BAB III
METODOLOGI PENELITIAN

3.1. Alat dan Bahan Penelitian

Jenis penelitian yang dilakukan adalah melakukan monitoring dan otomasi terhadap tanaman. Penelitian ini menggunakan metode eksperimen, di mana eksperimen ini dilakukan menggunakan alat dan bahan sebagai berikut :

1. Alat Penelitian

   Penelitian yang dilakukan ini, untuk tercapai sesuai dengan tujuan, dibutuhkan beberapa alat pendukung untuk penelitian, yaitu:

   a. PC/Laptop untuk melakukan perancangan dan penyusunan hasil dari penelitian.
   b. Tang Potong & Tang Gunting.
   c. Solder Uap & Timah.
   d. Multimeter Digital.

2. Bahan Penelitian

   Agar penelitian yang dilakukan tercapai sesuai dengan tujuan, dibutuhkan beberapa bahan pendukung untuk penelitian, yaitu:

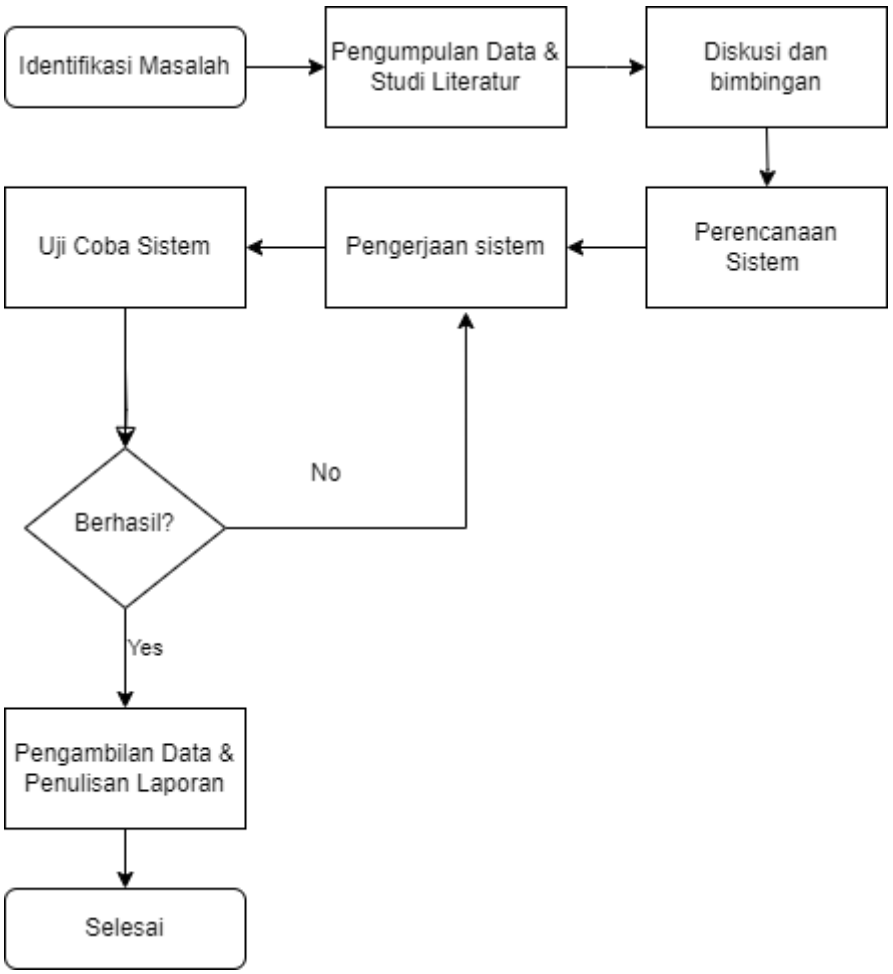
   a. Catu daya sebagai input tegangan
   b. Sensor Waktu (RTC), Sensor Kelembaban tanah *(Soil Moisture)*, Sensor Suhu (DHT 11) sebagai *Input* penelitian ini.
   c. Blynk IoT sebagai sarana Internet of Things.
   d. *Relay, Mini Water Pump,* LED sebagai *Output* dari penelitian ini.

e. Arduino ESP 32 sebagai mikrokontroller.

f. Kabel konektor.

3.2. Alur Penelitian

Selama penyusunan dan penulisan proposal ini penulis melakukan identifikasi masalah, pengumpulan bahan materi dari berbagai sumber, serta diskusi dan bimbingan sehingga menunjang proses perancangan. Adapun alur penelitian adalah sebagai berikut :



Gambar 3. 1 Alur Penelitian

1. Identifikasi Masalah

   Identifikasi masalah yang penulis lakukan adalah dengan melihat banyaknya tanaman yang layu akibat, kurangnya perawatan terhadap tanaman.

2. Pengumpulan Data dan Studi Literatur

   Penulis melakukan studi pustaka untuk mendapatkan referensi yang relevan dengan tujuan penelitian yaitu mempelajari sistem monitoring tanaman dengan teknologi *Internet of Things*.

3. Diskusi dan Bimbingan

   Penulis melakukan perencanaan dimulai dari pemilihan komponen yang akan digunakan dan perancangan konstruksi serta rangkaian pendukung lain

4. Perencanaan sistem

   Penulis melakukan perencanaan dimulai dari pemilihan komponen yang akan digunakan dan perancangan konstruksi serta rangkaian pendukung lainya.

5. Pembuatan sistem

   Setelah tahap perencanaan selesai, maka alat mulai dibuat sesuai dengan hasil perancangan.

6. Uji Coba sistem

   Dalam tahap ini akan diuji apakah sesuai dengan kriteria yang dikehendaki.

7. Pengambilan data dan penulisan laporan

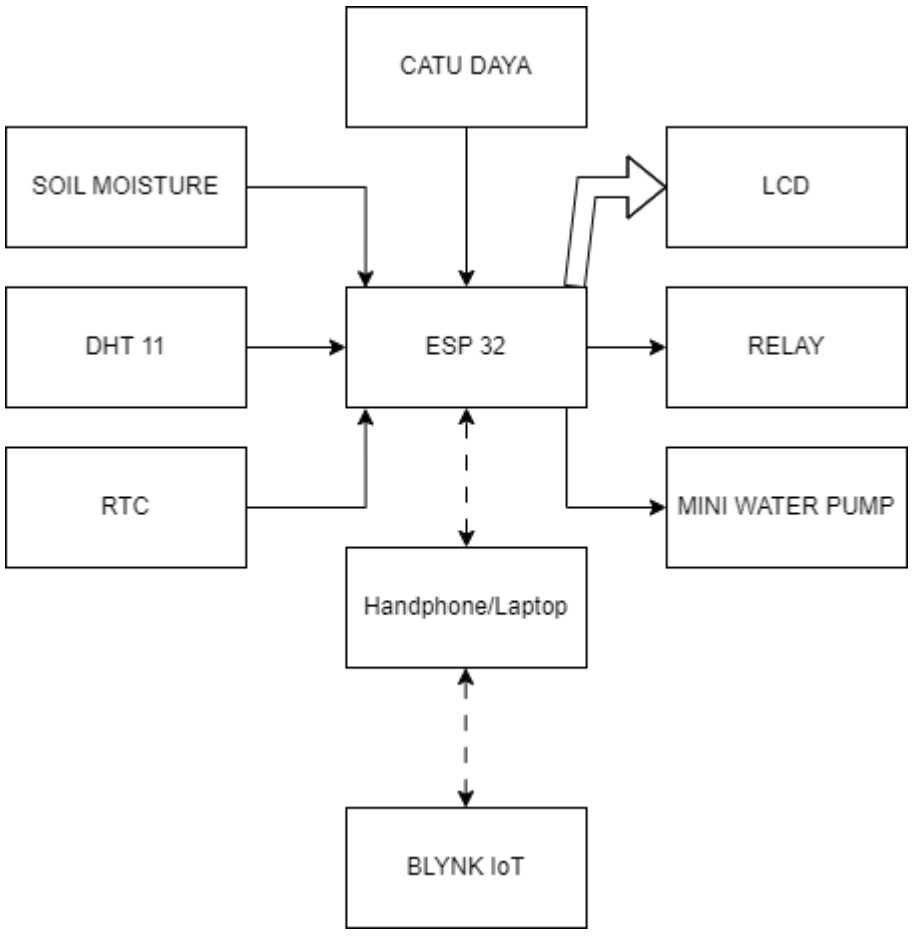
Penulisan laporan berdasarkan kepada hasil pengujian sistem yang telah dilakukan pada alat yang dibuat.

3.3. Deskripsi Sistem Penyiraman Tanaman Otomatis

Untuk menjelaskan dan menguraikan bagaimana sistem dapat bekerja maka dijelaskanlah sebagai berikut:

a. Block Diagram Sistem

Untuk menguraikan bagaimana sistem ini dapat berjalan dengan baik, maka dari itu dirancanglah blok diagram seperti pada gambar Gambar 3.2



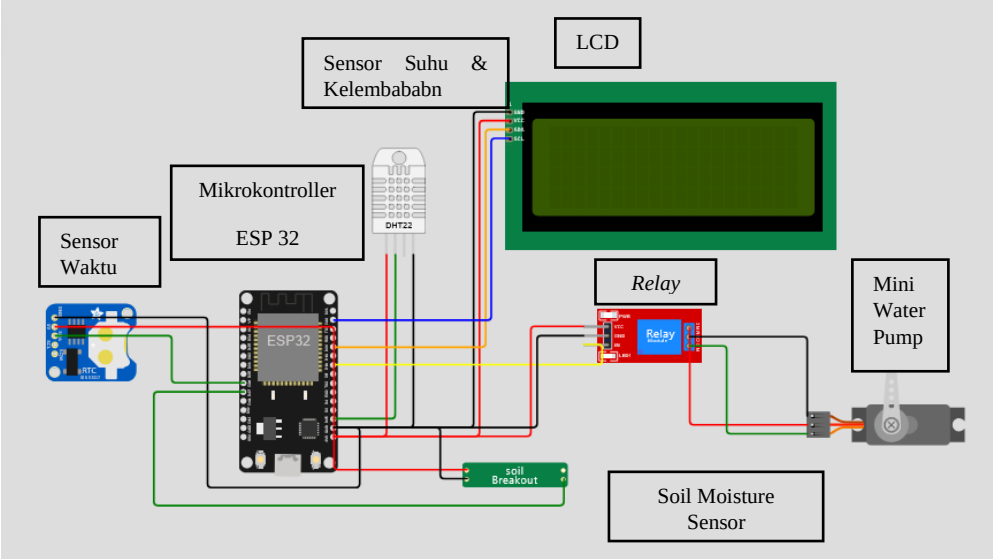
Gambar 3. 2 Diagram Rangkaian Sistem Penyiram tanaman otomatis

Diagram ini menunjukkan rancangan sistem otomatisasi penyiraman tanaman berbasis IoT *(Internet of Things)* yang menggunakan komponen-komponen elektronik seperti ESP32, sensor, dan aktuator. Berikut adalah penggunaan pin pada mikrokontroller ESP 32. ESP32 bertindak sebagai pengendali utama yang menghubungkan semua sensor dan aktuator.

Tabel 3.1. Kaki (Pin) yang digunakan Pada ESP 32

Komponen	Kaki (Pin) di ESP32	Fungsi
Sensor DHT11	GPIO 5	Data suhu dan kelembaban udara
Sensor Kelembaban Tanah	A0 (GPIO 36)	Membaca kelembaban tanah
RTC DS3231	SDA (GPIO 21)	Data komunikasi I2C (Serial Data)
LCD I2C 16x2	SCL (GPIO 22)	Clock komunikasi I2C
	SDA (GPIO 21)	Data komunikasi I2C
	SCL (GPIO 22)	Clock komunikasi I2C
Relay untuk pompa air	GPIO 13	Menghidupkan/mematikan pompa air
Tombol Manual Mode	GPIO 15	Mengaktifkan mode manual

Pembuatan sistem secara software menggunakan aplikasi wokwi. Berikut adalah gambaran sistem.



Gambar 3.3 sistem Penyiraman Tanaman Otomatis

Secara umum penelitian sistem penyiraman tanaman otomatis ini menggunakan *Mikrokontroller* ESP 32 berbasis Android dapat digambarkan seperti pada Gambar 3.3. Sistem yang dirancang terdiri dari beberapa komponen, yaitu Arduino ESP 32, sensor Suhu dan kelembaban, sensor Waktu, sensor kelembaban tanah, LCD dan Relay sebagai pemutus dan pembuka motor dc bekerja. ESP 32 ini merupakan mikrokontroler yang berfungsi sebagai pusat pengolahan data untuk memproses data yang diberikan oleh sensor-sensor yang digunakan. Sensor -sensor akan mengirim data ke mikrokontroller dan diproses, untuk menjalankan sistem ini.

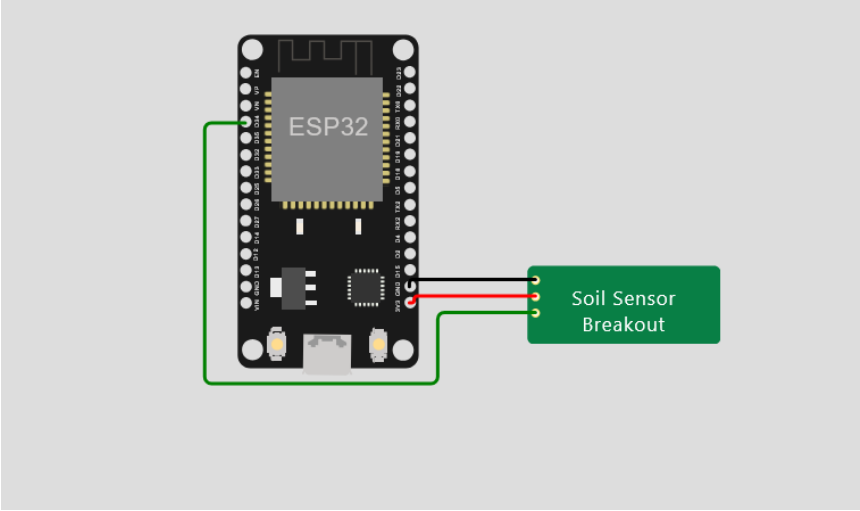
Apapun penjelasan dari hubungan block diagram dari setiap komponen adalah sebagai berikut:

1. Catu Daya

   Catu daya berfungsi sebagai sumber listrik utama yang mengalirkan energi ke seluruh komponen dalam sistem. Komponen ini memastikan bahwa semua perangkat seperti sensor, mikrokontroler, dan aktuator dapat beroperasi dengan baik.

2. Sensor *Soil Moisture*

   Sensor ini digunakan untuk mengukur tingkat kelembapan tanah. Informasi mengenai kadar air di dalam tanah akan dikirimkan ke ESP32 untuk dianalisis. Jika tanah terdeteksi kering, sistem akan mengaktifkan pompa air untuk menyiram taman. Pada gambar 3.4 Dapat dilihat penghubungan sensor ke mikrokontroller esp 32, menggunakan kaki A0(GPIO 36) untuk membaca kelembaban tanah.

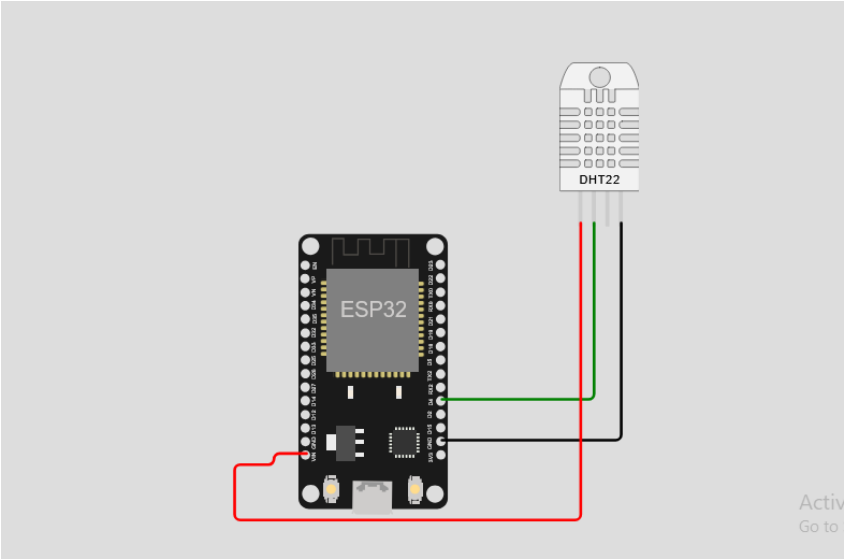


Gambar 3.4. Hubungan ESP 32 dengan Sensor *Soil Moisture*

3. DHT11 (Sensor Suhu dan Kelembapan)

   DHT11 berfungsi untuk mengukur suhu dan kelembapan udara di sekitar tanaman. Data ini dapat digunakan untuk mengontrol kondisi lingkungan

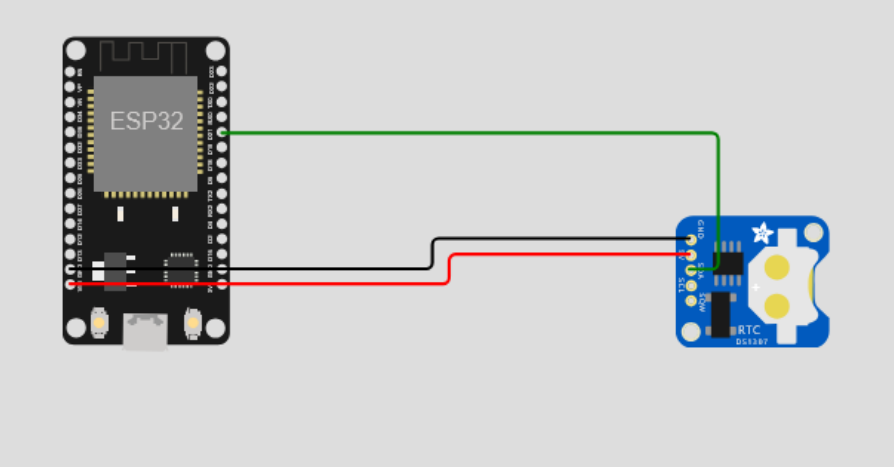
tanaman, serta memberikan informasi yang lebih lengkap mengenai status pertumbuhan tanaman kepada pengguna melalui platform IoT. Dapat dilihat pada gambar 3.5 Adalah proses penghubungan mikrokontroller ESP 32 dengan sensor DHT. Menggunakan kaki GPIO 5 untuk mengirim data suhu dan kelembaban udara.



Gambar 3.5. Hubungan ESP 32 dengan Sensor DHT

4. RTC *(Real Time Clock)*

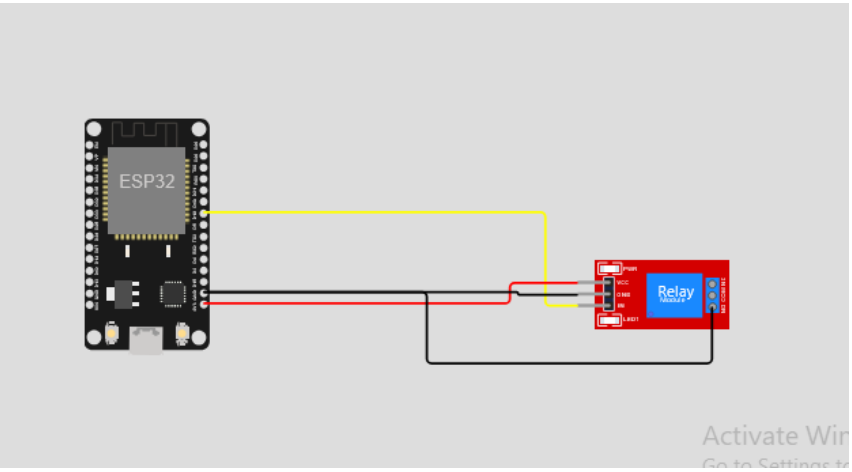
RTC digunakan untuk memberikan informasi waktu yang akurat kepada sistem. Komponen ini memastikan bahwa proses penyiraman dapat dilakukan pada waktu-waktu tertentu sesuai dengan jadwal yang telah ditentukan oleh pengguna atau berdasarkan kebutuhan tanaman. Pada gambar berikut merupakan hubungan kaki antara sensor RTC dengan Mikrokontroller esp 32. menggunakan kaki SDA (GPIO 21) untuk Data komunikasi I2C (Serial Data)



Gambar 3.6. Hubungan ESP 32 dengan Sensor RTC

5. Relay

Relay adalah sakelar elektronik yang digunakan untuk mengendalikan aliran listrik ke pompa air. Ketika ESP32 memutuskan untuk menyiram tanaman, relay akan diaktifkan untuk menyalakan pompa air. Relay bekerja sesuai dengan logika program dan keadaan kelembaban tanah untuk menyalakan pompa air. Adapun kaki yang digunakan untuk relay ini adalah GPIO 13. Dapat dilihat pada gambar 3.7 berikut.



Gambar 3.7. Hubungan ESP 32 dengan *Relay*

6. *Mini Water Pump*

   Pompa air ini digunakan untuk menyiram tanaman secara otomatis ketika tanah terdeteksi kering oleh sensor soil moisture. Pompa ini hanya akan aktif saat *relay* menyala. Dan ketika *relay* nonaktif maka pompa air tidak menyala. Pompa bekerja sesuai intruksi dan logika program.

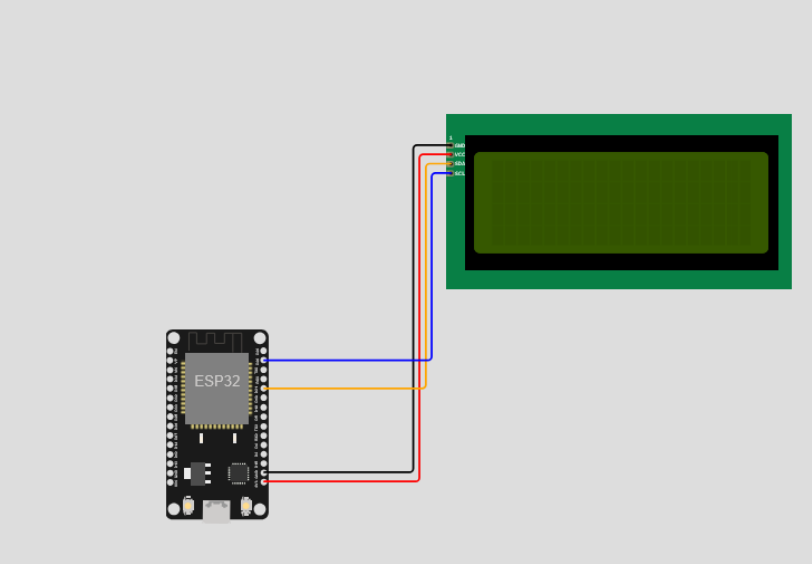
7. LCD

   LCD *(Liquid Crystal Display)* digunakan untuk menampilkan informasi penting seperti tingkat kelembapan tanah, suhu, kelembapan udara, dan status sistem. Hal ini memudahkan pengguna untuk mendapatkan informasi langsung dari sistem tanpa perlu membuka aplikasi Blynk.

Alur Kerja Sistem:

Sistem dimulai dengan pengukuran kelembapan tanah oleh sensor soil moisture. Jika kelembapan di bawah ambang batas yang telah ditentukan, ESP32 akan memerintahkan relay untuk mengaktifkan pompa air. Sementara itu, sensor DHT11 terus mengukur suhu dan kelembapan udara, memberikan gambaran lengkap tentang kondisi lingkungan di sekitar tanaman. RTC memastikan bahwa penyiraman dilakukan pada waktu yang tepat jika sistem diatur untuk bekerja secara terjadwal. Informasi tentang kelembapan tanah, suhu, dan kelembapan udara ditampilkan di layar LCD untuk monitoring lokal. Pengguna dapat memantau dan mengontrol seluruh sistem melalui aplikasi Blynk IoT yang terhubung ke ESP32, sehingga mereka bisa melakukan penyesuaian dari jarak jauh. Dengan sistem ini, penyiraman tanaman menjadi lebih efisien dan dapat diatur secara otomatis sesuai dengan kebutuhan tanaman, tanpa perlu intervensi manual yang

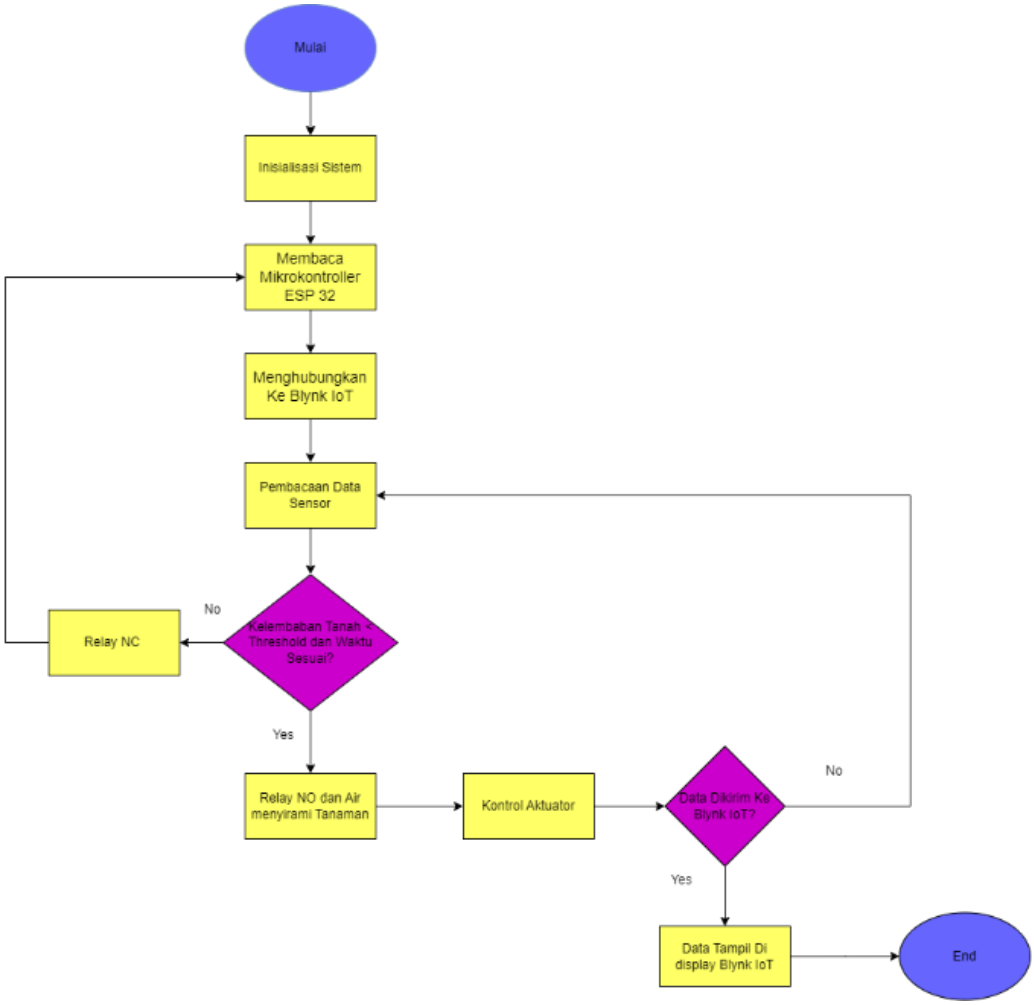
intensif. Adapun kaki yang digunakan untuk menghubungkan LCD dengan mikrokontroller ESP 32 adalah SCL (GPIO 22) Clock Komunikasi I2C, SDA (GPIO 21) untuk Data komunikasi I2C dan SCL(GPIO 22) untuk Clock Komunikasi I2C.



Gambar 3.8 Hubungan ESP 32 dengan LCD

b. Alur Kerja Sistem Monitoring

Alur kerja adalah gambar atau bagan yang memperlihatkan urutan atau langkah-langkah dari suatu program dan hubungan antar proses beserta pernyataannya. Tujuan utamanya adalah untuk menggambarkan suatu tahapan penyelesaian masalah secara sederhana, terurai, rapi dan jelas dengan menggunakan simbol-simbol yang standar. Adapun bentuk dari Alur Kerja Sistem Monitoring ini dapat dilihat pada gambar 3.9 :



Gambar 3.9 Alur Kerja Sistem Monitoring

Dari gambar diatas dapat dijabarkan sebagai berikut:

1. Mulai (*Start*)

   Sistem dimulai ketika perangkat dinyalakan. Ini merupakan langkah awal dalam menjalankan sistem otomasi penyiraman tanaman.

2. Inisialisasi Sistem

   Pada tahap ini, ESP32 akan menginisialisasi seluruh komponen yang digunakan, termasuk:

   - Sensor DHT11 untuk membaca suhu dan kelembapan udara.
   - Sensor Soil Moisture untuk membaca kelembapan tanah.

- RTC DS3231 untuk mengatur waktu penyiraman.
- LCD I2C untuk menampilkan informasi.
- Koneksi ke Blynk IoT untuk memungkinkan pemantauan data melalui aplikasi.

3. Membaca Mikrokontroler ESP32

   ESP32 mulai membaca kondisi sensor dan menyiapkan data untuk diproses lebih lanjut. Data yang dibaca meliputi:

   - Suhu dan kelembapan udara dari DHT11.
   - Kelembapan tanah dari *Soil Moisture Sensor*.
   - Waktu saat ini dari RTC DS3231.

4. Menghubungkan ke Blynk IoT

   ESP32 akan mencoba terhubung ke jaringan WiFi dan kemudian mengakses server Blynk IoT. Jika koneksi berhasil, sistem akan mulai mengirimkan data sensor secara berkala ke aplikasi Blynk untuk dipantau oleh pengguna.

5. Pembacaan data sensor

   Setelah terhubung ke Blynk, ESP32 akan membaca data dari sensor secara berkala untuk mendapatkan informasi real-time tentang kondisi lingkungan tanaman, seperti:

   - Suhu udara
   - Kelembapan udara
   - Kelembapan tanah

6. Keputusan: Apakah Kelembapan Tanah < *Threshold* dan Waktu Sesuai?

Sistem akan memeriksa dua parameter utama sebelum mengaktifkan penyiraman:

- Apakah kelembapan tanah di bawah *threshold* (misalnya 30%)?
- Apakah waktu saat ini sesuai dengan jadwal penyiraman?

Terdapat dua kemungkinan keputusan:

a. Jika tidak (kelembapan tanah cukup atau bukan waktu penyiraman)
    - Relay dalam kondisi NC (*Normally Closed*), yang berarti pompa tetap mati dan penyiraman tidak dilakukan.
    - Sistem kembali ke tahap pembacaan sensor dan terus memantau kondisi tanaman.

b. Jika ya (kelembapan tanah rendah dan waktu sesuai)
    - Relay NO (*Normally Open)* akan aktif dan menghidupkan pompa air.
    - Air akan menyiram tanaman sesuai kebutuhan.

7. Kontrol Aktuator (Relay dan Pompa Air)

   Jika sistem memutuskan untuk menyiram tanaman, relay akan aktif, yang menghidupkan pompa air untuk menyiram tanaman secara otomatis.

8. Keputusan: Apakah Data Dikirim ke Blynk IoT?

   Sistem memeriksa apakah data sensor dan status penyiraman berhasil dikirim ke aplikasi Blynk IoT.

   a. Jika tidak berhasil dikirim, sistem tetap berusaha mengirimkan data dalam siklus berikutnya.

b. Jika berhasil dikirim, data akan ditampilkan di aplikasi Blynk agar pengguna dapat memantau kondisi tanaman dari jarak jauh.

9. Data Tampil di *Display* Blynk IoT

Jika data berhasil dikirim ke Blynk, pengguna dapat melihatnya melalui aplikasi Blynk, yang menampilkan:

- Suhu dan kelembapan udara
- Kelembapan tanah
- Status pompa air (ON atau OFF)

10. Selesai *(End)*

Setelah data berhasil dikirim dan ditampilkan, sistem kembali ke tahap awal untuk terus melakukan pemantauan dan pengendalian penyiraman tanaman secara otomatis.

3.4. Perencanaan Perangkat Lunak (*Softwar*e)

Dalam perencanaan alat penyiram tanaman otomatis ini diperlukan perencanaan perangkat lunak, untuk membuat sistem ini dapat bekerja dan berjalan dengan baik, maka direncanakanlah program sebagai berikut :

a. Kode Program untuk terhubung ke BLYNK

```
#include <WiFi.h>
#include <BlynkSimpleEsp32.h>
#define BLYNK_TEMPLATE_ID "TMPL67w_tqG5I"
#define BLYNK_TEMPLATE_NAME "iot farming"
#define BLYNK_AUTH_TOKEN "-uJ--_jk9FIS1CbgV5T1ePhjUqpixcpW"
char auth[] = BLYNK_AUTH_TOKEN;
char ssid[] = "iPhone";
char pass[] = "deryak12";
```

Gambar 3.10 Code Program menghubungkan ESP 32 dengan blynk

Gambar 3.10 Code Program menghubungkan ESP 32 dengan Blynk mengirimkan data ke Blynk melalui ESP32, dan menampilkan hasilnya di

Serial Monitor. Pertama, library WiFi dan BlynkSimpleEsp32 diimpor agar ESP32 dapat terhubung ke jaringan WiFi dan aplikasi Blynk. Selanjutnya, informasi autentikasi Blynk dan kredensial WiFi dideklarasikan menggunakan BLYNK_TEMPLATE_ID, BLYNK_TEMPLATE_NAME, dan BLYNK_AUTH_TOKEN. Nama dan kata sandi WiFi ditetapkan dalam variabel ssid dan pass.

b. Kode Program untuk RTC (*Real-Time Clock*)

```
#include "RTClib.h"
RTC_DS3231 rtc;
void setup() {
    Serial.begin(115200);
    if (!rtc.begin()) {
        Serial.println("RTC tidak ditemukan!");
        while (1);
    }
    rtc.adjust(DateTime(__DATE__, __TIME__));
}
void loop() {
    DateTime now = rtc.now();
    Serial.print(now.hour()); Serial.print(":");
    Serial.print(now.minute()); Serial.print(":");
    Serial.println(now.second());
    delay(1000);
}
```

Gambar 3.11 Code Program menghubungkan ESP 32 dengan *Real Time Clock*

Berikut ini merupakan penjelasan dari gambar 3.11:

- #include "RTClib.h"

  Memuat pustaka (library) RTClib yang digunakan untuk mengakses modul RTC DS3231.

- RTC_DS3231 rtc;

  Mendeklarasikan objek rtc untuk mengakses fungsi-fungsi dari modul RTC DS3231.

- Serial.begin(115200);

  Menginisialisasi komunikasi serial dengan baud rate 115200.

- if (!rtc.begin()) { Serial.println("RTC tidak ditemukan!"); while (1); }

  Mengecek apakah modul RTC DS3231 terdeteksi.Jika tidak terdeteksi, program akan menampilkan pesan "RTC tidak ditemukan!" dan berhenti (while (1); membuat program tetap berjalan tanpa melakukan operasi lain).

- rtc.adjust(DateTime(__DATE__, __TIME__));

  Menyesuaikan waktu RTC dengan waktu saat kode dikompilasi.

- __DATE__ dan __TIME__ adalah makro bawaan yang menyimpan tanggal dan waktu saat program dikompilasi.

- DateTime now = rtc.now();

  Membaca waktu saat ini dari modul RTC DS3231.

- Serial.print(now.hour()); Serial.print(":"); Menampilkan jam di serial monitor.

- Serial.print(now.minute()); Serial.print(":"); Menampilkan menit di serial monitor.

- Serial.println(now.second()); Menampilkan detik di serial monitor dan pindah ke baris baru.

- delay(1000); Menunda eksekusi selama 1 detik, sehingga waktu diperbarui setiap detik.

c. Code Program untuk sensor DHT 11

Agar sensor DHT 11 dapat difungsikan dan deprogram dalam mikrokontroller ESP 32, maka diperlukan program seperti gambar

```
#include "DHT.h"
#define DHTPIN 5
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);
void setup() {
    Serial.begin(115200);
    dht.begin();
}
void loop() {
    float temp = dht.readTemperature();
    float humi = dht.readHumidity();
    Serial.print("Suhu: "); Serial.print(temp); Serial.print("C
    Serial.print("Kelembaban: "); Serial.print(humi); Serial.pri
    delay(2000);
}
```

Gambar 3.12 Code Program menghubungkan ESP 32 dengan *DHT 11*

Berikut ini merupakan penjelasan program dari gambar 3.12:

- pustaka DHT dimuat menggunakan #include "DHT.h", yang diperlukan agar ESP32 dapat berkomunikasi dengan sensor.
- #define DHTPIN 5 mendefinisikan bahwa sensor terhubung ke pin GPIO 5, dan #define DHTTYPE DHT11 menentukan jenis sensor yang digunakan.
- Objek DHT dht(DHTPIN, DHTTYPE); dibuat untuk mengakses fungsi pembacaan sensor.
- Dalam fungsi setup(), komunikasi Serial Monitor diaktifkan dengan Serial.begin(115200);, yang memungkinkan tampilan data sensor di komputer.
- dht.begin();, memastikan ESP32 siap membaca data suhu dan kelembaban.
- Pada fungsi loop(), suhu dan kelembaban udara dibaca menggunakan float temp = dht.readTemperature(); dan float humi = dht.readHumidity();. Jika pembacaan berhasil, nilai-nilai tersebut dikirimkan ke Serial Monitor

menggunakan Serial.print(), sehingga hasilnya ditampilkan dalam format seperti Suhu: 28.5C Kelembaban: 65%.

- Setelah setiap pembacaan, terdapat delay(2000); yang memberi jeda selama 2 detik sebelum sensor membaca ulang data.

d. Code program untuk *Soilmoisture* sensor

```
const int sensor_pin = A0;
int _moisture;
void setup() {
    Serial.begin(115200);
    Blynk.begin(auth, ssid, pass, "blynk.cloud", 80);
}
void loop() {
    Blynk.run();
    _moisture = (100 - ((analogRead(sensor_pin) / 4095.00) * 10(
    Serial.print("Kelembaban Tanah = ");
    Serial.print(_moisture);
    Serial.println("%");
    Blynk.virtualWrite(V2, _moisture);
    delay(1000);
}
```


Gambar 3.13 Code Program menghubungkan ESP 32 dengan *Soil Moisture Sensor*

Sensor kelembaban tanah dideklarasikan dengan variabel sensor_pin yang terhubung ke pin A0 pada ESP32. Variabel _moisture digunakan untuk menyimpan nilai pembacaan sensor. Pada fungsi setup(), komunikasi serial dengan baud rate 115200 diinisialisasi menggunakan Serial.begin(115200); agar data dapat ditampilkan di Serial Monitor. Selanjutnya, ESP32 terhubung ke jaringan WiFi dan Blynk melalui Blynk.begin(auth, ssid, pass, "blynk.cloud", 80);.

Di dalam fungsi loop(), Blynk.run(); memastikan ESP32 tetap terhubung ke server Blynk. Data kelembaban tanah dibaca menggunakan

analogRead(sensor_pin), dikonversi ke persentase dengan rumus (100 - ((analogRead(sensor_pin) / 4095.00) * 100));, lalu disimpan dalam variabel _moisture.

Hasil pembacaan ditampilkan di Serial Monitor dengan Serial.print("Kelembaban Tanah = ");. Data kelembaban kemudian dikirimkan ke Blynk menggunakan Blynk.virtualWrite(V2, _moisture);, di mana V2 adalah Virtual Pin di aplikasi Blynk. Akhirnya, delay(1000); memberikan jeda 1 detik sebelum membaca data kembali.

e. Code Program untuk *Relay* (Kontrol Penyiraman Otomatis)

```
int relay1 = 13;
void setup() {
    pinMode(relay1, OUTPUT);
    digitalWrite(relay1, HIGH);
}
void loop() {
    if (_moisture < 30) {
        digitalWrite(relay1, LOW);
        Serial.println("Pompa Menyala");
    } else {
        digitalWrite(relay1, HIGH);
        Serial.println("Pompa Mati");
    }
    delay(5000);
}
```

Gambar 3.14 Code Program menghubungkan ESP 32 dengan *Relay*

Kode ini digunakan untuk mengontrol *relay* yang mengaktifkan atau menonaktifkan pompa air berdasarkan kelembaban tanah yang diukur oleh sensor *soil moisture*.

Pertama, variabel relay1 dideklarasikan dan diatur pada pin 13 sebagai output. Dalam fungsi setup(), pin relay dikonfigurasi sebagai output menggunakan pinMode(relay1, OUTPUT); dan secara default diatur dalam

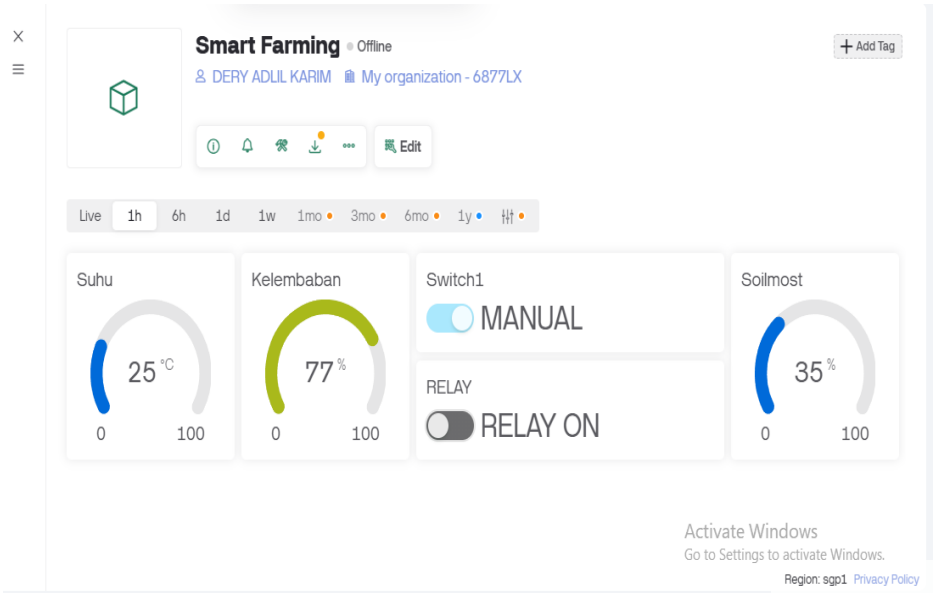
kondisi HIGH (digitalWrite(relay1, HIGH);) yang berarti pompa dalam keadaan mati.

Di dalam loop(), sistem membaca nilai kelembaban tanah dari variabel _moisture. Jika nilai _moisture kurang dari 30%, maka pompa air akan menyala dengan digitalWrite(relay1, LOW);, dan pesan "Pompa Menyala" ditampilkan di Serial Monitor. Sebaliknya, jika nilai _moisture lebih dari 30%, pompa akan mati dengan digitalWrite(relay1, HIGH); dan pesan "Pompa Mati" ditampilkan di Serial Monitor.

Akhirnya, *delay*(5000); memberikan jeda 5 detik sebelum pembacaan ulang kelembaban tanah dan eksekusi kembali program. Hal ini bertujuan untuk mengurangi frekuensi perubahan status pompa agar tidak terlalu sering hidup dan mati dalam waktu singkat.

Adapun, program diatas dapat diubah sesuai kebutuhan dan keinginan pengguna, apabila ingin mengubahnya, hanya perlu mengganti program dan menginputkan Kembali ke mikrokontroller ESP 32.

Perancangan proyek akhir Sistem penyiraman tanaman otomatis ini, tidak terlepas dari *internet of thing*, di sini penulis menggunakan Blynk sebagai Aplikasi Monitoring dan menjalankan sistemnya. Desain dapat dilihat pada gambar berikut :



Gambar 3.15 Desain Tampilan Blynk

Aplikasi Blynk IoT ini berfungsi sebagai monitoring sistem yang telah dibuat agar dapat berjalan menggunakan *internet of thing*, dan dapat dikontrol dimana saja berada. Blynk IoT dapat di akses menggunakan web di laptop, atau menggunakan *handphone*. Setelah semua program dan tampilan Blynk IoT selesai, program pada Arduino IDE dapat diinputkan ke Mikrokontroller.

3.5. Perencanaan kotak

Dalam penelitian ini, perencanaan perangkat keras sangat penting dilakukan, Adapun perencanaan sebagai berikut:

1. Perencanaan Kotak Sistem Kontrol

Gambar 3.16. Kotak Sistem Kontrol

Komponen yang Ditempatkan dalam Kotak. Kotak ini menampung berbagai komponen sistem kontrol, di antaranya:

a) ESP32 (Mikrokontroler utama).
b) Modul Relay 5V (Pengontrol pompa air).
c) Modul RTC DS3231 (Real-Time Clock untuk penjadwalan).
d) Power Supply (Adaptor 12V ke 5V/3.3V).
e) LCD I2C 16x2 (Layar tampilan data).
f) Kabel dan konektor (Menghubungkan sensor dan aktuator).

2. Pembuatan kotak *Protoype*

Kotak *prototype* ini digunakan untuk melakukan simulasi alat yang telah dirancang. Kotak ini dibuat menggunakan akrilik dengan rincian:

Panjang : 60 cm

Lebar : 30 cm

Tinggi : 15 cm

Gambar 3.17. kotak *Protoype*

Terdapat 3 bagian pada kotak *Protoype* ini

2. Tempat budidaya tanaman
3. Tempat penampungan air
4. Kotak sistem