

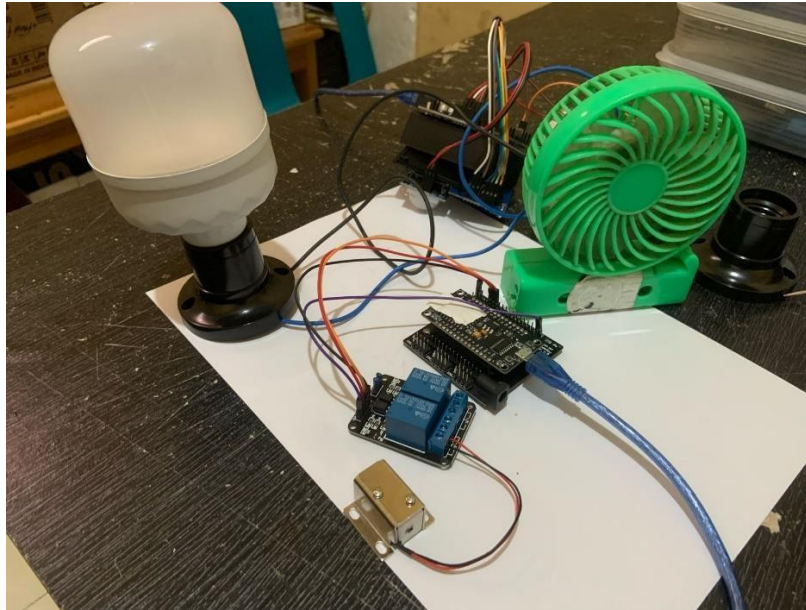
BAB IV

HASIL DAN PEMBAHASAN

4.1 Perancangan Sistem

Pada Implementasi sistem dilakukan dengan merakit seluruh komponen perangkat keras dan mengintegrasikannya dengan perangkat lunak berbasis *Internet of Things* (IoT) menggunakan protokol komunikasi MQTT. Sistem ini bertujuan untuk mengendalikan peralatan elektronik rumah seperti lampu, kipas, dan peralatan lain secara nirkabel melalui jaringan internet menggunakan modul ESP8266 sebagai *mikrokontroler* utama. Perangkat keras sistem terdiri dari ESP8266, relay, LED, kipas DC. ESP8266 akan terhubung ke jaringan WiFi dan berkomunikasi dengan *broker* MQTT untuk menerima perintah dari pengguna melalui aplikasi berbasis MQTT atau *dashboard* IoT.

Komunikasi MQTT memungkinkan sistem menerima dan mengirimkan data secara ringan dan cepat, sehingga pengguna dapat mengontrol peralatan elektronik rumah dari jarak jauh secara *real-time*. Selain itu, ESP8266 akan merespons setiap pesan yang diterima dari *topic* tertentu untuk mengaktifkan atau menonaktifkan perangkat yang dikendalikan. Implementasi sistem ini juga mencakup penggunaan antarmuka berbasis aplikasi atau *dashboard* MQTT seperti MQTT Dash, IoT MQTT Panel, atau *dashboard* web, yang dapat digunakan pengguna untuk mengontrol perangkat secara langsung dengan antarmuka yang sederhana dan mudah digunakan.



Gambar 4. 1 Tampilan Akhir Alat

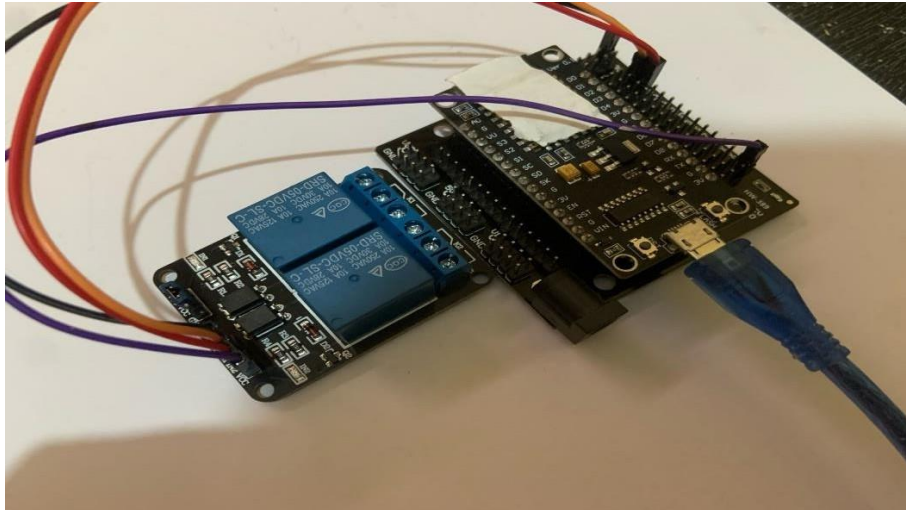
Pada gambar 4.1, terlihat bahwa perangkat keras sistem terdiri dari beberapa komponen yang saling terintegrasi dan bekerja secara otomatis berdasarkan logika yang telah diprogram. Sistem ini menggunakan ESP8266 sebagai pusat kendali utama (*mikrokontroler*) yang berfungsi untuk mengontrol peralatan elektronik rumah secara otomatis maupun manual melalui jaringan internet menggunakan komunikasi protokol MQTT. ESP8266 akan terhubung ke jaringan Wi-Fi dan berkomunikasi dengan *broker* MQTT, yang dapat diakses melalui aplikasi atau *platform* IoT. Pengguna dapat mengirimkan perintah melalui smartphone atau komputer, yang kemudian diterima oleh ESP8266 untuk mengendalikan perangkat elektronik seperti lampu LED, kipas DC, dan solenoid pintu. Protokol MQTT dipilih karena bersifat ringan, efisien, dan sangat cocok untuk aplikasi *Internet of Things* (IoT). Selain kendali perangkat, sistem ini juga dapat menampilkan informasi status perangkat melalui LCD I2C 16x2, yang memberikan tampilan lokal tentang kondisi perangkat, sehingga pengguna dapat memantau kondisi peralatan rumah secara langsung. Dengan menggabungkan ESP8266, protokol

MQTT, relay 2 channel, tampilan lokal LCD, serta pemantauan berbasis cloud, sistem ini memberikan solusi pengendalian dan pemantauan peralatan rumah yang cerdas, otomatis, dan terintegrasi melalui koneksi internet.

4.2 Implementasi Sistem

4.2.1 Pengujian Pengujian relay 2 channel

Pengujian relay 2 *channel* dilakukan untuk memastikan bahwa relay 2 *channel* berfungsi dengan baik dalam mengendalikan dua perangkat secara terpisah. Setiap *channel* relay dihubungkan dengan perangkat yang berbeda, yaitu lampu LED pada *channel* pertama dan kipas DC pada *channel* kedua. Tujuan pengujian adalah untuk memastikan bahwa masing-masing *channel* relay dapat mengendalikan perangkat secara terpisah. Metode pengujian dilakukan dengan mengirimkan perintah ON/OFF melalui aplikasi atau *platform* IoT (misal: Blynk IoT atau MQTT *Dashboard*) untuk mengendalikan masing-masing perangkat. *Channel* 1 digunakan untuk mengendalikan lampu LED, dan *Channel* 2 digunakan untuk mengendalikan kipas DC. Hasil pengujian menunjukkan bahwa lampu LED menyala/mati dengan perintah yang dikirim ke *Channel* 1, dan kipas DC menyala/mati dengan perintah ke *Channel* 2. Relay berfungsi dengan baik dalam mengendalikan perangkat secara terpisah tanpa gangguan.



Gambar 4. 2 Pengujian Pengujian Pengujian relay 2 channel

Pada gambar 4.2 pengujian relay 2 *channel* dilakukan untuk memastikan bahwa kedua *channel* pada modul relay dapat mengendalikan dua perangkat yang berbeda secara terpisah dan sesuai dengan perintah yang dikirimkan melalui *platform* IoT berbasis MQTT. Pada pengujian ini, *channel* 1 digunakan untuk mengendalikan lampu LED, sedangkan *channel* 2 digunakan untuk mengendalikan kipas DC. Pengujian dilakukan dengan mengirimkan perintah ON dan OFF melalui aplikasi MQTT *Dashboard* yang terhubung ke *broker* MQTT *broker.hivemq.com*. Perintah yang dikirim akan diterima oleh ESP8266, lalu diteruskan ke masing-masing *channel* relay untuk mengaktifkan atau menonaktifkan perangkat yang terhubung. Skenario pengujian dimulai dengan mengaktifkan dan menonaktifkan setiap *channel* secara bergantian. Ketika perintah ON dikirim ke *channel* 1, lampu LED menyala, dan saat perintah OFF dikirim, lampu mati. Hal yang sama dilakukan pada *channel* 2 untuk mengendalikan kipas DC. Selain itu, dilakukan juga pengujian dengan mengaktifkan kedua *channel* secara bersamaan untuk memastikan bahwa relay mampu bekerja tanpa konflik antar *channel*.

Hasil pengujian menunjukkan bahwa kedua *channel* relay merespons perintah dengan baik dan perangkat yang terhubung dapat beroperasi sesuai perintah yang diterima. Hal ini membuktikan bahwa relay 2 *channel* berfungsi dengan baik sebagai pengendali peralatan elektronik dalam sistem ini.

Tabel 4. 1 Data Pengujian Pengujian relay 2 channel

No	Channel	Perangkat yang Diuji	Perintah MQTT	Respon yang Diharapkan	Hasil
1	1	Lampu LED	ON	Lampu LED menyala	Berhasil
2	1	Lampu LED	OFF	Lampu LED mati	Berhasil
3	2	Kipas DC	ON	Kipas DC menyala	Berhasil
4	2	Kipas DC	OFF	Kipas DC mati	Berhasil
5	1 & 2	Lampu & Kipas	ON (keduanya)	Lampu LED dan Kipas DC menyala	Berhasil
6	1 & 2	Lampu & Kipas	OFF (keduanya)	Lampu LED dan Kipas DC mati	Berhasil

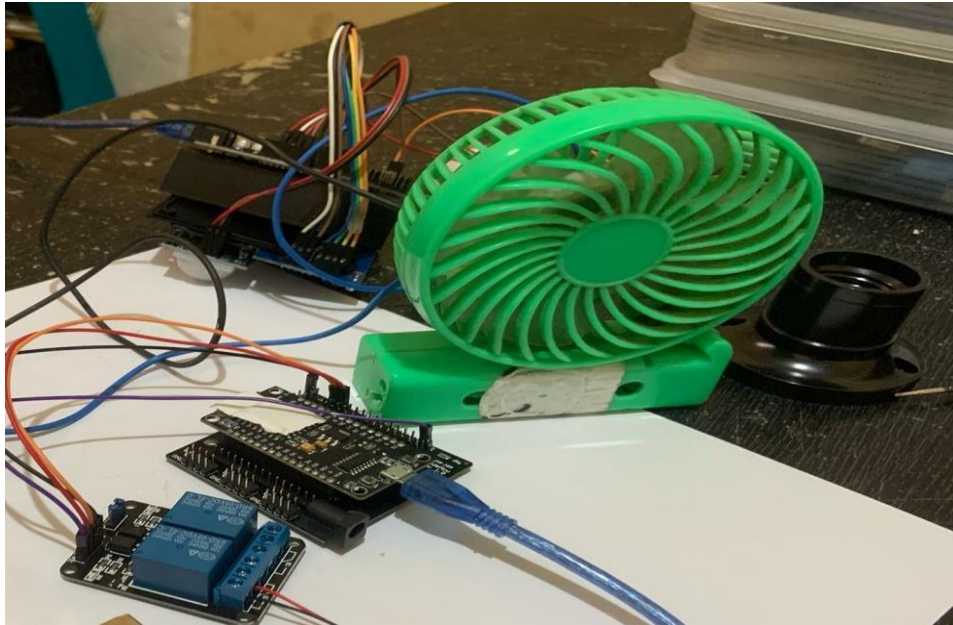
Berdasarkan Dari hasil pengujian di atas, dapat disimpulkan bahwa modul relay 2 *channel* mampu mengontrol perangkat dengan baik sesuai perintah dari *platform* IoT melalui komunikasi MQTT, baik secara terpisah maupun bersamaan. Seluruh skenario berhasil dijalankan dan perangkat merespons sesuai yang diharapkan.

4.2.2 Pengujian Kipas DC

Pengujian kipas DC dilakukan untuk memastikan bahwa sistem dapat mengendalikan kipas secara tepat melalui perintah yang dikirimkan dari aplikasi MQTT Client, seperti MQTT *Dashboard* atau aplikasi pendukung lainnya. Pengujian ini juga bertujuan untuk memastikan bahwa sistem mampu merespons perintah ON dan OFF dengan benar melalui relay yang telah terpasang pada

perangkat. Tujuan utama dari pengujian ini adalah untuk memastikan bahwa kipas DC dapat menyala saat perintah ON dikirim melalui MQTT, serta dapat mati saat perintah OFF dikirim. Selain itu, pengujian ini juga dilakukan untuk melihat sejauh mana sistem mampu merespons perintah dengan waktu tunda yang minimal.

Pengujian diawali dengan memastikan bahwa modul ESP8266 telah terhubung ke jaringan Wi-Fi serta *broker* MQTT publik, seperti *broker.hivemq.com*. Kipas DC dihubungkan ke salah satu *channel* relay yang telah dikonfigurasi, misalnya pada *channel* pertama. Setelah itu, aplikasi MQTT Client dijalankan, dan perangkat dihubungkan ke *broker* yang sama serta melakukan subscribe ke topik yang telah ditentukan untuk pengendalian kipas, seperti rumah/kontrol/kipas. Selanjutnya, dilakukan pengujian dengan mengirimkan pesan “ON” ke topik tersebut. Sistem diharapkan merespons dengan menyalakan kipas. Setelah berhasil, dilakukan pengujian lanjutan dengan mengirimkan pesan “OFF” ke topik yang sama untuk mematikan kipas. Selama proses ini, waktu jeda antara pengiriman perintah dan respons perangkat dicatat guna menilai performa sistem. Hasil pengujian menunjukkan bahwa sistem dapat mengendalikan kipas DC sesuai dengan perintah yang diberikan melalui MQTT. Respons sistem juga tergolong cepat, dengan waktu tunda yang sangat rendah dan tidak mengganggu fungsi utama sistem. Hal ini menunjukkan bahwa sistem pengendalian kipas berbasis IoT dengan komunikasi MQTT dan ESP8266 berjalan secara efektif dan sesuai dengan perancangan.



Gambar 4. 3 Pengujian Kipas DC

Berdasarkan Gambar 4.3 di atas, pengujian kipas DC dilakukan untuk memastikan bahwa sistem dapat mengendalikan kipas DC dengan tepat melalui perintah yang dikirimkan melalui komunikasi MQTT. Dalam pengujian ini, kipas DC dikendalikan menggunakan *channel 1* pada modul relay yang terhubung ke ESP8266. Perintah ON dan OFF dikirim melalui aplikasi *MQTT Dashboard* yang terhubung dengan *broker* MQTT publik. Saat perintah ON dikirim, kipas DC berhasil menyala, dan ketika perintah OFF dikirim, kipas DC berhasil mati. Pengujian juga dilakukan dengan mengirimkan perintah ON, menunggu beberapa detik, lalu mengirimkan perintah OFF. Hasilnya, kipas tetap menyala selama durasi yang telah ditentukan sebelum mati sesuai perintah. Pengujian berikutnya dilakukan dengan mengirimkan perintah ON secara berulang, dan setiap perintah berhasil direspons dengan kipas yang menyala kembali tanpa adanya keterlambatan atau kesalahan. Hasil pengujian menunjukkan bahwa sistem dapat mengendalikan kipas DC dengan andal dan responsif melalui perintah MQTT yang dikirim dari

platform IoT, yang dapat dilihat pada Tabel 4.2 di bawah ini:

Tabel 4. 2 Pengujian Sensor Suhu dan Kelembaban DHT22

No	Perintah MQTT	Status Kipas	Waktu Respon (detik)	Keterangan
1	ON	Menyala	0.3	Berfungsi normal
2	OFF	Mati	0.2	Berfungsi normal

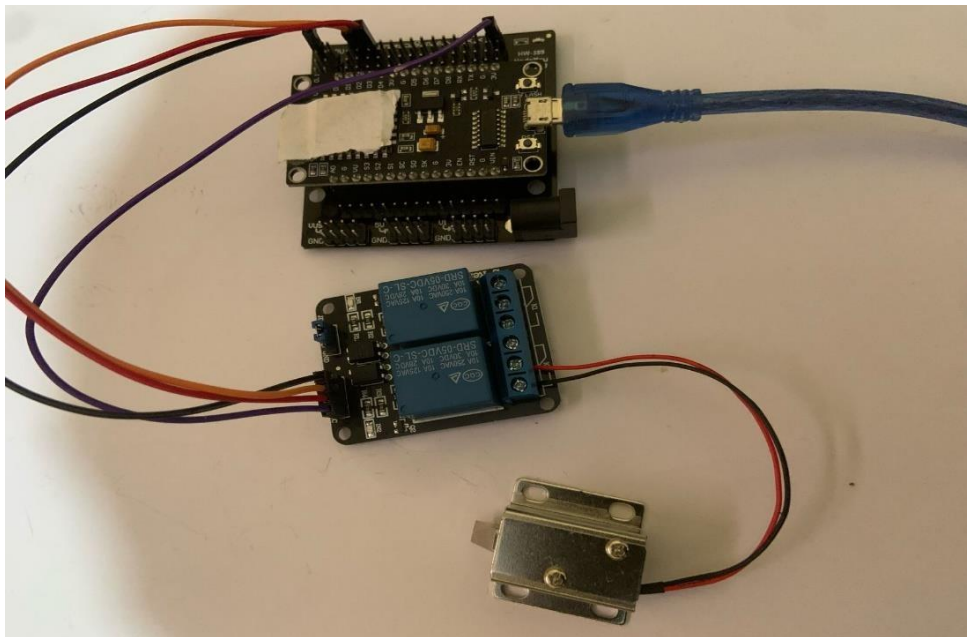
Hasil tersebut menunjukkan bahwa sistem pengendalian kipas DC berfungsi dengan baik dan responsif terhadap perintah yang diberikan. Hal ini membuktikan bahwa kipas DC dapat dikendalikan secara efektif dalam sistem pengendalian peralatan elektronik rumah berbasis IoT menggunakan ESP8266.

4.2.3 Pengujian Solenoid

Pengujian solenoid pintu dilakukan untuk memastikan bahwa sistem dapat mengendalikan solenoid dengan tepat melalui perintah yang dikirimkan melalui komunikasi MQTT. Solenoid pintu berfungsi untuk membuka atau mengunci pintu secara otomatis sesuai dengan perintah yang diterima. Dalam pengujian ini, solenoid dikendalikan menggunakan *channel* yang telah ditentukan pada modul relay yang terhubung ke ESP8266.

Tujuan dari pengujian ini adalah untuk memastikan bahwa solenoid pintu dapat berfungsi dengan baik saat perintah ON dikirim melalui MQTT, yang berarti pintu akan terbuka, dan dapat terkunci dengan mengirimkan perintah OFF melalui MQTT. Pengujian ini juga bertujuan untuk memastikan sistem dapat merespons perintah dengan waktu tunda yang minimal. Langkah pertama adalah memastikan bahwa ESP8266 terhubung ke jaringan Wi-Fi dan *broker* MQTT publik. Solenoid pintu harus terhubung ke *channel* relay yang sesuai, misalnya *channel* 2. Selanjutnya, buka aplikasi MQTT Client dan hubungkan ke *broker* yang sesuai,

kemudian subscribe ke topik yang digunakan untuk kontrol solenoid pintu, misalnya: rumah/kontrol/solenoid. Setelah itu, kirimkan perintah “ON” ke topik tersebut dan amati apakah solenoid berfungsi untuk membuka pintu. Kemudian, kirimkan perintah “OFF” dan periksa apakah solenoid mengunci pintu kembali. Selama proses ini, catat waktu respon dari pengiriman perintah hingga aksi solenoid untuk menilai performa sistem. Hasil pengujian menunjukkan bahwa solenoid pintu dapat berfungsi dengan baik. Saat perintah ON dikirim, solenoid pintu berfungsi untuk membuka pintu. Sebaliknya, saat perintah OFF dikirim, pintu kembali terkunci. Waktu respon sistem tergolong cepat, dengan waktu tunda yang minimal antara pengiriman perintah dan aksi solenoid.



Gambar 4. 4 Pengujian Selenoid

Berikut data pengukuran dari tegangan yang diukur dari Solenoid pintu dapat dilihat pada tabel 4.4 di bawah ini:

Tabel 4. 3 Data Solenoid

No	Kondisi Sistem	Tegangan Masuk (V)	Tegangan Keluar (V)	Keterangan
1	Solenoid dalam keadaan OFF	5V	0V	Solenoid tidak aktif
2	Solenoid dalam keadaan ON	5V	5V	Solenoid aktif dan membuka pintu
3	Solenoid setelah perintah OFF	5V	0V	Solenoid terkunci kembali

Berdasarkan Tabel 4.3 di atas, Berdasarkan Tabel 4.3 di atas, hasil pengukuran tegangan dilakukan melalui *mikrokontroler* ESP8266 pada solenoid pintu. Pengukuran dilakukan pada pin output solenoid yang terhubung dengan *channel* relay, yang mengendalikan solenoid untuk membuka dan mengunci pintu. Hasil pengukuran menunjukkan nilai sebagai berikut:

- a. Tegangan Masuk (Saat Solenoid OFF): 5V
- b. Tegangan Masuk (Saat Solenoid ON): 5V
- c. Tegangan Keluar (Saat Solenoid OFF): 0V
- d. Tegangan Keluar (Saat Solenoid ON): 5V

Hasil ini menunjukkan bahwa sistem berhasil mengendalikan solenoid pintu dengan tegangan yang sesuai. Solenoid bekerja dengan baik, mengunci dan membuka pintu sesuai dengan perintah yang diberikan melalui komunikasi MQTT.

4.2.4 Pengujian LED

Pengujian lampu LED dilakukan untuk memastikan bahwa sistem pengendalian peralatan elektronik rumah berbasis IoT dengan komunikasi MQTT dapat mengontrol lampu LED secara efektif melalui perintah dari aplikasi MQTT Client. Dalam sistem ini, lampu LED dikendalikan melalui salah satu *channel* pada

modul relay yang terhubung dengan ESP8266. Tujuan pengujian ini adalah untuk memastikan bahwa lampu LED dapat menyala saat perintah ON dikirim, padam saat perintah OFF dikirim, dan sistem mampu merespons setiap perintah dengan waktu tunda yang minimal.

Langkah awal dalam pengujian ini adalah memastikan ESP8266 telah terhubung ke jaringan Wi-Fi dan *broker* MQTT publik. Selanjutnya, lampu LED yang menjadi indikator dikoneksikan ke *channel* relay sesuai konfigurasi sistem. Setelah itu, aplikasi MQTT Client digunakan untuk mengirimkan perintah ON dan OFF ke topik yang digunakan untuk mengontrol lampu, seperti rumah/kontrol/lampu. Ketika perintah ON dikirim melalui aplikasi, lampu LED berhasil menyala. Sebaliknya, saat perintah OFF dikirim, lampu LED padam. Pengujian juga dilakukan dengan mengirimkan perintah ON secara berulang dalam waktu yang berbeda-beda, dan hasilnya menunjukkan bahwa sistem merespons dengan baik tanpa adanya keterlambatan atau kesalahan. Hasil pengujian menunjukkan bahwa sistem mampu mengendalikan lampu LED secara andal dan responsif melalui perintah MQTT yang dikirim dari aplikasi pengendali. Hal ini membuktikan bahwa fungsi pengendalian lampu dalam sistem dapat berjalan dengan optimal.

Tabel 4. 4 Data Pengujian LED

No	Kondisi Sistem	Tegangan Masuk (V AC)	Tegangan Keluar (V AC)	Keterangan
1	Lampu dalam keadaan OFF	170V	0V	Lampu tidak menyala, relay terbuka
2	Lampu dalam keadaan ON	170V	170V	Lampu menyala, relay tertutup
3	Setelah perintah OFF	170V	0V	Lampu padam kembali dengan baik



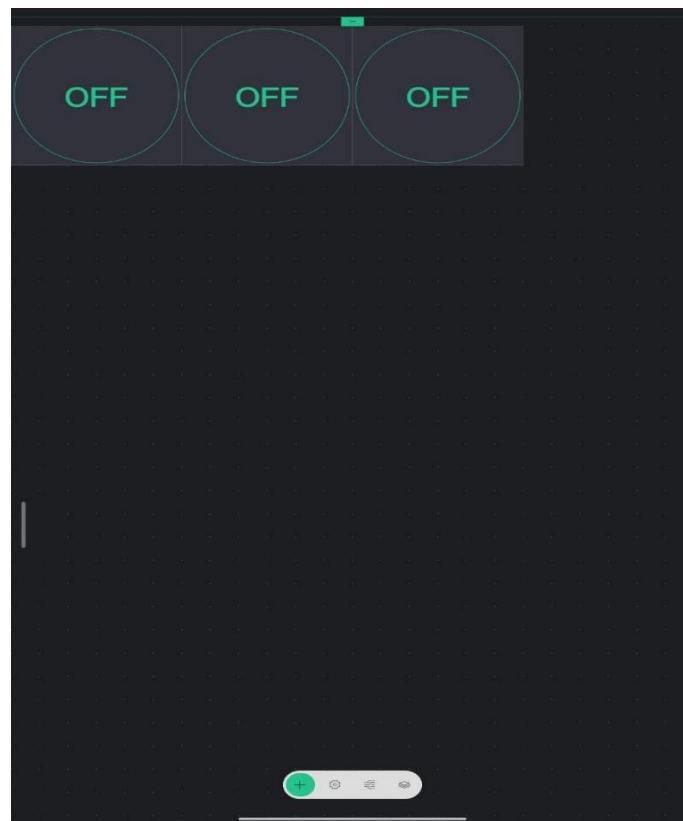
Gambar 4. 5 Pengujian LED

4.3 Pengujian Perangkat Lunak

4.3.1 Pengujian Software Blynk

Pengujian Blynk dilakukan untuk memastikan bahwa *platform* Blynk dapat digunakan sebagai antarmuka kontrol perangkat secara *real-time* dalam sistem pengendalian peralatan elektronik rumah berbasis IoT dengan komunikasi MQTT dan ESP8266. Dalam sistem ini, aplikasi Blynk digunakan untuk mengontrol kipas DC, lampu LED, dan solenoid pintu yang dihubungkan melalui modul relay 2 channel. Pengujian dilakukan dengan menghubungkan ESP8266 ke jaringan Wi-Fi dan mengintegrasikannya dengan akun Blynk menggunakan Auth Token yang telah dikonfigurasi sebelumnya. Masing-masing perangkat dikendalikan melalui widget tombol pada *dashboard* Blynk yang mewakili perintah ON dan OFF. Kipas DC dan lampu LED dikendalikan secara langsung melalui masing-masing *channel* relay, sementara solenoid pintu juga dikendalikan melalui perintah digital. Ketika

tombol ON pada aplikasi ditekan, ESP8266 menerima perintah dan mengaktifkan relay sehingga perangkat menyala. Saat tombol ditekan kembali ke OFF, relay nonaktif dan perangkat pun mati. Selama pengujian, seluruh perangkat merespons perintah dengan cepat dan akurat tanpa adanya keterlambatan yang signifikan.



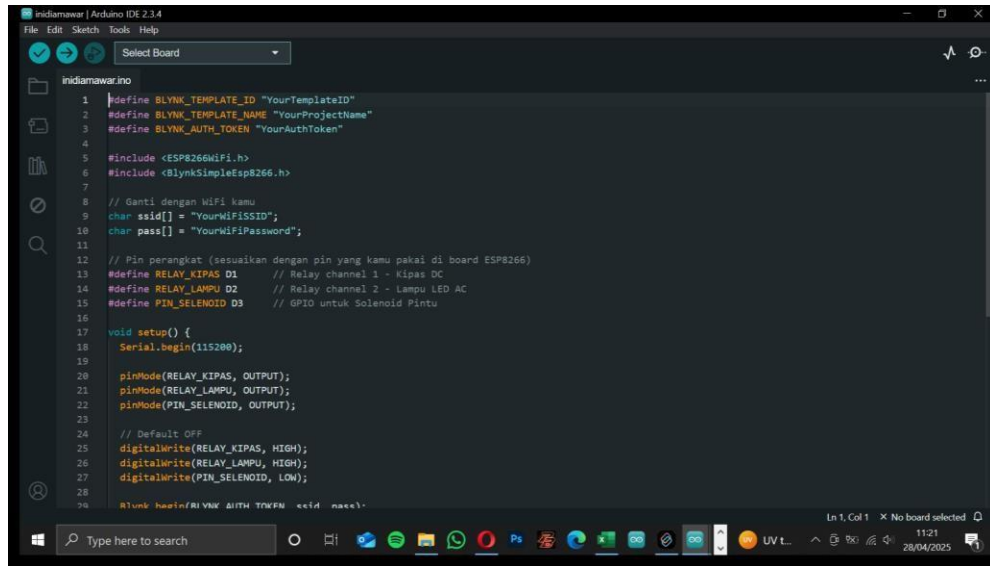
Gambar 4. 6 Tampilan Dashboard Blynk

Gambar ini hasil pengujian menunjukkan bahwa *platform* Blynk mampu bekerja secara andal sebagai antarmuka pengguna dalam mengontrol perangkat elektronik rumah. Aplikasi ini memudahkan pengguna dalam mengendalikan perangkat dari jarak jauh dengan hanya menggunakan smartphone dan koneksi internet. Dengan demikian, integrasi antara Blynk dan ESP8266 berjalan dengan baik dalam sistem pengendalian peralatan elektronik rumah berbasis IoT.

4.3.2 Pengujian Perangkat Arduino IDE

Berdasarkan Pengujian perangkat lunak menggunakan Arduino IDE dilakukan untuk memantau dan menganalisis respons ESP8266 dalam mengendalikan peralatan elektronik rumah secara *real-time* melalui komunikasi MQTT. Salah satu fitur penting yang digunakan dalam pengujian ini adalah Serial Monitor, yang memungkinkan pengguna untuk memantau data keluaran dari *mikrokontroler* secara langsung melalui antarmuka komputer. Pengujian ini bertujuan untuk memastikan bahwa ESP8266 dapat memproses perintah MQTT dengan baik dan mengaktifkan perangkat seperti kipas DC, lampu LED, dan solenoid pintu sesuai perintah yang diterima. Melalui Serial Monitor, pengguna dapat melihat status koneksi Wi-Fi, status koneksi ke *broker* MQTT, serta log perintah yang diterima dan dijalankan oleh *mikrokontroler*.

Dalam proses pengujian, baud rate pada Serial Monitor diatur sebesar 115200 untuk memastikan komunikasi data antara ESP8266 dan komputer berjalan lancar dan efisien. Pengaturan ini penting untuk meminimalkan gangguan atau kesalahan dalam pengiriman data, sehingga status perangkat dan perintah MQTT dapat dimonitor dengan akurat dan *real-time*. Hasil pengujian menunjukkan bahwa ESP8266 dapat menerima dan mengeksekusi perintah dari MQTT Client dengan baik, dan proses kontrol perangkat melalui MQTT dapat dipantau secara jelas melalui Arduino IDE.



Gambar 4. 7 Software Arduino IDE Versi 2.3.3

4.3.3 Pengujian Perangkat Arduino IDE

Pada tahap pengujian alat, dilakukan serangkaian uji coba terhadap seluruh komponen utama dalam sistem, yang bertujuan untuk memastikan setiap perangkat berfungsi sebagaimana mestinya sesuai dengan rancangan. Uji coba ini dilakukan dengan memberikan perintah melalui komunikasi MQTT dan mengamati respons dari masing-masing perangkat. Tabel uji coba dibagi menjadi dua bagian, yaitu tabel uji coba yang berhasil dan tabel uji coba yang tidak berhasil, agar mempermudah dalam mengidentifikasi perangkat yang bekerja dengan baik serta perangkat yang masih mengalami kendala.

Pada tabel uji coba yang berhasil, ditunjukkan bahwa beberapa perangkat seperti ESP8266, relay, kipas DC, LED, LCD I2C, dan sistem komunikasi MQTT berhasil merespons perintah dengan baik. Perangkat-perangkat tersebut menunjukkan performa yang stabil, tidak mengalami keterlambatan (delay), dan menjalankan fungsinya sesuai dengan logika program yang telah dirancang.

Tabel 4. 5 Uji coba Alat Yang Berhasil

No	Tanggal	Komponen	Perintah	Respons yang Diharapkan	Hasil Uji	Keterangan
1	10 Mei 2025	ESP8266 + WiFi	Dihidupkan	Terkoneksi ke WiFi dan MQTT	Berhasil	Koneksi stabil dalam ± 5 detik
2	10 Mei 2025	Relay + Kipas DC	MQTT: ON / OFF	Kipas menyala/mati sesuai perintah	Berhasil	Respon cepat dan konsisten
3	11 Mei 2025	Relay + LED	MQTT: ON / OFF	LED menyala/mati sesuai perintah	Berhasil	Tidak ada delay
4	11 Mei 2025	Solenoid (Kunci Pintu)	MQTT: BUKA / TUTUP	Solenoid aktif/nonaktif sesuai perintah	Berhasil	Pintu terbuka dan terkunci dengan baik
5	12 Mei 2025	Komunikasi MQTT	Perintah via dashboard HP	Perintah diterima ESP8266 secara real-time	Berhasil	Stabil tanpa gangguan koneksi
6	13 Mei 2025	Aplikasi Blynk	Tekan tombol ON/OFF di Blynk	Perintah berhasil diteruskan ke ESP8266	Berhasil	Tampilan user friendly, kontrol responsif

Sementara itu, pada tabel uji coba yang tidak berhasil, ditemukan beberapa kendala seperti pada pengunci pintu berbasis solenoid yang tidak aktif akibat daya yang kurang mencukupi, serta LCD I2C yang gagal menampilkan informasi karena kemungkinan kesalahan wiring. Selain itu, terdapat gangguan pada komunikasi MQTT saat koneksi internet melemah, serta relay yang kadang tidak merespons secara konsisten akibat kemungkinan gangguan sinyal atau suplai arus yang tidak stabil.

Tabel 4. 6 Uji Coba Yang Tidak Berhasil

No	Tanggal	Komponen	Perintah	Respons yang Diharapkan	Hasil Uji
1	14 Mei 2025	Solenoid (unit lain)	MQTT: BUKA / TUTUP	Solenoid aktif/nonaktif sesuai perintah	Gagal
2	14 Mei 2025	MQTT Broker	Kirim perintah saat sinyal lemah	Perintah tetap diterima ESP8266	Gagal
3	15 Mei 2025	Relay (unit lain)	Kirim sinyal dari ESP8266	Relay menyala/mati sesuai logika program	Sebagian

Melalui hasil pengujian ini, dapat disimpulkan bahwa sistem secara umum telah berfungsi dengan baik, namun masih diperlukan penyesuaian pada bagian tertentu agar sistem dapat bekerja secara optimal dan handal dalam penggunaan jangka panjang.