tiang perantara permanen atau yang dapat dilepas.

3.4.1 Rencana Model

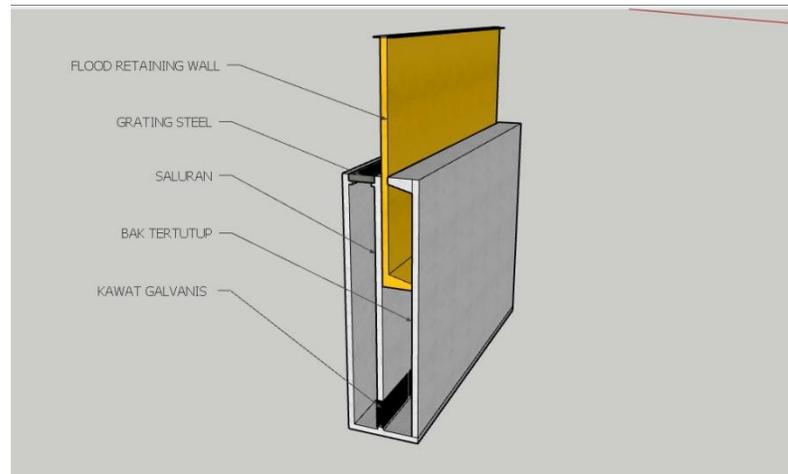


Gambar 3. 4 Ukuran model *Flood Retaining Wall* (FRW)

(Sumber : Hasil Analisis)

FRW terdiri dari dinding penghalang sebagai penahan banjir, dengan ketinggian 4 m & lebar 0,4 m yang tertanam di antara beton penghalang setinggi 4 m yang berfungsi sebagai bak tertutup selama operasinya. Seluruh struktur bertumpu pada pondasi atau dasar beton. Panjang pembatas yang diberikan adalah 5 m dan dapat diperpanjang sesuai kebutuhan. Sebuah grill besi dipasang diatas permukaan tanah, yang kemudian

dipasang juga kawat galvanis yang akan menyaring air sebelum masuk ke bak tertutup.



Gambar 3. 5 Bagian-bagian Dalam *Flood Retaining Wall* (FRW)

(Sumber : Hasil Analisis)

Berikut ini adalah bagian-bagian utama dari *Flood Retaining Wall* (FRW):

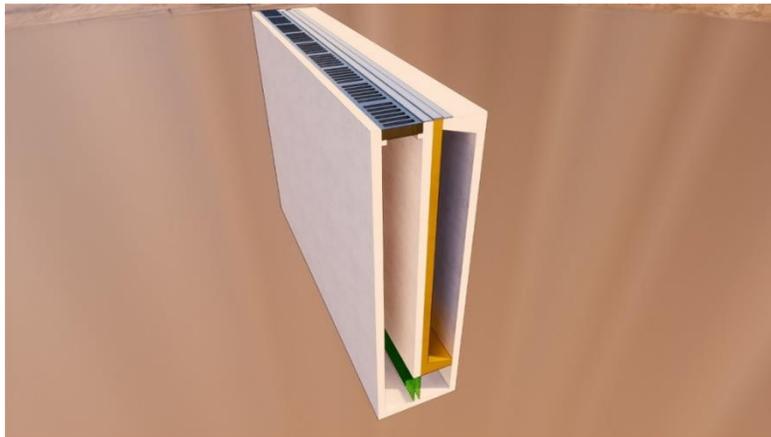
- Grill besi : Ini adalah komponen FRW yang melaluinya air akan di saring terlebih dahulu dari puing puing atau sampah yang terbawa banjir sebelum masuk ke bak tertutup.
- Kawat galvanis : Menyaring kembali air banjir untuk memastikan bahwa tidak ada partikel besar yang masuk ke dalam bak tertutup.
- Bak tertutup : Bak ini merupakan tempat air banjir bekerja untuk mengangkat atau membuat dinding penghalang ini mengapung.
- Dinding penghalang : Ini adalah komponen terpenting dari FRW. Dinding ini akan muncul sebagai perisai pelindung selama waktu banjir dan terkubur di tanah pada waktu air banjir surut.

3.4.2 Prinsip operasi

Dalam kondisi tidak banjir, semua bagian FRW tidak terlihat atau tersembunyi di dalam tanah. Ketika air banjir masuk kedalam saluran, tekanan di dalam bak tertutup akan sama dengan tekanan hidrostatik di saluran, FRW akan mengapung dan naik. Sehingga air banjir bisa terus naik tanpa membanjiri kawasan yang dilindungi FRW. Saat permukaan air surut kembali ke tingkat normal, air banjir di dalam bak tertutup dikeringkan dengan pompa. Saat air sudah tidak ada lagi di dalam bak tertutup, maka FRW akan kembali ke posisi semula di dalam bak. Dalam posisi istirahat tertutup, tutup penghalang disegel untuk mencegah masuknya limbah banjir atau puing-puing.

Langkah kerja FRW:

a) Kondisi normal



Gambar 3. 6 Flood Retaining Wall (FRW) Dalam Kondisi Normal

Sumber : Hasil Analisis

Setelah penghalang banjir dipasang, dinding penghalang tidak terlihat atau berada di bawah kondisi permukaan air normal. Dinding diperkuat untuk kekuatan benturan. Di atas, tutup baja mengunci di ruang bak tertutup dalam kondisi normal (tidak banjir).

b) Saat Pengaktifan



Gambar 3. 7 Proses Pengaktifan *Flood Retaining Wall (FRW)*

Sumber : Hasil Analisis

Setelah air naik kira-kira 10 cm di bawah permukaan banjir, cekungan penghalang terisi melalui saluran pengisi di dalam lubang. Dinding naik dan mengapung. Segera setelah bak tertutup terisi penuh, permukaan penutup akan "mengunci" penghalang ke posisi kedap air.

c) Bagian yang terangkat



Gambar 3. 8 *Flood Retaining Wall (FRW)* dalam kondisi banjir atau bekerja

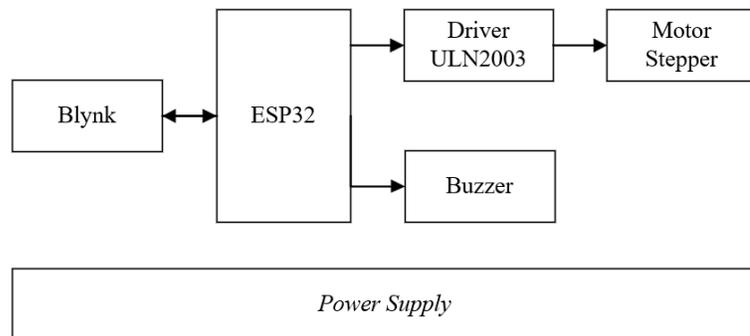
Sumber : Hasil Analisis

Air bisa naik lebih tinggi tanpa membanjiri kawasan yang di lindungi. Setelah ketinggian air mereda ke tingkat normal, saluran dikeringkan melalui pembuangan dengan katup satu arah atau dengan pompa. Setelah air meninggalkan bak, dinding kembali ke posisi istirahatnya di dalam bak tertutup. Tutup di atas dinding kemudian ditutup untuk mencegah masuknya limbah atau puing-puing.

3.5 Pengoperasian Berbasis IOT

3.5.1 Diagram Blok

Rancangan diagram blok difungsikan sebagai pedoman dasar mengenai cara kerja sistem *Flood Retaining Wall* (FRW). Setiap blok fungsi terdiri dari bagian input (masukan), pengolah dan pemroses data, dan output (keluaran). Rancangan diagram blok ditunjukkan oleh gambar 3.9 di bawah ini.



Gambar 3. 9 Diagram Blok Sistem

Sumber: Peneliti (2023)

Adapun penjelasan sistematika penjelasan setiap fungsi diagram blok yang telah dirancang, yaitu sebagai berikut:

1. Rangkaian *Power Supply* atau catu daya digunakan untuk memberi suplai daya atau menyalakan rangkaian sistem untuk bisa bekerja pada kondisi yang dirancang.

2. Pada sisi input terdapat I/O dari platform Blynk, yang berfungsi menghubungkan perangkat mikrokontroler dengan internet untuk bisa mengendalikan *Flood Retaining Wall* di mana saja dan kapan saja.
3. Unit pemroses atau mikrokontroler ESP32 berperan untuk menerima dan mengeksekusi perintah sistem yang dikirim platform IoT Blynk.
4. Pada output terdapat aktuator motor stepper yang berfungsi untuk melakukan pengendalian daripada naik dan turunnya *Flood Retaining Wall*. Pengendalian tersebut berdasarkan perintah yang dikirim oleh mikrokontroler ESP32 yang kemudian diterima oleh modul driver ULN2003.

3.5.2 Teknik Fabrikasi

Fabrikasi sistem IoT *Flood Retaining Wall* terbagi menjadi 3 proses, yaitu fabrikasi perancangan mekanik, perancangan sistem elektrik, dan perancangan sistem perangkat lunak (*software*). Berikut disajikan keseluruhan alat dan bahan yang diperlukan dalam bentuk tabel 3.1 di bawah ini.

Tabel 3. 1 Daftar Rincian Alat dan Bahan Rancang Bangun

No	Nama Bahan	Jumlah
A. Alat		
1.	Solder dan timah	1 pcs
2.	<i>Cutter</i> atau gunting	1 pcs
3.	Obeng set	1 pcs
4.	Kunci L set	1 pcs
5.	Lem tembak	1 pcs
6.	Tang cucut	1 pcs
7.	Tang potong	1 pcs
8.	Gergaji besi	1 pcs
9.	Lem	3 pcs
B. Bahan		
1.	As besi drat ulir \varnothing 6 mm	1 m
2.	<i>Linear bearing</i> \varnothing 6 mm	2 pcs
3.	Kopel as \varnothing 6 mm to \varnothing 6 mm	1 pcs
4.	Baut, mur, ring M4	1 pack
5.	Mur M6	6 pcs
6.	Benang wol	1 pcs

7.	Klem kabel bulat Ø 12 mm	1 pack
8.	Refill isi lem tembak	4 pcs
9.	Lem G (korea)	1 pcs
10.	<i>Double foam tape</i>	1 pcs
11.	Catu daya 5 VDC 3A	1 pcs
12.	Motor <i>stepper</i> 28BYJ-48 + <i>Driver</i> ULN2003	1 pcs
13.	ESP32	1 pcs
14.	<i>Buzzer</i> aktif	1 pcs
15.	Papan <i>breadboard</i>	2 pcs
16.	Terminal blok 2 pin	2 pcs
17.	Jack DC 5,5 x 2,1 mm	1 pcs
18.	Kabel AWG 26 + <i>dupont header</i>	8 m
19.	<i>Heatshrink</i> bakar Ø 2 mm	1 m
20.	Adaptor charger USB 5 V 1 A	1 pcs
21.	Kabel micro-USB 1 m	1 pcs

3.5.2.1 Perancangan Mekanik

Perancangan mekanik meliputi tahapan pembangun sistem mekanikal hingga dapat berfungsi semestinya. Perancangan mekanik terdiri dari peralatan penunjang proses cara kerja *Flood Retaining Wall*. Berikut merupakan penjelasan daripada perancangan mekanik *Flood Retaining Wall*.

1. Pembuatan mekanisme gerak *Flood Retaining Wall*

Flood Retaining Wall diberi lubang untuk memasukkan benang wol dengan konfigurasi silang pada kedua ujungnya agar seimbang terhadap titik yang sama. Pemberian lubang menggunakan solder agar presisi keempat titiknya. Benang wol diberi sisa panjang sekitar 1 meter agar dapat dikaitkan dengan penggulung as besi.



Gambar 3. 10 Flood Retaining Wall Terpasang Benang Wol

Sumber: Peneliti (2023)

2. Pembuatan *shaft* atau as putar

Menyesuaikan panjang as besi drat ulir sesuai panjang bangunan gedung dan memotong menggunakan gergaji besi. Setelah besi drat dipotong, maka perlu melakukan penyetelan dan pemasangan terhadap kopel as yang berperan menghubungkan putaran daripada motor stepper.

Dalam pemasangan as besi diperlukan suatu bantalan agar menjaga posisi as tersebut. Putaran as besi harus ditopang menggunakan *linear bearing* agar as tetap dapat berputar dengan bebas dan minim gesekan.



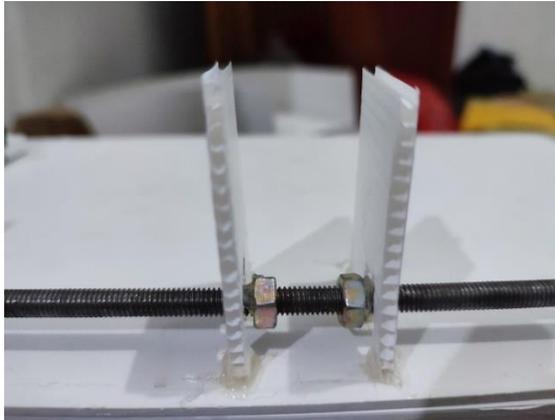
Gambar 3. 11 Pemasangan As Besi di Bangunan

Sumber: Peneliti (2023)

3. Pembuatan kerangka gerak as

Pergerakan as besi harus dijaga agar presisi secara berkelanjutan dan selalu sama

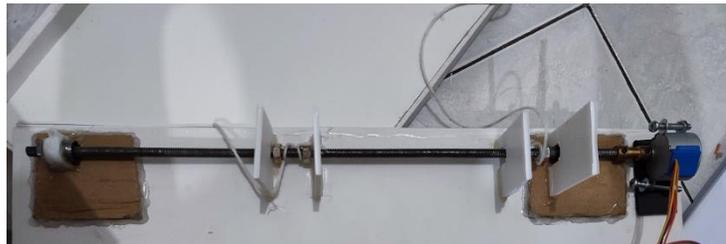
ketika menarik atau melepas gulungan benang wol. Titik penggulungan pada kedua sisi harus sama agar ketegangan benang antara kedua sisi seimbang. Pemberian pembatas pada titik penggulungan memungkinkan benang wol tidak keluar dari titik yang sudah ditentukan.



Gambar 3. 12 Pemasangan Pembatas Titik Penggulungan

Sumber: Peneliti (2023)

Pemasangan benang wol pada as besi harus diikat dengan kuat dan dilem agar posisi awal selalu terjaga tanpa terganggu oleh pergerakan putar as besi.



Gambar 3. 13 Pemasangan Tali *Flood Retaining Wall* Terhadap As

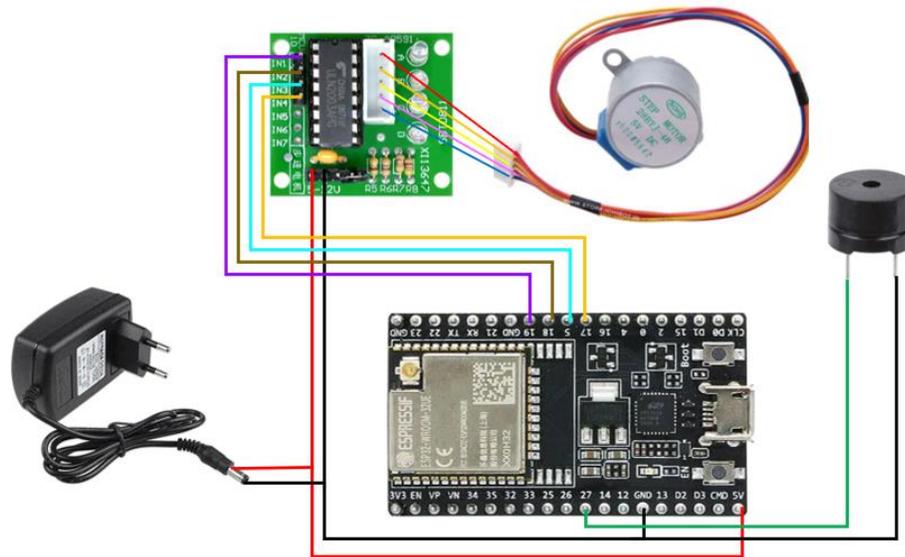
Sumber: Peneliti (2023)

3.5.2.2 Perancangan Elektrik

Perancangan sistem elektrikal merupakan sebuah proses yang melibatkan pemilihan komponen dan menghubungkan komponen-komponen tersebut. Komponen tersebut harus disesuaikan dengan kebutuhan fungsionalitas daripada *Flood Retaining Wall*.

Pada proses perakitan komponen elektrik, seluruh komponen dihubungkan menuju ke pusat kendali mikrokontroler sebagai sistem kontrol *Flood Retaining Wall*.

Pada perancangan ini terdapat tahapan merangkai sistem kontrol dan elektrik agar beberapa komponen agar saling terhubung. Rancangan ini bertujuan dalam memudahkan proses perakitan komponen menjadi satu sistem kerja. Diagram rangkaian elektronika ditunjukkan oleh gambar 3.14 di bawah ini.



Gambar 3. 14 Ilustrasi Integrasi Antar Komponen

Sumber: Peneliti (2023)

Adapun penjelasan konfigurasi pin mikrokontroler ESP32 ditunjukkan oleh tabel di bawah ini.

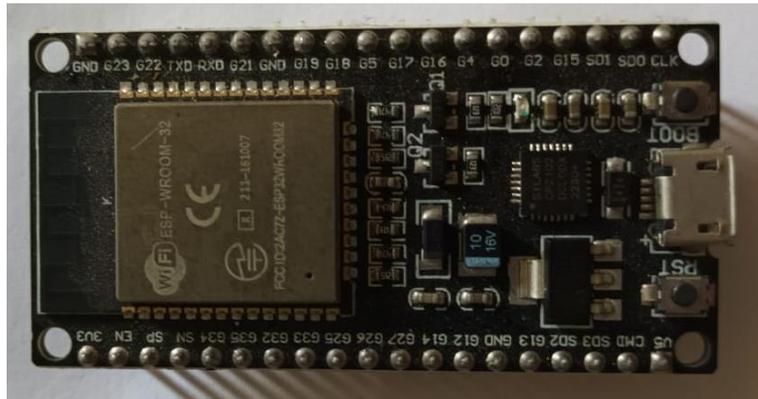
Tabel 3. 2 Konfigurasi Pin ESP32

IN1 ULN2003	Pin 19
IN2 ULN2003	Pin 18
IN3 ULN2003	Pin 5
IN4 ULN2003	Pin 17
Buzzer	Pin 27

3.5.2.2.1 Komponen Elektronika

Sistem elektrikal membutuhkan penyesuaian terhadap penggunaan dan fungsionalitas komponen elektronika. Komponen elektronika ini dipilih untuk menunjang sistem kerja *Flood Retaining Wall*. Berikut seluruh bahan komponen elektronika yang digunakan:

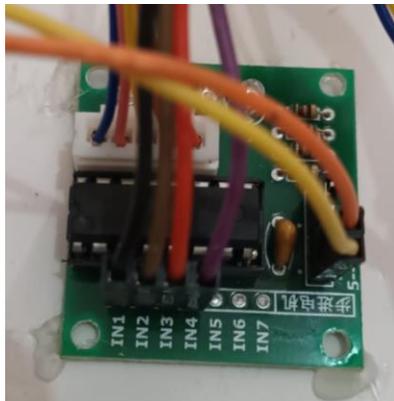
1. Mikrokontroler ESP32



Gambar 3. 15 Mikrokontroler ESP32

Sumber: Peneliti (2023)

2. *Driver* ULN2003



Gambar 3. 16 Driver motor stepper

Sumber: Peneliti (2023)

3. Motor *stepper* 28BYJ-48



Gambar 3. 17 Stepper 28BYJ-48

Sumber: Peneliti (2023)

4. *Buzzer* aktif



Gambar 3. 18 Buzzer Aktif

Sumber : Peneliti (2023)

5. Catu daya



Gambar 3. 19 Catu Daya Sistem

Sumber : Peneliti (2023)

6. Kabel AWG 26 atau kabel jumper



Gambar 3. 20 Diameter Kabel AWG 26

Sumber : Peneliti (2023)

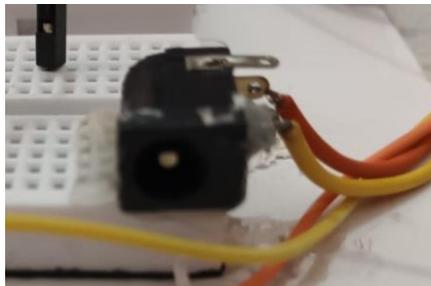
7. *Breadboard*



Gambar 3. 21 Papan Breadboard

Sumber : Peneliti (2023)

8. Jack DC *Female*



Gambar 3. 22 Jack DC Female Terpasang Pada Breadboard

Sumber : Peneliti (2023)

9. Terminal blok

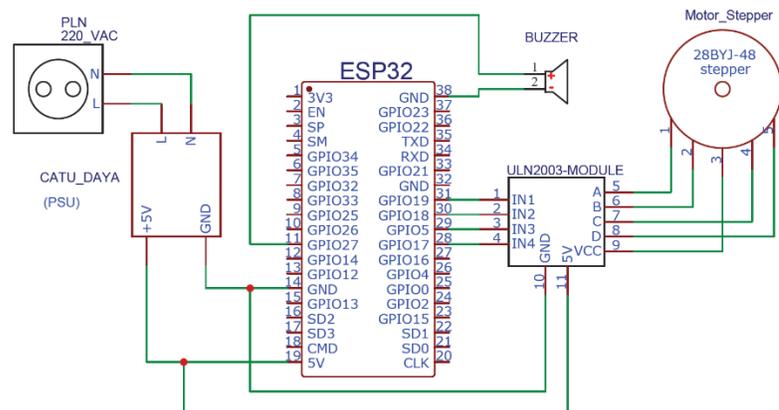


Gambar 3. 23 Terminal Blok Terpasang Pada Breadboard

Sumber : Peneliti (2023)

3.5.2.2.2 Integrasi Komponen Elektronika

Penghubung komponen pada sistem elektrikal merupakan proses menghubungkan seluruh komponen menjadi satu. Tujuan dari menghubungkan setiap komponen adalah untuk saling terhubung pada mikrokontroler ESP32 supaya dapat dikendalikan secara terpusat.



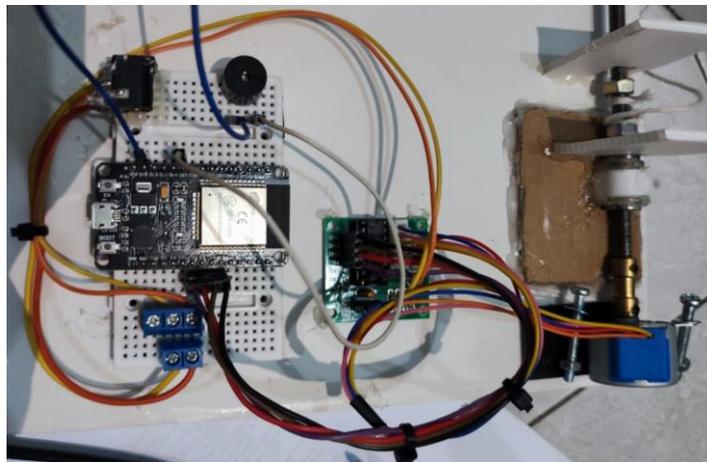
Gambar 3. 24 Diagram Sirkuit Rangkaian

Sumber : Peneliti (2023)

Penghubung pada setiap komponen memiliki alamat yang berbeda-beda. Pengalamatan pin memberi kemudahan dalam menghubungkan komponen dengan sistem kendali ESP32 dan program. Berikut merupakan pemberian alamat pada setiap komponen:

Tabel 3. 3 Konfigurasi Pin Komponen Elektronika Dengan ESP32

No	Komponen	Konfigurasi Pin	Pin ESP32
1.	Catu daya (PSU)	+ 5 V	Vin
		GND	GND
2.	Driver ULN2003	IN1	D19
		IN2	D18
		IN3	D5
		IN4	D17
		VCC	+ 5 V PSU
		GND	GND PSU
3.	Buzzer	Positif (+)	D27
		Negatif (-)	GND



Gambar 3. 25 Hasil Rangkaian Integrasi Komponen Elektronika

Sumber : Peneliti (2023)

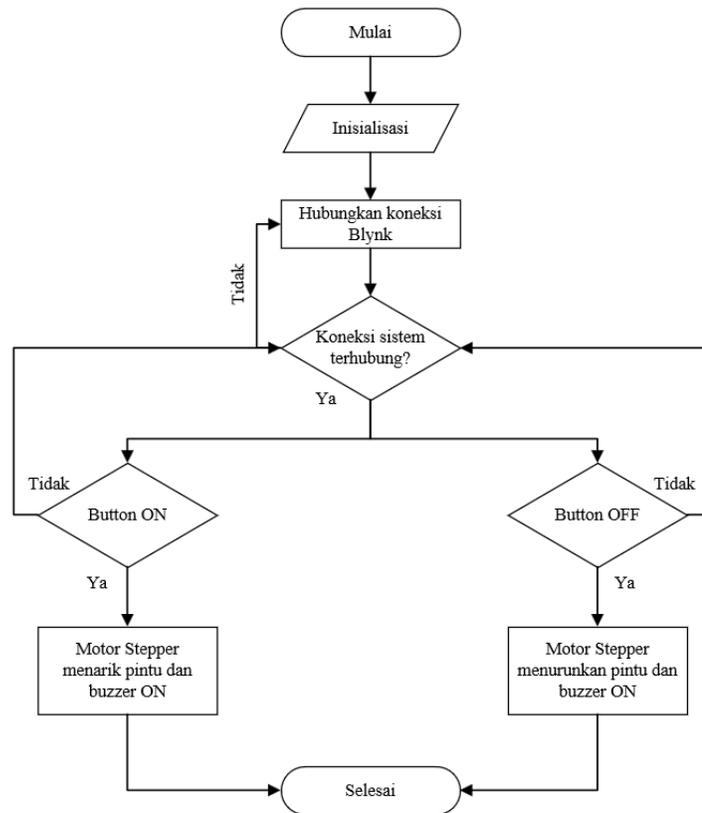
3.5.2.3 Perancangan Perangkat Lunak (Software)

Perancangan *software* meliputi 3 bagian utama, yaitu diagram alir (*flowchart*), antarmuka Blynk, dan program Arduino. Ketiga bagian tersebut dirancang untuk menjalankan kendali penuh ESP32 dalam menggerakkan aktuator motor stepper untuk membuka dan menutup *Flood Retaining Wall* yang terintegrasi dengan *Internet of Things* melalui platform Blynk.

3.5.2.3.1 Diagram Alir Water Flooding

Instruksi dimulai ketika sistem mendapat suplai tegangan dari catu daya (PSU), lalu ESP32 aktif. ESP32 akan melakukan inisialisasi awal program bersamaan dengan mencoba terhubung pada jaringan internet melalui WiFi. Setelah WiFi terhubung maka ESP32 akan mencoba berkomunikasi dengan Blynk yang kemudian status di Blynk menjadi “Online”.

Ketika status pada platform Blynk di *smartphone* berubah menjadi “Online”, maka perintah sudah dapat dieksekusi oleh pengguna dengan menekan “tombol ON”. Ketika “tombol ON” ditekan pada aplikasi Blynk, maka *Flood Retaining Wall* akan naik hingga menutup ke atas secara perlahan. Sebaliknya, ketika ditekan “tombol OFF”, maka *Flood Retaining Wall* akan otomatis berangsur turun hingga terbuka sepenuhnya.



Gambar 3. 26 Diagram Alir *Flood Retaining Wall*

Sumber : Peneliti (2023)

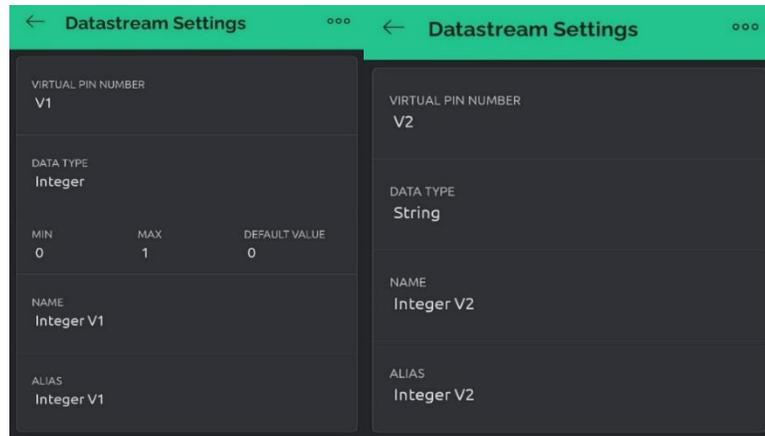
3.5.2.3.2 Perancangan Antarmuka Blynk

Sistem kendali dari *Flood Retaining Wall* akan melakukan kendali melalui kontrol nirkabel secara IoT menggunakan platform Blynk. Melalui Blynk *project*, sistem akan mengendalikan naik dan turunnya *Flood Retaining Wall* secara *real-time* menggunakan *smartphone*, sehingga dapat dikendalikan tanpa adanya batasan jarak dan waktu. Penjelasan sistematika perancangan antarmuka adalah sebagai berikut:

- 1) Pembuatan template Blynk dengan melakukan pengaturan memberikan nama *project* “Water Flooding” untuk informasi *project*, kemudian pemilihan jenis hardware mikrokontroler ESP32 dengan tipe koneksi yang digunakan WiFi. Setelah pembuatan *template*, maka akan mendapatkan pengaturan konfigurasi firmware untuk mengakses mikrokontroler menuju Blynk.

```
// Konfigurasi Blynk
#define BLYNK_TEMPLATE_ID "TMPL6pDUn_Kgb" // ID template
Blynk
#define BLYNK_TEMPLATE_NAME "Palang Pintu" // Nama
template Blynk
#define BLYNK_SERIAL // Aktifkan koneksi serial untuk
debug Blynk
```

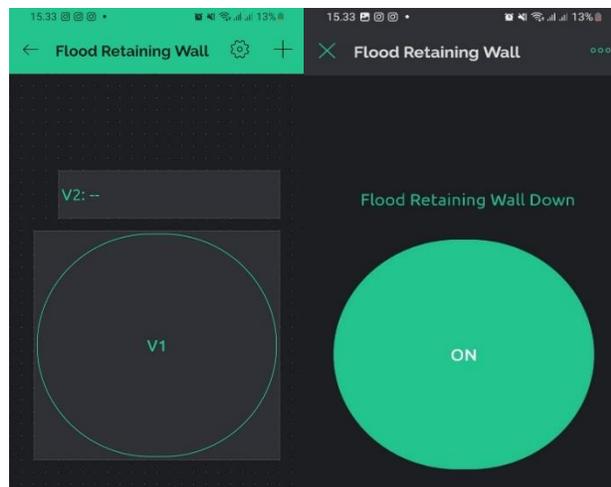
- 2) Pembuatan *datastream* melibatkan pembuatan tombol virtual dengan jenis tipe data "Integer". Nilai minimum dan maksimum untuk tipe data tombol virtual ini adalah 0 dan 1. Alamat tombol *virtual* Blynk yang digunakan adalah V1. kemudian, untuk memberikan informasi apakah pintu *Flood Retaining Wall* sedang naik atau turun, maka membuat *datastream* dengan tipe data *String*. *Datastream* ini akan menampilkan informasi dalam bentuk teks, dan memiliki alamat *virtual* Blynk adalah V2.



Gambar 3. 27 Datastream Tombol dan Datastream Informasi Pintu

Sumber : Peneliti (2023)

- 3) Pembuatan *template visual* pada kendali *Flood Retaining Wall* dengan menambahkan widget pada fitur Blynk. Tambahkan *widget tombol* pada *template* kendali, kemudian menghubungkan alamat V1 dari *datastream* untuk memberikan fungsi instruksi pada tombol. Pada informasi naik dan turunnya *Flood Retaining Wall*, maka tambahkan *widget label* dengan menghubungkan alamat V2 dari *datastream* untuk menampilkan teks ketika tombol bekerja.



Gambar 3. 28 Antarmuka Template dan Antarmuka Kendali Blynk

Sumber: Peneliti (2023)

3.5.2.3.3 Program Sistem Kendali Arduino IDE

Program sistem kendali dibuat menggunakan *software* Arduino IDE. Program bertujuan untuk melakukan integrasi seluruh komponen dalam satu kendali mikrokontroler ESP32. Komponen tersebut meliputi ESP32, *buzzer*, *driver* ULN 2003 yang terhubung dengan motor stepper 28BYJ-48, dan komunikasi nirkabel IoT Blynk.

- 1) Bentuk awal program yang dilakukan adalah melakukan konfigurasi Blynk *template*, mengakses *library*.

```
// Konfigurasi Blynk
#define BLYNK_TEMPLATE_ID "TMPL6pDUn_Kgb" // ID template
Blynk
#define BLYNK_TEMPLATE_NAME "Palang Pintu" // Nama template
Blynk
#define BLYNK_SERIAL // Aktifkan koneksi serial untuk debug
Blynk

// Include Libraries
#include <Stepper.h> // Library untuk mengontrol stepper
motor
#include <BlynkSimpleEsp32.h> // Library untuk
mengintegrasikan ESP32 dengan Blynk
```

- 2) Penyimpanan variabel jumlah langkah per revolusi motor stepper, mendefinisikan pin motor *stepper*, *buzzer*, dan variabel menyimpan interval waktu.

```
// Konfigurasi Stepper Motor dan Pin I/O
const int stepsPerRevolution = 2048; // Jumlah langkah per
revolusi untuk stepper motor

#define IN1 19
#define IN2 18
#define IN3 5
#define IN4 17
#define BUZZER_PIN 27 // Pin untuk mengontrol buzzer

Stepper myStepper(stepsPerRevolution, IN1, IN3, IN2, IN4);
// Objek Stepper

// Variabel Waktu dan Interval
unsigned long previousMillis = 0; // Waktu terakhir langkah
motor diambil
const long interval = 500; // Interval waktu antara setiap
langkah motor
```

```
const int stepDuration = 3; // Durasi setiap langkah motor
```

3) Mengenal autentikasi token Blynk, koneksi WiFi meliputi SSID dan password.

```
// Informasi Blynk dan WiFi
char auth[] = "ziMZyl9LkAWFoeSXqmvueValv892Mt0W"; // Token
otentikasi Blynk
char ssid[] = "xyzxyz"; //Isi Nama jaringan WiFi
char pass[] = "xyzxyz"; //isi password wifi
```

4) Deklarasi BLYNK_WRITE(V1) untuk mengendalikan sistem *Flood Retaining Wall*.

```
BLYNK_WRITE(V1) {
  int buttonState = param.asInt(); // Membaca nilai dari
pin virtual V1 dan menyimpannya dalam variabel buttonState

  if (buttonState == LOW) { // Jika tombol ditekan (nilai
LOW)
    // Pintu sedang naik
    Blynk.virtualWrite(V2, "Flood Retaining Wall Up"); //
Mengirim informasi "Pintu Naik" ke pin virtual V2 di Blynk

    // Menggerakkan stepper motor berlawanan jarum jam
dengan delay dan bunyi buzzer setiap langkah
    for (int i = 0; i < stepDuration; ++i) {
      myStepper.step(-stepsPerRevolution);
      delay(1000); // Delay setelah setiap langkah
      tone(BUZZER_PIN, 1000, 2000); // Buzzer nyala dengan
delay 1 detik
    }
  } else { // Jika tombol dilepaskan (nilai HIGH)
    // Pintu sedang turun
    Blynk.virtualWrite(V2, "Flood Retaining Wall Down");
// Mengirim informasi "Pintu Turun" ke pin virtual V2 di
Blynk

    // Menggerakkan stepper motor searah arah jarum jam
dengan delay dan bunyi buzzer setiap langkah
    for (int i = 0; i < stepDuration; ++i) {
      myStepper.step(stepsPerRevolution);
      delay(1000); // Delay setelah setiap langkah
      tone(BUZZER_PIN, 1000, 2000); // Buzzer nyala dengan
delay 2 detik
    }
  }
}
```

Fungsi BLYNK_WRITE(V1) melakukan inisiasi untuk terjadi perubahan pin virtual V1 di aplikasi Blynk. Nilai pin tersimpan sebagai integer dalam variabel

`buttonState` menggunakan `param.asInt()` nilai yang dikirim dari pin virtual V1. Ketika tombol virtual ditekan On, maka kondisi `if (buttonState == LOW)` dan menampilkan informasi string `Blynk.virtualWrite(V2, " Flood Retaining Wall Up ");`. Dalam instruksi `for (int i = 0; i < stepDuration; ++i)` perulangan `for`, variabel `i` dideklarasikan sebagai integer dan diinisiasi dengan nilai 0. Perulangan program akan terus berjalan selama nilai `i` kurang dari `stepDuration`. Setiap penambahan, nilai `i` akan diinkrement, dan pada setiap peningkatan, motor stepper (yang direpresentasikan oleh objek `myStepper`) akan digerakkan menggunakan perintah `myStepper.step(-stepsPerRevolution)` dan berputar berlawanan jarum jam. Proses ini akan terus berputar sampai nilai `i` tercapai pada `stepDuration` dengan `delay 300ms` setelah 1 `stepDuration` dan diiringi bunyi *buzzer* aktif selama 2s setelah delay.

Jika kondisi sudah terpenuhi, tekan tombol off maka memasuki `else` statement yaitu menjadi `(buttonState == HIGH)` dan menampilkan informasi string `Blynk.virtualWrite(V2, "Flood Retaining Wall Down");`. Cara kerja kondisi pintu turun sama dengan pintu naik, yang membedakan pada program adalah `myStepper.step(stepsPerRevolution)`, dimana nilai `stepsPerRevolution` bernilai positif, menyebabkan berputar searah jarum jam.

- 5) Fungsi `void setup()` menjalankan program sekali ketika mikrokontroler ESP32 pertama kali dihidupkan.

```
void setup() {  
  myStepper.setSpeed(13);  
  Blynk.begin(auth, ssid, pass);  
}
```

Pada fungsi `void setup()`, akan mengatur kecepatan pada motor stepper yaitu `myStepper.setSpeed(13);` diatur pada 13 RPM. Kemudian Blynk akan melakukan inisiasi koneksi dengan fungsi `Blynk.begin` dari *library*. Blynk di panggil untuk memulai koneksi dengan *server* Blynk dengan pernyataan token autentikasi (`auth`), SSID Wi-Fi (`ssid`), dan kata sandi Wi-Fi (`pass`). Setelah itu, maka mikrokontroler ESP32 akan berkomunikasi dengan aplikasi Blynk.

6) Fungsi `void loop()` menjalankan program berulang secara terus menerus.

```
void loop() {  
    Blynk.run();  
}
```

Program `void loop()` akan bertugas untuk menjalankan program secara terus menerus untuk menjaga koneksi dan komunikasi antara mikrokontroler ESP32 dengan *server* Blynk. Panggilan `Blynk.run()` untuk menjalankan memenuhi tanggapan dari perubahan pada aplikasi Blynk ketika melakukan komunikasi dan kendali pada tombol virtual.