

## BAB IV

### PROSES PEMBUATAN ALAT

#### 4.1 Pembuatan Perangkat Keras

Pembuatan perangkat keras untuk penerapan *Predictive maintenance* pada penyiraman otomatis berbasis ESP32 ini menggunakan mikrokontroler ESP32 sebagai pusat kendali utama. Sistem ini dirancang untuk memantau kondisi lingkungan dan melakukan prediksi kebutuhan penyiraman tanaman berdasarkan data dari sensor *Soil Moisture* menggunakan metode *Moving Average*.

Komponen utama yang digunakan dalam sistem ini meliputi ESP32, sensor kelembapan tanah (*Soil Moisture* sensor) yang berfungsi untuk mengetahui kadar air dalam media tanam, sensor suhu dan kelembapan udara DHT22 untuk memantau kondisi lingkungan sekitar tanaman, serta sensor intensitas cahaya BH1750 untuk mengetahui pencahayaan yang diterima tanaman.

Selain itu, digunakan *Real Time Clock* (RTC DS3231) yang berfungsi sebagai pencatat waktu untuk menyimpan data berdasarkan jam dan tanggal yang akurat. Komponen output yang digunakan adalah pompa air DC sebagai aktuator utama untuk penyiraman, yang dikendalikan oleh ESP32 melalui modul MOSFET. Modul MOSFET ini dipilih karena lebih efisien dan responsif dalam mengendalikan pompa dibandingkan dengan modul relay.

Untuk menampilkan informasi kondisi lingkungan dan status sistem, digunakan LCD 16x2 I2C yang menampilkan data seperti kelembapan tanah, suhu udara, serta status pompa (aktif atau tidak). Sistem juga terhubung ke aplikasi Blynk melalui koneksi Wi-Fi sehingga pengguna dapat memantau dan mengendalikan sistem secara jarak jauh melalui smartphone.

Seluruh komponen ini dirangkai pada PCB secara terintegrasi sehingga memudahkan dalam pemasangan dan pengujian alat. Pada Tabel 4.1 akan ditampilkan

daftar lengkap alat dan bahan yang digunakan dalam pembuatan perangkat keras sistem pot pintar ini.

#### 4.2 Alat dan Bahan

Proses selanjutnya yaitu melakukan persiapan alat dan bahan yang akan digunakan untuk perancangan sistem yang telah dibuat pada

*Tabel 4. 1 Komponen Penggunaan*

| No. | Nama Bahan                   | Jumlah |
|-----|------------------------------|--------|
| 1   | ESP32                        | 1 buah |
| 2   | Sensor BH1750                | 1 buah |
| 3   | <i>Soil Moisture</i>         | 1 buah |
| 4   | Sensor DHT22                 | 1 buah |
| 5   | <i>Buck Converter LM2596</i> | 1 buah |
| 6   | Mosfet                       | 1 buah |
| 7   | <i>Real Time Clock (RTC)</i> | 1 buah |
| 8   | LCD 16 x 2                   | 1 buah |
| 9   | Socket DC                    | 1 buah |
| 10  | Kabel data USB               | 1 buah |
| 11  | Adaptor 12 V 5 A             | 1 buah |

*Tabel 4. 2 Alat yang digunakan*

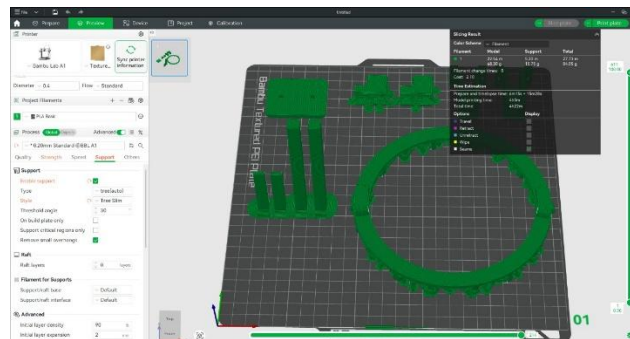
| No | Nama Alat  | Jumlah |
|----|------------|--------|
| 1  | Multimeter | 1 buah |
| 2  | Set obeng  | 1 buah |
| 3  | Gunting    | 1 buah |
| 4  | Cutter     | 1 buah |

|   |            |        |
|---|------------|--------|
| 5 | Alat tulis | 1 buah |
| 6 | Solder     | 1 buah |
| 7 | Bor        | 1 buah |

### 4.3 Langkah-Langkah Pembuatan Perangkat Keras (*Hardware*)

#### 4.3.1 Pembuatan 3D Penyiraman Melingkar

Pada perancangan sistem pot pintar ini, salah satu komponen utama adalah penyiram berbentuk melingkar yang berfungsi untuk mendistribusikan air secara merata di area media tanam. Desain penyiram melingkar dibuat khusus menggunakan perangkat lunak desain 3D dan diproses melalui aplikasi design untuk tahap slicing sebelum dicetak menggunakan printer 3D.



Gambar 4. 1 3D Penyiraman Melingkar

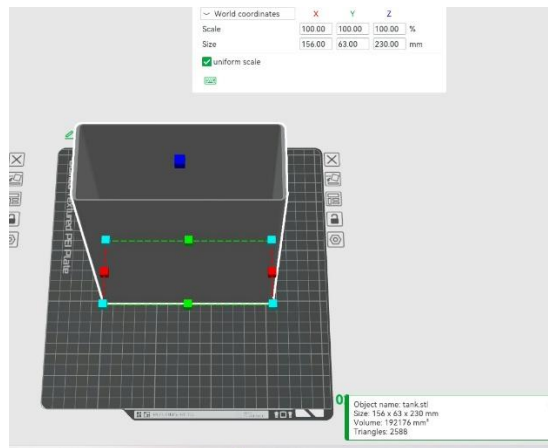
Gambar di atas menunjukkan tampilan preview hasil slicing, di mana setiap bagian komponen ditempatkan secara optimal pada build plate untuk efisiensi waktu cetak dan pemakaian material.



*Gambar 4. 2 Hasil 3D Penyiraman Melingkar*

Desain penyiraman melingkar dipilih karena mampu memberikan distribusi air yang merata ke seluruh area media tanam melalui lubang-lubang kecil yang tersebar di sepanjang lingkaran, sehingga seluruh bagian akar tanaman mendapatkan kelembapan yang seimbang. Sistem ini juga mengurangi risiko terjadinya genangan air pada satu sisi pot karena penyiraman dilakukan dari berbagai titik secara bersamaan. Selain itu, debit air yang keluar terbagi secara terukur di seluruh lingkaran, sehingga penggunaan air menjadi lebih efisien dan tidak terjadi pemborosan. Distribusi air yang merata ini juga membantu meningkatkan akurasi pembacaan sensor kelembapan tanah, yang sangat penting dalam penerapan metode *Predictive maintenance* untuk memprediksi kebutuhan penyiraman. Dari segi estetika dan integrasi desain, bentuk melingkar mudah disesuaikan dengan berbagai ukuran pot dan membuat sistem tampak rapi serta terintegrasi dengan baik.

### 4.3.2 Pembuatan 3D Penyimpanan Air



*Gambar 4. 3 3D Penyimpanan Air*

Tangki penyimpanan air yang dirancang pada gambar memiliki dimensi 156 mm x 63 mm x 230 mm dengan volume sekitar 192.176 mm<sup>3</sup>. Tangki ini dibuat untuk ditempatkan di bagian belakang pot, sehingga tidak mengganggu estetika tampilan keseluruhan sistem dan tetap mudah diakses untuk proses pengisian maupun perawatan. Peletakan di belakang pot juga membantu menjaga keseimbangan beban sistem, terutama ketika tangki terisi penuh.

Desain tangki dibuat dengan bentuk persegi panjang vertikal untuk memaksimalkan kapasitas penampungan air pada ruang yang relatif sempit. Posisi yang lebih tinggi dari sistem penyiraman memudahkan aliran air menuju pompa atau langsung ke pipa distribusi dengan memanfaatkan gravitasi, sehingga mengurangi beban kerja pompa. Tangki ini juga memiliki ruang yang cukup untuk memasang sensor level air jika diperlukan, sehingga dapat memantau ketersediaan air secara otomatis. Penempatan tangki di belakang pot dipilih dengan pertimbangan efisiensi

ruang, kemudahan integrasi dengan komponen lain, serta menjaga sistem tetap rapi dan terorganisir.

#### 4.3.3 Bentuk Keseluruhan Perangkat Keras (*Hardware*)

Pembuatan perangkat keras sistem pot pintar berbasis ESP32 dimulai dengan mempersiapkan seluruh komponen utama yang telah disebutkan pada Tabel 4.1 dan Tabel 4.2, untuk memastikan bahwa semua alat dan bahan tersedia serta sesuai dengan kebutuhan rancangan sistem. Komponen utama meliputi ESP32, sensor *Soil Moisture*, sensor DHT22 (suhu dan kelembapan udara), sensor cahaya BH1750, modul RTC, pompa DC, MOSFET, serta LCD I2C 16x2 sebagai indikator visual data lingkungan.

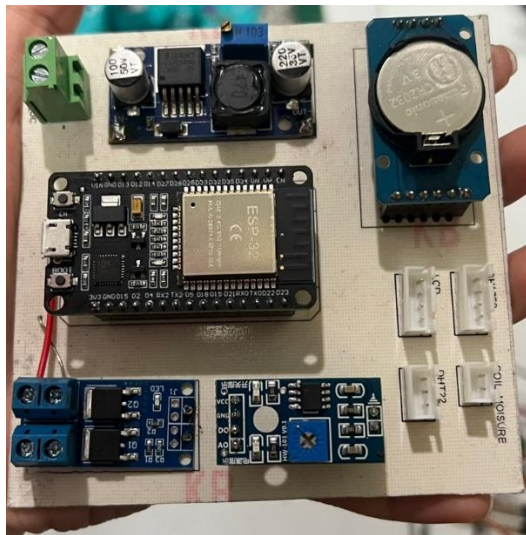
Setelah seluruh kelengkapan komponen tersedia, proses perakitan diawali dengan pemasangan sensor dan modul ke dalam papan PCB yang telah dirancang secara custom dengan ukuran 10 x 10 cm. Komponen-komponen tersebut disusun berdasarkan skema *Wiring* yang telah ditentukan, dan diintegrasikan dengan mikrokontroler ESP32 sebagai pusat kendali sistem. Rangkaian ini juga dilengkapi konektor power DC, *Buck Converter* sebagai penurun tegangan dari adaptor ke level yang sesuai, serta koneksi I2C untuk LCD dan sensor cahaya.



*Gambar 4. 4 Desain 3D Keseluruhan*

Desain sistem pot pintar ini juga mencakup aspek mekanik berupa wadah pot berbahan plastik dengan desain 3D kotak berukuran sekitar 16 x 16 x 25 cm. Di bagian depan pot, dibuat ruang khusus untuk menampung seluruh modul PCB dan LCD,. Di depan pot, terletak kotak penyimpanan untuk tempat peletakan sensor ke PCB.

Pada sistem ini, air tidak ditampung di dalam pot, melainkan dipisahkan ke wadah eksternal dan disalurkan melalui selang ke pompa DC. Pompa dikendalikan oleh sinyal dari MOSFET berdasarkan hasil perhitungan *Moving Average* dari data sensor. Nilai-nilai parameter lingkungan akan dikumpulkan secara berkala setiap 30 menit, kemudian dikirimkan ke aplikasi Blynk untuk pemantauan jarak jauh secara real-time. Data historis yang terkumpul akan digunakan untuk menerapkan metode *Predictive maintenance*, yaitu dengan memprediksi apakah dalam dua hari ke depan tanaman perlu disiram atau tidak, berdasarkan tren data parameter kelembapan dan lingkungan.



*Gambar 4. 5 PCB Komponen*

Komponen-komponen utama kemudian dipasang pada posisi yang telah ditentukan pada papan PCB untuk memastikan fungsionalitasnya dapat berjalan secara

optimal. Selanjutnya, proses perakitan dilakukan dengan hati-hati untuk memastikan penempatan rapi, kuat, dan meminimalkan potensi kesalahan dalam penyolderan.

Komponen utama yang digunakan terdiri dari mikrokontroler ESP32, sensor *Soil Moisture*, sensor suhu dan kelembapan DHT22, sensor cahaya BH1750, modul RTC DS3231, *Buck Converter*, MOSFET sebagai driver pompa DC, serta LCD 16x2 I2C sebagai penampil data. Semua komponen tersebut dipasang pada PCB berukuran 10 x 10 cm, dengan jalur koneksi disusun sesuai dengan layout rangkaian yang telah dirancang.

Pemasangan dimulai dari ESP32 sebagai pusat kendali sistem. Kemudian koneksi ke masing-masing sensor dilakukan sesuai jalur input-output pada pin ESP32. Penyolderan dilakukan dengan teliti untuk memastikan setiap koneksi terhubung dengan baik, terutama pada pin daya (VCC), ground (GND), serta pin data I/O. Jalur koneksi I2C (SCL dan SDA) juga diperhatikan dengan seksama karena digunakan untuk sensor BH1750, RTC, dan LCD.

Setelah seluruh komponen terpasang di PCB, dilakukan pengujian koneksi menggunakan multimeter untuk memastikan tidak terjadi konsleting atau sambungan putus. Kabel-kabel eksternal seperti pompa dan konektor adaptor juga dirangkai dan disambungkan ke modul MOSFET serta konektor input daya (DC jack).



*Gambar 4. 6 Hasil Cetak 3D*

Semua kabel yang terhubung ke sensor yang terletak di luar box PCB dirapikan dan diarahkan keluar melalui lubang khusus agar nantinya dapat langsung digunakan untuk pengukuran di media tanam. PCB beserta semua komponennya kemudian dipasang ke dalam box tertutup yang sudah disediakan pada sisi pot, di depan bos dilubangi untuk penempatan LCD agar informasi tetap dapat terlihat.

#### **4.4 Kalibrasi Sensor**

Kalibrasi sensor adalah proses penyesuaian atau penyesuaian hasil pembacaan sensor agar nilainya sesuai dengan kondisi fisik atau nilai rujukan yang sebenarnya. Tujuan kalibrasi adalah untuk meminimalkan kesalahan pembacaan dan memastikan data sensor yang digunakan dalam sistem prediktif memiliki tingkat akurasi yang baik.

##### **4.4.1 Kalibrasi Sensor DHT22**

Kalibrasi sensor DHT22 dilakukan untuk memastikan bahwa hasil pengukuran suhu dan kelembapan udara yang ditampilkan sensor sesuai dengan kondisi sebenarnya. Proses kalibrasi ini dilakukan dengan cara membandingkan pembacaan

sensor DHT22 terhadap alat ukur standar, yaitu HTC-2 Clock/Humidity Meter. Pengujian dilakukan pada kondisi ruangan yang sama.



*Gambar 4. 7 Kalibrasi Sensor DHT22*

Pada pengujian ini, sensor DHT22 menampilkan suhu sebesar 27°C dan kelembapan udara sebesar 64%, sedangkan alat ukur HTC-2 menunjukkan suhu 27,3°C dan kelembapan 65% pada kondisi yang sama. Sensor DHT22 memiliki tingkat akurasi yang cukup baik dan masih sesuai dengan spesifikasi toleransi pabrik. Dengan demikian, sensor dapat digunakan untuk mendukung sistem penyiraman otomatis berbasis ESP32.

*Tabel 4. 3 Akurasi Sensor DHT22*

| Parameter                | Satuan | Kisaran Output | Keterangan                        |
|--------------------------|--------|----------------|-----------------------------------|
| Suhu<br>(Temperature)    | °C     | -40 hingga 80  | Akurasi $\pm 0.5^{\circ}\text{C}$ |
| Kelembapan<br>Udara (RH) | %      | 0 hingga 100   | Akurasi $\pm 2-5\%$<br>RH         |

#### 4.4.2 Kalibrasi Sensor BH1750

Kalibrasi sensor BH1750 dilakukan untuk memastikan akurasi pembacaan intensitas cahaya (lux) dibandingkan dengan alat ukur standar (*lux meter*). Proses kalibrasi dilakukan dengan meletakkan sensor BH1750 dan lux meter pada posisi yang sama serta kondisi pencahayaan yang seragam.



*Gambar 4. 8 Kalibrasi Sensor BH1750*

Kalibrasi dilakukan dengan membandingkan hasil pembacaan sensor BH1750 dengan alat ukur standar berupa lux meter. Pada pengujian ditunjukkan bahwa sensor BH1750 membaca intensitas cahaya sebesar 55 lux.

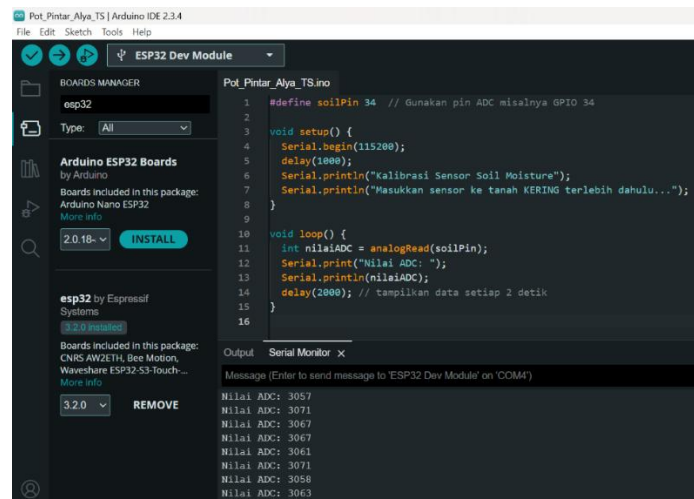


*Gambar 4. 9 Alat Ukur Intensitas Cahaya*

Nilai keluaran BH1750 kemudian dibandingkan dengan nilai lux meter, dan selisih perbedaan dicatat sebagai error, lux meter standar menunjukkan nilai 58 lux pada kondisi pencahayaan yang sama. Selisih antara kedua nilai tersebut menunjukkan adanya error kecil sekitar 5,17%, yang masih berada dalam batas toleransi sehingga sensor BH1750 dapat digunakan dengan cukup akurat untuk memantau intensitas cahaya pada sistem penyiraman otomatis berbasis ESP32.

#### 4.4.3 Kalibrasi Sensor *Soil Moisture*

Kalibrasi sensor *Soil Moisture* bertujuan untuk menentukan nilai ambang batas (range ADC) dari kondisi tanah sangat kering hingga sangat basah, sehingga hasil pembacaan dari sensor dapat diubah menjadi nilai kelembapan tanah dalam bentuk persentase (% kelembapan). *Soil Moisture* sensor tidak menghasilkan nilai dalam satuan persen, melainkan berupa nilai analog dari 0–4095 (pada ESP32). Maka diperlukan proses kalibrasi manual untuk mengkonversi nilai analog menjadi kelembapan tanah dalam bentuk persentase (%).



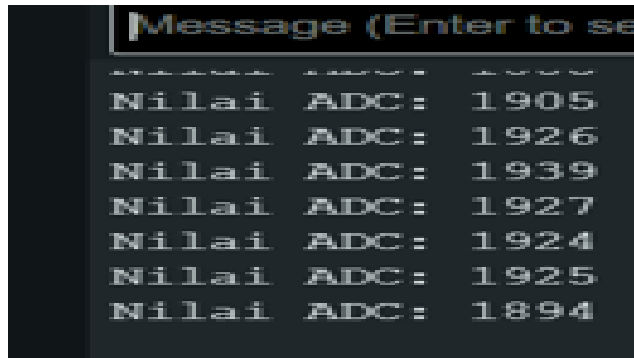
Gambar 4. 10 Nilai ADC Kering

Pada tahap awal kalibrasi sensor *Soil Moisture*, dilakukan pengujian untuk menentukan nilai pembacaan analog (nilai ADC) saat sensor berada dalam kondisi tanah yang sangat kering. Proses ini sangat penting karena nilai tersebut akan dijadikan sebagai acuan batas atas dalam perhitungan kelembapan tanah nantinya. Sensor dihubungkan ke pin analog GPIO34 pada ESP32, yang kemudian dibaca menggunakan fungsi `analogRead()`. Nilai hasil pembacaan ditampilkan secara berkala melalui Serial Monitor setiap dua detik.

Program diawali dengan inisialisasi komunikasi serial melalui `Serial.begin(115200)` agar data dapat ditampilkan ke komputer. Setelah jeda singkat selama satu detik, program memberikan instruksi kepada pengguna untuk memasukkan sensor ke dalam tanah kering. Kemudian, nilai ADC dari pin yang terhubung ke sensor dibaca dan dicetak ke Serial Monitor menggunakan `Serial.println()`.

Berdasarkan hasil yang ditampilkan di Serial Monitor, didapatkan nilai ADC berkisar antara 3057 hingga 3071 saat sensor berada di dalam tanah yang sangat kering. Nilai-nilai ini menunjukkan bahwa tegangan keluaran dari sensor cukup tinggi saat tanah tidak mengandung air, sesuai dengan prinsip kerja sensor resistif di mana konduktivitas listrik rendah menyebabkan nilai ADC menjadi tinggi. Rentang nilai ini nantinya akan digunakan sebagai `ADCKering` pada rumus konversi kelembapan tanah ke dalam bentuk persentase.

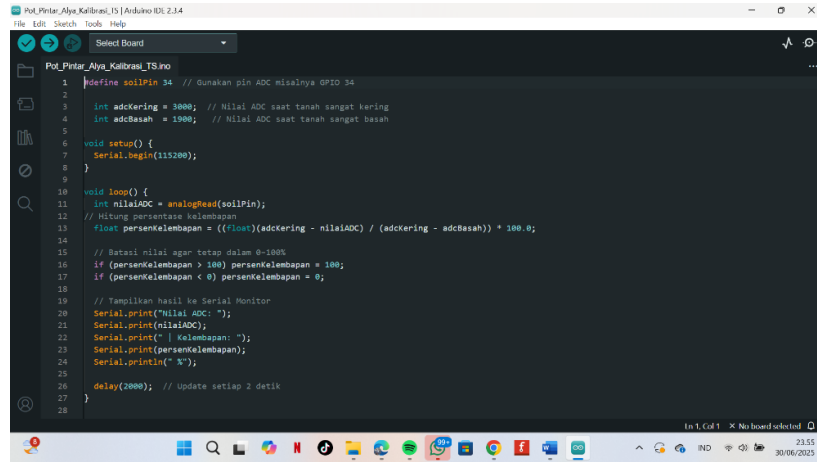
Dengan mengetahui nilai ADC pada kondisi kering, langkah selanjutnya adalah melakukan proses serupa pada kondisi tanah yang sangat basah untuk menentukan nilai `ADCBasah`. Kedua nilai ini akan menjadi batas atas dan bawah dalam rumus perhitungan kelembapan tanah secara real-time. Tahap kalibrasi ini penting dilakukan secara hati-hati agar hasil pembacaan sensor lebih akurat dan sesuai dengan kondisi sebenarnya di lapangan.



*Gambar 4. 11 Nilai ADC Basah*

Pada tahap ini, sensor diletakkan ke dalam media tanam yang telah disiram air hingga kondisi benar-benar basah dan jenuh. Program yang digunakan tetap sama dengan sebelumnya, yaitu membaca nilai analog dari pin GPIO34 dan menampilkannya ke Serial Monitor setiap dua detik. Berdasarkan hasil pengamatan pada Serial Monitor, terlihat bahwa nilai ADC yang terbaca berkisar antara 1894 hingga 1939. Rentang nilai ini menunjukkan bahwa ketika tanah berada dalam kondisi sangat basah, resistansi tanah menjadi rendah, sehingga tegangan keluarannya pun menurun, menyebabkan nilai ADC juga rendah.

Dengan hasil ini, maka nilai ADC Basah dapat diambil berdasarkan rata-rata atau nilai tengah dari rentang tersebut, misalnya sekitar **1915–1925**. Nilai ini kemudian akan dipasangkan dengan nilai sebelumnya pada tanah kering (sekitar 3060–3070) untuk menghasilkan rentang kelembapan yang proporsional. Sebagai contoh, pada implementasi program utama, nilai ADC Kering dapat ditetapkan sebagai 3000 dan ADC Basah sebagai 1920.



*Gambar 4. 12 Persen Kelembapan Tanah*

Setelah diperoleh nilai kalibrasi ADC pada kondisi tanah sangat kering dan sangat basah, langkah selanjutnya adalah mengimplementasikan konversi nilai ADC ke dalam bentuk persentase kelembapan tanah menggunakan rumus linear. Program menunjukkan bahwa pembacaan sensor dilakukan melalui pin analog GPIO34 (soilPin). Dua variabel ADCKering dan ADCBasah masing-masing diatur berdasarkan hasil kalibrasi sebelumnya, yaitu sekitar 3000 untuk kondisi kering dan 1900 untuk kondisi basah. Nilai-nilai ini digunakan sebagai batas atas dan bawah dalam proses transformasi ke skala 0–100%.

Setiap siklus pembacaan dimulai dengan mengambil data dari sensor melalui fungsi `analogRead(soilPin)`, yang menghasilkan nilai integer nilaiADC. Nilai ini kemudian dikonversi ke bentuk persentase melalui rumus:

$$\% \text{Kelembapan} = \frac{\text{ADCKering} - \text{nilaiADC}}{\text{ADCKering} - \text{ADCBasah}} \times 100 \quad (4.1)$$

Rumus ini bekerja berdasarkan prinsip bahwa nilai ADC lebih tinggi saat tanah kering, dan semakin rendah saat tanah basah. Dengan membalik arah selisih ( $ADCKering - nilaiADC$ ), hasilnya akan semakin besar mendekati 100% ketika tanah semakin basah, dan mendekati 0% ketika tanah semakin kering.

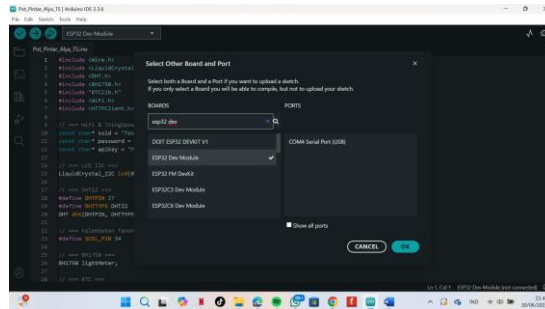
Untuk menjaga validitas data, program juga membatasi nilai persentase agar tetap berada di rentang 0 hingga 100 persen. Jika nilai kelembapan melebihi 100%, maka akan dibatasi ke 100%, dan jika bernilai negatif akibat fluktuasi data ADC, maka diatur menjadi 0%. Hal ini mencegah error logika dalam sistem pemantauan maupun penyiraman otomatis.

Hasil akhir dari pembacaan ditampilkan ke Serial Monitor dalam bentuk dua informasi utama: nilai ADC dan persentase kelembapan. Nilai ditampilkan setiap dua detik ( $delay(2000)$ ), sehingga dapat memantau kondisi tanah secara real-time. Dengan pengolahan ini, sistem pot pintar berbasis ESP32 dapat menginterpretasikan data analog dari sensor resistif menjadi bentuk numerik yang bermakna secara langsung, yaitu kelembapan tanah dalam persen, yang dapat digunakan sebagai dasar pengambilan keputusan penyiraman tanaman.

#### **4.5 Pembuatan Perangkat Lunak (*Software*)**

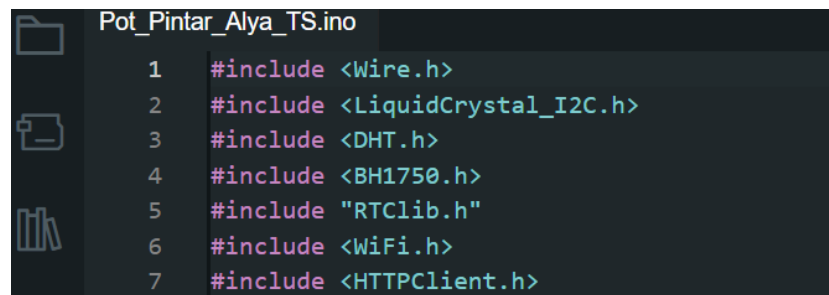
Pembuatan perangkat lunak berupa pembuatan pemrograman sistem monitoring dan controlling dari pembacaan sensor sampai dengan output hasil pembacaan pada platform Thinkspeak dan Blynk.

#### 4.5.1 Pembuatan Program pada Arduino IDE



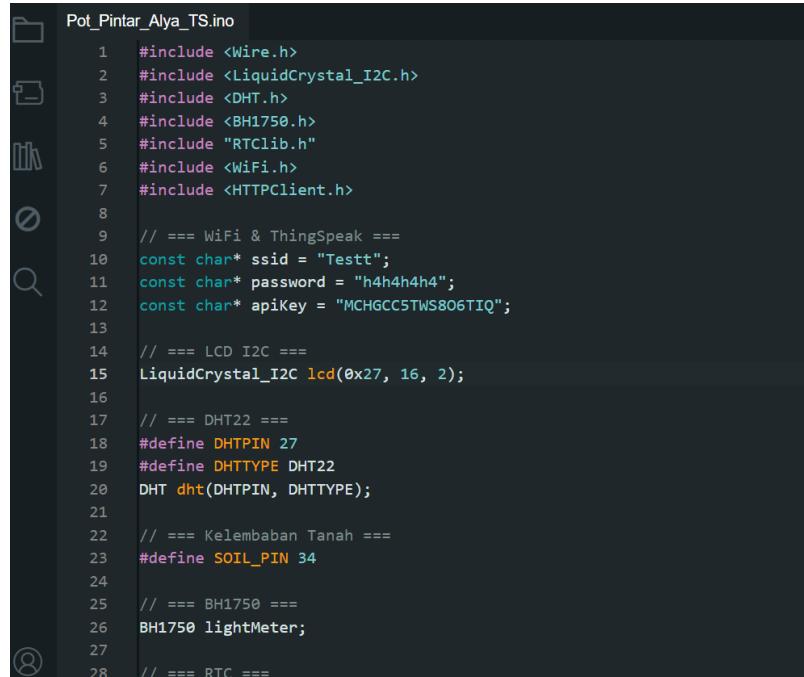
Gambar 4. 13 Pemilihan Board Arduino Ide

Pada tahap ini, perangkat lunak untuk mengendalikan sistem pot pintar dirancang menggunakan Arduino IDE. Proses pemrograman dimulai dengan membuka aplikasi Arduino IDE yang telah diinstal pada komputer. Setelah itu, dilakukan konfigurasi awal dengan memilih board yang sesuai, yaitu **ESP32 DEV Module**, melalui menu **Select Board** dan menentukan port komunikasi (COM port) yang terhubung dengan ESP32.



Gambar 4. 14 Library Arduino Ide

Langkah selanjutnya adalah mengimpor pustaka (library) yang diperlukan untuk mendukung kerja sensor-sensor dan aktuator dalam sistem pot pintar. Beberapa pustaka penting yang digunakan antara lain Wire.h, LiquidCrystal\_I2C.h, DHT.h, BH1750.h, RTCLib.h dan WiFi.h.



```

Pot_Pintar_Alya_TS.ino
1  #include <Wire.h>
2  #include <LiquidCrystal_I2C.h>
3  #include <DHT.h>
4  #include <BH1750.h>
5  #include "RTCLib.h"
6  #include <WiFi.h>
7  #include <HTTPClient.h>
8
9  // === WiFi & ThingSpeak ===
10 const char* ssid = "Testtt";
11 const char* password = "h4h4h4h4";
12 const char* apiKey = "MCHGCC5TWS806TIQ";
13
14 // === LCD I2C ===
15 LiquidCrystal_I2C lcd(0x27, 16, 2);
16
17 // === DHT22 ===
18 #define DHTPIN 27
19 #define DHTTYPE DHT22
20 DHT dht(DHTPIN, DHTTYPE);
21
22 // === Kelembaban Tanah ===
23 #define SOIL_PIN 34
24
25 // === BH1750 ===
26 BH1750 lightMeter;
27
28 // === RTC ===

```

*Gambar 4. 15 Program Arduino IDE*

Selain itu, pustaka HTTPClient.h digunakan untuk pengiriman data ke platform IoT yaitu ThingSpeak, sedangkan pustaka RTCLib.h dapat digunakan jika sistem melibatkan pencatatan waktu berbasis RTC (Real-Time Clock). Proses impor pustaka bertujuan untuk memastikan seluruh sensor dan komponen dapat dikenali dan dijalankan dengan baik sesuai konfigurasi dari masing-masing pengembang pustaka.

Setelah program berhasil diverifikasi tanpa adanya kesalahan, langkah berikutnya adalah melakukan proses unggah (upload) program ke papan ESP32. Proses ini dilakukan dengan cara menekan ikon Upload (→) di bagian atas Arduino IDE. Proses ini akan mengirimkan seluruh baris kode program ke memori ESP32 melalui sambungan USB.

```

// Kirim ke ThingSpeak setiap 20 detik
if (millis() - lastSend > intervalSend) {
    sendToThingSpeak(rh, suhu, kelembabanTanah, lux);
    lastSend = millis();
}

delay(300);

void sendToThingSpeak(float kelembabanUdara, float suhuUdara, float kelembabanTanah, float lux) {
    if (WiFi.status() == WL_CONNECTED) {
        HTTPClient http;

        String url = String("http://api.thingspeak.com/update?api_key=") + apiKey +
            "&field1=" + String(kelembabanUdara, 1) +
            "&field2=" + String(suhuUdara, 1) +
            "&field3=" + String(kelembabanTanah, 1) +
            "&field4=" + String(lux, 1);

        http.begin(url);
        int httpResponseCode = http.GET();

        if (httpResponseCode > 0) {
            Serial.print("Data sent to ThingSpeak. Code: ");
            Serial.println(httpResponseCode);
        } else {
            Serial.print("Failed to send. Error: ");

```

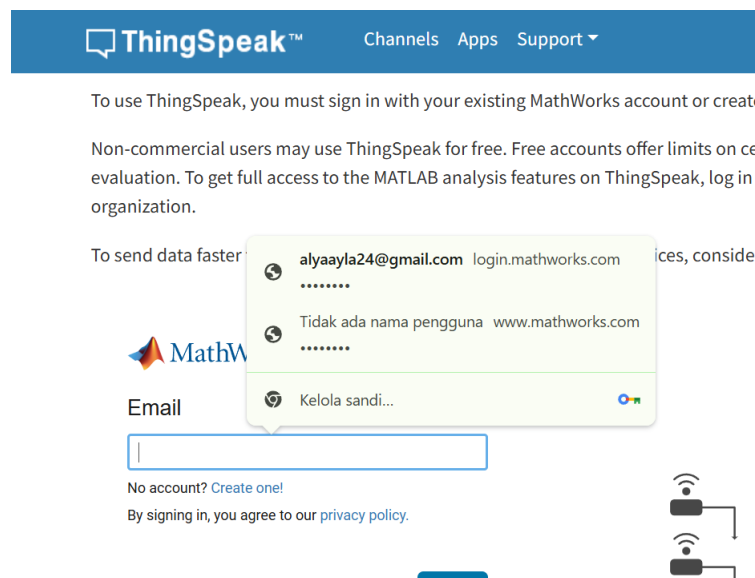
*Gambar 4. 16 Program Data Thinkspeak*

Pada bagian program ini, sistem melakukan proses pengiriman data sensor ke platform ThingSpeak setiap interval 20 detik. Interval pengiriman diatur melalui perbandingan waktu saat ini (millis()) dengan waktu terakhir pengiriman (lastSend). Ketika selisih waktu melebihi intervalSend, maka fungsi sendToThingSpeak() akan dipanggil dengan parameter berupa nilai kelembapan udara, suhu udara, kelembapan tanah, dan intensitas cahaya (lux). Di dalam fungsi sendToThingSpeak(), sistem akan terlebih dahulu memeriksa koneksi Wi-Fi dengan WiFi.status(). Jika terhubung, maka akan dibuat objek HTTPClient untuk menyusun URL request yang mengandung apiKey dan data masing-masing field. Nilai-nilai sensor tersebut dikonversi ke dalam format string dan ditambahkan ke URL sebagai parameter field (field1 hingga field4). Selanjutnya, data dikirimkan menggunakan metode http.GET(), dan respons dari ThingSpeak akan ditampilkan pada Serial Monitor sebagai konfirmasi keberhasilan atau kegagalan proses pengiriman. Bagian ini berfungsi penting dalam proses monitoring jarak jauh berbasis IoT karena memungkinkan data lingkungan dari pot pintar dikirim dan divisualisasikan secara real-time melalui ThingSpeak.

#### 4.5.2 Perangkat Lunak Platform Thingspeak

ThingSpeak merupakan platform berbasis cloud yang digunakan untuk memantau, menyimpan, dan menganalisis data sensor dari perangkat *Internet of Things* (IoT). Pada proyek ini, ThingSpeak berperan sebagai antarmuka pemantauan data secara real-time yang dikirimkan dari mikrokontroler ESP32[15].

Platform ini mendukung pengiriman dan visualisasi data dalam bentuk grafik kurva (line chart), serta menyediakan opsi untuk mengatur rentang waktu tampilan data berdasarkan jam, hari, bahkan minggu. ThingSpeak juga mendukung format API sehingga memungkinkan komunikasi dua arah, meskipun pada tugas akhir ini hanya digunakan untuk pengambilan (upload) data sensor ke server secara berkala.



Gambar 4. 17 Akun ThinkSpeak

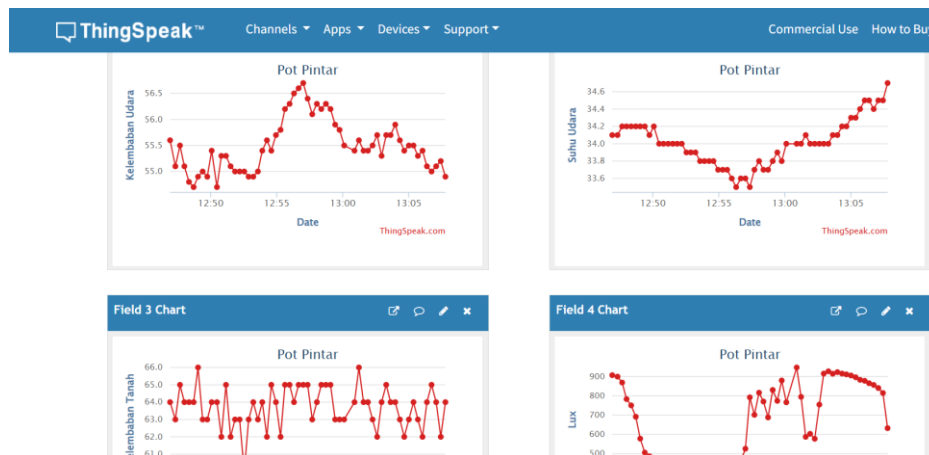
Langkah pertama dalam pemanfaatan platform ThingSpeak adalah membuat akun pada situs resmi [thingspeak.com](https://thingspeak.com). Setelah proses pendaftaran berhasil dan akun aktif, perlu membuat **channel baru** untuk menampung data dari sensor yang akan dikirimkan melalui mikrokontroler ESP32.

*Gambar 4. 18 Chanel ThinkSpeak*

Pada saat pembuatan channel, mengisi nama channel sesuai deskripsi, serta mengaktifkan field sesuai jumlah data yang akan dipantau, seperti Field 1 untuk kelembapan udara, Field 2 untuk suhu udara, Field 3 untuk kelembapan tanah, dan Field 4 untuk intensitas cahaya (lux). Setelah channel disimpan, ThingSpeak akan menyediakan **API Key** yang digunakan untuk otentikasi pengiriman data dari perangkat ke server ThingSpeak.

*Gambar 4. 19 API Key ThinkSpeak*

Langkah berikutnya adalah menghubungkan program pada Arduino IDE dengan platform ThingSpeak. Pada program, dilakukan konfigurasi Wi-Fi dan pembuatan struktur URL yang mengandung API Key serta nilai-nilai pembacaan sensor. Setiap data yang diperoleh dari sensor akan dikirim melalui koneksi HTTP secara berkala, sesuai interval yang ditentukan, menggunakan fungsi `http.GET()`. Setelah pengiriman berhasil, data tersebut akan otomatis ditampilkan dalam bentuk grafik visual pada halaman channel ThingSpeak yang telah dibuat.



*Gambar 4. 20 Grafik Data ThinkSpeak*

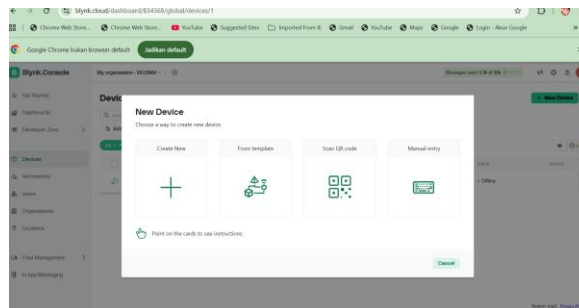
Monitoring data dapat dilakukan secara real-time melalui grafik pada tiap field yang menggambarkan tren perubahan lingkungan seperti suhu, kelembapan tanah, dan intensitas cahaya. Dengan demikian, ThingSpeak berfungsi sebagai antarmuka visual berbasis web yang sangat berguna untuk menganalisis performa sistem secara langsung dan jarak jauh tanpa perlu terhubung secara fisik ke perangkat ESP32.

#### **4.5.3 Perangkat Lunak Platform Blynk**

Sistem ini menggunakan platform Blynk sebagai antarmuka IoT untuk memantau dan mengendalikan pot pintar dari jarak jauh. Melalui Blynk, pengguna dapat melihat suhu (SU), kelembapan udara (KU), kelembapan tanah (KT), intensitas

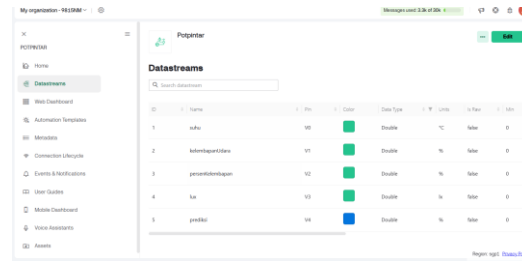
cahaya (lux), nilai prediksi (SMA-3), serta status/ kontrol pompa secara real-time. Tahapan pembuatan Blynk adalah sebagai berikut:

1. Masuk ke <https://blynk.cloud>, pilih Developer Zone → My Templates → New Template (jenis perangkat: *ESP32, WiFi*). Simpan Template ID dan Templat Name untuk dimasukkan ke program.



*Gambar 4. 21 Blynk Website*

2. Pada menu Datastreams, buat kanal virtual sebagai berikut (Gambar 4.xy):
  - V0 – suhu (Double, °C, min 0 max 100).
  - V1 – kelembapanUdara (Double, %, 0–100).
  - V2 – persenKelembapan tanah (Double, %, 0–100).
  - V3 – lux (Double, lx, 0–10000).
  - V4 – prediksi (Double, %, 0–100).
  - V5 – statusPompa (String, “ON/ OFF”).
  - V6 – pompa (Integer, 0/1) untuk sakelar manual.

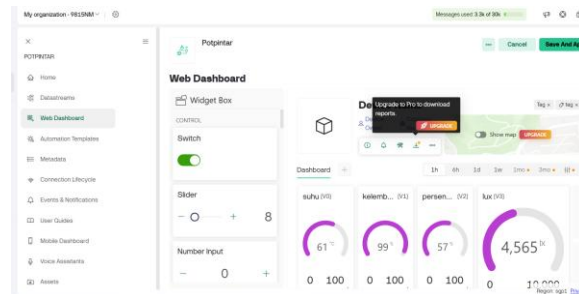


*Gambar 4. 22 Datastream Blynk*

### 3. Menyusun Dashboard

#### a. Web Dashboard

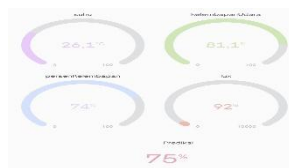
Tempatkan 4 *Gauge/Chart* untuk V0–V3, satu *Label/ Value* untuk V4 (Prediksi), satu *Label* untuk V5 (Status Pompa), dan satu *Switch* terhubung ke V6.



*Gambar 4. 23 Widget Blynk*

#### b. Mobile Dashboard (aplikasi Blynk)

Tata letak sama seperti web agar tampilan konsisten. Menghubungkan Perangkat (Device) ke Template Pada tab Devices → New Device → From Template, pilih template yang dibuat, beri nama perangkat PotPintar. Ambil Auth Token perangkat untuk dimasukkan ke program.



*Gambar 4. 24 Tampilan Blynk*